The R EdSurvey Package

Analyzing NAEP and TIMSS Data Using R: Day 4

Presenters: Sinan Yavuz, Ting Zhang, Emmanuel Sikali

October 2021

NAEP
NATIONAL ASSESSMENT OF EDUCATIONAL PROGRESS

Workshop Goal

Provide participants with an overview of the methods used to analyze national and international large-scale assessment data using the R package `EdSurvey`

Follow along in edsurvey_training_day4.R

Outline of EdSurvey Workshop - Part 1

Data Analysis and Modeling

- Achievement-level analysis
- Gap analysis
- Doing your own analyses using NAEP/TIMSS data

Other Topics

- How to get NAEP Restricted-Use Data;
- NAEP funding streams and opportunities
- Challenges and opportunities in NAEP that NCES has identified as needing innovative research and applications

Conclusing Remarks

Achievement Level Analysis

NAEP
NATIONAL ASSESSMENT OF EDUCATIONAL PROGRESS

Achievement Level Analysis and Benchmark Analysis

- NAEP
  - Intended to measure to what extent students' achievement matches the expected achievement defined in the NAEP assessment frameworks
- TIMSS
  - Uses *international benchmarks* as defined in the TIMSS assessment frameworks

- Related Documentation - Analyses-Using-Achievement-Levels-Based-on-Plausible-Values-NAEP-April-2017.pdf

Achievement Level Analysis and Benchmark Analysis

- NAEP
  - Three levels - ***Basic***, ***Proficient***, and ***Advanced*** - are defined for each subject and each grade, with cut scores for each level determined through a standard-setting process.
- TIMSS
  - Four levels - ***Low***, ***Intermediate***, ***High***, and ***Advanced*** - are defined for each subject and each grade, with cut scores for each level
- Standard-setting process presented in two ways
  - Discrete - percentage at an achievement level
  - Cumulative - percentage at or above an achievement level

Discrete vs. Cumulative - NAEP

- Discrete vs. Cumulative
  - Discrete: the percentage of students performing within each achievement level, counted separately from the other levels. These categories are the percentages of students scoring **below *Basic***, **at *Basic***, **at *Proficient***, and **at *Advanced***. The percentages at all mutually exclusive achievement levels add up to 100 percent
  - Cumulative: the percentage of students performing at or above each achievement level. These categories are percentages of students scoring **below *Basic***, **at or above *Basic***, **at or above *Proficient***, and **at *Advanced***. Except below Basic and at Advanced, the other two cumulative levels include students at the specific and all higher levels. Since they are not mutually exclusive, it is not meaningful to add all of these four percentages of cumulative achievement levels

Discrete vs. Cumulative - TIMSS

- Discrete vs. Cumulative
  - Discrete: the percentage of students performing within each benchmark, counted separately from the other levels. These categories are the percentages of students scoring **below *Low***, **at *Low***, **at *Intermediate***, **at *High***, and **at *Advanced***. The percentages at all mutually exclusive benchmarks add up to 100 percent
  - Cumulative: the percentage of students performing at or above each benchmark. These categories are percentages of students scoring **below *Low***, **at or above *Low***, **at or above *Intermediate***, **at or above *High***, and **at *Advanced***. Except **below *Low*** and **at *Advanced***, the other three cumulative levels include students at the specific and all higher levels. Since they are not mutually exclusive, it is not meaningful to add all of these four percentages of cumulative benchmarks

Loading NAEP and TIMSS

```
library(EdSurvey)
sdf <- readNAEP(system.file("extdata/data", "M36NT2PM.dat",
                            package="NAEPprimer"))

downloadTIMSS(years = c(2019), root = "C:/")

TIMSS19<- readTIMSS(paste0(edsurveyHome, "TIMSS/2019"),
                    countries = c("usa"), gradeLvl = "4")
```

Achievement Level Analysis

`achievementLevels()`: computes the percentages of students by achievement level (at or above the achievement level cut points). See details in `?achievementLevels`.

