Package: SGP (via r-universe)

December 6, 2024

Type Package

Title Student Growth Percentiles & Percentile Growth Trajectories

Version 2.2-0.0

Date 2024-10-6

Maintainer Damian W. Betebenner <dbetebenner@nciea.org>

Depends R (>= 4.1.0)

Suggests SGPdata (>= 28.0-0), knitr, rmarkdown

Imports Cairo, callr, collapse, colorspace, crayon, datasets, data.table (>= 1.14.0), digest, doParallel, equate (>= 2.0-5), foreach, graphics, grid, grDevices, gridBase, iterators, gtools, jsonlite, matrixStats, methods, parallel, quantreg, randomNames (>= 0.0-5), rngtools (>= 1.5), RSQLite, sn (>= 1.0-0), splines, stats, svglite, toOrdinal, utils

Description An analytic framework for the calculation of norm- and criterion-referenced academic growth estimates using large scale, longitudinal education assessment data as developed in Betebenner (2009) <doi:10.1111/j.1745-3992.2009.00161.x>.

SystemRequirements (PDF)LaTeX (https://www.latex-project.org/) with 'pdfpages' package for studentGrowthPlot option in visualizeSGP to bind together student growth plots into school catalogs

URL https://sgp.io, https://github.com/CenterForAssessment/SGP, https://CRAN.R-project.org/package=SGP

BugReports https://github.com/CenterForAssessment/SGP/issues

VignetteBuilder knitr

LazyData Yes

LazyDataCompression xz

License GPL-3

ByteCompile TRUE

NeedsCompilation no

Author Damian W. Betebenner [aut, cre], Adam R. Van Iwaarden [aut], Ben Domingue [aut], Yi Shang [aut], Jonathan Weeks [ctb], John Stewart [ctb], Jinnie Choi [ctb], Xin Wei [ctb], Hi Shin Shim [ctb], Xiaoyuan Tan [ctb] (Arizona Department of Education), Carrie Giovannini [ctb] (Arizona Department of Education), Sarah Polasky [ctb] (Arizona State University), Rebecca Gau [ctb] (Arizona Charter School Association), Jeffrey Dean [ctb] (University of Arkansas), William Bonk [ctb] (Colorado Department of Education), Marie Huchton [ctb] (Colorado Department of Education), Allison Timberlake [ctb] (Georgia Department of Education), Qi Qin [ctb] (Georgia Department of Education), Melissa Fincher [ctb] (Georgia Department of Education), Kiran Athota [ctb] (Georgia Department of Education), Travis Allen [ctb] (Georgia Department of Education), Glenn Hirata [ctb] (Hawaii Department of Education), Glenn Nochi [ctb] (Hawaii Department of Education), Joshua Lee [ctb] (Hawaii Department of Education), Ayaka Nukui [ctb] (Idaho Department of Education), Carissa Miller [ctb] (Idaho Department of Education), Matthew Raimondi [ctb] (Elgin Area School District U46 (Illinois)), Wes Bruce [ctb] (Indiana Department of Education), Robert Hochsegang [ctb] (Indiana Department of Education), Tony Moss [ctb] (Kansas State Department of Education), Xuewen Sheng [ctb] (Kansas State Department of Education), Kathy Flanagan [ctb] (Massachusetts Department of Elementary and Secondary Education), Robert Lee [ctb] (Massachusetts Department of Elementary and Secondary Education), Ji Zeng [ctb] (Michigan Department of Education), Steve Viger [ctb] (Michigan Department of Education), Joe DeCastra [ctb] (Mississippi Department of Education), Ken Thompson [ctb] (Mississippi Department of Education), Soo Yeon Cho [ctb] (Missouri Department of Education), Jeff Halsell [ctb] (Clark County School District, Nevada), Selcuk Ozdemir [ctb] (Washoe County School District, Nevada), Roger Silva [ctb] (Nevada Department of Education), Deb Wiswell [ctb] (New Hampshire Department of Education), Katya Levitan-Reiner [ctb] (New Haven Public Schools), Catherine McCaslin [ctb] (New Haven Public Schools), Joshua Marland [ctb] (New York Education Department), W Joshua Rew [ctb] (Oregon Department of Education), Jason Becker [ctb] (Rhode Island Department of Education), Jessica Bailey [ctb] (Rhode Island Department of Education), Ana Karantonis [ctb] (Rhode Island Department of Education), Deborah Jonas [ctb] (Virginia Department of Education), Juan D'Brot [ctb] (West Virginia Department of Education), Nate Hixson [ctb] (West Virginia Department of Education), Deb Came [ctb] (Washington Office of Superintendent of Public Instruction), Ashley Colburn [ctb] (Washington Office of Superintendent of Public Instruction), Nick Hassell [ctb] (Washington Office of Superintendent of Public Instruction),

Krissy Johnson [ctb] (Washington Office of Superintendent of Public Instruction), Daniel Bush [ctb] (Wisconsin Department of Education), Justin Meyer [ctb] (Wisconsin Department of Education), Joseph Newton [ctb] (Wisconsin Department of Education), Nick Stroud [ctb] (Wisconsin Department of Education), John Paul [ctb] (Wyoming Department of Education), Michael Flicek [ctb] (Michael Flicek Projects LLC working with Wyoming Department of Education), Phyllis Clay [ctb] (Albuquerque Public Schools), Peter Kinyua [ctb] (Albuquerque Public Schools), Brendan Houng [ctb] (University of Melbourne, Australia, NAPLAN), Leslie Rosale [ctb] (Ministry of Education, Guatemala), Nathan Wall [ctb] (eMetric working with Nevada Department of Education and South Dakota Department of Education), Julia English [ctb] (Massachusetts DESE), Sarah Jo Torgrimson [ctb] (Massachusetts DESE), Narek Sahakyan [ctb] (World Class Instruction and Design (WIDA))

Repository CRAN

Date/Publication 2024-10-06 19:10:06 UTC

Config/pak/sysreqs libcairo2-dev libfontconfig1-dev libfreetype6-dev make texlive libpng-dev

Contents

SGP-package *SGP: Student Growth Percentiles & Percentile Growth Trajectories*

Description

SGP contains classes and functions to calculate student growth percentiles and percentile growth projections/trajectories following methodology found in Betebenner (2008, 2009). The package contains two primary functions, studentGrowthPercentiles and studentGrowthProjections, and numerous higher level functions that make use of them including: prepareSGP, analyzeSGP, combineSGP, summarizeSGP, visualizeSGP and outputSGP. These functions are used to calculate and visualize student growth percentiles and percentile growth projections/trajectories for students using large scale, longitudinal assessment data. These norm- and criterion-referenced growth values are currently used in a number of states for many purposes including diagnostic and accountability. The functions employ quantile regression (using the quantreg package) to estimate the conditional density for current achievement using each student's achievement history. Percentile growth projections/trajectories are calculated using the coefficient matrices derived from the student growth percentile analyses. These quantities are summarized in a variety of ways to describe student growth. Beginning with version 1.4-0.0, the SGP package also calculate time dependent SGPs (SGPt) and allows for student growth projections to be calculated across assessment transitions by equating the two tests.

Details

Package:	SGP
Type:	Package
Version:	2.2-0.0
Date:	2024-10-6
License:	GPL-3
LazyLoad:	yes

Calculation of student growth percentiles and percentile growth trajectories/projections is typically performed by grade and subject. Data for growth percentile calculation must be specifically formatted. See sgpData for an example data set. Batch R syntax for performing analyses across all grades and years is provided in the examples of the studentGrowthPercentiles and studentGrowthProjections using the higher level functions prepareSGP, analyzeSGP, combineSGP, summarizeSGP, and visualizeSGP.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>, Adam Van Iwaarden <avaniwaarden@nciea.org>, Ben Domingue <ben.domingue@gmail.com> and Yi Shang <shangyi@gmail.com>

References

Betebenner, D. W. (2008). Toward a normative understanding of student growth. In K. E. Ryan & L. A. Shepard (Eds.), *The Future of Test Based Accountability* (pp. 155-170). New York: Routledge.

Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. Educational Measurement: Issues and Practice, 28(4):42-51.

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.

Castellano, K. E. & McCaffrey, D. F. (2017). The Accuracy of Aggregate Student Growth Percentiles as Indicators of Educator Performance. Educational Measurement: Issues and Practice, 36(1):14-27.

Koenker, R. (2005). *Quantile regression*. Cambridge: Cambridge University Press.

Shang, Y., VanIwaarden, A., & Betebenner, D. W. (2015). Covariate measurement error correction for Student Growth Percentiles using the SIMEX method. Educational Measurement: Issues and Practice, 34(1):4-14.

---

abcSGP — *Perform 6 step sequence: prepareSGP, analyzeSGP, combineSGP, summarizeSGP, visualizeSGP, and outputSGP*

Description

Utility function to perform sequence of 6 steps going from data preparation, prepareSGP, SGP data analysis, analyzeSGP, data combining, combineSGP, data summary, summarizeSGP, data visualization visualizeSGP and data output outputSGP.

Usage

```
abcSGP(sgp_object,
 state=NULL,
 steps=c("prepareSGP", "analyzeSGP", "combineSGP",
  "summarizeSGP", "visualizeSGP", "outputSGP"),
 years=NULL,
 content_areas=NULL,
 grades=NULL,
 prepareSGP.var.names=NULL,
 prepareSGP.create.additional.variables=FALSE,
  prepareSGP.create.achievement.level=TRUE,
 sgp.percentiles=TRUE,
 sgp.projections=TRUE,
```

```
  sgp.projections.lagged=TRUE,
  sgp.percentiles.baseline=TRUE,
  sgp.projections.baseline=TRUE,
  sgp.projections.lagged.baseline=TRUE,
  sgp.use.my.coefficient.matrices=NULL,
  sgp.minimum.default.panel.years=NULL,
  sgp.target.scale.scores=FALSE,
  sgp.target.scale.scores.only=FALSE,
  sgp.test.cohort.size=NULL,
  return.sgp.test.results=FALSE,
  simulate.sgps=TRUE,
  calculate.simex=NULL,
  calculate.simex.baseline=NULL,
     calculate.srs=NULL,
     calculate.srs.baseline=NULL,
  goodness.of.fit.print=TRUE,
  parallel.config=NULL,
  save.intermediate.results=FALSE,
  save.old.summaries=FALSE,
  sgPlot.demo.report=FALSE,
  sgp.config=NULL,
  sgp.summaries=NULL,
  summary.groups=NULL,
  data_supplementary=NULL,
  confidence.interval.groups=NULL,
  plot.types=c("bubblePlot", "studentGrowthPlot", "growthAchievementPlot"),
  outputSGP.output.type=c("LONG_Data",
     "LONG_FINAL_YEAR_Data",
     "WIDE_Data",
     "INSTRUCTOR_Data"),
  outputSGP.directory="Data",
   verbose.output=FALSE,
  sgp.sqlite=FALSE,
  sgp.percentiles.equated=NULL,
     sgp.percentiles.equating.method=NULL,
  sgp.percentiles.calculate.sgps=TRUE,
     get.cohort.data.info=FALSE,
  SGPt=NULL,
   fix.duplicates=NULL)
```

Arguments

sgp_object	A list containing LONG formatted data. See sgpData_LONG for an exemplar. By including the name of the state in the object name (e.g., Idaho_SGP), the function will detect what state is associated with the data and supply that to the 'state' argument of the function so that state meta-data located in the SGP-stateData object can be utilized. NOTE: Data preparation must be meticulous to utilize this enhanced functionality.

`state` Acronym indicating state associated with the data for access to embedded knot and boundaries, cutscores, CSEMs, and other relevant state level data. This can be supplied to the function automatically by including the full state name (e.g, New_Hampshire_SGP) in the name of the object supplied to `sgp_object`.

`steps` Vector containing all or some subset of `prepareSGP`, `analyzeSGP`, `combineSGP`, `summarizeSGP`, `visualizeSGP` indicating what steps the user wants accomplished. Default is to perform all steps.

`years` A vector indicating year(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer the year(s) in which to perform growth percentile analyses based upon the assumption of having at least three years of panel data for analyses.

`content_areas` A vector indicating content area(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer the content area(s) available for analyses.

`grades` A vector indicating grades for which to calculate student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer all the grade progressions for student growth percentile and student growth projections/trajectories analyses.

`prepareSGP.var.names`
list supplied to prepareSGP mapping provided variable names to variable names required as part of the SGP package. See `prepareSGP` for more details. Defaults to NULL.

`prepareSGP.create.additional.variables`
Boolean variable indicating whether prepareSGP should create addition variables (e.g., `HIGH_NEED_STATUS`) if they are missing. Defaults to FALSE.

`prepareSGP.create.achievement.level`
Boolean variable indicating whether prepareSGP should create ACHIEVEMENT_LEVEL variable if it is missing. Defaults to TRUE.

`sgp.percentiles`
Boolean variable indicating whether to calculate student growth percentiles. Defaults to TRUE.

`sgp.projections`
Boolean variable indicating whether to calculate student growth projections. Defaults to TRUE.

`sgp.projections.lagged`
Boolean variable indicating whether to calculate lagged student growth projections often used for growth to standard analyses. Defaults to TRUE.

`sgp.percentiles.baseline`
Boolean variable indicating whether to calculate baseline student growth percentiles and/or coefficient matrices. Defaults to TRUE.

`sgp.projections.baseline`
Boolean variable indicating whether to calculate baseline student growth projections. Defaults to TRUE.

`sgp.projections.lagged.baseline`
Boolean variable indicating whether to calculate lagged baseline student growth projections. Defaults to TRUE.

`sgp.use.my.coefficient.matrices`
: Boolean variable indicating whether to use embedded coefficient matrices to calculate SGPs. One should be careful to remove previously calculated SGPs prior to recalculating SGPs.

`sgp.minimum.default.panel.years`
: Integer indicating the minimum number of panel years to begin with in the calculation of student growth percentiles. The default is NULL (converted to 3 years).

`sgp.target.scale.scores`
: Boolean variable passed to `combineSGP` indicating whether target scale scores associated with SGP_TARGETs should be calculated as part of the `combineSGP` run. Defaults to FALSE.

`sgp.target.scale.scores.only`
: Boolean variable passed to `combineSGP` indicating whether ONLY target scale scores associated with SGP_TARGETs should be calculated as part of the `combineSGP` run. Defaults to FALSE.

`sgp.test.cohort.size`
: Integer indicating the maximum number of students sampled from the full cohort to use in the calculation of student growth percentiles. Intended to be used as a test of the desired analyses to be run. The default, NULL, uses no restrictions (no tests are performed, and analyses use the entire cohort of students).

`return.sgp.test.results`
: Boolean variable passed to `analyzeSGP` and `studentGrowthPercentiles` indicating whether the results from the cohort sample subset (if specified using the above argument) should be returned for inspection. Defaults to FALSE. If TRUE, only the sample subset of the data used will be returned in the SGP object's @Data slot. Alternatively, user can supply the character "ALL_DATA" to the argument to return the entire original data.

`simulate.sgps`
: Boolean variable indicating whether to simulate SGP values for students based on test-specific Conditional Standard Errors of Measurement (CSEM). Test CSEM data must be available for simulation. Must be set to TRUE for confidence interval construction. Defaults to TRUE in `abcSGP` only.

`calculate.simex`
: A character state acronym or list including state/csem variable, csem.data.vnames, csem.loss.hoss, simulation.iterations, lambda and extrapolation method. Returns both SIMEX adjusted SGP (`SGP_SIMEX`) as well as the percentile ranked SIMEX SGP (`RANK_SIMEX`) values as suggested by Castellano and McCaffrey (2017). Defaults to NULL, no simex calculations performed. Alternatively, setting the argument to TRUE sets the list up with state=state, lambda=seq(0,2,0.5), simulation.iterations=50, simex.sample.size=25000, extrapolation="linear" and save.matrices=TRUE.

`calculate.simex.baseline`
: A character state acronym or list including state/csem variable, csem.data.vnames, csem.loss.hoss, simulation.iterations, lambda and extrapolation method. Defaults to NULL, no simex calculations performed. Alternatively, setting the argument to TRUE uses the same defaults as above along with `simex.use.my.coefficient.matrices = TRUE`. This argument is passed to `analyzeSGP`.

`calculate.srs` A character state acronym or list including [FILL IN LATER]. Creates a longitudinal data set based upon a stratified random sample of variables and proportions for the United States (default) or provided by the user. The argument defaults to NULL, Alternatively, setting the argument to TRUE uses the defaults specified above. This argument is passed to `analyzeSGP`.

`calculate.srs.baseline`
A character state acronym or list including [FILL IN LATER]. Calculates SGPs based upon previosly established coefficient matrices derived from a stratified random sample of data. Defaults to NULL, no stratified random sample SGPs are calculated. Alternatively, setting the argument to TRUE uses the defaults specified above. This argument is passed to `analyzeSGP`.

`goodness.of.fit.print`
Boolean variable passed to `analyzeSGP` indicating whether to print goodness of fit results.

`parallel.config`
A named list with, at a minimum, two elements indicating 1) the BACKEND package to be used for parallel computation and 2) the WORKERS list to specify the number of processors to be used in each major analysis. The BACKEND element can be set = to `FOREACH` or `PARALLEL`. Please consult the manuals and vignettes for information of these packages! The `analyzeSGP` help page contains more thorough explanation and examples of the `parallel.config` setup.
The parallel.config list is passed to `analyzeSGP`, `combineSGP`, `summarizeSGP` and `visualizeSGP`. The WORKERS list can accordingly contain elements for PERCENTILES, PROJECTIONS, LAGGED_PROJECTIONS, BASELINE_MATRICES, BASELINE_PERCENTILES for `analyzeSGP`, SUMMARY for `summarizeSGP` and GA_PLOTS and SG_PLOTS for `visualizeSGP`. See those functions help pages for details.

`save.intermediate.results`
Should intermediate results of `abcSGP` be saved after each of `prepareSGP`, `analyzeSGP`, `combineSGP`, and `summarizeSGP`. Default is FALSE.

`save.old.summaries`
A Boolean argument (defaults to FALSE which will delete the `@Summary` slot before creating new summaries) indicating whether the call to `summarizeSGP` should save existing summaries in the `@Summary` slot.

`sgPlot.demo.report`
A Boolean variable (defaults to FALSE) indicating whether to produce only the demonstration student report catalog. Default is to produce reports for entire current year data.

`sgp.config` Configuration passed to `analyzeSGP` and `combineSGP` for user specified SGP analyses. See `analyzeSGP` documentation for details on format of configuration argument.

`sgp.summaries` A list giving the summaries requested for each group analyzed based upon the `summary.group` argument. Default is NULL allowing the `summarizeSGP` function to produce the list of summaries automatically.

`summary.groups` A list consisting of 8 types of groups across which all summaries are taken: `institution`, `content`, `time`, `institution_type`, `institution_level`, `demographic`, and `institution_inclusion`. Summaries generated in summarizeSGP are

for all possible combinations of the 8 types of group. See documentation for summarizeSGP for more detail.

data_supplementary
: A list argument (or NULL, the default) providing additional multiple membership lookup tables for summarizeSGP. See sgpData_INSTRUCTOR_NUMBER for an example. Supplied data is embedded in the @Data_Supplementary slot.

confidence.interval.groups
: A subset of the groups provided in the summary.groups argument indicating which groups to provide confidence intervals for. See documentation for summarizeSGP for more detail.

plot.types
: A character vector passed to visualizeSGP indicating the types of plots to produce. Currently supported plots include bubblePlots, studentGrowthPlots, and growthAchievementPlots.

outputSGP.output.type
: An argument passed to outputSGP indicating the output types to be produced. Defaults to LONG_Data, LONG_FINAL_YEAR_Data, WIDE_Data, and INSTRUCTOR_Data.

outputSGP.directory
: A a file path indicating where to save output files. Defaults to Data.

verbose.output
: A Boolean argument indicating whether the function should output verbose diagnostic messages.

sgp.sqlite
: A Boolean argument (defaults to FALSE) indicating whether a SQLite database file of the essential SGP data should be created from the @Data slot and subsequently used to extract data subsets for analyses conducted in order to reduce the amount of RAM memory required. See full argument description in analyzeSGP.

sgp.percentiles.equated
: A Boolean argument (defaults to NULL, which calculates equated results if in the equating year) passed to analyzeSGP indicating whether equating should be used on the most recent year of test data provided. Equating allows for student growth projections to be calculated in across assessment transitions where the scale for the assessment changes.

sgp.percentiles.equating.method
: Character vector argument passed to analyzeSGP indicating type(s) of equating method to used if sgp.percentiles.equated=TRUE. Default is NULL indicating 'equipercentile' equating. Options include 'identity', 'mean', 'linear', and 'equipercentile'.

sgp.percentiles.calculate.sgps
: Boolean argument passed to analyzeSGP indicating whether student growth percentiles are produced as part of calls to the studentGrowthPercentiles function. Default is TRUE. Setting to FALSE produces only coefficient matrices.

get.cohort.data.info
: Boolean argument passd to analyzeSGP indicating whether to create norm group cohort information based upon sgp configurations provided or calculated in analyzeSGP.

SGPt
: Argument (defaults to NULL) indicating whether to perform time dependent SGP analyses (SGPt).

`fix.duplicates` Argument to control how `analyzeSGP` and `combineSGP` deal with duplicate records based upon the key of VALID_CASE, CONTENT_AREA, YEAR, and ID. The function currently warns of duplicate records and doesn't modify data. If set to 'KEEP.ALL', `analyzeSGP` tries to fix the duplicate individual records by adding a '_DUP_***' suffix to the duplicate ID before running `studentGrowthPercentiles` in order to create unique records based upon the key. If needed, the `@Data` slot will be extended as necessary to accomodate additional student records and SGP results in `combineSGP`.

Value

Function returns a list containing the input long data set in the `@Data` slot as a data.table keyed using `VALID_CASE`, `CONTENT_AREA`, `YEAR`, `ID`, SGP results including student growth percentile and student growth projections/trajectories in the SGP slot, and summary results in the `@Summary` slot.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

See Also

`prepareSGP`, `analyzeSGP`, `combineSGP`, `summarizeSGP`, `studentGrowthPercentiles`, and `studentGrowthProjections`

Examples

```
## Not run:
## Runs all 5 steps

Demonstration_SGP <- abcSGP(sgp_object=sgpData_LONG, state="DEMO")


## Or letting the function detect the state.

Demonstration_SGP <- abcSGP(sgpData_LONG)


###
###  Example uses of the parallel.config argument
###

Demonstration_SGP <- abcSGP(sgpData_LONG,
 parallel.config=list(
            BACKEND="PARALLEL", TYPE="PSOCK",
            WORKERS=list(
             PERCENTILES=8, BASELINE_PERCENTILES=8, PROJECTIONS=7, LAGGED_PROJECTIONS=6,
               SUMMARY=8,
               GA_PLOTS=8, SG_PLOTS=8)
 )
)
```

```
## End(Not run)
```

---

analyzeSGP *Analyze student data to produce student growth percentiles and student growth projections*

Description

Wrapper function used to produce student growth percentiles and student growth projections (both cohort and baseline referenced) using long formatted data like that provided by prepareSGP.

