# Package 'SGP'

April 17, 2009

**Version** 0.0-4

**Date** 2009-02-15

**Title** Student Growth Percentile and Percentile Growth Projection/Trajectory Functions

**Depends** R (>= 2.7.0), splines, quantreg, grid, gtools

**Author** Damian W. Betebenner <dbetebenner@nciea.org>, with contributions from Jonathan Weeks, Jinnie Choi, Marie Huchton, Xin Wei, and Hi Shin Shim

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

**Description** Functions to calculate growth percentiles and percentile growth projections/trajectories for students using large scale, longitudinal assessment data. Functions use quantile regression to estimate the conditional density associated with each students achievement history. Percentile growth projections/trajectories are calculated using the coefficient matrices derived from the quantile regression analyses and specify what percentile growth is required for students to reach future achievement targets.

**LazyLoad** Yes

**LazyData** Yes

**License** file LICENSE

**Repository** CRAN

**Date/Publication** 2009-04-14 17:27:05

## R topics documented:

SGP-package *Functions to calculate student growth percentiles and percentile growth projections/trajectories following methodology found in Betebenner (2008)*

## Description

SGP contains two functions, `studentGrowthPercentiles` and `studentGrowthProjections`, used to calculate student growth percentiles and percentile growth projections/trajectories for students using large scale, longitudinal assessment data. These norm referenced growth values are currently used in state testing and accountability systems. The functions employ quantile regression (using the `quantreg` package) to estimate the conditional density associated with each student's achievement history. Percentile growth projections/trajectories are calculated using the coefficient matrices derived from the student growth percentile analyses.

## Details

| | |
|---|---|
| Package: | SGP |
| Type: | Package |
| Version: | 0.0-4 |
| Date: | 2009-2-15 |
| License: | Creative Commons Attribution + ShareAlike (by-sa) |
| LazyLoad: | yes |

Calculation of student growth percentiles 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 is provided in the examples of the `studentGrowthPercentiles` and `studentGrowthProjections` functions. Calculation of percentile growth projections/trajectories follows calculation of student growth percentiles and requires coefficient matrices derived during student growth percentile estimation.

## Author(s)

Damian W. Betebenner ⟨DBetebenner@nciea.org⟩

## References

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

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.

sgpData *Longitudinal Assessment Data*

## Description

A dataset comprising a subset of four years of state assessment data suitable for student growth percentile and percentile growth projection/trajectory analyses. The dataset is used in examples provided in the documentation for the `studentGrowthPercentiles` and `studentGrowthProjections` functions and serves as an exemplar for user construction of their own datasets for growth percentile analyses.

## Usage

```
data(sgpData)
```

## Format

A data frame of student level observations for four years on the following variables.

**`ID`** Student ID

**`GRADE_04`** Student Grade Tested 2004, possibly missing

**`GRADE_05`** Student Grade Tested 2005, possibly missing

**`GRADE_06`** Student Grade Tested 2006, possibly missing

**`GRADE_07`** Student Grade Tested 2007, possibly missing

**`SS_04`** Student Scale Score 2004, possibly missing

**`SS_05`** Student Scale Score 2005, possibly missing

**`SS_06`** Student Scale Score 2006, possibly missing

**`SS_07`** Student Scale Score 2007, possibly missing

## Note

All datasets used with `studentGrowthPercentiles` and `studentGrowthProjections` must be specifically formatted as *wide* format files. The first variable/column is the student ID variable. The next set of columns provide the grade of the student across all the years provided in the dataset (possibly missing). The last set of columns provide the scales scores of the student in the respective grades.

## Source

Anonymized state assessment data

## See Also

`studentGrowthPercentiles`, `studentGrowthProjections`

studentGrowthPercentiles
*Student Growth Percentiles*

## Description

Function to calculate student growth percentiles using up to 8 panels of large scale assessment data. Outputs growth percentiles for each student and supplies various options as function arguments. Output from this function is used to calculate percentile growth projections/trajectories.

## Usage

```
studentGrowthPercentiles(student.data,
                         num.panels,
                         num.prior = num.panels - 1,
                         subset.grade,
                         percentile.cuts = c(1, 35, 65, 99),
                         use.my.knots.and.boundaries = FALSE,
                         print.other.gp = FALSE,
                         rq.method = "br",
                         convert.0and100 = TRUE,
                         percuts.digits=2,
                         save.matrices = TRUE,
                         isotonize = TRUE,
                         convert.using.loss.hoss=TRUE,
                         goodness.of.fit=TRUE,
                         sgp.function.labels)
```

## Arguments

| | |
|---|---|
| student.data | Data frame containing longitudinal student data in wide format. Data must be formatted so that student ID is the first variable/column, student grade variables for each year, 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. The function renames variables internally. |
| num.panels | Number of panels in the data frame. |
| 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. |
| 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.