- Each NAEP data set coded with year's cut points
  - use `showCutPoints()` to print a summary

```
showCutPoints(sdf)
```

```
## Achievement Levels:
##   Mathematics:  262, 299, 333
```

```
showCutPoints(TIMSS19)
```

```
## Achievement Levels:
##   Low International Benchmark:  400
##   Intermediate International Benchmark:  475
##   High International Benchmark:  550
##   Advanced International Benchmark:  625
```

Discrete vs Cumulative - NAEP

```
ach <- achievementLevels("composite", data = sdf,
                         returnCumulative = TRUE)
ach
```

```
##
## AchievementVars: composite
##
## Achievement Level Cutpoints:
## 262 299 333
##
## Plausible values: 5
## jrrIMax: 1
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## full data n: 17606
## n used: 16915
##
##
## Discrete
##  composite_Level      N      wtdN   Percent StandardError
##      Below Basic 5731.2 5779.5052 34.132690     0.9744207
```

Discrete vs Cumulative - NAEP

- To get only discrete or only cumulative summary tables

```
ach$discrete
ach$cumulative
```

- You can identify your cut-points

```
achievementLevels("composite", data = sdf,
                  returnCumulative = TRUE,
                  cutpoints = c(250,300,350))
```

Discrete vs Cumulative - TIMSS

- We can summarize similar tables for TIMSS

```
achievementLevels("mmat", data = TIMSS19, returnCumulative = TRUE)
```

```
##
## AchievementVars: mmat
##
## Achievement Level Cutpoints:
## 400 475 550 625
##
## Plausible values: 5
## jrrIMax: 1
## Weight variable: 'totwgt'
## Variance method: jackknife
## JK replicates: 150
## full data n: 10115
## n used: 8776
##
##
## Discrete
##                              mmat_Level      N      wtdN   Percent StandardError
##        Below Low International Benchmark  628.0  257375.1  6.835698     0.5524873
##           At Low International Benchmark 1511.6  624576.3 16.588297     0.7504469
##  At Intermediate International Benchmark 2704.4 1151681.9 30.587841     0.6580029
##          At High International Benchmark 2732.8 1207739.0 32.076676     1.0201220
##      At Advanced International Benchmark 1199.2  523790.1 13.911488     0.7492847
```

Additional Covariates

- A covariate can be added

```
ach1 <- achievementLevels(c("composite", "dsex"), data = sdf)
ach1$discrete
```

```
##   composite_Level   dsex      N      wtdN   Percent StandardError
## 1     Below Basic   Male 2880.8 2865.6455 16.923973     0.5590578
## 2        At Basic   Male 3266.2 3236.4034 19.113601     0.4993938
## 3   At Proficient   Male 1877.2 1910.7861 11.284749     0.4091708
## 4     At Advanced   Male  461.8  499.1392  2.947824     0.2579418
## 5     Below Basic Female 2850.4 2913.8597 17.208717     0.6094830
## 6        At Basic Female 3429.4 3343.8146 19.747951     0.4428114
## 7   At Proficient Female 1788.8 1783.9704 10.535800     0.4126107
## 8     At Advanced Female  360.4  378.8444  2.237385     0.1944887
```

Aggregate by Additional Covariates

- Aggregate by selected characteristics
  - the percentage distribution of students by achievement levels (**discrete** or **cumulative**) and selected characteristics (specified in `aggregateBy`)

```
ach2 <- achievementLevels(c("composite", "dsex"),
                          aggregateBy = "dsex", data = sdf)
ach2$discrete
```

```
##   composite_Level   dsex      N      wtdN   Percent StandardError
## 1     Below Basic   Male 2880.8 2865.6455 33.666050     1.0951825
## 2        At Basic   Male 3266.2 3236.4034 38.021772     0.9537470
## 3   At Proficient   Male 1877.2 1910.7861 22.448213     0.7257305
## 4     At Advanced   Male  461.8  499.1392  5.863965     0.5081607
## 5     Below Basic Female 2850.4 2913.8597 34.604399     1.1154848
## 6        At Basic Female 3429.4 3343.8146 39.710456     0.8650729
## 7   At Proficient Female 1788.8 1783.9704 21.186066     0.8148916
## 8     At Advanced Female  360.4  378.8444  4.499079     0.3888590
```

Aggregate by Subject Scale

- Aggregate by a subject scale or subscale
  - the percentage distribution of students by selected characteristics *within* a specific achievement level.

```
ach3 <- achievementLevels(c("composite", "dsex"),
                          aggregateBy = "composite", data = sdf)
ach3$discrete
```

```
##   composite_Level   dsex      N      wtdN  Percent StandardError
## 1     Below Basic   Male 2880.8 2865.6455 49.58289     0.9486797
## 2        At Basic   Male 3266.2 3236.4034 49.18383     0.8020508
## 3   At Proficient   Male 1877.2 1910.7861 51.71616     1.1913055
## 4     At Advanced   Male  461.8  499.1392 56.85063     2.0076502
## 5     Below Basic Female 2850.4 2913.8597 50.41711     0.9486797
## 6        At Basic Female 3429.4 3343.8146 50.81617     0.8020508
## 7   At Proficient Female 1788.8 1783.9704 48.28384     1.1913055
## 8     At Advanced Female  360.4  378.8444 43.14937     2.0076502
```