Usage

```
analyzeSGP(sgp_object,
           state=NULL,
           years=NULL,
           content_areas=NULL,
           grades=NULL,
           sgp.percentiles=TRUE,
           sgp.projections=TRUE,
           sgp.projections.lagged=TRUE,
           sgp.percentiles.baseline=TRUE,
           sgp.projections.baseline=TRUE,
           sgp.projections.lagged.baseline=TRUE,
           sgp.percentiles.baseline.max.order=3,
           sgp.percentiles.srs.baseline.max.order=3,
           sgp.projections.baseline.max.order=3,
           sgp.projections.lagged.baseline.max.order=3,
           sgp.projections.max.forward.progression.years=3,
           sgp.projections.max.forward.progression.grade=NULL,
           sgp.projections.use.only.complete.matrices=NULL,
           sgp.minimum.default.panel.years=NULL,
           sgp.use.my.coefficient.matrices=NULL,
           sgp.use.my.sgp_object.baseline.coefficient.matrices=NULL,
           sgp.test.cohort.size=NULL,
           return.sgp.test.results=FALSE,
           simulate.sgps=TRUE,
           calculate.simex=NULL,
           calculate.simex.baseline=NULL,
           calculate.simex.srs.baseline=NULL,
           calculate.srs=NULL,
           calculate.srs.baseline=NULL,
           goodness.of.fit.print=TRUE,
           sgp.config=NULL,
           sgp.config.drop.nonsequential.grade.progression.variables=TRUE,
           sgp.baseline.panel.years=NULL,
```

```
        sgp.baseline.config=NULL,
        trim.sgp.config=TRUE,
        parallel.config=NULL,
        verbose.output=FALSE,
        print.other.gp=NULL,
        sgp.projections.projection.unit="YEAR",
        get.cohort.data.info=FALSE,
        sgp.sqlite=FALSE,
        sgp.percentiles.equated=NULL,
        sgp.percentiles.equating.method=NULL,
        sgp.percentiles.calculate.sgps=TRUE,
        SGPt=NULL,
        fix.duplicates=NULL,
        ...)
```

Arguments

sgp_object	An object of class SGP containing long formatted data in the @Data slot (from prepareSGP).
state	Acronym indicating state associated with the data for access to embedded knot and boundaries, cutscores, CSEMs, and other state related assessment data.
years	A vector indicating year(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer the year(s) based upon the assumption of having at least three years of panel data for analyses.
content_areas	A vector indicating content area(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If left missing the function will use the data to infer the content area(s) available for analyses.
grades	A vector indicating grades for which to calculate student growth percentiles and/or student growth projections/trajectories. If left missing the function will use the data to infer all the grade progressions for student growth percentile and student growth projections/trajectories analyses.
sgp.percentiles	Boolean variable indicating whether to calculate student growth percentiles. Defaults to TRUE.
sgp.projections	Boolean variable indicating whether to calculate student growth projections. Defaults to TRUE.
sgp.projections.lagged	Boolean variable indicating whether to calculate lagged student growth projections often used for growth to standard analyses. Defaults to TRUE.
sgp.percentiles.baseline	Boolean variable indicating whether to calculate baseline student growth percentiles and/or coefficient matrices. Defaults to TRUE.
sgp.projections.baseline	Boolean variable indicating whether to calculate baseline student growth projections. Defaults to TRUE.

`sgp.projections.lagged.baseline`
: Boolean variable indicating whether to calculate lagged baseline student growth projections. Defaults to TRUE.

`sgp.percentiles.baseline.max.order`
: Integer indicating the maximum order to calculate baseline student growth percentiles (regardless of maximum coefficient matrix order). Also the max order of baseline coefficient matrices to be calculated if requested. Default is 3. To utilize the maximum matrix order, set to NULL.

`sgp.percentiles.srs.baseline.max.order`
: Integer indicating the maximum order to calculate baseline, stratified random sample, student growth percentiles (regardless of maximum coefficient matrix order). Also the max order of baseline SRS coefficient matrices to be calculated if requested. Default is 3. To utilize the maximum matrix order, set to NULL.

`sgp.projections.baseline.max.order`
: Integer indicating the maximum order to calculate baseline student growth projections (regardless of maximum coefficient matrix order). Default is 3. To utilize the maximum matrix order, set to NULL.

`sgp.projections.lagged.baseline.max.order`
: Integer indicating the maximum order to calculate lagged baseline student growth projections (regardless of maximum coefficient matrix order). Default is 3. To utilize the maximum matrix order, set to NULL.

`sgp.projections.max.forward.progression.years`
: Integer indicating the maximum number of years forward that cohort based projections will be established for. Default is 3 years.

`sgp.projections.max.forward.progression.grade`
: Integer indicating the maximum grade forward that cohort based projections will be established for. Default is NULL, the highest grade.

`sgp.projections.use.only.complete.matrices`
: Boolean argument (defaults to TRUE/NULL) indicating whether to produce projections only when a complete set of coefficient matrices is available.

`sgp.minimum.default.panel.years`
: Integer indicating the minimum number of panels years to use for default sgp analyses. Default value is NULL (converted to 3) years of data.

`sgp.use.my.coefficient.matrices`
: Argument, defaults to NULL, indicating whether to use coefficient matrices embedded in argument supplied to 'sgp_object' to calculate student growth percentiles.

`sgp.use.my.sgp_object.baseline.coefficient.matrices`
: Argument, defaults to NULL (FALSE), indicating whether to utilize baseline matrices embedded in supplied `sgp_object` and not utilize baseline matrices embedded in `SGPstateData`.

`sgp.test.cohort.size`
: Integer indicating the maximum number of students sampled from the full cohort to use in the calculation of student growth percentiles. Intended to be used as a test of the desired analyses to be run. The default, NULL, uses no restrictions (no tests are performed, and analyses use the entire cohort of students).

`return.sgp.test.results`
: Boolean variable passed to `studentGrowthPercentiles` indicating whether the results from the cohort sample subset (if specified using the above argument) should be returned for inspection. Defaults to FALSE. If TRUE, only the sample subset of the data used will be returned in the SGP object's @Data slot. Alternatively, user can supply the character "ALL_DATA" to the argument to return the entire original data.

`simulate.sgps`
: Boolean variable indicating whether to simulate SGP values for students based on test-specific Conditional Standard Errors of Measurement (CSEM). Test CSEM data must be available for simulation and included in `SGPstateData`. This argument must be set to TRUE for confidence interval construction. Defaults to TRUE.

`calculate.simex`
: A character state acronym or list including state/csem variable, csem.data.vnames, csem.loss.hoss, simulation.iterations, lambda and extrapolation method. Returns both SIMEX adjusted SGP (`SGP_SIMEX`) as well as the percentile ranked SIMEX SGP (`RANK_SIMEX`) values as suggested by Castellano and McCaffrey (2017). Defaults to NULL, no simex calculations performed. Alternatively, setting the argument to TRUE sets the list up with state=state, lambda=seq(0,2,0.5), simulation.iterations=50, simex.sample.size=25000, extrapolation="linear" and save.matrices=TRUE.

`calculate.simex.baseline`
: A character state acronym or list including state/csem variable, csem.data.vnames, csem.loss.hoss, simulation.iterations, lambda and extrapolation method. Defaults to NULL, no simex calculations performed. Alternatively, setting the argument to TRUE uses the same defaults as above along with `simex.use.my.coefficient.matrices = TRUE`, which assumes baseline SIMEX coefficient matrices are available.

`calculate.simex.srs.baseline`
: A character state acronym or list including state/csem variable, csem.data.vnames, csem.loss.hoss, simulation.iterations, lambda and extrapolation method. Defaults to NULL, no simex calculations performed for stratified random sample (SRS). Alternatively, setting the argument to TRUE uses the same defaults as above along with `simex.use.my.coefficient.matrices = TRUE`, which assumes baseline SIMEX coefficient matrices are available.

`calculate.srs`
: A character state acronym or list including [FILL IN LATER]. Creates a longitudinal data set based upon a stratified random sample of variables and proportions for the United States (default) or provided by the user. The argument defaults to NULL, Alternatively, setting the argument to TRUE uses the defaults specified above.

`calculate.srs.baseline`
: A character state acronym or list including [FILL IN LATER]. Calculates SGPs based upon previosly established coefficient matrices derived from a stratified random sample of data. Defaults to NULL, no stratified random sample SGPs are calculated. Alternatively, setting the argument to TRUE uses the defaults specified above.

`goodness.of.fit.print`
: Boolean variable indicating whether to print out Goodness of Fit figures as PDF into a directory labeled Goodness of Fit. Defaults to TRUE.

sgp.config
: If years, content_areas, and grades are missing, user can directly specify a list containing three vectors: baseline.content.areas, baseline.panel.years, and baseline.grade.sequences. This advanced option is helpful for analysis of non-traditional grade progressions and other special cases. See examples for use cases.

sgp.config.drop.nonsequential.grade.progression.variables
: Boolean variable (defaults to TRUE) indicating whether non-sequential grade progression variables should be dropped when sgp.config is processed. For example, if a grade progression of c(3,4,6) is provided, the data configuration will assume (default is TRUE) that data for a missing year needs to be dropped prior to applying studentGrowthPercentiles or studentGrowthProjections to the data.

sgp.baseline.panel.years
: A vector of years to be used for baseline coefficient matrix calculation. Default is to use most recent five years of data.

sgp.baseline.config
: A list containing three vectors: sgp.content.areas, sgp.panel.years, sgp.grade.sequences indicating how baseline student growth percentile analyses are to be conducted. In almost all cases this value is calculated by default within the function but can be specified directly for advanced use cases. See source code for more detail on this configuration option.

trim.sgp.config
: A Boolean variable indicating whether the arguments content_areas, years and grades should be used to 'trim' any manually supplied configuration for analysis supplied by 'sgp.config'.

parallel.config
: A named list with, at a minimum, two elements indicating 1) the BACKEND package to be used for parallel computation and 2) the WORKERS list to specify the number of processors to be used in each major analysis. The BACKEND element can be set = to FOREACH or PARALLEL. Please consult the manuals and vignettes for information of these packages!

  TYPE is a third element of the parallel.config list that provides necessary information when using FOREACH or PARALLEL packages as the backend. With BACKEND="FOREACH", the TYPE element specifies the flavor of 'foreach' backend. As of version 1.0-1.0, only "doParallel" is supported. If BACKEND = "PARALLEL", the parallel package will be used. This package combines deprecated parallel packages snow and multicore. Using the "snow" implementation of parallel the function will create a cluster object based on the TYPE element specified and the number of workers requested (see WORKERS list description below). The TYPE element indicates the users preferred cluster type (either "PSOCK" for socket cluster of "MPI" for an OpenMPI cluster). If Windows is the operating system, this "snow" implementation must be used and the TYPE element must = "PSOCK". Defaults are assigned based on operating system if TYPE is missing based on system OS. Unix/Mac OS defaults to the "multicore" to avoid worker node pre-scheduling and appears to be more efficient in these operating systems.

  The WORKERS list must contain, at a minimum, a single number of processors (nodes) desired or available. If WORKERS is specified in this man-

ner, then the same number of processors will be used for each analysis type (sgp.percentiles, sgp.projections, ... sgp.projections.lagged.baseline). Alternatively, the user may specify the numbers of processors used for each analysis. This allows for better memory management in systems that do not have enough RAM available per core. The choice of the number of cores is a balance between the number of processors available, the amount of RAM a system has and the size of the data (sgp_object). Each system will be different and will require some tailoring. One rule of thumb used by the authors is to allow for 4GB of memory per core used for running large state data. The SGP Demonstration (and data that size) requires more like 1-2GB per core. As an example, PERCENTILES=4 and PROJECTIONS=2 might be used on a quad core machine with 4 GB of RAM. This will use all 4 cores available for the sgp.percentiles analysis and 2 cores for the sgp.projections analysis (which requires more memory than available). The WORKERS list accepts these elements: PERCENTILES, PROJECTIONS (for both cohort and baseline referenced projections), LAGGED_PROJECTIONS (for both cohort and baseline referenced lagged projections), BASELINE_MATRICES (used to produce the baseline coefficient matrices when not available in SGPstateData - very computationally intensive), BASELINE_PERCENTILES (SGP calculation only when baseline coefficient matrices have already been produced and are available - NOT very computationally intensive).

Alternatively, the name of an external CLUSTER.OBJECT (PSOCK or MPI) set up by the user outside of the function can be used.

Example use cases are provided below.

`verbose.output` A Boolean argument (defaults to FALSE) indicating whether the function should output verbose diagnostic messages.

`print.other.gp` A Boolean argument (defaults to FALSE) indicating whether the function should output SGP of all orders.

`sgp.projections.projection.unit`

A character vector argument indicating whether the studentGrowthProjections function should produce projections relative to future grades or future years. Options are "YEAR" and "GRADE", with default being "YEAR".

`get.cohort.data.info`

A Boolean argument (defaults to FALSE) indicating whether a summary of all cohorts to be submitted to the `studentGrowthPercentiles` and `studentGrowthProjections` functions should be performed prior to analysis.

`sgp.sqlite` A Boolean argument (defaults to FALSE) indicating whether a SQLite database file of the essential SGP data should be created from the `@Data` slot and subsequently used to extract data subsets for analysis with `studentGrowthPercentiles` and `studentGrowthProjections` functions. If the size of the `@Data` object is greater than 1 GB `sgp.sqlite` is set to TRUE internally. When TRUE, this can substantially reduce the amount of RAM memory required to conduct analyses. If set to TRUE the file "TMP_SGP_Data.sqlite" will be created in the R temporary directory (see`?tempdir` for information). This file is deleted by default although one may keep it if the argument is specified as the character "KEEP".

`sgp.percentiles.equated`

A Boolean argument (defaults to NULL/FALSE) indicating whether equating

should be used on the most recent year of test data provided. Equating allows for student growth projections to be calculated in across assessment transitions where the scale for the assessment changes.

`sgp.percentiles.equating.method`
: A character vector (defaults to NULL/'equipercentile') indicating the type of equating method to use. Options include any combination of 'identity', 'mean', 'linear', and 'equipercentile'.

`sgp.percentiles.calculate.sgps`
: A Boolean argument (defaults to TRUE) indicating whether to calculate percentiles in calls to studentGrowthPercentiles function. Setting to FALSE would indicate desire to calculate only coefficient matrices and no percentiles.

`SGPt`
: An argument supplied to implement time-dependent SGP analyses (SGPt). Default is NULL giving standard, non-time dependent argument. If set to TRUE, the function assumes the variables 'TIME' and 'TIME_LAG' are supplied as part of the panel.data. To specify other names, supply a list of the form: list(TIME='my_time_name', TIME_LAG='my_time_lag_name'), substituting your variable names.

`fix.duplicates`
: Argument to control how duplicate records based upon the key of VALID_CASE, CONTENT_AREA, YEAR, and ID are dealt with. If set to 'KEEP.ALL', the function tries to fix the duplicate individual records by adding a '_DUP_***' suffix to the duplicate ID before running `studentGrowthPercentiles` in order to create unique records based upon the key. See `combineSGP` for additional info on `fix.duplicates` functionality.

`...`
: Arguments to be passed to `studentGrowthPercentiles` or `studentGrowthProjections` for finer control over SGP calculations. NOTE: arguments can only be passed to one lower level function at a time, and only student growth percentiles OR projections can be created but not both at the same time.

Value

Function returns a list containing the long data set in the `@Data` slot as a `data.table` keyed using `VALID_CASE`, `CONTENT_AREA`, `YEAR`, `ID` and the student growth percentile and/or student growth projection/trajectory results in the `SGP` slot.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

See Also

`prepareSGP`, `combineSGP`

Examples

```
## Not run:
## analyzeSGP is Step 2 of 5 of abcSGP
Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- prepareSGP(Demonstration_SGP)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP)
```

```
## Or (explicitly pass state argument)

Demonstration_SGP <- prepareSGP(sgpData_LONG)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP, state="DEMO")

###
###  Example uses of the sgp.config argument
###

#  Use only 3 years of Data, for grades 3 to 6
#  and only perform analyses for most recent year (2012)

my.custom.config <- list(
MATHEMATICS.2013_2014 = list(
 sgp.content.areas=rep("MATHEMATICS", 3), # Note, must be same length as sgp.panel.years
 sgp.panel.years=c('2011_2012', '2012_2013', '2013_2014'),
 sgp.grade.sequences=list(3:4, 3:5, 4:6)),
READING.2013_2014 = list(
 sgp.content.areas=rep("READING", 3),
 sgp.panel.years=c('2011_2012', '2012_2013', '2013_2014'),
 sgp.grade.sequences=list(3:4, 3:5, 4:6)))

Demonstration_SGP <- prepareSGP(sgpData_LONG)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP,
 sgp.config=my.custom.config,
 sgp.percentiles.baseline = FALSE,
 sgp.projections.baseline = FALSE,
 sgp.projections.lagged.baseline = FALSE,
 simulate.sgps=FALSE)


##  Another example sgp.config list:

#  Use different CONTENT_AREA priors, and only 1 year of prior data
my.custom.config <- list(
MATHEMATICS.2013_2014.READ_PRIOR = list(
 sgp.content.areas=c("READING", "MATHEMATICS"),
 sgp.panel.years=c('2012_2013', '2013_2014'),
 sgp.grade.sequences=list(3:4, 4:5, 5:6)),
READING.2013_2014.MATH_PRIOR = list(
 sgp.content.areas=c("MATHEMATICS", "READING"),
 sgp.panel.years=c('2012_2013', '2013_2014'),
 sgp.grade.sequences=list(3:4, 4:5, 5:6)))


## An example showing multiple priors within a single year

Demonstration_SGP <- prepareSGP(sgpData_LONG)

DEMO.config <- list(
READING.2012_2013 = list(
 sgp.content.areas=c("MATHEMATICS", "READING", "MATHEMATICS", "READING", "READING"),
 sgp.panel.years=c('2010_2011', '2010_2011', '2011_2012', '2011_2012', '2012_2013'),
```

```
 sgp.grade.sequences=list(c(3,3,4,4,5), c(4,4,5,5,6), c(5,5,6,6,7), c(6,6,7,7,8))),
MATHEMATICS.2012_2013 = list(
 sgp.content.areas=c("READING", "MATHEMATICS", "READING", "MATHEMATICS", "MATHEMATICS"),
 sgp.panel.years=c('2010_2011', '2010_2011', '2011_2012', '2011_2012', '2012_2013'),
 sgp.grade.sequences=list(c(3,3,4,4,5), c(4,4,5,5,6), c(5,5,6,6,7), c(6,6,7,7,8))))

Demonstration_SGP <- analyzeSGP(
  Demonstration_SGP,
  sgp.config=DEMO.config,
  sgp.projections=FALSE,
  sgp.projections.lagged=FALSE,
  sgp.percentiles.baseline=FALSE,
  sgp.projections.baseline=FALSE,
  sgp.projections.lagged.baseline=FALSE,
  sgp.config.drop.nonsequential.grade.progression.variables=FALSE)


###
###  Example uses of the parallel.config argument
###

##  Windows users must use a snow socket cluster:
#  possibly a quad core machine with low RAM Memory
#  4 workers for percentiles, 2 workers for projections.
#  Note the PSOCK type cluster is used for single machines.

Demonstration_SGP <- prepareSGP(sgpData_LONG)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP,
 parallel.config=list(
  BACKEND="PARALLEL", TYPE="PSOCK",
  WORKERS=list(PERCENTILES=4,
                   PROJECTIONS=2,
                   LAGGED_PROJECTIONS=2,
                   BASELINE_PERCENTILES=4))

##  New parallel package - only available with R 2.13 or newer
#  Note there are up to 16 workers, and MPI is used,
#  suggesting this example is for a HPC cluster, possibly Windows OS.
 ...
 parallel.config=list(
  BACKEND="PARALLEL", TYPE="MPI",
  WORKERS=list(PERCENTILES=16,
                   PROJECTIONS=8,
                   LAGGED_PROJECTIONS=6,
                   BASELINE_PERCENTILES=12))
 ...

## FOREACH use cases:
 ...
 parallel.config=list(
  BACKEND="FOREACH", TYPE="doParallel",
  WORKERS=3)
 ...
```

```
#  NOTE:  This list of parallel.config specifications is NOT exhaustive.
#  See examples in analyzeSGP documentation for some others.0

###
###  Advanced Example: restrict years, recalculate baseline SGP
###  coefficient matrices, and use parallel processing
###

#  Remove existing DEMO baseline coefficient matrices from
#  the SGPstateData object so that new ones will be computed.

SGPstateData$DEMO$Baseline_splineMatrix <- NULL

#  set up a customized sgp.config list

 . . .

#  set up a customized sgp.baseline.config list

 . . .

#  to be completed


## End(Not run)
```

---

baselineSGP *Analyze student data to produce student growth percentiles and coefficient matrices from a baseline (i.e. multiple cohort) norm group*

Description

Utility function/exemplar used to produce student growth percentiles using long formatted data like that provided by prepareSGP. Used as part of analyzeSGP for baseline referenced student growth percentile analyses.

Usage

```
baselineSGP(sgp_object,
       state=NULL,
       years=NULL,
       content_areas=NULL,
       grades=NULL,
       exclude.years=NULL,
       sgp.config=NULL,
       sgp.baseline.config=NULL,
       sgp.baseline.panel.years=NULL,
```

```
        sgp.percentiles.baseline.max.order=3,
        return.matrices.only=FALSE,
        calculate.baseline.sgps=TRUE,
        calculate.simex.baseline=NULL,
        goodness.of.fit.print=TRUE,
        parallel.config=NULL,
        SGPt=NULL,
...)
```

Arguments

sgp_object	An object of class SGP containing long formatted data in the @Data (from prepareSGP) slot.
state	Acronym indicating state associated with the data for access to embedded knot and boundaries.
years	A vector indicating year(s) in which to produce baseline referenced student growth percentiles.
content_areas	A vector indicating content area in which to produce baseline referenced student growth percentiles.
grades	A vector indicating which grades to calculate baseline referenced student growth percentiles.
exclude.years	A vector indicating which years to exclude from the calculations?
sgp.config	If years, content_areas, and grades are missing, user can directly specify a list containing three vectors: baseline.content.areas, baseline.panel.years, and baseline.grade.sequences. This advanced option is helpful for analysis of non-traditional grade progressions and other special cases. See analyzeSGP for use cases.
sgp.baseline.config	A list containing three vectors: sgp.content.areas, sgp.panel.years, sgp.grade.sequences indicating how baseline student growth percentile analyses are to be conducted. In most cases this value will be calculated by default within the function but can be specified directly for advanced use cases. See source code for more detail on this configuration option.
sgp.baseline.panel.years	A character vector indicating the years to be used for the calculation of baseline SGPs. Default is to use most recent five years of data.
sgp.percentiles.baseline.max.order	Integer indicating the maximum order to calculate baseline student growth percentiles (regardless of maximum coefficient matrix order). Default is 3. To utilize the maximum matrix order, set to NULL.
return.matrices.only	Boolean variable indicating whether the function will only return baseline referenced coefficient matrices. Defaults to FALSE.
calculate.baseline.sgps	Boolean variable indicating whether the function will calculate baseline referenced student growth percentiles from baseline referenced coefficient matrices. Defaults to TRUE.

`calculate.simex.baseline`
: A list including state/csem variable, csem.data.vnames, csem.loss.hoss, simulation.iterations, lambda and extrapolation method. Defaults to NULL, no simex calculations performed. Alternatively, setting the argument to TRUE sets the list up with state=state, lambda=seq(0,2,0.5), simulation.iterations=50, simex.sample.size=25000, extrapolation="linear" and save.matrices=TRUE.

`goodness.of.fit.print`
: Boolean variable indicating whether the function will export goodness of fit plots if baseline referenced student growth percentiles are calculated. Defaults to TRUE.

`parallel.config`
: parallel configuration argument allowing for parallel analysis by 'tau'. Defaults to NULL.

`SGPt`
: Argument supplied to generate time dependent SGPs. Defaults to NULL/FALSE.

`...`
: Arguments to be passed internally to `studentGrowthPercentiles` for finer control over SGP calculations.