`percentile.cuts`
: Additional percentile cuts (supplied as a vector) associated with each student's conditional distribution. Default is to supply the 1st, 35th, 65th, and 99th growth percentile cuts (scale scores associated with those growth percentiles) for each student. If percentile cuts are not wanted, set `percentile.cuts=NULL`.

`use.my.knots.and.boundaries`
: Boolean argument (default is FALSE) specifying whether the growth percentile function calculates B-splines using pre-specified knots and boundaries from the user or whether the program creates its own and saves those results to the directory 'Knots_Boundaries'. *KNOTS* are vectors of cut-points used in B-spline interpolations and *BOUNDARIES* are vectors of length 2 specifying minimum and maximum values.

  If, `use.my.knots.and.boundaries=FALSE`, then knots and boundaries are created and saved to the subdirectory 'Knots_Boundaries' using the naming convention 'boundaries_SUBJECT_gNUMBER', and 'knots_SUBJECT_gNUMBER', where 'SUBJECT' is the subject tested and 'NUMBER' is the grade number. 'SUBJECT' and 'NUMBER' must be specified in `knot.boundary.labels` if `use.my.knots.and.boundaries=FALSE`.

  If `knots.and.boundaries=TRUE` then directory 'Knots_Boundaries' exists and contains knots and boundaries given in the form 'knots_SUBJECT_gNUMBER' and 'boundaries_SUBJECT_gNUMBER'. See example for how this can be accomplished within a larger master file.

`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.

`rq.method`
: An argument that defines the estimation method used in the quantile regression calculations. The default is the the `"br"` method referring to the Barrodale and Roberts l1 estimation detailed in Koenker (2005) and in the help for the quantile regression package.

`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.

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

`save.matrices`
: Boolean argument (defaults to TRUE) indicating whether the coefficient matrices derived by the quantile regression analyses are saved to the 'Coefficient_Matrices' directory for later use (e.g., for use in the percentile growth projection/trajectory analyses).

`isotonize`
: Boolean argument (defaults to TRUE) indicating whether quantile regression results are isotonized to prevent quantile crossing following the methods derived by Dette & Volgushev (2008).

`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 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 placed in the 'Goodnessi_of_Fit' directory and test whether the observed growth percentile distribution follows a normal distribution.

`sgp.function.labels`
: If `save.matrices=TRUE` or `save.knots.boundaries=TRUE`, the function requires the user to specify a list, `sgp.function.labels`, of the form `list(my.year= , my.subject= , my.grade= ).` The user specified values are used to save the coefficient matrices and/or knots/boundaries in an orderly fashion using an appropriate combination of year, subject and grade.

## 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 the specified grade using `subset.grade`. Function will subset those students with non-canonical grade progressions. If percentile growth projection/trajectory analyses are also desired, coefficient matrices from the quantile regression analyses must be saved for use in those analyses. The example provides a generic file for use in analyzing assessment data across multiple grades.

## Value

Function returns a data frame containing student IDs and the associated student growth percentiles. If percentile.cuts or other order growth percentiles are requested then additional variables are supplied in the returned data frame.

## Note

If using user specified knots and boundaries (`use.my.knots.and.boundaries=TRUE`), knot and boundary vectors must be available within the Knots_Boundaries directory using a specific naming conventions. See the example for syntax.

## Author(s)

Damian W. Betebenner ⟨dbetebenner@nciea.org⟩

## References

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

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.

Dette, H. & Volgushev, S. Non-crossing non-parametric estimates of quantile curves. Journal of the Royal Statistical Society B, 70(3), 609-627.