Aggregate by a Variable Combination

- Aggregate by more than one variable
  - `iep` Student classified as having a disability

```
dsex_iep <- achievementLevels(c("composite", "dsex", "iep"),
                              aggregateBy = c("dsex", "iep"),
                              data = sdf)
```

- Divides into number of `dsex` levels times number of `iep` levels, where each category adds up to 100 across achievement levels.

```
searchSDF("dsex",data = sdf, levels = TRUE)
searchSDF("iep",data = sdf, levels = TRUE)
```

Aggregate by a Variable Combination

```
dsex_iep$discrete
```

```
##    composite_Level   dsex iep      N       wtdN    Percent StandardError
## 1      Below Basic   Male Yes  810.2  753.47862 66.4635116     2.0061208
## 2         At Basic   Male Yes  281.6  282.52828 24.9215056     2.0783210
## 3    At Proficient   Male Yes   72.8   85.69544  7.5590995     1.4614600
## 4      At Advanced   Male Yes    9.4   11.97026  1.0558833     0.7673700
## 5      Below Basic Female Yes  471.2  465.33346 76.4954517     2.9245271
## 6         At Basic Female Yes  108.8  106.71734 17.5430994     2.0864253
## 7    At Proficient Female Yes   31.2   34.36986  5.6500084     1.6430596
## 8      At Advanced Female Yes    2.8    1.89454  0.3114405     0.2601418
## 9      Below Basic   Male  No 2067.6 2111.69806 28.6261355     1.0630715
## 10        At Basic   Male  No 2982.6 2952.86086 40.0289211     1.0125447
## 11   At Proficient   Male  No 1804.4 1825.09062 24.7408909     0.7840337
## 12     At Advanced   Male  No  452.4  487.16896  6.6040524     0.5558956
## 13     Below Basic Female  No 2379.0 2448.49754 31.3451478     1.2051321
## 14        At Basic Female  No 3318.8 3236.55190 41.4336531     0.9207178
## 15   At Proficient Female  No 1757.4 1749.56228 22.3975264     0.8954779
## 16     At Advanced Female  No  356.8  376.79678  4.8236727     0.4233201
```

Summary

- Two methods to calculate percentages
  - **discrete** - percentage at an achievement level
  - **cumulative** - percentage at or above an achievement level
- You can use Covariates in one of two ways
  - the percentage distribution of students by selected characteristics *within* a specific achievement level. (`aggregateBy` includes a subject scale or subscale)
  - the percentage distribution of students by achievement levels and selected characteristics (specified in `aggregateBy`)
- See the cut points with `showCutPoints()`

Self-Reflection - achievementLevels

Ask yourself: Use EdSurvey functions to create a summary table using `achievementLevels()` with these parameters:

```
# for reference
help(package = "EdSurvey")
searchSDF("text", sdf)
levelsSDF("myvar", sdf)
```

- `composite` math performance across a variable
- aggregated by language other than English spoken in home
- the percentage of students performing **at or above** each achievement level

Self-Reflection - achievementLevels

Scenario Result:

```
exerciseAL <- achievementLevels(c("composite", "b018201"), aggregateBy = c(
                                data = sdf, returnCumulative = TRUE)
exerciseAL$cumulative
```

```
##           composite_Level             b018201      N       wtdN   Percent StandardError
## 1             Below Basic               Never 2881.2 2741.81706 29.967754     1.1059320
## 2       At or Above Basic               Never 6642.8 6407.40734 70.032246     1.1059320
## 3  At or Above Proficient               Never 2726.0 2694.99580 29.456003     0.9354285
## 4             At Advanced               Never  490.6  522.99356  5.716261     0.5301935
## 5             Below Basic     Once in a while  992.8  993.62918 29.841115     1.2790694
## 6       At or Above Basic     Once in a while 2335.2 2336.10292 70.158885     1.2790694
## 7  At or Above Proficient     Once in a while 1024.8 1037.94788 31.172114     1.4081560
## 8             At Advanced     Once in a while  205.4  218.33310  6.557077     0.7432284
## 9             Below Basic       Half the time  495.4  594.63374 41.495053     2.0402629
## 10      At or Above Basic       Half the time  682.6  838.38946 58.504947     2.0402629
## 11 At or Above Proficient       Half the time  247.6  334.71472 23.357244     2.0037955
## 12            At Advanced       Half the time   51.2   69.94572  4.880990     0.9963799
## 13            Below Basic All or most of time 1006.0 1225.65906 46.291461     1.6649333
## 14      At or Above Basic All or most of time 1127.0 1422.04104 53.708539     1.6649333
## 15 At or Above Proficient All or most of time  341.0  455.24954 17.194151     1.4260673
## 16            At Advanced All or most of time   47.4   59.51578  2.247829     0.5702864
```