Value

If `return.matrices.only` is set to TRUE function returns a list containing the baseline referenced coefficient matrices. Otherwise function returns the SGP object provided with the sgp_object argument with the baseline referenced coefficient matrices, growth percentiles, etc. embedded.

Author(s)

Adam Van Iwaarden <avaniwaarden@nciea.org>, Ben Domingue <ben.domingue@gmail.com> and Damian W. Betebenner <dbetebenner@nciea.org>

See Also

`prepareSGP`, `analyzeSGP`, `combineSGP`

Examples

```
## Not run:
##  Calculate baseline referenced SGPs
## (using coefficient matrices embedded in SGPstateData)

Demonstration_SGP <- prepareSGP(sgpData_LONG)
Demonstration_SGP <-  baselineSGP(Demonstration_SGP)


## Calculate baseline referenced coefficient matrices

SGPstateData[["DEMO"]][["Baseline_splineMatrix"]] <- NULL
Demonstration_SGP <- prepareSGP(sgpData_LONG)
DEMO_Baseline_Matrices <- baselineSGP(
    Demonstration_SGP,
    return.matrices.only=TRUE,
    calculate.baseline.sgps=FALSE)
```

```
## Calculate baseline referenced coefficient matrices and
## baseline referenced SGPs with 4 years of data

SGPstateData[["DEMO"]][["Baseline_splineMatrix"]] <- NULL

sgpData_LONG_4_YEAR <- subset(sgpData_LONG, YEAR!="2013_2014")

Demonstration_SGP <- prepareSGP(sgpData_LONG_4_YEAR)
Demonstration_SGP <- baselineSGP(Demonstration_SGP)

## End(Not run)
```

---

bubblePlot *Core bubblePlot function for SGP*

Description

Function to create bubble plots associated with student growth percentile and percentile growth trajectory results. The function is adaptable to many representations but is used in conjunction with results derived from summarizeSGP to represent summary level results of growth against achievement (usually, median student growth percentile against percentage at/above proficient). The function has MANY options and users are advised to read this documentation thoroughly as well as investigate the source code for the function itself to see what the many different representations that are possible. The function has the ability to produce interactive data tips using the pdf2 package available on R-Forge. This package is NOT installed as part of the SGP package and must be installed separately from the package to take advantage of this functionality of the bubblePlot function. To install pdf2 from the R prompt type: `install.packages("pdf2",repos="http://R-Forge.R-project.org")`. The use of the pdf2 package is scheduled to be deprecated as it is no longer maintained by the creator and requires use of a pre 2.14 version of R.

Usage

```
bubblePlot(
 bubble_plot_data.X,
 bubble_plot_data.Y,
 bubble_plot_data.SUBSET=NULL,
 bubble_plot_data.INDICATE=NULL,
        bubble_plot_data.BUBBLE_CENTER_LABEL=NULL,
 bubble_plot_data.SIZE,
 bubble_plot_data.LEVELS=NULL,
 bubble_plot_data.BUBBLE_TIPS_LINES,
 bubble_plot_labels.X=c("Growth", "Median Student Growth Percentile"),
 bubble_plot_labels.Y=c("Achievement", "Percent at/above Proficient"),
 bubble_plot_labels.SIZE=c(50, 100, 500, 1000),
 bubble_plot_labels.LEVELS=NULL,
 bubble_plot_labels.BUBBLE_TIPS_LINES=list("Median SGP (Count)",
```

```
 "Percent at/above Proficient"),
bubble_plot_labels.BUBBLE_TITLES,
bubble_plot_titles.MAIN="Growth and Achievement",
bubble_plot_titles.SUB1="State School Performance",
bubble_plot_titles.SUB2="Growth & Current Achievement",
bubble_plot_titles.LEGEND1="School Size",
bubble_plot_titles.LEGEND2_P1=NULL,
bubble_plot_titles.LEGEND2_P2=NULL,
bubble_plot_titles.NOTE=NULL,
bubble_plot_configs.BUBBLE_MIN_MAX=c(0.03, 0.03),
bubble_plot_configs.BUBBLE_X_TICKS=seq(0,100,10),
bubble_plot_configs.BUBBLE_X_TICKS_SIZE=c(rep(0.6, 5), 1, rep(0.6, 5)),
       bubble_plot_configs.BUBBLE_X_BANDS=NULL,
       bubble_plot_configs.BUBBLE_X_BAND_LABELS=NULL,
bubble_plot_configs.BUBBLE_Y_TICKS=seq(0,100,10),
bubble_plot_configs.BUBBLE_Y_TICKS_SIZE=rep(0.6, 11),
       bubble_plot_configs.BUBBLE_Y_BANDS=NULL,
       bubble_plot_configs.BUBBLE_Y_BAND_LABELS=NULL,
bubble_plot_configs.BUBBLE_SUBSET_INCREASE=0,
bubble_plot_configs.BUBBLE_SUBSET_ALPHA=list(Transparent=0.3, Opaque=0.95),
bubble_plot_configs.BUBBLE_COLOR="deeppink2",
       bubble_plot_configs.BUBBLE_COLOR_GRADIENT_REVERSE=FALSE,
bubble_plot_configs.BUBBLE_TIPS=TRUE,
bubble_plot_configs.BUBBLE_PLOT_DEVICE="PDF",
bubble_plot_configs.BUBBLE_PLOT_FORMAT="print",
bubble_plot_configs.BUBBLE_PLOT_LEGEND=FALSE,
bubble_plot_configs.BUBBLE_PLOT_TITLE=TRUE,
bubble_plot_configs.BUBBLE_PLOT_SUMMARY_STATISTICS=TRUE,
bubble_plot_configs.BUBBLE_PLOT_BACKGROUND_LABELS=c("Growth", "Achievement"),
bubble_plot_configs.BUBBLE_PLOT_EXTRAS="BASE_LINE",
     bubble_plot_configs.BUBBLE_PLOT_DIMENSION=NULL, ## List of WIDTH and HEIGHT
bubble_plot_configs.BUBBLE_PLOT_NAME="bubblePlot.pdf",
bubble_plot_configs.BUBBLE_PLOT_PATH="Figures",
bubble_plot_pdftk.CREATE_CATALOG=FALSE)
```

Arguments

`bubble_plot_data.X`
: A vector of X coordinates for the bubbles to be plotted.

`bubble_plot_data.Y`
: A vector of Y coordinates for the bubbles to be plotted.

`bubble_plot_data.SUBSET`
: A Boolean vector indicating a subset of the bubbles to be highlighted and plotted. When BUBBLE_TIPS are indicated, only subsetted bubbles will show bubble tips. To further accentuate highlight bubbles, their radius can be altered using the bubble_plot_configs.BUBBLE_SUBSET_INCREASE argument. Default value is `bubble_plot_data.SUBSET=NULL`.

`bubble_plot_data.INDICATE`
: A Boolean vector indicating whether to attached a label to to further highlight

in a manner suitable for printing. Usually done for few bubbles. Default value is bubble_plot_data.INDICATE=NULL.

bubble_plot_data.BUBBLE_CENTER_LABEL
: A character vector to label the interior of the bubbles with. Usually a vector of singleton characters. Default value is bubble_plot_data.BUBBLE_CENTER_LABEL=NULL.

bubble_plot_data.SIZE
: A vector indicating the size of each of the bubbles plotted.

bubble_plot_data.LEVELS
: A vector (usually a factor) indicating categories to which the bubbles belong. Default value is bubble_plot_data.LEVELS=NULL.

bubble_plot_data.BUBBLE_TIPS_LINES
: A list of arbitrary length indicating the different values supplied when bubble tips are requested. Default value is bubble_plot_data.BUBBLE_TIPS_LINES=list(paste(MEDIAN_SGP ' (', MEDIAN_SGP_COUNT, ')', sep=''), round(PERCENT_AT_ABOVE_PROFICIENT)).

bubble_plot_labels.X
: A vector of length 2 where the 1st component is, usually a one word summary for the axis (e.g., Growth) and the 2nd component of the vector is a label for the axis (e.g., Median Student Growth Percentile). Default value is bubble_plot_labels.X=c('Growth', 'Median Student Growth Percentile').

bubble_plot_labels.Y
: A vector of length 2 where the 1st component is, usually a one word summary for the axis (e.g., Achievement) and the 2nd component of the vector is a label for the axis (e.g., Percent at/above Proficient). Default value is bubble_plot_labels.Y=c('Achievemen 'Percent at/above Proficient').

bubble_plot_labels.SIZE
: A vector of quantities giving breaking points for the size bubbles indicated in the legend of the plot. Default value is bubble_plot_labels.SIZE=c(50, 100, 500, 1000).

bubble_plot_labels.LEVELS
: A vector of quantities giving level labels associated with bubble_plot_data.LEVELS. These labels will appear in the right legend of the plot. Default value is bubble_plot_labels.LEVELS=NU

bubble_plot_labels.BUBBLE_TIPS_LINES
: A list of labels that appear in the mouse over data tips. Should be of same length as the list from bubble_plot_data.BUBBLE_TIPS_LINES. Default value is bubble_plot_labels.BUBBLE_TIPS_LINES=list('Median SGP (Count)', 'Percent at/above Proficient').

bubble_plot_labels.BUBBLE_TITLES
: A character vector with of the same length as number of points plotted indicated what name should appear on each mouse over bubble tip (e.g., the school name associated with the bubble. Default value is bubble_plot_labels.BUBBLE_TITLES=SCHOOL_NAME,

bubble_plot_titles.MAIN
: The main title of the bubble plot. Default value is bubble_plot_titles.MAIN='Growth and Achievement'.

bubble_plot_titles.SUB1
: The right upper title of the bubble plot. Default value is bubble_plot_titles.SUB1='State School Performance'.

bubble_plot_titles.SUB2
: The right lower title of the bubble plot. Default value is bubble_plot_titles.SUB2='Growth & Current Achievement'.

bubble_plot_titles.LEGEND1
: The title of the upper legend to the right of the bubble plot. Default value is bubble_plot_titles.LEGEND1='School Size'.

bubble_plot_titles.LEGEND2_P1
: The 1st line of the title of the lower legend of the bubble plot. Default value is bubble_plot_titles.LEGEND2_P1=NULL.

bubble_plot_titles.LEGEND2_P2
: The 2nd line of the title of the lower legend of the bubble plot. Default value is bubble_plot_titles.LEGEND2_P2=NULL.

bubble_plot_titles.NOTE
: A note, message, description, etc to be placed in lower half of the legend. Default value is bubble_plot_titles.NOTE=NULL. Note that this can only be used if there are not LEVELS (and therefore no second legend).

bubble_plot_configs.BUBBLE_MIN_MAX
: A vector of length two indicating min and max values for the bubbles in inches. Default value is bubble_plot_configs.BUBBLE_MIN_MAX=c(0.03, 0.03).

bubble_plot_configs.BUBBLE_X_TICKS
: A vector indicating what x coordinates to display for the x axis of the bubble plot. Default value is bubble_plot_configs.BUBBLE_X_TICKS=seq(0,100,10).

bubble_plot_configs.BUBBLE_X_TICKS_SIZE
: A vector, the same length as bubble_plot_configs.BUBBLE_X_TICKS indicating the character expansion (i.e., cex) associated with the characters. Default value is bubble_plot_configs.BUBBLE_X_TICKS_SIZE=c(rep(0.6, 5), 1, rep(0.6, 5)).

bubble_plot_configs.BUBBLE_X_BANDS
: A vector of cutpoints used to separate the plot background into horizontal rectangular regions.

bubble_plot_configs.BUBBLE_X_BAND_LABELS
: A character vector of labels used to labels the rectangles produced with bubble_plot_configs.BUBBLE_X

bubble_plot_configs.BUBBLE_Y_TICKS
: A vector indicating what x coordinates to display for the x axis of the bubble plot. Default value is bubble_plot_configs.BUBBLE_Y_TICKS=seq(0,100,10).

bubble_plot_configs.BUBBLE_Y_TICKS_SIZE
: A vector, the same length as bubble_plot_configs.BUBBLE_Y_TICKS indicating the character expansion (i.e., cex) associated with the characters. Default value is bubble_plot_configs.BUBBLE_Y_TICKS_SIZE=rep(0.6, 11).

bubble_plot_configs.BUBBLE_Y_BANDS
: A vector of cutpoints used to separate the plot background into horizontal rectangular regions.

bubble_plot_configs.BUBBLE_Y_BAND_LABELS
: A character vector of labels used to labels the rectangles produced with bubble_plot_configs.BUBBLE_X

bubble_plot_configs.BUBBLE_SUBSET_INCREASE
: Default value is bubble_plot_configs.BUBBLE_SUBSET_INCREASE=0.

bubble_plot_configs.BUBBLE_SUBSET_ALPHA
: Default value is bubble_plot_configs.BUBBLE_SUBSET_ALPHA=list(Transparent=0.3, Opaque=0.95).

bubble_plot_configs.BUBBLE_COLOR
: Default value is bubble_plot_configs.BUBBLE_COLOR="deeppink2".

bubble_plot_configs.BUBBLE_COLOR_GRADIENT_REVERSE
: Boolean argument (defaults to FALSE) indicating whether to reverse color gradient associated with bubbles.

bubble_plot_configs.BUBBLE_TIPS
: Default value is bubble_plot_configs.BUBBLE_TIPS=TRUE.

bubble_plot_configs.BUBBLE_PLOT_DEVICE
: Default value is bubble_plot_configs.BUBBLE_PLOT_DEVICE='PDF'.

bubble_plot_configs.BUBBLE_PLOT_FORMAT
: Default value is bubble_plot_configs.BUBBLE_PLOT_FORMAT='print'.

bubble_plot_configs.BUBBLE_PLOT_LEGEND
: Default value is bubble_plot_configs.BUBBLE_PLOT_LEGEND=FALSE.

bubble_plot_configs.BUBBLE_PLOT_TITLE
: Default value is bubble_plot_configs.BUBBLE_PLOT_TITLE=TRUE.

bubble_plot_configs.BUBBLE_PLOT_SUMMARY_STATISTICS
: Default value is bubble_plot_configs.BUBBLE_PLOT_SUMMARY_STATISTICS=TRUE. Controls whether summary statistics are printed in the legend.

bubble_plot_configs.BUBBLE_PLOT_BACKGROUND_LABELS
: Default value is bubble_plot_configs.BUBBLE_PLOT_BACKGROUND_LABELS=c('Growth', 'Achievement').

bubble_plot_configs.BUBBLE_PLOT_EXTRAS
: Default value is bubble_plot_configs.BUBBLE_PLOT_EXTRAS='BASE_LINE' which prints a vertical line at 50 on the plot.

bubble_plot_configs.BUBBLE_PLOT_DIMENSION
: Default value is bubble_plot_configs.BUBBLE_PLOT_DIMENSION=NULL.

bubble_plot_configs.BUBBLE_PLOT_NAME
: Default value is bubble_plot_configs.BUBBLE_PLOT_NAME='bubblePlot.pdf'.

bubble_plot_configs.BUBBLE_PLOT_PATH
: Default value is bubble_plot_configs.BUBBLE_PLOT_PATH=paste('Figures', sep='').

bubble_plot_pdftk.CREATE_CATALOG
: Default value is bubble_plot_pdftk.CREATE_CATALOG=FALSE.

Details

Typical use of the function is as part of visualizeSGP function. However, function can be used more generically for diverse plots showing many dimensions of data simultaneously.

Value

Function creates a bubble chart and writes the result as a PDF to bubble_plot_configs.BUBBLE_PATH.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

---

bubblePlot_Styles *bubblePlot_Styles providing various uses of the core bubblePlot function*

Description

Function includes a number of "styles" associated with bubblePlot to create bubble plots depicting a variety of relationships often of interest to stakeholders. The bubblePlot function itself is adaptable to many representations but is most often used in conjunction with results derived from summarizeSGP to represent summary level results of growth against achievement (usually, median student growth percentile against percentage at/above proficient).

Usage

```
bubblePlot_Styles(
                sgp_object,
  state,
  bPlot.years=NULL,
  bPlot.content_areas=NULL,
  bPlot.districts=NULL,
  bPlot.schools=NULL,
  bPlot.instructors=NULL,
  bPlot.styles=c(1),
  bPlot.levels=NULL,
  bPlot.level.cuts=NULL,
  bPlot.full.academic.year=TRUE,
  bPlot.minimum.n=10,
  bPlot.anonymize=FALSE,
  bPlot.prior.achievement=TRUE,
  bPlot.draft=FALSE,
  bPlot.demo=FALSE,
  bPlot.output="PDF",
  bPlot.format="print",
  bPlot.folder="Visualizations/bubblePlots")
```

Arguments

sgp_object	An object of class SGP containing long formatted data in the @Data slot that will be used for the production of student growth and achievement plots and system growth and achievement plots, summary data from summarizeSGP in the @Summary slot for bubble plots.
state	Acronym indicating state associated with the summaries for access to assessment program information embedded in SGPstateData.

`bPlot.years` A vector indicating year(s) in which to produce bubblePlots using data available in `sgp_object`. If missing the function will use the last year available in the data to produce bubblePlots.

`bPlot.content_areas`
A vector indicating content area(s) to produce bubblePlots using data available in `sgp_object`. If missing the function will produce plots for all available content areas provided in the data.

`bPlot.districts`
A vector indicating districts to produce bubblePlots using data available in `sgp_object`. If missing the function will produce plots for all available districts provided in the data where districts represent a relevant unit to be represented by the specific `bubblePlot` style.

`bPlot.schools` A vector indicating schools to produce bubblePlots using data available in `sgp_object`. If missing the function will produce plots for all available schools provided in the data where districts represent a relevant unit to be represented by the specific `bubblePlot` style.

`bPlot.instructors`
A vector indicating instructors to produce bubblePlots using data available in `sgp_object`. If missing the function will produce plots for all available instructors provided in the data where schools and districts represent relevant units to be represented by the specific `bubblePlot` style.

`bPlot.styles` A vector of positive integers indicating the types of bubblePlots to produce using data available in `sgp_object`. See associated documentation for example plots.

`bPlot.levels` A character vector of levels to be used to color bubbles in the bubblePlot. See associated documentation for example plots.

`bPlot.level.cuts`
A vector of cuts to be used to distinguish levels used to color bubbles in the bubblePlot. See associated documentation for example plots.

`bPlot.full.academic.year`
A Boolean argument (defaults to TRUE) indicating whether bubblePlots should use full academic year results if available.

`bPlot.minimum.n`
A positive integer (defaults to 10) indicating the minimum size for summary values to be displayed in the bubblePlots.

`bPlot.anonymize`
A Boolean argument (defaults to FALSE) indicating whether to anonymize bubblePlots school and district names that appear in the plots and data tips of the plots. For student level anonymization, the function utilizes the randomNames package to produce gender and ethnic correct names based upon gender and ethnicity codes available in `@Data`.

`bPlot.prior.achievement`
A Boolean argument (defaults to TRUE) indicating whether to produce bubblePlots using prior achievement as well as current achievement as the vertical dimension of the bubblePlot.

`bPlot.draft` A Boolean argument (defaults to FALSE) indicating whether to put an indicator on the chart noting that the results are draft and to not distribute.

bPlot.demo	A Boolean argument (defaults to FALSE) indicating whether to produce demo student level plots (styles 150 and/or 153) for instructors.
bPlot.output	Argument indicating the desired type of output format for bubble plots. Either 'PDF' (default) or 'PNG'.
bPlot.format	Either "print" or "presentation" indicating whether to optimize the plot for print form (light background) or presentation form (dark background).
bPlot.folder	Character vector indicating where bubblePlots should be placed. Default folder is "Visualizations/bubblePlots".

Details

bubblePlot_Styles is an omnibus function containing a number of applications of the bubblePlot function with data output from SGP analyses. As added styles are constructed they will be added to the function allowing user to construct plots of their liking. User wishing to participate and provide or suggest specific styles of their own should contact the package maintainer. Styles representing summary level data (e.g., growth and achievement for schools) are currently assigned numbers from 1 to 99 and styles representing individual level data (e.g., growth and achievement for students within a single grade) are currently assigned numbers from 100 to 199.

1:	Growth and achievement bubblePlots for the entire state with bubbles representing schools.
100:	Growth and achievement bubblePlots by grade for students in selected schools and districts.

Value

Function produces *numerous* PDF bubble charts in the styles specified by the function.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

---

capwords — *Function for converting all caps to mixed case. Useful in data cleaning.*

Description

The function capwords converts characters to mixed case character as intelligently as possible and leading/trailing spaces.

Usage

```
capwords(x,
   special.words = c("ELA","I", "II", "III", "IV",  "CCSD", "CUSD", "CUD", "USD", "PSD",
         "UD", "ESD", "DCYF", "EMH", "HS", "MS", "ES", "SES", "IEP", "ELL", "MAD",
          "PARCC", "SBAC", "SD", "SWD", "US", "SGP", "SIMEX", "SS", "SAT", "PSAT",
            "WIDA", "ACCESS", "WIDA-ACCESS"))
```

Arguments

x	A character string to be converted to mixed case.
special.words	A character vector (see default above), specifying words to not convert to mixed case.

Value

Returns a mixed case character string.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

Examples

```
capwords("TEST") ## Test
capwords("TEST1 TEST2") ## Test1 Test2
capwords("O'NEIL") ## O'Neil
capwords("JOHN'S") ## John's

## Use sapply for converting character vectors

test.vector <- paste("TEST", 1:10, sep="")
sapply(test.vector, capwords)


## With factors, convert levels instead of the entire vector

test.factor <- factor(paste("TEST", rep(letters[1:10], each=50)))
levels(test.factor) <- sapply(levels(test.factor), capwords)
levels(test.factor)
```

---

combineSGP *Combine student data and SGP results*

Description

Utility function that merges student long data in the @Data slot with results from student growth percentiles and/or student growth projections calculations. Default values of this function are designed to be used following use of other utility functions: prepareSGP and analyzeSGP. Function is integrated with cutscores embedded in SGPstateData to calculate growth-to-standard SGP targets and their associated scale scores with catch-up/keep-up to proficient status and/or move-up/stay-up to advanced status as well as the scale scores associated with these targets.

Usage

```
combineSGP(sgp_object,
 state=NULL,
 years=NULL,
 content_areas=NULL,
 sgp.percentiles=TRUE,
        sgp.percentiles.baseline=TRUE,
 sgp.projections=TRUE,
 sgp.projections.baseline=TRUE,
 sgp.projections.lagged=TRUE,
        sgp.projections.lagged.baseline=TRUE,
 sgp.target.scale.scores=FALSE,
 sgp.target.scale.scores.only=FALSE,
    sgp.target.scale.scores.merge=FALSE,
 sgp.target.content_areas=NULL,
 max.sgp.target.years.forward=3,
 update.all.years=FALSE,
 sgp.config=NULL,
 sgp.percentiles.equated=NULL,
 SGPt=NULL,
  fix.duplicates=NULL,
 parallel.config=NULL)
```

Arguments

sgp_object	An object of class SGP containing slots @Data (from prepareSGP) and @SGP (from analyzeSGP).
state	Acronym for which state is to be used for the lagged projections and growth to standard analyses. Function will try to guess state name from passed sgp_object is missing.
years	A vector of years indicating years of data to merge with @Data. If missing, merge will use all available years of student growth percentile results.
content_areas	A vector of content areas indicating content areas of student growth percentile data to merge with @Data. If missing, merge will use all available content areas of student growth percentile results.
sgp.percentiles	A Boolean variable indicating whether to combine student growth percentiles.
sgp.percentiles.baseline	A Boolean variable indicating whether to combine baseline student growth percentiles.
sgp.projections	A Boolean variable indicating whether to combine current year student growth projections and calculate catch-up/keep-up and move-up/stay-up values.
sgp.projections.baseline	A Boolean variable indicating whether to combine current year baseline student growth projections and calculate catch-up/keep-up and move-up/stay-up values.