## See Also

studentGrowthProjections, sgpData

## Examples

```
################################################################################
####
#### Code for master file which computes student growth percentiles for 2007
#### using provided sgpData dataset
####
################################################################################

###
### Load student growth percentile (SGP) library
###

require(SGP)

###
### Calculate student growth percentiles
###

## 444444444444444444444444444444444444444444444444444444444444444

sgp_reading_2007_g4 <- studentGrowthPercentiles(
             student.data = sgpData, ### SGP Library data set
             num.panels = 4,
             num.prior = 1,
             subset.grade = 4,
             percuts.digits = 0,
             sgp.function.labels = list(my.year=2007, my.subject="Reading", my.grade=4))

## save(sgp_reading_2007_g4, file="Results_Data/sgp_reading_2007_g4.Rdata")

## 555555555555555555555555555555555555555555555555555555555555555

sgp_reading_2007_g5 <- studentGrowthPercentiles(
             student.data = sgpData,
             num.panels = 4,
             num.prior = 2,
             subset.grade = 5,
             percuts.digits = 0,
             use.my.knots.and.boundaries=TRUE,
             sgp.function.labels = list(my.year=2007, my.subject="Reading", my.grade=5))

## save(sgp_reading_2007_g5, file="Results_Data/sgp_reading_2007_g5.Rdata")

## 666666666666666666666666666666666666666666666666666666666666666
```

```
sgp_reading_2007_g6 <- studentGrowthPercentiles(
             student.data = sgpData,
             num.panels = 4,
             num.prior = 3,
             subset.grade = 6,
             percuts.digits = 0,
             use.my.knots.and.boundaries=TRUE,
             sgp.function.labels = list(my.year=2007, my.subject="Reading", my.grade=6))

## save(sgp_reading_2007_g6, file="Results_Data/sgp_reading_2007_g6.Rdata")

## 7777777777777777777777777777777777777777777777777777777777

sgp_reading_2007_g7 <- studentGrowthPercentiles(
             student.data = sgpData,
             num.panels = 4,
             num.prior = 3,
             subset.grade = 7,
             percuts.digits = 0,
             use.my.knots.and.boundaries=TRUE,
             sgp.function.labels = list(my.year=2007, my.subject="Reading", my.grade=7))

## save(sgp_reading_2007_g7, file="Results_Data/sgp_reading_2007_g7.Rdata")

###
### Bind grade results and save output
###

sgp_reading_2007_gall <- rbind(sgp_reading_2007_g4,
                               sgp_reading_2007_g5,
                               sgp_reading_2007_g6,
                               sgp_reading_2007_g7)

save(sgp_reading_2007_gall, file="Results_Data/sgp_reading_2007_gall.Rdata")
write.table(sgp_reading_2007_gall, file="Results_Data/sgp_reading_2007_gall.dat",
            sep=",", row.names=FALSE, quote=FALSE)

###
### End of Master Program for Student Growth Percentiles
###
```

---

`studentGrowthProjections`

*Student Growth Projections*

---

## Description

Function to calculate percentile growth projections/trajectories using up to 8 panels of large scale assessment data. Function produces growth percentiles sufficient for each student to reach a set of predefined scale score cut points in 1, 2, 3 and 4 time periods (most often years).

## Usage

```
studentGrowthProjections(student.data,
                         num.panels,
                         max.num.scores,
                         proj.function.labels,
                         num.prior.scores,
                         subset.grade,
                         chunk.size = 10000,
                         convert.0and100 = TRUE,
                         percentile.trajectories=c(1, 35, 65, 99),
                         isotonize = TRUE,
                         projcuts.digits=2)
```

## Arguments

`student.data` Data frame containing longitudinal student data in wide format. Data must be formatted so that student ID is the first variable/column, student grade variables for each year, 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. The function renames variables internally.

`num.panels` Number of panels in the data frame

`max.num.scores`
The MAXIMUM number of scores one wishes to use in the FIRST YEAR projection analysis. Function will perform analyses using `1, 2, ... max.num.scores` for the first year projections/trajectories. Note `max.num.scores < num.panels` and, in most cases, `max.num.scores = num.panels-1`. Current function implementation limits `max.num.scores` to at most 4.

`proj.function.labels`
List of labels used to retrieve the appropriate coefficient matrices, knots, and boundaries for the projections/trajectories. The list corresponds to the list used to save the coefficient matrices and knots and boundaries in the growth percentile analyses. The list is of the form `list(my.year= , my.subject= , my.grade= )`.

`num.prior.scores`
Number of prior scores used to calculate each of the 1, 2, 3 and 4 year projections. A list of length `max.number.scores` consisting of vectors of length 4.

The first component of the list (a vector of length 4, a 4-tuple) indicates how many prior scores are used in the calculation of the 1, 2, 3, and 4 year projections, respectively, in the case when *1 prior score* is utilized. The first element of the 4-tuple is used to calculate the 1 year projection. The second element of the 4-tuple indicates how many scores are used to calculate the 2 year projections. The third element of the 4-tuple indicates how many scores are used to calculate the 3 year projections. And the fourth element of the 4-tuple indicates how many scores are used to calculate the 4 year projections.