`sgp.projections.lagged`
: A Boolean variable indicating whether to combine lagged student growth projections and calculate catch-up/keep-up and move-up/stay-up values.

`sgp.projections.lagged.baseline`
: A Boolean variable indicating whether to combine lagged baseline student growth projections and calculate catch-up/keep-up move-up/stay-up values.

`sgp.target.scale.scores`
: A Boolean variable indicating whether scale scores from calculated SGP targets should be calculated.

`sgp.target.scale.scores.only`
: A Boolean variable indicating whether combineSGP should skip merging and only calculate scale scores from calculated SGP targets. Default is FALSE.

`sgp.target.scale.scores.merge`
: An argument (defaults to FALSE) indicating whether & how target scale scores should be merged with the LONG data in the @Data slot.

`sgp.target.content_areas`
: A Boolean variable indicating whether content area associated with SGP targets should be calculated.

`max.sgp.target.years.forward`
: A integer indicating the number of years forward from the lagged (last year's) score to project forward for growth to standard calculations. Default is 3 years from the present, 4 years from the lagged year, which is the standard in most growth to standard calculations used by state departments of education.

`update.all.years`
: A Boolean argument defaulting to FALSE indicating whether `combineSGP` should delete previously merged variables calculated in `analyzeSGP` and re-merge all available data.

`sgp.config`
: Argument (defaults to NULL) passed utilized only for target scale score calculation. If targets for end of course tests are required, user must specify configurations directly. See code from `testSGP` number 3 for an example.

`sgp.percentiles.equated`
: Boolean variable indicating whether equated percentiles are being merged (defaults to NULL which is converted to FALSE unless otherwise explicitly declared. If scale score targets are being calculated, linkages will be passed to scale score targets for calculation.

`SGPt`
: Argument used to pass SGPt configuration to `getTargetScaleScore` for `studentGrowthProjections` calculation of scale score targets.

`fix.duplicates`
: Argument to control how duplicate records based upon the key of VALID_CASE, CONTENT_AREA, YEAR, and ID are dealt with. If set to 'KEEP.ALL', the function tries to fix the duplicate individual records. If present and a '_DUP_***' suffix has been added to the duplicate ID before running `studentGrowthPercentiles`, then the `@Data` slot will be extended as necessary to accomodate additional student records and SGP results.

`parallel.config`
: Parallel configuration only used when 'sgp.target.scale.scores' is set to TRUE. Default is NULL consistent with no targets being calculated. To utilize parallel processing in the calculation of SGP scale score targets user must specify a

list designating a backend (e.g, BACKEND='PARALLEL') and a number of workers (e.g., 'WORKERS=list(SGP_SCALE_SCORE_TARGETS=4)').

Value

Function returns a list containing the input long data set in the `@Data` slot as a `data.table` keyed using `VALID_CASE`, `CONTENT_AREA`, `YEAR`, `ID` merged with student growth percentiles and/or straight/lagged projection targets and catch-up/keep-up and move-up/stay-up status with, if requested, the scale scores associated with such targets.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

See Also

prepareSGP, analyzeSGP

Examples

```
## Not run:
## combineSGP is Step 3 of 5 of abcSGP
Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- prepareSGP(Demonstration_SGP)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP)
Demonstration_SGP <- combineSGP(Demonstration_SGP)

## End(Not run)
```

---

courseProgressionSGP *Identify potential course progressions for SGP analyses*

Description

Utility function used to analyze supplied long data or an existing SGP class object to identify potential course progressions suitable for analysis with analyzeSGP, studentGrowthPercentiles, etc. See examples for more information.

Usage

```
courseProgressionSGP(
 sgp_object,
 lag.direction=c("FORWARD", "BACKWARD"),
 year)
```

Arguments

sgp_object	Either a panel data set in long form or an object of class SGP. See embedded sgpData_LONG data set for an exemplar.
lag.direction	Character string indicating whether the progressions should be produced prospectively or retrospectively.
year	Character indicating the value of YEAR that is the focus of analysis.

Value

Function returns a nested list class object. The final node of each nested list is a data.table which summarizes the number of students with a particular GRADE by CONTENT_AREA by YEAR course progression

Author(s)

Adam Van Iwaarden <avaniwaarden@nciea.org> and Damian W. Betebenner <dbetebenner@nciea.org>

See Also

sgpData_LONG

Examples

```
## Not run:
## Run courseProgressionSGP on the subset of the long data that contains
## ONLY mathematics related records (would realistically also contain EOCT math courses)
Math_Data <- subset(SGPdata::sgpData_LONG, CONTENT_AREA == "MATHEMATICS")
Math_Progressions <- courseProgressionSGP(Math_Data, lag.direction= "BACKWARD", year="2015_2016")

##  Examine results for Adcademic Year 2015-2016, 5th grade Mathematics.
Math_Progressions[['BACKWARD']][['2015_2016']][['MATHEMATICS.05']]
Math_Progressions[['BACKWARD']][['2015_2016']][['MATHEMATICS.05']][COUNT>100]

## End(Not run)
```

---

createKnotsBoundaries *Function to create Knots and Boundaries from supplied data in LONG format.*

Description

The function createKnotsBoundaries creates Knots, Boundaries and Loss/Hoss scores for subsequent use and embedding in SGPstateData. Function can be called separately but is usually called as part of prepareSGP. See examples below.

Usage

```
createKnotsBoundaries(tmp.data,
  knot.cut.percentiles=c(0.2,0.4,0.6,0.8))
```

Arguments

tmp.data	Data supplied to function in LONG format. See sgpData_LONG for an exemplar. tmp.data must contain, at least, variables 'VALID_CASE', 'CONTENT_AREA', 'SCALE_SCORE', 'GRADE'.
knot.cut.percentiles	A numeric vector of quantiles of 'SCALE_SCORE' to be used as the cut points. Default is to use the 20th, 40th, 60th, and 80th percentiles (i.e., c(0.2,0.4,0.6,0.8)).

Value

Returns a list containing Knots, Boundaries and Loss/Hoss scores.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

Examples

```
## Not run:
### Run on supplied long data

DEMO_Knots_Boundaries <- createKnotsBoundaries(sgpData_LONG)

### Run as part of prepareSGP

### First NULL out knots and boundaries embedded in SGPstateData

SGPstateData[["DEMO"]][["Achievement"]][["Knots_Boundaries"]] <- NULL
Demonstration_SGP <- prepareSGP(sgpData_LONG)

## End(Not run)
```

---

getStateAbbreviation *Function for converting state/organization abbreviations to long form and back.*

Description

The function getStateAbbreviation converts state/organization abbreviations to their long form and back. For example ID gets converted to Idaho.

Usage

```
getStateAbbreviation(
                    supplied.name,
                  SGPfunction=NULL,
                  type="ABBREVIATION")
```

Arguments

supplied.name	A character string state/organization abbreviation or long form name.
SGPfunction	SGP package function from which getStateAbbreviation is being called. Defaults to NULL.
type	Either Abbreviation or Long indicating whether the desired output is an Abbreviation (long form name is supplied) or the long form name (abbreviation is supplied). The default is Abbreviation.

Value

Returns a character string.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

Examples

```
getStateAbbreviation("IDAHO") ## ID
getStateAbbreviation("ID", type="Long") ## Idaho
```

---

gofPrint *Function to produce Goodness of Fit plots from a SGP object.*

Description

The function gofPrint is used to output Goodness of Fit plots from R grobs stored in a SGP object.

Usage

```
gofPrint(
  sgp_object,
  years = NULL,
  content_areas = NULL,
  grades = NULL,
  sgp_types = NULL,
  norm_group = NULL,
  output_format = c("PDF", "SVG", "DECILE_TABLES"),
  output_path = "Goodness_of_Fit",
  ...)
```

Arguments

sgp_object	SGP object resulting from SGP analyses
years	The academic year(s) for which fit plots should be produced
content_areas	The content area(s) for which fit plots should be produced
grades	The grade(s) for which fit plots should be produced
sgp_types	The SGP type for which fit plots should be produdced (e.g., cohort or baseline referenced, SIMEX adjusted, etc.).
norm_group	The specific norm group (cohort or course progression) for which fit plots should be produced
output_format	The graphical file type of plots that should be produced and/or the decile tables as a .Rdata file. Defaults are PDF, SVG and decile tables. PNG file type is also available.
output_path	Directory path for plot/table output
...	Additional arguments to pass to svglite

Value

Returns a graphical output of the fit plot(s) requested based on the arguments provided. Can also save the decile table data in Rdata format.

Author(s)

Adam Van Iwaarden <avaniwaarden@nciea.org> and Damian W. Betebenner <dbetebenner@nciea.org>

Examples

```
  ## Not run:
    gofPrint(Demonstration_SGP)

## End(Not run)
```

---

gofSGP *Function for producing goodness of fit plots using existing SGP object*

Description

gofSGP creates goodness-of-fit plots in either PDF or PNG for showing SGP distribution by prior achievement level and prior scale score decile. These plots expand upon the plots currently produced with the studentGrowthPercentiles function.

Usage

```
gofSGP(
 sgp_object,
 state=NULL,
 years=NULL,
 content_areas=NULL,
 content_areas_prior=NULL,
 grades=NULL,
 ceiling.floor=TRUE,
 use.sgp="SGP",
 output.format="PDF",
 color.scale="reds.and.blues")
```

Arguments

sgp_object	The SGP object from which the goodness-of-fit data will be used.
state	The 'state' for the sgp_object. Derive from sgp_object name if not explicitly supplied.
years	The years that goodness-of-fit plots are requested. Default is to use all years available.
content_areas	The content area(s) that goodness-of-fit plots are requested. Default is to use all content areas available.
content_areas_prior	The content area(s) of the prior year which growth by achievement level is being produced that goodness-of-fit plots are requested. Default is to use all content areas available.
grades	The grade(s) that goodness-of-fit plots are requested. Default is to use all grade available.
ceiling.floor	Boolean variable to explictly control display of ceiling/floor portion of goodness of fit plot.
use.sgp	Character vectors (defaults to 'SGP') indicating what student growth percentile variable to calculate goodness-of-fit plots for.
output.format	Character vectors (defaults to 'PDF') indicating what driver to use to output results. Options currently include 'PDF', 'PNG' and 'SVG'. 'SVG' is currently experimental.
color.scale	Character vectors (defaults to 'red') indicating what color palette to use for creating percentile distribution table.

Value

Returns output ('PDF' , 'PNG' or 'SVG') associated with goodness-of-fit analyses.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

See Also

studentGrowthPercentiles

Examples

```
## Not run:
Demonstration_SGP <- abcSGP(sgpData_LONG)
gofSGP(Demonstration_SGP)

## End(Not run)
```

---

growthAchievementPlot *growthAchievementPlot for SGP*

Description

Function to create growth and achievement plots depicting system level results associated with student growth percentile results. The charts show, simultaneously, norm- and criterion-referenced student achievement (i.e., status) as well as norm- and criterion-referenced student growth. These charts are those shown on the cover of the December 2009 Issue of *Educational Measurement: Issues and Practice*. See Betebenner (2009) and Betebenner (2012) for more details

Usage

```
growthAchievementPlot(
    gaPlot.sgp_object,
    gaPlot.students=NULL,
    gaPlot.percentile_trajectories,
    gaPlot.achievement_percentiles=c(.01, seq(.05, .95, by=.05), .99),
    gaPlot.show.scale.transformations=TRUE,
    gaPlot.grade_range,
    gaPlot.max.order.for.progression=NULL,
 gaPlot.start.points="Achievement Level Cuts",
    gaPlot.back.extrapolated.cuts=NULL,
 gaPlot.subtitle = TRUE,
    gaPlot.SGPt=NULL,
    state,
    content_area,
    year,
    format="print",
    baseline=FALSE,
    equated=NULL,
 output.format="PDF",
    output.folder,
    assessment.name)
```

Arguments

`gaPlot.sgp_object`
: The sgp_object containing system information for constructing the growth and achievement plot. Object is calculated using `abcSGP` or (at least) `prepareSGP` followed by `analyzeSGP`. The function requires coefficient matrices in order to display percentile growth trajectories.

`gaPlot.students`
: Either NULL (the default) or a list of student IDs for whom one wishes to generate growth and achievement plots.

`gaPlot.percentile_trajectories`
: A vector indicating the growth percentile trajectories to be depicted on the plot. If missing, the percentile trajectories will be the trajectories associated with the state supplied. If no state is supplied, the percentile trajectories will be 10, 35, 50, 65, 90.

`gaPlot.achievement_percentiles`
: A vector of percentiles that achievement (i.e., status) percentiles will be depicted across the range of grades.

`gaPlot.show.scale.transformations`
: A Boolean arguments (defaults to TRUE) indicating whether to show the scale as a vertical axis if a scale transformation is applied to the supplied data.

`gaPlot.grade_range`
: The grade range for which to demonstrate plot. If missing, function uses supplied state to derive grade range.

`gaPlot.max.order.for.progression`
: The maximum coefficient matrix order to use for each progression. Default is NULL which utilizes the maximum order available with the coefficient matrices.

`gaPlot.start.points`
: Either 'Achievement Level Cuts' or 'Achievement Percentiles' defining where the growth percentiles trajectories will start in the growth achievement plots.

`gaPlot.back.extrapolated.cuts`
: Scale score in final year for which nth percentile growth culminates from back-extrapolated scale scores.

`gaPlot.subtitle`
: Boolean variable (defaults to TRUE) indicating whether subtitle is printed on the growth achievement chart indicating student starting point.

`gaPlot.SGPt`
: A boolean variable indicating whether time dependent coefficient matrices (SGPt) are to be used in student growth projection calculation.

`state`
: A two letter acronym for the state associated with the supplied data.

`content_area`
: A character vector indicating the content area to produce the growth and achievement chart. Note that the supplied content area must match that used in the internal labeling of content area for the sgp_object.

`year`
: The year for which to produce the growth and achievement plots.

`format`
: Either "print" (the default) for light background or "presentation" for dark background slides.

baseline	Boolean variable (defaults to FALSE) indicating whether to create percentile trajectories using baseline referenced coefficient matrices.
equated	Boolean variable (defaults to FALSE) indicating whether to utilize linked matrices to generate growth and achievement plots.
output.format	A character vector indicating which output format to use. Currently support 'PDF' (the default) and 'PNG'. The call to growthAchievementPlot in visualizeSGP outputs in both formats.
output.folder	A character vector indicating where to put the produced growth and achievement plot.
assessment.name	A character vector indicating the assessment.name. If missing, the supplied state is used to determine the assessment.name.

Details

Typical use of the function is as part of visualizeSGP function. However, function can be used to produce growth and achievement charts.

Value

Function creates a growth and achievement chart and writes the result as a PDF to output.folder.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

References

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.

Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. Educational Measurement: Issues and Practice, 28(4):42-51.

---

outputSGP *Output student data and SGP results for a variety of purposes*

Description

Utility function used to export student data and SGP results for a variety of purposes. Current functionality exports data in wide format for data visualization purposes. See source code for detailed functionality.

Usage

```
outputSGP(sgp_object,
 state=NULL,
        output.type=c("LONG_Data", "LONG_FINAL_YEAR_Data", "WIDE_Data",
   "INSTRUCTOR_Data"),
        baseline.sgps=FALSE,
        outputSGP_SUMMARY.years=NULL,
        outputSGP_SUMMARY.content_areas=NULL,
        outputSGP_INDIVIDUAL.years=NULL,
        outputSGP_INDIVIDUAL.content_areas=NULL,
        outputSGP.anonymize=FALSE,
        outputSGP.student.groups=NULL,
        outputSGP.directory="Data",
 outputSGP.translate.names=TRUE,
 outputSGP.projection.years.for.target=3,
 outputSGP.pass.through.variables=NULL)
```

Arguments

sgp_object	An object of class SGP containing data to be exported.
state	Acronym for which state is to be used for the lagged projections and growth to standard analyses. Function will try to guess state name from passed sgp_object is missing.
years	A vector indicating the years to be included in the output. Default is to use all years.
content_areas	A vector indicating the content areas to be included in the output. Default is to use all content areas.
output.type	A character vector indicating what output type is requested. Currently LONG_Data, WIDE_Data, INSTRUCTOR_Data, SchoolView, RLI are supported modes of output. LONG_Data exports the contents of the @Data slot in a pipe delimited format. LONG_FINAL_YEAR_Data exports the contents of the last year of the @Data slot in a pipe delimited format. WIDE_Data exports a reshaped version of the @Data slot where each row is a unique student record. INSTRUCTOR_Data uses the @Data_Supplementary$INSTRUCTOR_NUMBER table to export a long student by instructor number table. SchoolView exports tables used for representation in SchoolView. RLI exports used for representation by RLI. The default exports LONG_Data, LONG_FINAL_YEAR_Data, WIDE_Data and INSTRUCTOR_Data.
baseline.sgps	Boolean vector indicating whether to output baseline SGPs for cohort referenced SGPs.
outputSGP_SUMMARY.years	A character vector indicating the year to be used for output file construction for summary tables.
outputSGP_SUMMARY.content_areas	A character vector indicating the content areas to be used for output file construction for summary tables.

outputSGP_INDIVIDUAL.years
: A character vector indicating the year to be used for output file construction for individual level file.

outputSGP_INDIVIDUAL.content_areas
: A character vector indicating the content areas to be used for output file construction for individual level file.

outputSGP.anonymize
: A Boolean variable indicating whether to anonymize output files.

outputSGP.student.groups
: A list of variables to be used for student groups in individual and summary tables.

outputSGP.directory
: A a file path indicating where to save output files. Defaults to Data.

outputSGP.translate.names
: A Boolean argument, defaults to TRUE, indicating whether data output should refer to 'names.provided' or 'names.sgp' in @Names slot of supplied SGP object. This argument allows for the the conversion of variable naming conventions from the SGP package back to that used by the state/organization.

outputSGP.projection.years.for.target
: An integer argument indicating what projection to supply with regard to the number of years projected forward.

outputSGP.pass.through.variables
: A character vector of variables in @Data that are to be merged with output in RLI output.type. Default is NULL or no.

Value

Function writes data in multiple formats including .Rdata, .txt (pipe delimited) and zipped versions of .txt.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

See Also

abcSGP, prepareSGP, analyzeSGP, combineSGP, summarizeSGP

Examples

```
## Not run:
Demonstration_SGP <- prepareSGP(sgpData_LONG)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP)
Demonstration_SGP <- combineSGP(Demonstration_SGP)
outputSGP(Demonstration_SGP)

## Output current year

outputSGP(Demonstration_SGP, output.type="LONG_FINAL_YEAR_Data")
```

```
## End(Not run)
```

---

prepareSGP *Prepare data for SGP analyses*

Description

Utility function/exemplar used to embed supplied long data into a list object as a keyed data.table. NOTE: This function also serves the purposes of running many checks on the SGP object you construct to make sure it is up to date and in the best shape possible. If you have an older object that you wish to make sure is up to date with the latest version of the SGP package, running prepareSGP on an object is never bad thing to do. See examples for more information.

Usage

```
prepareSGP(data,
 data_supplementary=NULL,
 state=NULL,
 var.names=NULL,
 create.additional.variables=TRUE,
 fix.duplicates=NULL,
  create.achievement.level=TRUE)
```

Arguments

data	A panel data set in long form or an object of class SGP. See embedded sgpData_LONG data set for an exemplar.
data_supplementary	Supplementary data (e.g., student teacher lookup tables) to be embedded in SGP object in slot @Data_Supplementary. Data must be embedded in a list. Default is no data supplied.
state	A two letter acronym indicating the state associated with the data. If not supplied, the function will try to infer what the state is from the data object name supplied.
var.names	A list or a data.frame that includes all required columns that do not match the SGP conventions, as well as all secondary columns needed for summarizing and reporting.
create.additional.variables	Boolean argument indicating whether prepareSGP should create additional variables often used in analyses. For example, the function can create a variable HIGH_NEED_STATUS identifying the top and bottom quartile of students in each school by year by content area by grade grouping.

`fix.duplicates` Argument to control how `prepareSGP` deals with duplicate records based upon the key of VALID_CASE, CONTENT_AREA, YEAR, and ID. The function currently warns of duplicate records and doesn't modify data. If set to TRUE, `prepareSGP` tries to fix the duplicate individual records by adding a '_DUP_***' suffix to the duplicate ID in order to create unique records based upon the key.

`create.achievement.level`
Boolean argument indicating whether prepareSGP should create the ACHIEVEMENT_LEVEL variable if it is missing. Defaults to TRUE.

Value

Function returns an object of class `SGP`. The long data is in the `data` slot.

Author(s)

Adam Van Iwaarden <avaniwaarden@nciea.org>, Damian W. Betebenner <dbetebenner@nciea.org>, and Ben Domingue <ben.domingue@gmail.com>

See Also

sgpData_LONG

Examples

```
## Not run:
## prepareSGP is Step 1 of 5 of abcSGP
Demonstration_SGP <- prepareSGP(sgpData_LONG)

## Running prepareSGP on an already create SGP object as part of a annual update

Demonstration_SGP <- prepareSGP(Demonstration_SGP)


## Running prepareSGP on a long data set without creating addition variables

Demonstration_SGP <- prepareSGP(sgpData_LONG, create.additional.variables=FALSE)

## End(Not run)
```

---

rliSGP *Wrapper function associated with SGP analyses for Renaissance Learning Incorporated (RLI) interim STAR assessments.*

Description

rliSGP is a wrapper function design to expedite SGP analyses on Renaissance Learning Incorporated (RLI) interim STAR assessments.

Usage

```
rliSGP(
 sgp_object,
  additional.data=NULL,
 state=NULL,
 content_areas=c("MATHEMATICS", "MATHEMATICS_SPANISH", "READING",
   "READING_SPANISH", "READING_UNIFIED", "EARLY_LITERACY",
   "EARLY_LITERACY_SPANISH"),
  testing.window,
  eow.or.update="UPDATE",
 update.save.shell.only=FALSE,
  configuration.year,
 sgp.percentiles.baseline=TRUE,
 sgp.projections.baseline=TRUE,
 sgp.projections.lagged.baseline=FALSE,
 sgp.target.scale.scores=TRUE,
  update.ids=NULL,
 SGPt=TRUE,
 simulate.sgps=FALSE,
 save.intermediate.results=FALSE,
 coefficient.matrices=NULL,
 goodness.of.fit.print=FALSE,
 return.updated.shell=FALSE,
 fix.duplicates="KEEP.ALL",
 eow.calculate.sgps=FALSE,
 score.type="RASCH",
 cutscore.file.name="Cutscores.csv",
 get.cohort.data.info=FALSE,
  use.latest.rliMatrices=TRUE,
  parallel.config=NULL)
```

Arguments

sgp_object	An object of class SGP or a data.frame from which an SGP object can be produced. If data.frame is supplied, the current window's data will be created from the last window present in the data.frame.
additional.data	Current window's data for UPDATE or EOW (end of window) processing. Can be NULL if new data all fed into argument 'sgp_object' as data.frame.
state	Abbreviation for STATE/ORGANIZATION being analyzed. Currently only available for 'RLI' or 'RLI_UK'.
content_areas	Character vector indicating which content areas rliSGP should be run for ('MATHEMATICS', 'READING', 'READING_UNIFIED', and/or 'EARLY_LITERACY'). Default is all.
testing.window	Argument indicating testing window being analyzed. Either 'FALL', 'WINTER', 'SPRING' indicating what testing window's data is being analyzed.

eow.or.update Argument indicating whether end-of-window or within-window updates are being done. Either 'EOW' (end of window) or 'UPDATE' (within window).

update.save.shell.only
Boolean argument indicating whether End of Window processing should only produce/save RLI_SGP_UPDATE_SHELL and bypass coefficient matrix production.

configuration.year
Year associated with SGP analyses (configuration year) to be used.

sgp.percentiles.baseline
Boolean variable passed to updateSGP to control whether baseline student growth percentiles are calculated. Default is TRUE.

sgp.projections.baseline
Boolean variable passed to updateSGP to control whether baseline student growth projections are calculated. Default is TRUE.

sgp.projections.lagged.baseline
Boolean variable passed to updateSGP to control whether lagged baseline student growth projections are calculated. Default is FALSE.

sgp.target.scale.scores
Boolean variable passed to updateSGP to control whether scale score targets are calculated. Default is TRUE.

update.ids Argument that allows ID to be updated based upon a supplied two column data frame where the first column represents the ORIGINAL/OLD ID and the second column the NEW ID that is to replace it. NOTE that ID needn't be replaced in either 'sgp_object' nor 'additional.data' that are provided as part of the 'rliSGP' call. Substitution will occur on entire data set before analysis. Default is NULL, no ID replacement is performed.

SGPt Argument indicating whether to perform time dependent SGP (SGPt) analyses. Current default is TRUE based upon transition to SGPt in Summer 2015.

simulate.sgps A Boolean argument indicating whether to calculate SGP standard errors based upon CSEMs provided by either an embedded CSEM-scale score lookup or an embedded variable. Default is FALSE.

save.intermediate.results
Boolean variable passed to updateSGP indicating whether to save intermediate results from prepareSGP, analyzeSGP, and combineSGP as part of the rliSGP analyses.

coefficient.matrices
List of baseline matrices to be used for analyses (overrides what exists in SGPstateData). Defaults to NULL (use those in SGPstateData).

goodness.of.fit.print
Boolean argument passed to updateSGP indicating whether to print goodness of fit plots associated with outcome. Default is FALSE.

return.updated.shell
Boolean argument (defaults to FALSE) indicating whether to return sgp_object as part of call to rliSGP.

fix.duplicates Character argument (defaults to 'KEEP.ALL') indicating how to deal with duplicates submitted as part of data for analysis. Setting 'fix.duplicates' to null sets no modification of duplicates.

eow.calculate.sgps
: Boolean argument passed to updateSGP further passed to analyzeSGP indicating whether student growth percentiles are calculated as part of EOW analyses. Default is FALSE which results in just coefficient matrices being calculated.

score.type
: Either RASCH (default) or STAR indicating what type of score is being supplied for analysis. The selection ensures the correct cutscores, matrices, and knots/boundaries are used for analyses.

cutscore.file.name
: Cutscore file name and path for input and creation of RLI Cutscores. Default is Cutscores.csv.

get.cohort.data.info
: Boolean argument passed to analyzeSGP indicating whether to calculate norm group cohort counts.

use.latest.rliMatrices
: Boolean argument indicating whether to ensure use of latest version of RLImatrices (defaults to TRUE).

parallel.config
: Argument passed to abcSGP or updateSGP to activate parallel processing with an appropriate backend.

Value

Saves relevant results to disc for use in subsequent analyses.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

Examples

```
## Not run:
####### For Within Window processing

testSGP('RLI')

####### For End-of-Window processing



## End(Not run)
```

---

setNamesSGP *Function for renaming data (typically from outputSGP) from a state/organization naming conventions to those used in the SGP package*

Description

setNamesSGP renames a dataset (without copying or returning the data).

Usage

```
setNamesSGP(
 data,
 state=NULL)
```

Arguments

data	The dataset to be renamed. For example longitudinal data exported from outputSGP that uses a state's naming conventions.
state	The 'state' for the data Derive from data name if not explicitly supplied.

Value

NULL. Simply renames the data. Note that the state must be included in the SGPstateData and have a Variable_Name_Lookup entry to work.

Author(s)

Adam R. Van Iwaarden <avaniwaarden@nciea.org>

See Also

outputSGP

Examples

```
## Not run:
load("Data/Demonstration_SGP_LONG_Data.Rdata")
setNamesSGP(Demonstration_SGP_LONG_Data)

## End(Not run)
```

SGP-class *Class "SGP"*

Description

The formal S4 class for SGP. This class stores the data object for use with the functions analyzeSGP, combineSGP, summarizeSGP, and visualizeSGP. The SGP class contains and organizes all the results associated with SGP analyses. is.SGP tests for membership for this class.

Details

list.null: combines class list and class NULL

Usage

Objects can be created by calls of the form `new("SGP", ...)`, but this is not encouraged. To instantiate a new instance of SGP class use the function `prepareSGP` instead.

```
is.SGP(x)
```

Slots

`Data`: A data.table including student-level data in a (long) format. For annual testing, each VALID_CASE, CONTENT_AREA, YEAR, ID combination represents a unique case in the data. For instances with multiple tests within a year, each VALID_CASE, CONTENT_AREA, YEAR, ID, WITHIN_YEAR combination represents a unique case in the data. See sgpData_LONG for an exemplar data set

`Data_Supplementary`: A list (possibly NULL) providing additional data.tables containing student level multiple-membership lookup tables. For example, sgpData_INSTRUCTOR_NUMBER provides student teacher linkages and can be embedded in this slot using a list that contains it.

`Names`: A data.frame with five columns: 'names.provided', 'names.sgp', 'names.type', 'names.info', 'names.output'. This data.frame is used as a lookup table to translate state specific variable names to SGP variable names as well as provide information for summarizeSGP on the types of summary tables to produce.

`SGP`: A list including the output from `analyzeSGP`

`Summary`: A list including the output from `summarizeSGP`

`Version`: A list of meta-data including the version of the SGP package used to construct the SGP object and the date the object was created.

Author(s)

Jonathan P. Weeks <weeksjp@gmail.com>, Adam Van Iwaarden <avaniwaarden@nciea.org> and Damian W. Betebenner <dbetebenner@nciea.org>

See Also

`prepareSGP`

`SGPstateData` *State assessment program data from large scale state assessments for use with SGP package*

Description

An environment (an object of class environment) containing information on state assessment programs, organized by state. Currently the environment contains achievement level cutscores and labels for the state assessments, assessment name and abbreviation, growth cutscores and labels, information on vertical scaling, conditional standard errors of measurement (CSEMs), knots and boundaries, and state specific configurations currently being used for SGP analyses at the agency (state, district, national) level. The cutscores, in particular, are used to calculate growth-to-standard/projection

values. States currently included in the data set are Arizona (AZ), Arkansas (AR), California (CA), Colorado (CO), Connecticut (CT), Georgia (GA), Hawaii (HI), Idaho (ID), Indiana (IN), Kansas (KS), Maine (ME), Massachusetts (MA), Minnesota (MN), Mississippi (MS), Missouri (MO), Nebraska (NE), Nevada (NV), New Hampshire (NH), New Mexico (NM), New Jersey (NJ), New York (NY), Oregon (OR), Rhode Island (RI), South Dakota (SD), Utah (UT), Vermont (VT), Virginia (VA), West Virginia (WV), Wisconsin (WI), Archdioces of Baltimore (AOB), Colorado English Language Assessment (CELA), Demonstration (DEMO), Albuquerque (ABQ), Australia NAPLAN (NAPLAN), Guatemala (GUA), Renaissance Learning Incorporated (RLI), Renaissance Learning Incorporated UK (RLI_UK), and New Haven (NEW_HAVEN).

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

Source

State assessment data and technical assessment documentation

`splineMatrix-class` *Class "splineMatrix"*

Description

The formal S4 class for coefficient matrices produced from the `studentGrowthPercentiles` function. This class stores the B-spline knots and boundaries used by the coefficient matrix object for the production of student growth percentiles and projections.

Details

splineMatrix: This class contains the S3 `matrix` class, inheriting its methods. The slot `Knots` should be one or more lists of numeric vector(s) used in the internal call to `bs`, which generates a B-spline basis matrix from student scores. There are typically with 4 values for the knots. Similarly, `Boundaries` are used in `bs` for the `Boundary.knots` argument. This is always two values which are at or slightly beyond the lowest and highest observed student scores. `Content_Areas` and `Grade_Progression` provide information about the data (sub)set used to produce the matrix.

Objects from the Class

Objects can be created by calls of the form `new("splineMatrix", ...)`, but this is not encouraged. Previously produced coefficient matrices MUST be bound to the IDENTICAL knots and boundaries used to create them. Use the function `studentGrowthPercentiles` instead.

Slots

`.Data`: A coefficient matrix derived from `studentGrowthPercentiles`.

`Knots`: A list(s) of numeric values used as the `knots` to generate the B-spline basis matrix in `studentGrowthPercentiles`.

`Boundaries`: A list(s) of numeric values used as the `Boundary.knots`to generate the B-spline basis matrix in `studentGrowthPercentiles`.

`Content_Areas`: A list of time dependent content area names included in the data used to produce the coefficient matrix.

`Grade_Progression`: A list of the time dependent grades included in the data used to produce matrices.

`Time`: A list of the Times (e.g., years) measurements occurred included in the data used to produce matrices.

`Time_Lags`: A list of the time lags/differences between Time (e.g., years) included in the data used to produce matrices.

`Version`: A list including the version of the SGP package used to construct the splineMatrix object and the date the object was created.

Author(s)

Adam Van Iwaarden <avaniwaarden@nciea.org>, Ben Domingue <ben.domingue@gmail.com> and Damian W. Betebenner <dbetebenner@nciea.org>

See Also

`studentGrowthPercentiles`

`studentGrowthPercentiles`

Student Growth Percentiles

Description

Function to calculate student growth percentiles using large scale assessment data. Outputs growth percentiles for each student and supplies various options as function arguments. Results from this function are utilized to calculate percentile growth projections/trajectories using the `studentGrowthProjections` function.

Usage

```
studentGrowthPercentiles(panel.data,
                         sgp.labels,
                         panel.data.vnames=NULL,
                         additional.vnames.to.return=NULL,
                         grade.progression,
                         content_area.progression,
```

```
  year.progression,
  year_lags.progression,
  num.prior,
  max.order.for.percentile=NULL,
  return.additional.max.order.sgp=NULL,
  subset.grade,
  percentile.cuts,
  growth.levels,
  use.my.knots.boundaries,
  use.my.coefficient.matrices,
  calculate.confidence.intervals,
  print.other.gp=FALSE,
  print.sgp.order=FALSE,
  calculate.sgps=TRUE,
  rq.method="br",
  rq.method.for.large.n="fn",
  max.n.for.coefficient.matrices=NULL,
  knot.cut.percentiles=c(0.2,0.4,0.6,0.8),
  knots.boundaries.by.panel=FALSE,
  exact.grade.progression.sequence=FALSE,
  drop.nonsequential.grade.progression.variables=TRUE,
  convert.0and100=TRUE,
  sgp.quantiles="Percentiles",
  sgp.quantiles.labels=NULL,
  sgp.loss.hoss.adjustment=NULL,
  sgp.cohort.size=NULL,
  sgp.less.than.sgp.cohort.size.return=NULL,
  sgp.test.cohort.size=NULL,
  percuts.digits=0L,
  isotonize=TRUE,
  convert.using.loss.hoss=TRUE,
  goodness.of.fit=TRUE,
  goodness.of.fit.minimum.n=NULL,
  goodness.of.fit.output.format="GROB",
  return.prior.scale.score=TRUE,
  return.prior.scale.score.standardized=TRUE,
  return.norm.group.identifier=TRUE,
  return.norm.group.scale.scores=NULL,
  return.norm.group.dates=NULL,
  return.norm.group.preference=NULL,
return.panel.data=identical(parent.frame(), .GlobalEnv),
  print.time.taken=TRUE,
  parallel.config=NULL,
  calculate.simex=NULL,
  sgp.percentiles.set.seed=314159,
  sgp.percentiles.equated=NULL,
  SGPt=NULL,
  SGPt.max.time=NULL,
```

```
verbose.output=FALSE)
```

Arguments

panel.data	REQUIRED. Object of class list, data.frame, or matrix containing longitudinal student data in wide format. If supplied as part of a list, data should be contained in panel.data$Panel_Data. Data must be formatted so that student ID is the first variable/column, student grade/time variables for each time period, from earliest to most recent, are the next variables/columns, and student scale score variables for each year, from earliest to latest, are the remaining variables/columns. See sgpData for an exemplar data set. NOTE: The column position of the variables IS IMPORTANT, NOT the names of the variables.
sgp.labels	REQUIRED. A list, sgp.labels, of the form list(my.year= , my.subject=) or list(my.year= , my.subject= , my.extra.label). The user-specified values are used to save the student growth percentiles, coefficient matrices, knots/boundaries, and goodness of fit results in an orderly fashion using an appropriate combination of year & subject & grade. Except in special circumstances, supplying my.year and my.subject are sufficient to uniquely label derivative output.
panel.data.vnames	Vector of variables to use in student growth percentile calculations. If not specified, function attempts to use all available variables.
additional.vnames.to.return	A list of the form list(VARIABLE_NAME_SUPPLIED=VARIABLE_NAME_TO_BE_RETURNED) indicating data to be returned with results from studentGrowthPercentiles analyses.
grade.progression	Preferred argument to specify a student grade/time progression in the data. For example, 3:4 would indicate to subset the data where the two most recent grades for which data are available are 3 and 4, respectively. The argument allows for non-sequential grade progressions to be analyzed with automatic removal of columns where "holes" occur in the supplied grade.progression. For example, for the grade.progression c(7,8,10), the penultimate GRADE and SCALE_SCORE column in the supplied panel.data would be removed. The argument can also be combined with an appropriate panel.data.vnames argument to remove a year of data would analyze students progressing from 7 to 8 to 10.
content_area.progression	Character vector of content area names of same length as grade.progression to be provided if not all identical to 'my.subject' in sgp.labels list. Vector will be used to populate the @Content_Areas slot of the splineMatrix class coefficient matrices. If missing, 'sgp.labels$my.subject' is repeated in a vector length equal to grade.progression.
year.progression	Character vector of years associated with grade and content area progressions. If missing then the year.progression is assumed to end in 'my.year' provided in sgp.labels and be of the same length as grade.progression. Vector will be used to populate the @Years slot of the splineMatrix class coefficient matrices.

`year_lags.progression`
: A numeric vector indicating the time lags/span between observations in the columns supplied to `studentGrowthPercentiles`. The default, NULL, allows the function to calculate the lags/differences based upon the supplied years.

`num.prior`
: Number of prior scores one wishes to use in the analysis. Defaults to `num.panels-1`. If `num.prior=1`, then only 1st order growth percentiles are computed, if `num.prior=2`, then 1st and 2nd order are computed, if `num.prior=3`, 1st, 2nd, and 3rd ... NOTE: specifying `num.prior` is necessary in some situations (in early grades for example) where the number of prior data points is small compared to the number of panels of data.

`max.order.for.percentile`
: A positive integer indicating the maximum order for percentiles desired. Similar limiting of number of priors used can be accomplished using the `grade.progression` argument.

`return.additional.max.order.sgp`
: A positive integer (defaults to NULL) indicating the order of an additional SGP to be returned: `SGP_MAX_ORDER_N`.

`subset.grade`
: Student grade level for sub-setting. If the data fed into the function contains multiple grades, setting `subset.grade=5` selects out those students in grade five in the most recent year of the data. If no sub-setting is desired, argument do not include the `subset.grade` argument. If `grade.progression` is supplied, then a subset grade is implicitly specified.

`percentile.cuts`
: Additional percentile cuts (supplied as a vector) between 1 and 99 associated with each student's conditional distribution. Default is to provide NO growth percentile cuts (scale scores associated with those growth percentiles) for each student.

`growth.levels`
: A two letter state acronym or a list of the form `list(my.cuts= , my.levels= )` specifying a vector of cuts between 1 and 99 (e.g., 35, 65) and the associated qualitative levels associated with the cuts (e.g., low, typical, and high). Note that the length of my.levels should be one more than the length of my.cuts. To add your growth levels to the `SGPstateData` data set, please contact the package administrator.

`use.my.knots.boundaries`
: A list of the form `list(my.year= , my.subject= )` specifying a set of pre-calculated knots and boundaries for B-spline calculations. Most often used to utilize knots and boundaries calculated from a previous analysis. Knots and boundaries are stored (and must be made available) with `panel.data` supplied as a list in `panel.data$Knots_Boundaries$my.subject.my.year`. As of SGP_0.0-6 user can also supply a two letter state acronym to utilize knots and boundaries within the `SGPstateData` data set supplied with the SGP package. To add your knots and boundaries to the `SGPstateData` data set, please contact the package administrator. If missing, function automatically calculates knots, boundaries, and loss.hoss values and stores them in `panel.data$Knots_Boundaries $my.subject.my.year` where `my.subject` and `my.year` are provided by `sgp.labels`.

`use.my.coefficient.matrices`
: A list of the form `list(my.year= , my.subject= )` specifying a set of pre-calculated coefficient matrices to use for student growth percentile calculations.

Can be used to calculate baseline referenced student growth percentiles or to calculate student growth percentiles for small groups of excluded students without recalculating an entire set of data. If missing, coefficient matrices are calculated based upon the provided data and stores them in `panel.data$Coefficient_Matrices$my.subject.my.year` where `my.subject` and `my.year` are provided by `sgp.labels`.

`calculate.confidence.intervals`

A character vector providing either a state acronym or a variable name from the supplied panel data. If a state acronym, CSEM tables from the embedded `SGPstateData` (note: CSEM data must be embedded in the `SGPstateData` set. To have your state CSEMs embed in the `SGPstateData` set, please contact the package administrator) will be used. If a variable name, the supplied panel data must contain a variable providing student level CSEMs (e.g., with adaptive testing). NOTE: If a variable name is supplied, the user must also use the argument `panel.data.vnames` indicating what variables in the supplied `panel.data` will be used for the `studentGrowthPercentiles` analysis. For greater control, the user can also supply a list of the form `list(state= , confidence.quantiles= , simulation.iterations= , distribution= , round= )` or `list(variable= , confidence.quantiles= , simulation.iterations= , distribution= , round= )` specifying the `state` or `variable` to use, `confidence.quantiles` to report from the simulated SGPs calculated for each student, `simulation.iterations` indicating the number of simulated SGPs to calculate, `distribution` indicating whether to the the `Normal` or `Skew-Normal` to calculate SGPs, and `round` (defaults to 1, which is an integer - see `round_any` from `plyr` package for details) giving the level to round to. If requested, simulations are calculated and simulated SGPs are stored in `panel.data$Simulated_SGPs`.

`print.other.gp` Boolean argument (defaults to FALSE) indicating whether growth percentiles of all orders should be returned. The default returns only the highest order growth percentile for each student.

`print.sgp.order`

Boolean argument (defaults to FALSE) indicating whether the order of the growth percentile should be provided in addition to the SGP itself.

`calculate.sgps` Boolean argument (defaults to TRUE) indicating whether student growth percentiles should be calculated following coefficient matrix calculation.

`rq.method` Argument defining the estimation method used in the quantile regression calculations. The default is the `"br"` method referring to the Barrodale and Robert's L1 estimation detailed in Koenker (2005) and in the help for the quantile regression (`quantreg`) package.

`rq.method.for.large.n`

Argument defining the estimation method used in the quantile regression calculations when norm group cohort size exceeds 300,000 students. The default is the `"fn"` method referring to the Frisch-Newton estimation detailed in Koenker (2005) and in the help for the quantile regression (`quantreg`) package.

`max.n.for.coefficient.matrices`

Argument the defines a size threshold above which a subset of data is taken with a number of cases equal to the sgp.subset.size.threshold argument. Default is NULL, no subset is taken.

knot.cut.percentiles
: Argument that specifies the quantiles to be used for calculation of B-spline knots. Default is to place knots at the 0.2, 0.4, 0.6, and 0.8 quantiles.

knots.boundaries.by.panel
: Boolean argument (defaults to FALSE) indicating whether knots and boundaries should be calculated by panel in supplied panel data instead of aggregating across panel. If panels are on different scales, then different knots and boundaries may be required to accommodate quantile regression analyses.

exact.grade.progression.sequence
: Boolean argument indicating whether the grade.progression supplied is used exactly (TRUE) as supplied or whether lower order analyses are run as part of the whole analysis (FALSE–the default).

drop.nonsequential.grade.progression.variables
: Boolean argument indicating whether to drop variables that do not occur with a non-sequential grade progress. For example, if the grade progression 7, 8, 10 is provided, the penultimate variable in `panel.data` is dropped. Default is TRUE.

convert.0and100
: Boolean argument (defaults to TRUE) indicating whether conversion of growth percentiles of 0 and 100 to growth percentiles of 1 and 99, respectively, occurs. The default produces growth percentiles ranging from 1 to 99.

sgp.quantiles
: Argument to specify quantiles for quantile regression estimation. Default is Percentiles. User can additionally submit a vector of quantiles (between 0 and 1). Goodness of fit output only available currently for PERCENTILES.

sgp.quantiles.labels
: Argument to specify integer labels associated with provided 'sgp.quantiles'. Integer labels must a vector of length 1 longer than the length of 'sgp.quantiles'.

sgp.loss.hoss.adjustment
: Argument to control whether SGP is calculated using which.max for values associated with the hoss embedded in SGPstateData. Providing two letter state acronym utilizes this adjustment whereas supply NULL (the default) uses no adjustment.

sgp.cohort.size
: Argument to control the minimum cohort size used to calculate SGPs and associated coefficient matrices. NULL (the default) uses no restriction. If not NULL, argument should be an integer value.

sgp.less.than.sgp.cohort.size.return
: If non-NULL, indicates whether a data set should be returned with the indicated character string in place of the SGP that would be calculated. If set to TRUE, then character string: `< sgp.cohort.size students in cohort. No SGP Calculated`.

sgp.test.cohort.size
: Integer indicating the maximum number of students sampled from the full cohort to use in the calculation of student growth percentiles. Intended to be used as a test of the desired analyses to be run. The default, NULL, uses no restrictions (no tests are performed, and analyses use the entire cohort of students).

percuts.digits
: Argument specifying how many digits (defaults to 2) to print percentile cuts (if asked for) with.

`isotonize` Boolean argument (defaults to TRUE) indicating whether quantile regression results are isotonized to prevent quantile crossing following the methods derived by Chernozhukov, Fernandez-Val and Glichon (2010).

`convert.using.loss.hoss`
Boolean argument (defaults to TRUE) indicating whether requested percentile cuts are adjusted using the lowest obtainable scale score (LOSS) and highest obtainable scale score (HOSS). Those percentile cuts above the HOSS are replaced with the HOSS and those percentile cuts below the LOSS are replaced with the LOSS. The LOSS and HOSS are obtained from the loss and hoss calculated with the knots and boundaries used for spline calculations.

`goodness.of.fit`
Boolean argument (defaults to TRUE) indicating whether to produce goodness of fit results associated with produced student growth percentiles. Goodness of fit results are grid.grobs stored in `panel.data$Goodness_of_Fit $my.subject.my.year` where `my.subject` and `my.year` are provided by `sgp.labels`.

`goodness.of.fit.minimum.n`
Integer argument (defaults to 250) indicating the minimum number of observations necessary before goodness of fit plots are constructed."

`goodness.of.fit.output.format`
Character argument (defaults to graphical object 'GROB') indicating output format for goodness of fit plots. Options include: 'GROB', 'PDF', 'PNG', 'SVG'.

`return.prior.scale.score`
Boolean argument (defaults to TRUE) indicating whether to include the prior scale score in the SGP data output. Useful for examining relationship between prior achievement and student growth.

`return.prior.scale.score.standardized`
Boolean argument (defaults to TRUE) indicating whether to include the standardized prior scale score in the SGP data output. Useful for examining relationship between prior achievement and student growth.