If calculation of a given prediction is not appropriate, then the component of the vector should be NA. For example, for students currently in the penultimate grade that is assessed, only a 1 year projection is calculated. Thus, the latter three elements of the vector are NA, c(??,NA,NA,NA)

The second component of the list (another 4-tuple) indicates the number of prior scores to be used in the calculation of the 1, 2, 3, and 4 year projections in the case when *2 prior scores* are utilized. Similarly, the third and fourth component of the list indicate how many prior scores are to be used in the calculation of the 1, 2, 3 and 4 year projections in the case when *3 and 4 priors scores*, respectively, are utilized.

The num.prior.scores used for projection calculation are based upon the quantile regression based coefficient matrices calculated when growth percentiles are calculated. If max.num.scores=1, then num.prior.scores is USUALLY list(c(1,2,3,4)). That is, because there is only 1 actual student score, only 1 score is used to calculate the 1st year projection. However, for the 2nd year, because there are now 2 scores available (the actual score used for the 1 year projection AND the 1 year projection itself), 2 scores are used to get the 2 year projection. However, in order to use 2 scores to get the 2 year projection, a prior 2nd order analysis (i.e., using 3 panels of data) must exist that relates the 2 predictor scores with the dependent variable. Hence, the values are dependent BOTH upon the number of predictors available AND also upon the prior analyses performed. Similarly, for the 3rd and 4th year projections, 3 and 4 scores, respectively, are used to get the 3 and 4 year projections.

When not supplied, the function tries to set the appropriate values. These are:

```
max.num.scores=1:  num.prior.scores=list(c(1,2,3,4))
max.num.scores=2:  num.prior.scores=list(c(1,2,3,4),
                                         c(2,3,4,4))
max.num.scores=3:  num.prior.scores=list(c(1,2,3,4),
                                         c(2,3,4,4),
                                         c(3,4,4,4))
max.num.scores=4:  num.prior.scores=list(c(1,2,3,4),
                                         c(2,3,4,4),
                                         c(3,4,4,4),
                                         c(4,4,4,4))
```

subset.grade Student grade level for sub-setting. If the data fed into the function contain multiple grades, setting subset.grade=5 selects out those students in grade five in the most recent year of the data. If missing, all data supplied are analyzed without any sub-setting occurring.

chunk.size An integer (default of 10,000) indicating what the number of rows in the block of data (so that the program doesn't overwhelm memory) fed into the internal growth projection function.

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.

`percentile.trajectories`
: Returns a vector of percentile trajectories (defaults to 1st, 35th, 65th, and 99th) associated with 1, 2, 3 and 4 year trajectories for each student in addition to the percentiles sufficient to reach the performance thresholds. If no percentile trajectories are desired, set `percentile.cuts=NULL`.

`isotonize`
: Boolean argument (defaults to TRUE) indicating whether quantile regression results are isotonized to prevent quantile crossing following the methods derived by Dette & Volgushev (2008).

`projcuts.digits`
: The number of digits (defaults to 2) percentile trajectories (if requested) are formatted.

## Value

Function returns a data frame consisting of student IDs and the associated percentile growth projections/trajectories. Depending upon the number of performance level cut-scores and percentile trajectories specified, the data frame contains 1, 2, 3 and 4 year projections/trajectories for each performance level cut-point supplied (enumerated 'LEVEL1', 'LEVEL2', 'LEVEL3', ...) and each percentile cut specified (enumerated, for example, 'CUT_1', 'CUT_35', 'CUT_65', 'CUT_99').

## Note

Use of this function assumes prior calculation of student growth percentiles with `save.matrices=TRUE` (saving the coefficient matrices associated with student growth percentile calculations within the 'Coefficient_Matrices' directory). Additionally percentile growth projections/trajectories require future achievement levels (e.g., performance level cut-scores) against which percentile growth projections/trajectories are calculated. These values must be available in the 'Cutscores' directory using specific naming conventions. See the example for syntax on how to create and save cut-score vectors.

## Author(s)

Damian W. Betebenner ⟨dbetebenner@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.

Dette, H. & Volgushev, S. Non-crossing non-parametric estimates of quantile curves. Journal of the Royal Statistical Society B, 70(3), 609-627.