`return.norm.group.identifier`
Boolean argument (defaults to TRUE) indicating whether to include the content areas and years that form students' specific norm group in the SGP data output.

`return.norm.group.scale.scores`
Boolean argument (defaults to NULL) indicating whether to return a semi-colon separated character vector of the scores associated with the SGP_NORM_GROUP to which the student belongs.

`return.norm.group.dates`
Boolean argument or character string (defaults to NULL) indicating whether to return a semi-colon separated character vector of the dates associated with time dependent SGPt calculations. If TRUE is supplied, 'DATE' is the assumed name for the date variable.

`return.norm.group.preference`
A single numeric value (defaults to NULL). When multiple SGPs will be produced for some students and a system is required to identify the preferred SGP that will be matched with the student in the `combineSGP` function. This argument provides a ranking that specifies how preferable SGPs produced from the analysis in question is relative to other possible analyses. LOWER NUMBERS CORRESPOND WITH HIGHER PREFERENCE.

`return.panel.data`
: Boolean argument indicating whether to return the original data provided in `panel.data$Panel_Data` in the SGP list of results. Defaults to 'identical(parent.frame(), .GlobalEnv)': If the parent environment from which the function is called is .GlobalEnv, then FALSE, otherwise TRUE.

`print.time.taken`
: Boolean argument (defaults to TRUE) indicating whether to print message indicating information on `studentGrowthPercentiles` analysis and time taken.

`parallel.config`
: parallel configuration argument allowing for parallel analysis by 'tau'. Defaults to NULL.

`calculate.simex`
: A character state acronym or list including state/csem variable, csem.data.vnames, csem.loss.hoss, simulation.iterations, simulation.sample.size, lambda and extrapolation method. Returns both SIMEX adjusted SGP (`SGP_SIMEX`) as well as the percentile ranked SIMEX SGP (`RANK_SIMEX`) values as suggested by Castellano and McCaffrey (2017). Defaults to NULL, no simex calculations performed.

`sgp.percentiles.set.seed`
: An integer (or NULL) argument indicating whether to set.seed to make analyses fully reproducible. To turn off, set argument to NULL. Default is 314159.

`sgp.percentiles.equated`
: An object containing information (linkages, year, ...) on equating done for calculating student growth percentiles.

`SGPt`
: An argument supplied to implement time-dependent SGP analyses (SGPt). Default is NULL giving standard, non-time dependent argument. If set to TRUE, the function assumes the variables 'TIME' and 'TIME_LAG' are supplied as part of the panel.data. To specify other names, supply a list of the form: list(TIME='my_time_name', TIME_LAG='my_time_lag_name'), substituting your variable names.

`SGPt.max.time`
: Boolean argument (defaults to NULL/FALSE) indicating whether cuts/trajectories should be calculated based upon the maximum Time value in the matrices. Such cuts are sometimes used to provide within window trajectories.

`verbose.output`
: A Boolean argument indicating whether the function should output verbose diagnostic messages.

Details

Typical use of the function is to submit a data frame to the function containing records of all students across all grades, allowing the function to subset out specific grade progressions using `grade.progression`. Additional uses include using pre-calculated results to recalculate SGPs for baseline referencing. `studentGrowthPercentiles` examples provide code for use in analyzing assessment data across multiple grades.

Value

Function returns an object of class list containing objects: Coefficient_Matrices, Goodness_of_Fit, Knots_Boundaries, Panel_Data, SGPercentiles, Simulated_SGPs.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

References

Betebenner, D. W. (2008). Toward a normative understanding of student growth. In K. E. Ryan & L. A. Shepard (Eds.), *The Future of Test Based Accountability* (pp. 155-170). New York: Routledge.

Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. Educational Measurement: Issues and Practice, 28(4):42-51.

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.

Castellano, K. E. & McCaffrey, D. F. (2017). The Accuracy of Aggregate Student Growth Percentiles as Indicators of Educator Performance. Educational Measurement: Issues and Practice, 36(1):14-27.

Chernozhukov, V., Fernandez-Val, I. and Galichon, A. (2010), Quantile and Probability Curves Without Crossing. Econometrica, 78: 1093-1125.

Koenker, R. (2005). *Quantile regression*. Cambridge: Cambridge University Press.

Shang, Y., VanIwaarden, A., & Betebenner, D. W. (2015). Covariate measurement error correction for Student Growth Percentiles using the SIMEX method. Educational Measurement: Issues and Practice, 34(1):4-14.

See Also

studentGrowthProjections, sgpData, sgpData_LONG, SGPstateData

Examples

```
## Not run:
## Calculate 4th grade student growth percentiles using included sgpData

require(SGPdata)
sgp_g4 <- studentGrowthPercentiles(
  panel.data=sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  percentile.cuts=c(1,35,65,99),
  subset.grade=4,
  num.prior=1)

## NOTE: "grade.progression" can be used in place of "subset.grade" and "num.prior"

sgp_g4_v2 <- studentGrowthPercentiles(
  panel.data=sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  percentile.cuts=c(1,35,65,99),
  grade.progression=c(3,4))

identical(sgp_g4$SGPercentiles, sgp_g4_v2$SGPercentiles)
```

```
## Established state Knots and Boundaries are available in the supplied SGPstateData
## file and used by supplying the appropriate two letter state acronym.

sgp_g4_DEMO <- studentGrowthPercentiles(
  panel.data=sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  use.my.knots.boundaries="DEMO",
  grade.progression=c(3,4))


## Sample code for running non-sequential grade progression analysis.

sgp_g8_DEMO <- studentGrowthPercentiles(
  panel.data=sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  use.my.knots.boundaries="DEMO",
  grade.progression=c(5,6,8))


## NOTE: Unless specified with 'goodness.of.fit.output.format'
## Goodness of Fit results are stored as graphical objects in the
## Goodness_of_Fit slot. To view or save (using any R output device) try:
## Load 'grid' package to access grid.draw function

require(grid)
grid.draw(sgp_g4$Goodness_of_Fit$READING.2015[[1]]["PLOT"])

require(grid)
pdf(file="Grade_4_Reading_2015_GOF.pdf", width=8.5, height=8)
grid.draw(sgp_g4$Goodness_of_Fit$READING.2015[[1]]["PLOT"])
dev.off()

# Other grades

sgp_g5 <- studentGrowthPercentiles(
  panel.data=sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  percentile.cuts=c(1,35,65,99),
  grade.progression=3:5)

sgp_g6 <- studentGrowthPercentiles(
  panel.data=sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  percentile.cuts=c(1,35,65,99),
  grade.progression=3:6)

sgp_g7 <- studentGrowthPercentiles(
  panel.data=sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  percentile.cuts=c(1,35,65,99),
  grade.progression=3:7)
```

```
sgp_g8 <- studentGrowthPercentiles(
  panel.data=sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  percentile.cuts=c(1,35,65,99),
  grade.progression=4:8)

## All output of studentGrowthPercentiles (e.g., coefficient matrices) is contained
## in the object.  See, for example, names(sgp_g8), for all included objects.
## Results are stored in the slot SGPercentiles.

# Combine all results

sgp_all <- rbind(
  sgp_g4$SGPercentiles$READING.2015,
  sgp_g5$SGPercentiles$READING.2015,
  sgp_g6$SGPercentiles$READING.2015,
  sgp_g7$SGPercentiles$READING.2015,
  sgp_g8$SGPercentiles$READING.2015)

# Save SGP results to .csv file

write.csv(sgp_all, file="sgp_all.csv", row.names=FALSE, quote=FALSE, na="")


## NOTE: studentGrowthPercentiles ADDs results to the current SGP object.
## This allows one to "recycle" the object for multiple grades and subjects as desired.

# Loop to calculate all SGPs for all grades without percentile cuts
# but with growth levels and goodness of fit plots exported automatically as PDFs, PNGs, SVGs,
# and DECILE_TABLES (10x10 table at bottom left of goodness of fit plots)

my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)
my.sgpData <- list(Panel_Data=sgpData)   ### Put sgpData into Panel_Data slot

for (i in seq_along(my.grade.sequences)) {
 my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  growth.levels="DEMO",
  goodness.of.fit="DEMO",
        goodness.of.fit.output.format=c("PDF", "PNG", "SVG", "DECILE_TABLES"),
  grade.progression=my.grade.sequences[[i]])
}

#  Save Student Growth Percentiles results to a .csv file:

write.csv(my.sgpData$SGPercentiles$READING.2015,
file="2015_Reading_SGPercentiles.csv", row.names=FALSE, quote=FALSE, na="")

## Loop to calculate all SGPs for all grades using 2010 to 2013 data

my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)

for (i in seq_along(my.grade.sequences)) {
```

```
 my.sgpData_2009 <- studentGrowthPercentiles(panel.data=my.sgpData,
  panel.data.vnames=c("ID", "GRADE_2010",
   "GRADE_2011", "GRADE_2012", "GRADE_2013",
   "SS_2010", "SS_2011", "SS_2012", "SS_2013"),
  sgp.labels=list(my.year=2013, my.subject="Reading"),
  grade.progression=my.grade.sequences[[i]])
}


## Loop to calculate all SGPs for all grades WITH 80

my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)

for (i in seq_along(my.grade.sequences)) {
 my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
 sgp.labels=list(my.year=2015, my.subject="Reading"),
 calculate.confidence.intervals=list(state="DEMO",
 confidence.quantiles=c(0.1, 0.9), simulation.iterations=100,
  distribution="Normal", round=1),
 grade.progression=my.grade.sequences[[i]])
}

### Example showing how to use pre-calculated coefficient
### matrices to calculate student growth percentiles

my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)
my.sgpData <- list(Panel_Data=sgpData)   ### Put sgpData into Panel_Data slot

for (i in seq_along(my.grade.sequences)) {
 my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  growth.levels="DEMO",
  grade.progression=my.grade.sequences[[i]])
}

percentiles.1st.run <- my.sgpData$SGPercentiles$READING.2015

### my.sgpData has as full set of coefficient matrices for Reading, 2015. To view these

names(my.sgpData$Coefficient_Matrices$READING.2015)

## Let's NULL out the SGPercentiles slot and recreate the percentiles
## using the embedded coefficient matrices

my.sgpData$SGPercentiles$READING.2015 <- NULL

for (i in seq_along(my.grade.sequences)) {
 my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
  sgp.labels=list(my.year=2015, my.subject="Reading"),
  use.my.knots.boundaries=list(my.year=2015, my.subject="Reading"),
  use.my.coefficient.matrices=list(my.year=2015, my.subject="Reading"),
  growth.levels="DEMO",
  grade.progression=my.grade.sequences[[i]])
```

```
}

percentiles.2nd.run <- my.sgpData$SGPercentiles$READING.2015

identical(percentiles.1st.run, percentiles.2nd.run)


## End(Not run)
```

---

studentGrowthPlot *Create a student growth and achievement chart*

Description

Function used to produce individual student growth and achievement chart (an achievement time lines indicating student growth) based upon output from student growth percentile and student growth projection analyses. Function is integrated with SGPstateData to accommodate state specific scales and nomenclature including performance level names. See Betebenner (2012) for discussion

Usage

```
studentGrowthPlot(Scale_Scores,
 Plotting_Scale_Scores,
 Achievement_Levels,
 SGP,
 SGP_Levels,
 Grades,
 Content_Areas,
 Cuts,
 Plotting_Cuts,
 SGP_Targets,
 SGP_Scale_Score_Targets,
 Plotting_SGP_Scale_Score_Targets,
 Cutscores,
 Years,
 Report_Parameters)
```

Arguments

Scale_Scores	A vector of historical scale scores.
Plotting_Scale_Scores	A vector of scale scores used as the vertical coordinates for plotting. If supplied, Scale_Scores are used for text and Plotting_Scale_Scores are used for the actual coordinates.
Achievement_Levels	A vector of historical performance levels.

SGP	A vector of historical student growth percentiles.
SGP_Levels	A vector of historical growth (SGP) levels (e.g., low, typical, high).
Grades	A vector of historical grades student was tested in.
Content_Areas	A vector of historical content areas student was tested in.
Cuts	A list of cuts scores for NY1, NY2, and NY3.
Plotting_Cuts	A list of plotting cuts scores for NY1, NY2, and NY3. Plotting cuts are identical to Cuts for states with a vertical scale and are transformed for non-vertical scale states.
SGP_Targets	A list of CUKU, CUKU_Current, MUSU, MUSU_Current targets.
SGP_Scale_Score_Targets	A list of CUKU, CUKU_Current, MUSU, MUSU_Current scale score targets.
Plotting_SGP_Scale_Score_Targets	A list of CUKU, CUKU_Current, MUSU, MUSU_Current scale score targets for plotting that are transformed when no vertical scale exists.
Cutscores	A data.frame of long formatted achievement level cutscores.
Years	A vector of years corresponding to supplied scale scores.
Report_Parameters	A list containing arguments: Current_Year, Content_Area, State, Denote_Content_Area, SGP_Targets, and Configuration.

Details

Function currently used as part of SGP package to produce student growth charts for states. Function is usually called from the higher level function visualizeSGP which allows for the creation of a student growth plot catalog for each school with student reports organized by grade and student name.

Value

Returns a student growth plot graphical object that is usually exported in either PDF or PNG format.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

References

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.

studentGrowthPlot_Styles

studentGrowthPlot_Styles providing base templates for the core studentGrowthPlot function

Description

Function includes five "styles" associated with studentGrowthPlot to create PDF student growth plots depicting the growth and achievement for an individual student over time. The five styles display one to five content area student growth plots on a single page. The one, two, and three content area plots are rendered on 8.5 by 11 PDFs and the four and five content area plots are rendered on 11 by 17 PDFs. These charts are currently being used in multiple states to report student growth results. This function is called by visualizeSGP to generate individual student reports. This function may expand in the future to incorporate other possible individual student reports.

Usage

```
studentGrowthPlot_Styles(
           sgPlot.data,
           sgPlot.sgp_object,
           sgPlot.cutscores,
           state,
           last.year,
           content_areas,
           districts,
           schools,
           reports.by.student,
           reports.by.instructor,
           reports.by.school,
           sgPlot.years,
           sgPlot.demo.report,
           sgPlot.folder,
           sgPlot.folder.names,
           sgPlot.anonymize,
           sgPlot.front.page,
           sgPlot.header.footer.color,
           sgPlot.fan,
    sgPlot.sgp.targets,
           sgPlot.cleanup,
           sgPlot.baseline,
    sgPlot.sgp.targets.timeframe,
           sgPlot.zip,
           sgPlot.output.format,
           sgPlot.linkages)
```

Arguments

sgPlot.data	Wide formatted individual student report data used to produce student growth plots. To view structure of wide formatted data, utilize the sgPlot.save.sgPlot.data option with visualizeSGP to save wide formatted student growth plot data.
sgPlot.sgp_object	SGP object containing coefficient matrices and knots and boundaries for percentile trajectory calculations associated with JSON output.
sgPlot.cutscores	List of cutscores, possibly transformed, for plotting studentGrowthPlots
state	Acronym indicating state associated with the summaries for access to assessment program information embedded in SGPstateData.
last.year	Argument indicating the final year represented in the student growth plots.
content_areas	Argument providing the content areas depicted in the student growth plots.
districts	A vector of district numbers indicating which districts student growth plots should be produced for.
schools	A vector of school numbers indicating which schools student growth plots should be produce for.
reports.by.student	A Boolean variable passed to studentGrowthPlot_Styles indicating whether separate individual plots will be produced or separate reports and a summary catalog containing those reports will be produced.
reports.by.instructor	A Boolean variable passed to studentGrowthPlot_Styles indicating whether individual plots will be collated and bundled as a summary catalog by instructor.
reports.by.school	A Boolean variable passed to studentGrowthPlot_Styles indicating whether individual plots will be collated and bundled as a summary catalog by school. Prior to version 0.9-9.7, this was the only way of bundling reports and was thus the default.
sgPlot.years	A vector of all years over which student growth plots are being produced.
sgPlot.demo.report	A Boolean argument indicating whether a demonstration report catalog (with anonymized individual, school, and district names) is to be produced.
sgPlot.folder	A character argument specifying the folder into which the student growth reports will be placed.
sgPlot.folder.names	Either names or number indicating whether names or numbers should be used as folder names.
sgPlot.anonymize	A Boolean argument indicating whether individual, school, and district names should be anonymized.
sgPlot.front.page	A character vector indicating the file, the the base directory, that should be used as the front page for the student growth plots.

`sgPlot.header.footer.color`
: A color (as a character) specifying the header/footer color of the report.

`sgPlot.fan`
: A Boolean argument indicating whether the projection fan indicating growth rates necessary to reach 1 years targets be displayed.

`sgPlot.sgp.targets`
: A Boolean argument indicating whether the sgp targets are to be displayed.

`sgPlot.cleanup`
: A Boolean argument indicating whether to cleanup/remove files produced as part of pdfLaTeX build.

`sgPlot.baseline`
: A Boolean argument indicating whether to use baseline referenced student growth percentiles in student growth plots.

`sgPlot.sgp.targets.timeframe`
: An integer argument indicating the number of years forward associated with SGP targets.

`sgPlot.zip`
: A Boolean argument indicating whether to school folders.

`sgPlot.output.format`
: Argument indicating the desired type of output format for student growth plots. Either 'PDF', 'PNG', or 'PDF_PIECES'.

`sgPlot.linkages`
: Argument passing linkage information (defaults to NULL) associated with assessment scale transition.

Details

`studentGrowthPlot_Styles` is a utility function containing five templates for displaying `studentGrowthPlot`s. The templates display two or three content areas depending upon the availability of test data for the state. This function will expand in the future to allow for other types of individual growth and achievement reports.

Value

Function produces *numerous* (potentially hundreds of thousands of) PDF/PNG student growth plots.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

References

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.

studentGrowthProjections
Student Growth Projections

Description

Function to calculate percentile growth projections/trajectories using large scale assessment data and results derived from student growth percentile calculation. Function can produce percentile growth trajectories, as well as growth percentiles, sufficient for each student to reach a set of predefined scale score cut.

Usage

```
studentGrowthProjections(panel.data,
                         sgp.labels,
                         grade.progression,
                         content_area.progression=NULL,
                         year_lags.progression=NULL,
                         grade.projection.sequence=NULL,
                         content_area.projection.sequence=NULL,
                         year_lags.projection.sequence=NULL,
                         max.forward.progression.years=NULL,
                         max.forward.progression.grade=NULL,
                         max.order.for.progression,
                         use.my.knots.boundaries,
                         use.my.coefficient.matrices,
                         panel.data.vnames,
                         achievement.level.prior.vname=NULL,
                         performance.level.cutscores,
                         calculate.sgps=TRUE,
                         convert.0and100=TRUE,
                         trajectories.chunk.size=50000L,
                         sgp.projections.equated=NULL,
                         projection.unit="YEAR",
                         projection.unit.label=NULL,
                         percentile.trajectory.values=NULL,
                  percentile.trajectory.values.max.forward.progression.years=NULL,
                         return.percentile.trajectory.values=NULL,
                         return.projection.group.identifier=NULL,
                         return.projection.group.scale.scores=NULL,
                         return.projection.group.dates=NULL,
                         isotonize=TRUE,
                         lag.increment=0L,
                         lag.increment.label=NULL,
                         sgp.exact.grade.progression=FALSE,
                         projcuts.digits=NULL,
                         sgp.projections.use.only.complete.matrices=NULL,
```

```
                    SGPt=NULL,
                    print.time.taken=TRUE)
```

Arguments

panel.data	Object of class list containing longitudinal student data in wide format in panel.data$Panel_Data. See studentGrowthPercentiles for data requirements. List object must also contain panel.data$Knots_Boundaries and panel.data$Coefficient_Ma See sgpData for an exemplar data set. NOTE: The column position of the variables IS IMPORTANT, NOT the names of the variables.
sgp.labels	REQUIRED. A list, sgp.labels, of the form list(my.year= , my.subject=). The user-specified values are used to save the percentile growth projections/trajectories and identify coefficient matrices and knots & boundaries for calculation if use.my.coefficient.matrices or use.my.knots.boundaries is missing. Partly replaces previous argument proj.function.labels.
grade.progression	REQUIRED. Argument to specify a student grade/time progression in the data to be used for percentile growth projection/trajectory calculation. This argument helps in replacing previous arguments num.panels, max.num.scores, and num.prior.scores.
content_area.progression	Argument to specify a student content area progression in the data supplied for percentile growth projection/trajectory calculation. Defaults to NULL and is calculated from supplied argument 'sgp.labels'.
year_lags.progression	Argument to specify a student year progression lags in the data supplied for percentile growth projection/trajectory calculation. Defaults to NULL assuming annual increment calculated from supplied argument 'sgp.labels' and 'grade.progression'.
grade.projection.sequence	Argument to manually supply grade sequence over which projection is made. Defaults to NULL and is calculated from available data.
content_area.projection.sequence	Argument to manually supply content area sequence over which projection is made. Defaults to NULL and assumes current year content area is repeated going forward.
year_lags.projection.sequence	Argument to manually supply year lags sequence over which projection is made. Length of supplied sequence should be 1 less than length of supplied 'grade.projection.sequence' and 'content_area.projection.sequence'. Defaults to NULL and assumes annual (1 year lags).
max.forward.progression.years	The MAXIMUM number of years/grades/time periods to project forward conditional upon available coefficient matrices. If missing/NULL (the default), function will project forward as far as allowed by available coefficient matrices.
max.forward.progression.grade	The MAXIMUM grade to project forward based upon available coefficient matrices. If missing/NULL (the default), function will project forward as far as allowed by available coefficient matrices.

max.order.for.progression
: Argument to specify the maximum coefficient matrix order to be used for percentile growth projection/trajectory calculation. If missing, the function utilizes the highest matrix order available.

use.my.knots.boundaries
: A list of the form `list(my.year= , my.subject= )` specifying the set of precalculated knots and boundaries for B-spline calculations. Knot and boundaries are stored (and must be made available) with `panel.data` supplied as a list in `panel.data$Knots_Boundaries$my.year.my.subject`. As of SGP_0.0-6.9 user can also supply a two letter state acronym to utilize knots and boundaries within the `SGPstateData` data set supplied with the SGP package. If missing, function tries to retrieve knots and boundaries from `panel.data$Knots_Boundaries$my.year.my.subject` where `my.year` and `my.subject` are provided by `sgp.labels`.

use.my.coefficient.matrices
: A list of the form `list(my.year= , my.subject= )` specifying the set of precalculated coefficient matrices to use for percentile growth projection/trajectory calculations. Coefficient matrices are stores (and must be available) with `panel.data` supplied as a list in `panel.data$Coefficient_Matrices $my.year.my.subject`. If missing, function tries to retrieve coefficient matrices from `panel.data$Coefficient_Matrices$my.year.my.subject` where `my.year` and `my.subject` are provided by `sgp.labels`.

panel.data.vnames
: Vector of variables to use in percentile growth projection/trajectory calculations. If not specified, function attempts to use all available variables.

achievement.level.prior.vname
: Character vector indicating variable is supplied panel data corresponding to the prior achievement level to be added to the output. Used in the production of growth to standard analyses.

performance.level.cutscores
: Argument for supplying performance level cutscores to be used for determining growth-to-standard percentile growth trajectory calculations. Argument accepts a two letter state acronym (e.g., "CO") that retrieves cutscores that are automatically embedded in a data set contained in the SGP package. Argument also accepts a subject specific list of the form:

```
performance.level.cutscores <- list(
       Reading=list(GRADE_3=c(cut1, cut2, cut3),
                    GRADE_4=c(cut1, cut2, cut3),
                     . . .
                    GRADE_8=c(cut1, cut2, cut3)),
          Math=list(GRADE_3=c(cut1, cut2, cut3),
                     . . .
                    GRADE_7=c(cut1, cut2, cut3),
                    GRADE_8=c(cut1, cut2, cut3)))
```

Note that the subject name must match that provided by `sgp.labels`. If cuts are not desired leave the cutscore unspecified, which is the default. If your state's

cutscores are not included in the SGPstateData data set or are incorrect, please contact <dbetebenner@nciea.org> to have them added or corrected!

calculate.sgps Boolean argument (defaults to TRUE) indicating whether to calculate student growth projections. Currently used to bypass calculations in analyzeSGP when scale changes occur.

convert.0and100
Boolean argument (defaults to TRUE) indicating whether conversion of growth percentiles of 0 and 100 to growth percentiles of 1 and 99, respectively, occurs. The default produces growth percentiles ranging from 1 to 99.

trajectories.chunk.size
To enhance speed, large data sets are broken up into smaller data sets before trajectories and cuts are calculated. The default chunk.size is 50,000 and is utilized when data sets of over 150 percent of the trajectory.chunk.size are passed to .get.percentile.trajectories and .get.trajectories.and.cuts.

sgp.projections.equated
A list (defaults to NULL) specifying the Year and Linkages associated with the equating used for projection purposes.

projection.unit
Argument specifying the units in which the projections/trajectories and cuts are reported. Either "GRADE", the default, or "YEAR".

projection.unit.label
Argument specifying the label associated with the units in which the projections/trajectories and cuts are reported. Defaults to NULL utilizing the value associated with the argument supplied to projection.unit.

percentile.trajectory.values
An integer argument with values ranging from 1 to 100 that returns a vector of percentile trajectory cuts (default is NULL, no percentile trajectory values returned). The returned values are the lower bound for the interval associated with the percentile(s) requested. For example, by specifying 'percentile.trajectory.values=1' the user would receive the conditional .005 quantile associated with each student's distribution. Supplied values can also be a vector of targets that will return Scale Scores associated with those targets.

percentile.trajectory.values.max.forward.progression.years
An integer argument indicating the number of years associated with the trajectory calculated by percentile.trajectory.values. Supplied value can also be a vector of targets that will trim the trajectory to fit with a designated number of years.

return.percentile.trajectory.values
A Boolean variable indicating whether to return the percentile trajectory values when they are supplied as targets via panel.data. Default is NULL/FALSE.

return.projection.group.identifier
A Boolean variable indicating whether to return the projection group identifier to distinguish different projection trajectories. Defaults to NULL/FALSE.

return.projection.group.scale.scores
A Boolean variable indicating whether to return the projection group scale scores for each student. Defaults to NULL/FALSE.

return.projection.group.dates
: A Boolean variable indicating whether to return the projection group dates associated with the students' scale scores. Defaults to NULL/FALSE.

isotonize
: Boolean argument (defaults to TRUE) indicating whether quantile regression results are isotonized to prevent quantile crossing following the methods derived by Chernozhukov, Fernandez-Val and Glichon (2010).

lag.increment
: A non-negative integer (defaults to 0) indicating the lag associated with the data supplied for projections. Only relevant if Cutscores or Knots and Boundaries are year dependent.

lag.increment.label
: Overrides default ways in which lagged targets and straight targets are labelled ("" and "_CURRENT", respectively).

sgp.exact.grade.progression
: A Boolean argument (defaults to FALSE) indicating whether to use the exact grade progression supplied or all orders up to the grade progression supplied.

projcuts.digits
: The number of digits (defaults to NULL/0) percentile trajectories (if requested) are formatted.

sgp.projections.use.only.complete.matrices
: A Boolean argument (defaults to TRUE) indicating whether projections should be calculated even when all matrices are not available for the given content_area.projection.sequenc and grade.projection.sequence.

SGPt
: Character vector indicating the name of the 'DATE' variable to be used for time dependent student growth projection calculations. Defaults to NULL, non-time-dependent SGP calculations.

print.time.taken
: Boolean argument (defaults to TRUE) indicating whether to print message indicating information on studentGrowthProjections analysis and time taken.

Value

Function returns the input panel.data list object with the additional percentile growth trajectories/percentiles stored in panel.data$SGProjections$my.year.my.subject consisting of student IDs and the associated percentile growth projections/trajectories and cuts. The data frame contains projections/trajectories for each performance level cut-point supplied and each percentile cut the user specifies.

Note

Use of this function assumes prior calculation of student growth percentiles, making the coefficient matrices available within the panel.data$Coefficient_Matrices list object. Additionally, if cutscores are desired they must be supplied explicitly by the user (as detailed above in performance.level.cutscores) or included in the SGPstateData data set. If your state's cutscores are not included or are incorrect, please contact <dbetebenner@nciea.org> to have cutscores added or corrected!

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

References

Betebenner, D. W. (2008). Toward a normative understanding of student growth. In K. E. Ryan & L. A. Shepard (Eds.), *The Future of Test Based Accountability* (pp. 155-170). New York: Routledge.

Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. Educational Measurement: Issues and Practice, 28(4):42-51.

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York: Routledge.

Chernozhukov, V., Fernandez-Val, I. and Galichon, A. (2010), Quantile and Probability Curves Without Crossing. Econometrica, 78: 1093-1125.

See Also

studentGrowthPercentiles, sgpData

Examples

```
## Not run:
## First calculate SGPs for 2014
my.grade.sequences <- list(3:4, 3:5, 3:6, 3:7, 4:8)
my.sgpData <- list(Panel_Data = sgpData)

for (i in seq_along(my.grade.sequences)) {
 my.sgpData <- studentGrowthPercentiles(panel.data=my.sgpData,
  sgp.labels=list(my.year=2014, my.subject="Reading"),
  use.my.knots.boundaries="DEMO",
  grade.progression=my.grade.sequences[[i]])
}

## Calculate Growth Projections

my.grade.progressions <- list(3, 3:4, 3:5, 3:6, 4:7)

for (i in seq_along(my.grade.progressions)) {
 my.sgpData <- studentGrowthProjections(panel.data=my.sgpData,
  sgp.labels=list(my.year=2014, my.subject="Reading"),
  projcuts.digits=0,
  projection.unit="GRADE",
  performance.level.cutscores="DEMO",
  percentile.trajectory.values=c(25, 50, 75),
  grade.progression=my.grade.progressions[[i]])
}

##  Save the Student Growth Projections Results to a .csv file:

write.csv(my.sgpData$SGProjections$READING.2014,
file= "2014_Reading_SGProjections.csv", row.names=FALSE, quote=FALSE)

## End(Not run)
```

summarizeSGP *Summarize student scale scores, proficiency levels and student growth percentiles according to user specified summary group variables*

Description

Utility function used to produce summary tables using long formatted data that contain student growth percentiles. An exemplar is provided from the successive execution of prepareSGP, analyzeSGP and combineSGP.

Usage

```
summarizeSGP(sgp_object,
             state,
             years,
             content_areas,
             sgp.summaries=NULL,
             summary.groups=NULL,
             confidence.interval.groups=NULL,
             produce.all.summary.tables=FALSE,
             summarizeSGP.baseline=NULL,
             projection.years.for.target=NULL,
             save.old.summaries=FALSE,
             highest.level.summary.grouping="STATE",
             parallel.config=NULL)
```

Arguments

sgp_object	An object of class SGP containing long formatted data in the @Data slot. If summaries of student growth percentiles are requested, those quantities must first be produced (possibly by first using analyzeSGP) and subsequently combined with the @Data data (possibly with combineSGP).
state	Acronym indicating state associated with the summaries for access to assessment program information embedded in SGPstateData.
years	A character vector indicating year(s) in which to produce summary tables associated with student growth percentile and percentile growth trajectory/projection analyses. If missing the function will use the data to calculate years and produce summaries for the most recent three years.
content_areas	A character vector indicating content area(s) in which to produce student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer the content area(s) available for analyses.
sgp.summaries	A list giving the summaries requested for each group analyzed based upon the summary.group argument. The default (produced internal to summarizeSGP) summaries include:

MEDIAN_SGP	The group level median student growth percentile.
MEDIAN_SGP_COUNT	The number of students used to compute the median.
PERCENT_AT_ABOVE_PROFICIENT	The percentage of students at or above proficient.
PERCENT_AT_ABOVE_PROFICIENT_COUNT	The number of students used to compute the percentage at/above proficien
PERCENT_AT_ABOVE_PROFICIENT_PRIOR	The percentage of students at or above proficient in the prior year.
PERCENT_AT_ABOVE_PROFICIENT_PRIOR_COUNT	The number of students used to compute the percentage at/above proficien

NOTE: The internal function `percent_in_category()` summary function requires a variable that MUST be a factor with proficiency categories as levels. The function utilizes the SGPstateData with the provided state name in an attempt to identify achievement levels and whether or not they are considered proficient.

`summary.groups` A list consisting of 8 elements indicating the types of groups across which all summaries are taken (Inclusion means that summaries will be calculated for levels of the associated variable). For state data, if the list is not explicitly provided, the function will attempt to determine levels based upon meta data supplied in the `@Names` slot of the provided SGP object. See `prepareSGP` for more information on supplied meta-data.

institution:	State, District and/or School.
content area:	Variable indicating content area (default is CONTENT_AREA) if content area summaries are of
time:	Variable indicating time (default is YEAR) if time summaries are of interest. NOTE: Cross year
institution_type:	Variable(s) indicating the type of institution institution (default EMH_LEVEL) if summaries by
institution_level:	Variable(s) indicating levels within the institution (default GRADE) if summaries by institution l
demographic:	Demographics variables if summaries by demographic subgroup are of interest.
institution_inclusion:	Variables indicating inclusion for institutional calculations.
growth_only_summary:	Variables indicating whether to calculate summaries only for those students with growth in addit

All group slots MUST be included in the list, although `NULL` can be provided if a grouping subset is not desired. All possible combinations of the group variables are produced.

`confidence.interval.groups`
A list consisting of information used to calculate group confidence intervals:

TYPE:	
VARIABLES:	
QUANTILES	
GROUP	
content	
time	
institution_type	
institution_level	
demographic	
institution_inclusion	
growth_only_summary	The growth only summary variables if confidence intervals by growth only summary group are desired

For CSEM analysis this argument requires that simulated SGPs have been produced (see analyzeSGP for more information). List slots set to NULL will not produce confidence intervals. NOTE: This is currently an experimental functionality and is very memory intensive. Groups to be included should be identified selectively! The default 95% confidence intervals are provided in the selected summary tables as two additional columns named LOWER_MEDIAN_SGP_95_CONF_BOUND and UPPER_MEDIAN_SGP_95_CONF_BOUND.

produce.all.summary.tables

A Boolean variable, defaults to FALSE, indicating whether the function should produce ALL possible summary table. By default, a set of approximately 70 tables are produced that are used in other parts of the packages (e.g., bubblePlots).

summarizeSGP.baseline

A Boolean variable, defaults to FALSE, indicating whether the function should utilize baseline sgp for summary table production. By default, a set of approximately 100 tables are produced that are used in other parts of the packages (e.g., bubblePlots).

projection.years.for.target

An integer/NULL argument (defaults to NULL) indicating SGP_TARGET variables to summarize based upon years projected forward. Default is years for projections used by state (or 3 years) which is what is generally used by most states.

save.old.summaries

A Boolean argument, defaults to FALSE, indicating whether to save the @Summary slot (if not NULL) prior to calculating new summaries. By defaulting to FALSE, the function overwrites previous (e.g., last year's summaries) summaries.

highest.level.summary.grouping

A character vector indicating the highest level for summary groups, defaults to 'STATE'.

parallel.config

A named list with, at a minimum, two elements indicating 1) the BACKEND package to be used for parallel computation and 2) the WORKERS list to specify the number of processors to be used in each major analysis. The BACKEND element can be set = to FOREACH or PARALLEL. Please consult the manuals and vignettes for information of these packages! The analyzeSGP help page contains more thorough explanation and examples of the parallel.config setup.

TYPE is a third element of the parallel.config list that provides necessary information when using FOREACH or PARALLEL packages as the backend. With BACKEND="FOREACH", the TYPE element specifies the flavor of 'foreach' backend. As of version 1.0-1.0, only "doParallel" is supported. TYPE=NA (default) produces summaries sequentially. If BACKEND = "PARALLEL", the parallel package will be used. This package combines deprecated parallel packages snow and multicore. Using the "snow" implementation of parallel the function will create a cluster object based on the TYPE element specified and the number of workers requested (see WORKERS list description below). The TYPE element indicates the users preferred cluster type (either "PSOCK"

for socket cluster of "MPI" for an OpenMPI cluster). If Windows is the operating system, this "snow" implementation must be used and the TYPE element must = "PSOCK". Defaults are assigned based on operating system if TYPE is missing based on system OS. Unix/Mac OS defaults to the "multicore" to avoid worker node pre-scheduling and appears to be more efficient in these operating systems.

The WORKERS element is a list with SUMMARY specifying the number of processors (nodes) desired or available. For example, SUMMARY=2 may be used on a dual core machine to use both cores available. (NOTE: choice of the number of cores is a balance between the number of processors available and the amount of RAM a system has; each system will be different and may require some adjustment).

Default is FOREACH as the back end, TYPE=NA and WORKERS=1, which produces summary tables sequentially: 'list(BACKEND="FOREACH", TYPE=NA, WORKERS=list(SUMMARY=1))'

Example parallel use cases are provided below.

Details

Function makes use of the `foreach` package to parallel process summary tables of student data. The proper choice of parallel backend is dependent upon the user's operating system, software and system memory capacity. Please see the `foreach` documentation for details. By default, the function will process the summary tables sequentially.

Value

Function returns lists containing the summary tables as data.table objects in the `@Summary` slot of the SGP data object. Each `institution` has a slot in the `@Summary` list.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

See Also

prepareSGP, analyzeSGP, combineSGP

Examples

```
## Not run:
## summarizeSGP is Step 4 of 5 of abcSGP
Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- prepareSGP(Demonstration_SGP)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP)
Demonstration_SGP <- combineSGP(Demonstration_SGP)
Demonstration_SGP <- summarizeSGP(Demonstration_SGP)

###  Example uses of the parallel.config argument

##  Windows users  must use the parallel package and R version >= 2.13:
```

```
#  Note the number of workers is 8, and PSOCK type cluster is used.
#  This example is would be good for a single workstation with 8 cores.
 . . .
 parallel.config=list(
  BACKEND="PARALLEL", TYPE="PSOCK",
  WORKERS=list(SUMMARY=2))
 . . .

#  doParallel package - only available with R 2.13 or newer
 . . .
 parallel.config=list(
  BACKEND="FOREACH", TYPE="doParallel",
  WORKERS=list(SUMMARY=6))
 . . .

##  parallel package - only available with R 2.13 or newer
#  Note the number of workers is 50, and MPI is used,
#  suggesting this example is for a HPC cluster usage.
 . . .
 parallel.config=list(
  BACKEND="PARALLEL", TYPE="MPI"),
  WORKERS=list(SUMMARY=50))
 . . .

#  NOTE:  This list of parallel.config specifications is NOT exhaustive.
#  See examples in analyzeSGP documentation for some others.

## End(Not run)
```

---

testSGP *Test SGP Package functions*

Description

testSGP runs large scale tests of the SGP package to test for consistent performance across releases.

Usage

```
testSGP(TEST_NUMBER,
 save.results=TRUE,
 test.option=list(),
 memory.profile=FALSE,
 stop.fail=TRUE)
```

Arguments

TEST_NUMBER	An integer indicating the test to be run. Type 'testSGP()' to see list and description of available tests.
save.results	A Boolean variable, defaulting to FALSE, indicating whether the results of the analysis is saved to the working directory.

test.option	A character string (defaults to NULL) supplying a test option to the given test specified by TEST_NUMBER. Argument is test specific. See source code for testSGP for possible arguments.
memory.profile	A Boolean variable indicating whether to use memory profiling via Rprof. Experimental. Defaults to FALSE.
stop.fail	A Boolean variable indicating whether to stop the function if a test fails Defaults to TRUE.

Value

Returns output associated with functions being run.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

See Also

abcSGP

Examples

```
## Not run:
## testSGP(0): Test of studentGrowthPercentiles, studentGrowthProjections, and sgpData
testSGP(0)

## testSGP(1) & testSGP('1b') runs abcSGP for all years in sgpData_LONG with/without sqliteSGP
testSGP(1)
testSGP('1b')

## testSGP(2): Various tests of updateSGP functionality.
## testSGP('2a'): Test of updateSGP performing SGP analyses in two steps:
## Create what_sgp_object: 2010-2011 to 2013-2014 then add with_sgp_data_LONG 2014-2015 using
## overwrite.existing.data=FALSE and sgp.use.my.coefficient.matrices=FALSE.
## testSGP('2b'): Test of updateSGP performing SGP analyses in two steps:
##  Create what_sgp_object: 2010-2011 to 2013-2014 then add with_sgp_data_LONG 2014-2015 using
## overwrite.existing.data=TRUE and sgp.use.my.coefficient.matrices=FALSE.
## testSGP('2c'): Test of updateSGP performing SGP analyses in two steps:
## Create what_sgp_object: 2010-2011 to 2013-2014 then add with_sgp_data_LONG 2014-2015 using
## overwrite.existing.data=TRUE and sgp.use.my.coefficient.matrices=TRUE.
## testSGP('2d'): Test of updateSGP performing SGP analyses in two steps:
## Create what_sgp_object: 2010-2011 to 2013-2014 then add with_sgp_data_LONG 2014-2015 using
## overwrite.existing.data=FALSE and sgp.use.my.coefficient.matrices=TRUE.
testSGP('2a')
testSGP('2b')
testSGP('2c')
testSGP('2d')

## testSGP(3) runs abcSGP on grade progressions including End of Course Tests in Mathematics
## (Algebra I and Algebra II) and Reading (Grade 9 Literature and American Literature)
testSGP(3)
```

```
## testSGP(4) runs prepareSGP and analyzeSGP with simex adjustment for measurement error
testSGP(4)

## testSGP(5) runs abcSGP assuming at test transition in the most recent year. NOTE YET COMPLETED
testSGP(5)

## testSGP(6) runs a basic baseline SGP analysis including the construction of baseline matrices.
testSGP(6)

## End(Not run)
```

---

updateSGP *Function to update SGP object with additional year's analyses*

Description

updateSGP takes an object of class SGP and adds in additional data (usually an additional year's data) and runs analyses on that additional year's data including the results in the supplied SGP object.

Usage

```
updateSGP(
 what_sgp_object=NULL,
 with_sgp_data_LONG=NULL,
 with_sgp_data_INSTRUCTOR_NUMBER=NULL,
 state=NULL,
 steps=c("prepareSGP",
  "analyzeSGP",
  "combineSGP",
  "summarizeSGP",
  "visualizeSGP",
  "outputSGP"),
 years=NULL,
 content_areas=NULL,
 grades=NULL,
 sgp.percentiles=TRUE,
 sgp.projections=TRUE,
 sgp.projections.lagged=TRUE,
 sgp.percentiles.baseline=TRUE,
 sgp.projections.baseline=TRUE,
 sgp.projections.lagged.baseline=TRUE,
 sgp.test.cohort.size=NULL,
 return.sgp.test.results=FALSE,
 simulate.sgps=TRUE,
 save.old.summaries=NULL,
 save.intermediate.results=TRUE,
```

```
    calculate.simex=NULL,
    calculate.simex.baseline=NULL,
    sgp.use.my.coefficient.matrices=NULL,
    sgp.target.scale.scores=FALSE,
    sgp.target.scale.scores.only=FALSE,
    overwrite.existing.data=TRUE,
    update.old.data.with.new=TRUE,
    output.updated.data=TRUE,
    sgPlot.demo.report=TRUE,
    plot.types=c("bubblePlot", "studentGrowthPlot", "growthAchievementPlot"),
    outputSGP.output.type=c("LONG_Data",
       "LONG_FINAL_YEAR_Data",
       "WIDE_Data",
       "INSTRUCTOR_Data"),
    outputSGP.directory="Data",
    sgp.config=NULL,
    goodness.of.fit.print=TRUE,
    parallel.config=NULL,
    sgp.sqlite=FALSE,
    SGPt=NULL,
    sgp.percentiles.equated=NULL,
    sgp.percentiles.equating.method=NULL,
    sgp.percentiles.calculate.sgps=TRUE,
    fix.duplicates=NULL,
    get.cohort.data.info=FALSE,
    ...)
```

Arguments

what_sgp_object	The SGP object to which the additional data will be added and analyzed. This object must be specified.
with_sgp_data_LONG	The additional data in LONG format to be added to the supplied SGP object. The additional data must be in the same form as the data in the @Data slot. If with_sgp_data_LONG is not supplied, the function with update the sgp_object supplied in 'what_sgp_object' using the embedded coefficient matrices, essentially re-doing the analyses.
with_sgp_data_INSTRUCTOR_NUMBER	The addition INSTRUCTOR_NUMBER data in LONG format to be added to the supplied SGP object. The additional data must be in the same format as the data in the @Data_Supplementary[['INSTRUCTOR_NUMBER']] slot. Default is NULL, no INSTRUCTOR_NUMBER data is supplied.
state	The 'state' for the sgp_object. Derived from sgp_object name if not explicitly supplied.
steps	A vector indicting the steps abcSGP will perform as part of the update. Defaults to all steps: prepareSGP, analyzeSGP, combineSGP, summarizeSGP, visualizeSGP, outputSGP.

`years` If only 'what_sgp_object' is supplied, years specifies the years to be run among those in the provided sgp_object.

`content_areas` If only 'what_sgp_object' is supplied, content_areas specifies the content areas to be run among those provided by the coefficient matrices in the sgp_object. Default is to run all analyses associated with the coefficient matrices.

`grades` A vector indicating grades for which to calculate student growth percentiles and/or student growth projections/trajectories. If missing the function will use the data to infer all the grade progressions for student growth percentile and student growth projections/trajectories analyses. This argument is passed to either `abcSGP` or `analyzeSGP` depending on the update context.

`sgp.percentiles`
Boolean variable indicating whether to calculate student growth percentiles (if analyzeSGP is included in the 'steps' argument). Defaults to TRUE.

`sgp.projections`
Boolean variable indicating whether to calculate student growth projections (if analyzeSGP is included in the 'steps' argument). Defaults to TRUE.

`sgp.projections.lagged`
Boolean variable indicating whether to calculate lagged student growth projections often used for growth to standard analyses (if analyzeSGP is included in the 'steps' argument). Defaults to TRUE.

`sgp.percentiles.baseline`
Boolean variable indicating whether to calculate baseline student growth percentiles and/or coefficient matrices (if analyzeSGP is included in the 'steps' argument). Defaults to TRUE.

`sgp.projections.baseline`
Boolean variable indicating whether to calculate baseline student growth projections (if analyzeSGP is included in the 'steps' argument). Defaults to TRUE.

`sgp.projections.lagged.baseline`
Boolean variable indicating whether to calculate lagged baseline student growth projections (if analyzeSGP is included in the 'steps' argument). Defaults to TRUE.

`sgp.test.cohort.size`
Integer indicating the maximum number of students sampled from the full cohort to use in the calculation of student growth percentiles. Intended to be used as a test of the desired analyses to be run. The default, NULL, uses no restrictions (no tests are performed, and analyses use the entire cohort of students).

`return.sgp.test.results`
Boolean variable passed to `analyzeSGP` and `studentGrowthPercentiles` indicating whether the results from the cohort sample subset (if specified using the above argument) should be returned for inspection. Defaults to FALSE. If TRUE, only the sample subset of the data used will be returned in the SGP object's @Data slot. Alternatively, user can supply the character "ALL_DATA" to the argument to return the entire original data.