## See Also

`studentGrowthPercentiles`, `sgpData`

## Examples

```
###############################################################################
####
#### Code for master file which computes percentile growth
#### projections/trajectories associated with 2004 to 2006
#### data using provided sgpData
####
###############################################################################

###
### Establish and Save Reading Cut Scores
###

if (is.na(file.info("Cutscores")$isdir)) {
    dir.create("Cutscores")
}

cutscores_Reading_g3 <- c(465.5, 525.5, 655.5)
cutscores_Reading_g4 <- c(516.5, 571.5, 670.5)
cutscores_Reading_g5 <- c(537.5, 587.5, 690.5)
cutscores_Reading_g6 <- c(542.5, 599.5, 695.5)
cutscores_Reading_g7 <- c(566.5, 619.5, 715.5)

save(cutscores_Reading_g3, file="Cutscores/cutscores_Reading_g3.Rdata")
save(cutscores_Reading_g4, file="Cutscores/cutscores_Reading_g4.Rdata")
save(cutscores_Reading_g5, file="Cutscores/cutscores_Reading_g5.Rdata")
save(cutscores_Reading_g6, file="Cutscores/cutscores_Reading_g6.Rdata")
save(cutscores_Reading_g7, file="Cutscores/cutscores_Reading_g7.Rdata")

###
### Load Data
###

data(sgpData)

###
### Create subset of data for 2004 to 2006
###  (for growth to standard comparisons)
###

attach(sgpData)
sgpData <- data.frame(ID, GRADE_04, GRADE_05, GRADE_06, SS_04, SS_05, SS_06)
detach(sgpData)

###
### Submit to Growth Projection Function
###

##
## 333333333333333333333333333333333333333333333333333333333
##
```

```
##
## 3rd graders receiving projections
##

sgp_g2s_reading_2007_g3 <- studentGrowthProjections(
             student.data = sgpData,
             num.panels = 3,
             max.num.scores = 1,
             num.prior.scores=list(c(1,2,3,3)),
             proj.function.labels = list(my.year=2007, my.subject="Reading", my.grade=3),
             subset.grade = 3,
             percentile.trajectories=c(1,35,50,65,99),
             projcuts.digits=0)

## save(sgp_g2s_reading_2007_g3, file="Results_Data/sgp_g2s_Reading_2007_g3.Rdata")

##
## 4444444444444444444444444444444444444444444444444444444444
##

sgp_g2s_reading_2007_g4 <- studentGrowthProjections(
             student.data = sgpData,
             num.panels = 3,
             max.num.scores = 2,
             num.prior.scores=list(c(1,2,3,NA), c(2,3,3,NA)),
             proj.function.labels = list(my.year=2007, my.subject="Reading", my.grade=4),
             subset.grade = 4,
             percentile.trajectories=c(1,35,50,65,99),
             projcuts.digits=0)

## save(sgp_g2s_reading_2007_g4, file="Results_Data/sgp_g2s_Reading_2007_g4.Rdata")

##
## 5555555555555555555555555555555555555555555555555555555555555
##

sgp_g2s_reading_2007_g5 <- studentGrowthProjections(
             student.data = sgpData,
             num.panels = 3,
             max.num.scores = 3,
             num.prior.scores=list(c(1,2,NA,NA), c(2,3,NA,NA), c(3,3,NA,NA)),
             proj.function.labels = list(my.year=2007, my.subject="Reading", my.grade=5),
             subset.grade = 5,
             percentile.trajectories=c(1,35,50,65,99),
             projcuts.digits=0)

## save(sgp_g2s_reading_2007_g5, file="Results_Data/sgp_g2s_Reading_2007_g5.Rdata")

##
## 6666666666666666666666666666666666666666666666666666666666666
##

sgp_g2s_reading_2007_g6 <- studentGrowthProjections(
```

```
                student.data = sgpData,
                num.panels = 3,
                max.num.scores = 3,
                num.prior.scores=list(c(1,NA,NA,NA), c(2,NA,NA,NA), c(3,NA,NA,NA)),
                proj.function.labels = list(my.year=2007, my.subject="Reading", my.grade=6),
                subset.grade = 6,
                percentile.trajectories=c(1,35,50,65,99),
                projcuts.digits=0)

## save(sgp_g2s_reading_2007_g6, file="Results_Data/sgp_g2s_Reading_2007_g6.Rdata")

###
### Combine all the different grade growth projection files into a single file
### and write the results to a comma separated data set
###

sgp_g2s_reading_2007_gall <- rbind(sgp_g2s_reading_2007_g3,
                                   sgp_g2s_reading_2007_g4,
                                   sgp_g2s_reading_2007_g5,
                                   sgp_g2s_reading_2007_g6)

save(sgp_g2s_reading_2007_gall, file="Results_Data/sgp_g2s_reading_2007_gall.Rdata")
write.table(sgp_g2s_reading_2007_gall, file="Results_Data/sgp_g2s_reading_2007_gall.dat",
            sep=",", row.names=FALSE, quote=FALSE)


###
### End of Master Program for Percentile Growth Projections/Trajectories
###
```

# Index