`simulate.sgps` Boolean variable indicating whether to simulate SGP values for students based on test-specific Conditional Standard Errors of Measurement (CSEM). Test CSEM

data must be available for simulation and included in SGPstateData. This argument must be set to TRUE for confidence interval construction. Defaults to TRUE. This argument is passed to analyzeSGP.

save.old.summaries

A Boolean argument (defaults to NULL/TRUE which will save the @Summary slot before creating new summaries) indicating whether the call to summarizeSGP should save existing summaries in the @Summary slot. If overwrite.existing.data is TRUE, save.old.summaries will default to FALSE unless explictly set to TRUE.

save.intermediate.results

A Boolean argument (defaults to FALSE) indicating whether results should be save to the current directory after each step of the analysis.

calculate.simex

A character state acronym or list including state/csem variable, csem.data.vnames, csem.loss.hoss, simulation.iterations, lambda and extrapolation method. Returns both SIMEX adjusted SGP (SGP_SIMEX) as well as the percentile ranked SIMEX SGP (RANK_SIMEX) values as suggested by Castellano and McCaffrey (2017). Defaults to NULL, no simex calculations performed. Alternatively, setting the argument to TRUE sets the list up with state=state, lambda=seq(0,2,0.5), simulation.iterations=50, simex.sample.size=25000, extrapolation="linear" and save.matrices=TRUE.

calculate.simex.baseline

A character state acronym or list including state/csem variable, csem.data.vnames, csem.loss.hoss, simulation.iterations, lambda and extrapolation method. Defaults to NULL, no simex calculations performed. Alternatively, setting the argument to TRUE uses the same defaults as above along with simex.use.my.coefficient.matrices = TRUE. This argument is passed to analyzeSGP.

sgp.use.my.coefficient.matrices

A Boolean argument (defaults to FALSE/NULL) passed to analyzeSGP indicating whether previous coefficient matrices should be used as part of the analyses.

sgp.target.scale.scores

A Boolean argument (defaults to FALSE/NULL) passed to combineSGP indicating whether to calculate scale scores associated with SGP targets as part of the analyses.

sgp.target.scale.scores.only

A Boolean argument (defaults to FALSE/NULL) passed to combineSGP indicating whether ONLY to calculate scale scores associated with SGP targets as part of the combineSGP.

overwrite.existing.data

A Boolean argument (defaults to TRUE) indicating whether updateSGP should overwrite existing data/results from an earlier run as part of updateSGP.

update.old.data.with.new

A Boolean argument (defaults to TRUE) indicating whether updateSGP should add new data supplied in argument with_SGP_Data_LONG to existing longitudinal data or reduce data set to run analyses on only that which is provided.

output.updated.data

A Boolean argument (defaults to TRUE) indicating whether updateSGP should use outputSGP to save the new data added in with_sgp_data_LONG separately

in a new directory called "Updated_Data". Only relevant when 'overwrite.existing.data is FALSE and sgp.use.my.coefficient.matrices is TRUE. Output type controlled by the outputSGP.output.type argument.

sgPlot.demo.report
: A Boolean argument (defaults to TRUE) indicating whether updateSGP should produce just the demo student growth plots or those associated with all students in the last year.

plot.types
: A character vector (defaults to 'c(bubblePlot, studentGrowthPlot, growthAchievementPlot)') indicating what plot types to export from visualizeSGP.

outputSGP.output.type
: Specifies the type of output generated as part of intermediate step when adding addition data and using old coefficient matrices. Defaults are the defaults of outputSGP, LONG_Data, LONG_FINAL_YEAR_Data, WIDE_Data, and INSTRUCTOR_Data.

outputSGP.directory
: A a file path indicating where to save output files. Defaults to Data.

sgp.config
: List of analysis control parameters passed to analyzeSGP. For details on this argument see document for that function.

goodness.of.fit.print
: A Boolean variable passed to analyzeSGP indicating whether to print goodness of fit plots.

parallel.config
: Parallel computation configuration passed to abcSGP, analyzeSGP, combineSGP, and summarizeSGP. See documentation under those functions for details.

sgp.sqlite
: A Boolean argument (defaults to FALSE) indicating whether a SQLite database file of the essential SGP data should be created from the @Data slot and subsequently used to extract data subsets for analyses conducted in order to reduce the amount of RAM memory required. See full argument description in analyzeSGP.

SGPt
: Argument (defaults to NULL) indicating whether time dependent student growth percentile (SGPt) are calculate.

sgp.percentiles.equated
: Argument (defaults to NULL) passed to abcSGP and analyzeSGP indicating whether equated SGP are to be calculated.

sgp.percentiles.equating.method
: Character vector argument passed to analyzeSGP indicating type(s) of equating method to used if sgp.percentiles.equated=TRUE. Default is NULL indicating 'equipercentile' equating. Options include 'identity', 'mean', 'linear', and 'equipercentile'.

sgp.percentiles.calculate.sgps
: Boolean argument passed to abcSGP and analyzeSGP indicating whether to produce student growth percentiles as part of call to studentGrowthPercentiles. Default is TRUE. Setting to FALSE produces only coefficient matrices.

fix.duplicates
: Argument to control how analyzeSGP and combineSGP deal with duplicate records based upon the key of VALID_CASE, CONTENT_AREA, YEAR, and ID. The function currently warns of duplicate records and doesn't modify data.

If set to 'KEEP.ALL', analyzeSGP tries to fix the duplicate individual records by adding a '_DUP_***' suffix to the duplicate ID before running studentGrowthPercentiles in order to create unique records based upon the key. If needed, the @Data slot will be extended as necessary to accomodate additional student records and SGP results in combineSGP.

get.cohort.data.info
: Boolean argument passed to analyzeSGP indicating whether to calculate norm group cohort counts.

...
: Arguments to be passed to abcSGP for finer control over SGP calculations. For example, 'parallel.config' can be supplied to abcSGP for parallel computation.

Value

Returns and object of class SGP including additional analyses.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org>

See Also

prepareSGP and abcSGP

Examples

```
## Not run:
### Run analyses on all but final year's of data

Demonstration_Data_LONG <- subset(sgpData_LONG, YEAR
Demonstration_Data_LONG_2013_2014 <- subset(sgpData_LONG, YEAR

Demonstration_SGP <- abcSGP(
        sgp_object=Demonstration_Data_LONG,
        sgPlot.demo.report=TRUE)


### Run updateSGP on Demonstration_SGP and the 2013_2014 data

Demonstration_SGP <- updateSGP(
        what_sgp_object=Demonstration_SGP,
        with_sgp_data_LONG=Demonstration_Data_LONG_2013_2014)

## End(Not run)
```

visualizeSGP *Visualize data from SGP analyses*

Description

Utility function to produce a variety of graphical displays associated with student growth percentile/percentile growth trajectory results. Function currently includes facility to produce individual student growth and achievement plots, interactive bubble plots depicting summary growth and achievement data, and growth and achievement charts showing system level growth and achievement data as shown on the cover of *Educational Measurement: Issues and Practice* as part of Betebenner (2009) and Betebenner (2012).

Usage

```
visualizeSGP(
        sgp_object,
      plot.types=c("bubblePlot", "studentGrowthPlot", "growthAchievementPlot"),
        state,
        bPlot.years=NULL,
        bPlot.content_areas=NULL,
        bPlot.districts=NULL,
        bPlot.schools=NULL,
        bPlot.instructors=NULL,
        bPlot.styles=c(1),
        bPlot.levels=NULL,
        bPlot.level.cuts=NULL,
        bPlot.full.academic.year=TRUE,
        bPlot.minimum.n=10,
        bPlot.anonymize=FALSE,
        bPlot.prior.achievement=TRUE,
        bPlot.draft=FALSE,
        bPlot.demo=FALSE,
        bPlot.output="PDF",
        bPlot.format="print",
        bPlot.folder="Visualizations/bubblePlots",
        sgPlot.save.sgPlot.data=FALSE,
        sgPlot.years=NULL,
        sgPlot.content_areas=NULL,
        sgPlot.districts=NULL,
        sgPlot.schools=NULL,
        sgPlot.reports.by.school=TRUE,
        sgPlot.instructors=NULL,
        sgPlot.reports.by.instructor=FALSE,
        sgPlot.students=NULL,
        sgPlot.reports.by.student=FALSE,
        sgPlot.reports.group.vars=
         list(DISTRICT_NUMBER="DISTRICT_NUMBER", SCHOOL_NUMBER="SCHOOL_NUMBER"),
```

```
        sgPlot.header.footer.color="#4CB9CC",
        sgPlot.front.page=NULL,
        sgPlot.folder="Visualizations/studentGrowthPlots",
        sgPlot.folder.names="number",
        sgPlot.fan=TRUE,
        sgPlot.custom.trajectory=FALSE,
        sgPlot.sgp.targets=FALSE,
        sgPlot.sgp.targets.timeframe=3,
        sgPlot.anonymize=FALSE,
        sgPlot.cleanup=TRUE,
        sgPlot.demo.report=FALSE,
        sgPlot.produce.plots=TRUE,
        sgPlot.baseline=NULL,
        sgPlot.zip=TRUE,
        sgPlot.output.format="PDF",
        sgPlot.year.span=5,
        sgPlot.plot.test.transition=TRUE,
        gaPlot.years=NULL,
        gaPlot.content_areas=NULL,
        gaPlot.students=NULL,
        gaPlot.format="print",
        gaPlot.baseline=NULL,
        gaPlot.max.order.for.progression=NULL,
        gaPlot.start.points="Achievement Level Cuts",
        gaPlot.back.extrapolated.cuts=NULL,
        gaPlot.SGPt=NULL,
        gaPlot.folder="Visualizations/growthAchievementPlots",
        parallel.config=NULL)
```

Arguments

sgp_object	An object of class SGP containing long formatted data in the @Data slot that will be used for the production of student growth and achievement plots and system growth and achievement plots, summary data from summarizeSGP in the Summary slot for bubble plots. Alternatively, a properly formatted WIDE data set can be provided for production of student growth plots. Such data sets are produced from an SGP object when the sgPlot.save.sgPlot.data argument is set to TRUE.
plot.types	A character vector indicating what types of plots to construct. Currently available plots include bubblePlot, studentGrowthPlot, and growthAchievementPlot.
state	Acronym indicating state associated with the summaries for access to assessment program information embedded in SGPstateData.
bPlot.years	A vector indicating year(s) in which to produce bubblePlots using data available in sgp_object. If missing the function will use the last year available in the data to produce bubblePlots.
bPlot.content_areas	A vector indicating content area(s) to produce bubblePlots using data available in sgp_object. If missing the function will produce plots for all available

content areas provided in the data.

`bPlot.districts`
: A vector indicating districts to produce bubblePlots using data available in `sgp_object`. Consult `bubblePlot` styles to determine which bubblePlots styles accept specification for districts. Default is to produce plots for all available districts in the data.

`bPlot.schools`
: A vector indicating schools to produce bubblePlots using data available in `sgp_object`. Consult `bubblePlot` styles to determine which `bubblePlot` styles accept specification for schools. Default is to produce plots for all available schools in the data.

`bPlot.instructors`
: A vector indicating instructors to produce bubblePlots using data available in `sgp_object`. If missing the function will produce plots for all available instructors provided in the data where schools and districts represent relevant units to be represented by the specific `bubblePlot` style.

`bPlot.styles`
: A vector of positive integers indicating the types of bubblePlots to produce using data available in `sgp_object`. See associated documentation for example plots.

`bPlot.levels`
: A character vector of levels to be used to color bubbles in the `bubblePlot`. See associated documentation for example plots.

`bPlot.level.cuts`
: A vector of cuts to be used to distinguish levels used to color bubbles in the `bubblePlot`. See associated documentation for example plots.

`bPlot.full.academic.year`
: A Boolean argument (defaults to TRUE) indicating whether bubblePlots should use full academic year results if available.

`bPlot.minimum.n`
: A positive integer (defaults to 10) indicating the minimum size for summary values to be displayed in the bubblePlots.

`bPlot.anonymize`
: A Boolean argument (defaults to FALSE) indicating whether to anonymize bubblePlots school and district names that appear in the plots and data tips of the plots. For student level anonymization, the function utilizes the randomNames package to produce gender and ethnic correct names based upon gender and ethnicity codes available in `sgp_object@Data`.

`bPlot.prior.achievement`
: A Boolean argument (defaults to TRUE) indicating whether to produce bubblePlots using prior achievement as well as current achievement as the vertical dimension of the `bubblePlot`.

`bPlot.draft`
: A Boolean argument (defaults to FALSE) indicating whether to put an indicator on the chart noting that the results are draft and to not distribute.

`bPlot.demo`
: A Boolean argument (defaults to FALSE) indicating whether to produce demo student level plots (styles 150 and/or 153) for instructors.

`bPlot.output`
: Argument indicating the desired type of output format for bubble plots. Either 'PDF' (default) or 'PNG'.

bPlot.format Either "print" or "presentation" indicating whether to optimize the plot for print form (light background) or presentation form (dark background).

bPlot.folder Character vector indicating where bubblePlots should be placed. Default folder is "Visualizations/bubblePlots".

sgPlot.save.sgPlot.data
A Boolean argument indicating whether wide formatted data should be save as part of individual student report production. For operational student report production construction, using a wide formatted file for a large state can save in having to reshape the same long file multiple times.

sgPlot.years A vector indicating year for which to produce studentGrowthPlots. The supplied year indicates the *final* year associated with each student's studentGrowthPlot. If missing the function will use the last year available in the data to produce studentGrowthPlots.

sgPlot.content_areas
A vector indicating the content areas for which to produce studentGrowthPlots. If missing, the function will utilize all available content areas.

sgPlot.districts
A vector indicating which districts to produce studentGrowthPlots for. If missing the function will use *all* available districts in the data to produce studentGrowthPlots.

sgPlot.schools A vector indicating which schools to produce studentGrowthPlots for. If missing the function will use *all* available schools in the data to produce studentGrowthPlots. If both sgPlot.districts and sgPlot.schools are provided the function produces studentGrowthPlots for ALL students in the districts and schools provided.

sgPlot.reports.by.school
A Boolean variable indicating whether the reports should be collated as single reports in a single folder by school and deposited into a district/school/grade folder hierarchy. The default is TRUE which puts the reports into their appropriate district/school/grade folder.

sgPlot.instructors
A vector indicating which instructors to produce studentGrowthPlots for. If NULL and the argument sgPlot.reports.by.instructor is TRUE, the argument function will use *all* available instructors in the data to produce studentGrowthPlots. If sgPlot.districts and/or sgPlot.schools are provided the function produces studentGrowthPlots for ALL students in the districts and/or schools provided.

sgPlot.reports.by.instructor
A Boolean variable indicating whether the reports should be collated as single reports in a single folder by school and deposited into a district/school/grade folder hierarchy. The default is TRUE which puts the reports into their appropriate district/school/grade folder.

sgPlot.students
A vector of student IDs indicating which students to produce studentGrowthPlots for. If missing the function will use *all* available students in the data to produce studentGrowthPlots.

sgPlot.reports.group.vars
A list of variables used to group student reports. Default is list(DISTRICT_NUMBER='DISTRICT_NUMBER' SCHOOL_NUMBER='SCHOOL_NUMBER')

sgPlot.reports.by.student
: A Boolean variable indicating whether the reports should be collated as single reports in a single folder or deposited into a district/school/grade folder hierarchy. The default is FALSE which puts the reports into their appropriate district/school/grade slot.

sgPlot.header.footer.color
: Character vector (default is blue) indicating the color of the header/footer associated with the studentGrowthPlot. Another good color is goldenrod2.

sgPlot.front.page
: A path to a PDF to be used as the front page to the studentGrowthPlot. The default is missing so that no front page is attached to the studentGrowthPlot.

sgPlot.folder
: Character vector indicating where studentGrowthPlots should be placed. Note that studentGrowthPlots are placed within nested folders within this folder. Default folder is "Visualizations/studentGrowthPlots".

sgPlot.folder.names
: Either "name" or "number" (the default) indicating how the nested folder structure will be labeled that holds the studentGrowthPlots.

sgPlot.fan
: A Boolean argument (defaults to TRUE) indicating whether to produce projection fan on studentGrowthPlots.

sgPlot.custom.trajectory
: A Boolean argument (defaults to FALSE) or a character vector indicating the variable name containing the custom trajectory coordinates to be plotted.

sgPlot.sgp.targets
: A Boolean argument (defaults to TRUE) indicating whether to indicate SGP growth targets on studentGrowthPlots.

sgPlot.sgp.targets.timeframe
: An integer argument specifying the number of years forward associated with targets to be added to the student growth plots.

sgPlot.anonymize
: A Boolean argument (defaults to FALSE) indicating whether to anonymize studentGrowthPlots student, school and district names. For student level anonymization, the function utilizes the randomNames package to produce gender and ethnicity based names based upon gender and ethnicity codes available in sgp_object@Data.

sgPlot.cleanup
: A Boolean argument (defaults to TRUE) indicating whether to remove files produced by pdfLaTeX to produce studentGrowthPlot catalogs.

sgPlot.demo.report
: A Boolean argument (defaults to TRUE) indicating whether to just produce a sample studentGrowthPlot catalogs. Note: When producing studentGrowthPlots for an entire state, considerable resources are required to produce this many reports. We are actively working on parallelizing this functionality to reduce report production time by two orders of magnitude.

sgPlot.produce.plots
: A Boolean argument (defaults to TRUE) indicating whether to produce studentGrowthPlots. Useful when one just wants to produce wide formatted data without the actual student growth plots.

sgPlot.baseline
: Argument (defaults to NULL) indicating whether to use baseline referenced SGPs for student growth plot construction. If not set by user, argument will be set using SGPstateData which contains information on whether state is a cohort or baseline referenced system.

sgPlot.zip
: A Boolean argument (defaults to TRUE) indicating whether to zip school folders containing studentGrowthPlots.

sgPlot.output.format
: Argument indicating the desired type of output format for student growth plots. Either 'PDF' (default), 'PNG', 'PDF_PIECES', or 'JSON'.

sgPlot.year.span
: Integer argument (defaults to 5) indicating the number of years to display for the pdf student growth plot.

sgPlot.plot.test.transition
: Boolean argument (defaults to TRUE) indicating whether test transition data will be used IF it's available. Setting to FALSE ignore previous test transition data, which would indicate a desire to 'forget the past'.

gaPlot.years
: A vector indicating the year(s) for which to produce growthAchievementPlots. If missing the function will use the last year available in the data to produce the growthAchievementPlots.

gaPlot.content_areas
: A vector indicating content area(s) to produce growthAchievementPlots using data available in sgp_object. If missing the function will produce plots for all available content areas provided in the data.

gaPlot.students
: A vector of student IDs indicating which students to produce growthAchievementPlots for. If missing the function will use *all* available students in the data to produce growthAchievementPlots.

gaPlot.format
: Either 'print' or 'presentation' indicating whether to optimize the plot for print form (light background) or presentation form (dark background).

gaPlot.baseline
: Argument (defaults to NULL) indicating whether to calculate growth and achievement plots using percentile trajectories derived from baseline referenced coefficient matrices. If not set by user, argument will be set using SGPstateData which contains information on whether state is a cohort or baseline referenced system.

gaPlot.max.order.for.progression
: The maximum coefficient matrix order to use for each progression. Default is NULL which utilizes the maximum order available with the coefficient matrices.

gaPlot.start.points
: Either 'Achievement Level Cuts' or 'Achievement Percentiles' defining where the percentile trajectories of the growth achievement plot will start from.

gaPlot.folder
: Character vector indicating where growthAchievementPlots should be placed. The default folder is 'Visualizations/growthAchievementPlots'.

gaPlot.back.extrapolated.cuts
: A numeric value or Boolean vector indicating whether to create back extrapolated cutscores based upon nth percentile growth.

`gaPlot.SGPt` Boolean indicator for growth achievement plot indicating whether time dependent SGPs are to be used for student growth projections `SGPt`

`parallel.config`

A named list with, at a minimum, two elements indicating 1) the BACKEND package to be used for parallel computation and 2) the WORKERS list to specify the number of processors to be used in each major analysis. The BACKEND element can be set = to `FOREACH` or `PARALLEL`. Please consult the manuals and vignettes for information of these packages! The `analyzeSGP` help page contains more thorough explanation and examples of the `parallel.config` setup.

TYPE is a third element of the `parallel.config` list that provides necessary information when using FOREACH or PARALLEL packages as the backend. With BACKEND="FOREACH", the TYPE element specifies the flavor of 'foreach' backend. As of version 1.0-1.0, only "doParallel" is supported. TYPE=NA (default) produces summaries sequentially. If BACKEND = "PARALLEL", the `parallel` package will be used. This package combines deprecated parallel packages `snow` and `multicore`. Using the "snow" implementation of `parallel` the function will create a cluster object based on the TYPE element specified and the number of workers requested (see WORKERS list description below). The TYPE element indicates the users preferred cluster type (either "PSOCK" for socket cluster of "MPI" for an OpenMPI cluster). If Windows is the operating system, this "snow" implementation must be used and the TYPE element must = "PSOCK". Defaults are assigned based on operating system if TYPE is missing based on system OS. Unix/Mac OS defaults to the "multicore" to avoid worker node pre-scheduling and appears to be more efficient in these operating systems.

The WORKERS element is a list with GA_PLOTS (growth achievement plots) and SG_PLOTS (student growth plots) specifying the number of processors to be used. NOTE: choice of the number of cores is a balance between the number of processors available and the amount of RAM a system has; each system will be different and may require some adjustment.

Default is FOREACH as the back end, TYPE=NA and both plot WORKERS=1, which produces plots sequentially: 'list(BACKEND="FOREACH", TYPE=NA, WORKERS=list(GA_PLOTS=1, SG_PLOTS=1))'

Examples of various parallel configurations can be found in the examples for `analyzeSGP` and `summarizeSGP`.

Value

Function produces *numerous* (potentially hundreds of thousands) of pdf files in a folder structure specified by the user and supplied through arguments to the function.

Author(s)

Damian W. Betebenner <dbetebenner@nciea.org> and Adam Van Iwaarden <avaniwaarden@nciea.org>

References

Betebenner, D. W. (2012). Growth, standards, and accountability. In G. J. Cizek, *Setting Performance Standards: Foundations, Methods & Innovations. 2nd Edition* (pp. 439-450). New York:

Routledge.

Betebenner, D. W. (2009). Norm- and criterion-referenced student growth. Educational Measurement: Issues and Practice, 28(4):42-51.

See Also

bubblePlot, bubblePlot_Styles, studentGrowthPlot, growthAchievementPlot

Examples

```
## Not run:
## visualizeSGP is Step 5 of 5 of abcSGP
Demonstration_SGP <- sgpData_LONG
Demonstration_SGP <- prepareSGP(Demonstration_SGP)
Demonstration_SGP <- analyzeSGP(Demonstration_SGP)
Demonstration_SGP <- combineSGP(Demonstration_SGP)
Demonstration_SGP <- summarizeSGP(Demonstration_SGP)
visualizeSGP(Demonstration_SGP)

## Produce a DEMO catalog of student growth plots

visualizeSGP(
 sgp_object=Demonstration_SGP,
 plot.types="studentGrowthPlot",
 state="DEMO",
 sgPlot.demo.report=TRUE)

## Production of sample student growth and achievement plots

visualizeSGP(
 sgp_object=Demonstration_SGP,
 plot.types="studentGrowthPlot",
 state="DEMO",
 sgPlot.districts=470,
 sgPlot.schools=c(6418, 8008),
 sgPlot.header.footer.color="#4CB9CC")

## End(Not run)
```

Index