Gap Analysis: Estimating the Difference in Two Statistics

NAEP
NATIONAL ASSESSMENT OF EDUCATIONAL PROGRESS

Gap Analysis

`gap()`: estimate the difference in a statistic for two groups in the population. A gap occurs when one group outperforms another group and the difference in the two statistics are statistically significant.

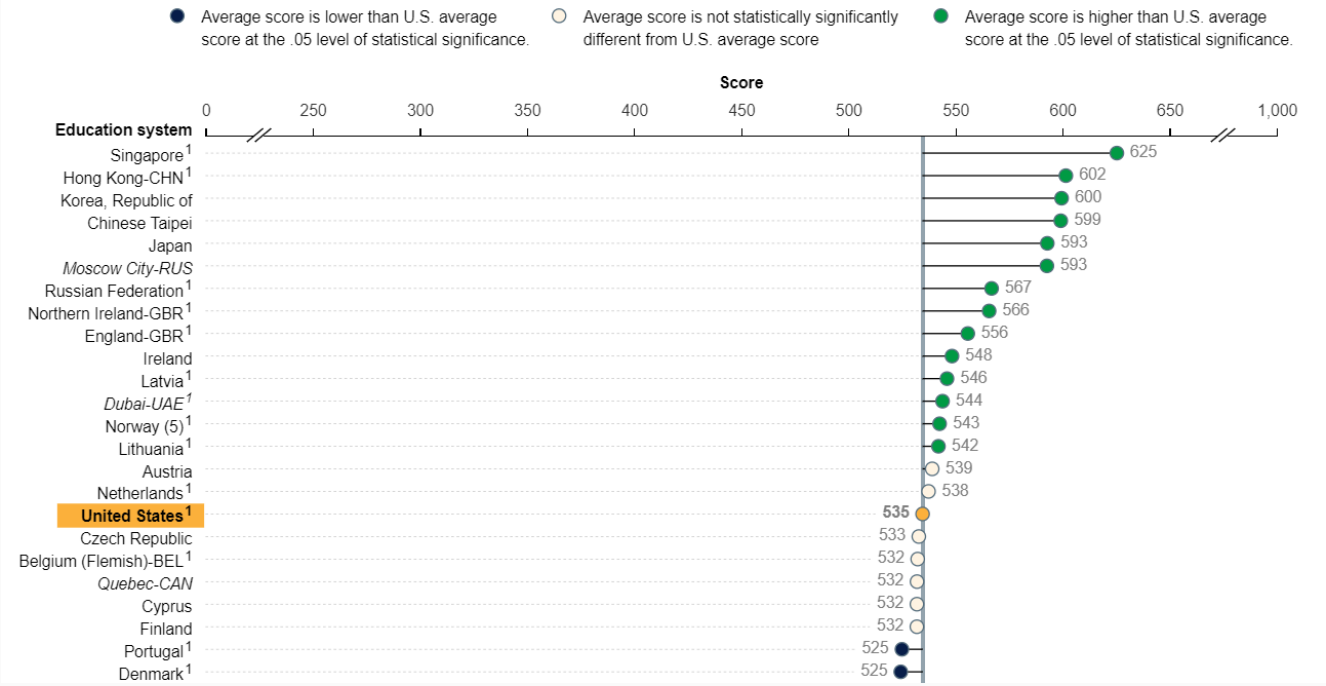
- Statistics can be any of
  - mean scores
  - student group percentages
  - achievement level percentages
  - percentiles
- Variance estimation

$$Var(\theta_A - \theta_B) = Var(\theta_A) + Var(\theta_B) - 2Cov(\theta_A, \theta_B)$$

- Related Documentation - EdSurvey-Gap.pdf
- Related Documentation - EdSurvey-TIMSS.pdf

Typical Gap Comparisons

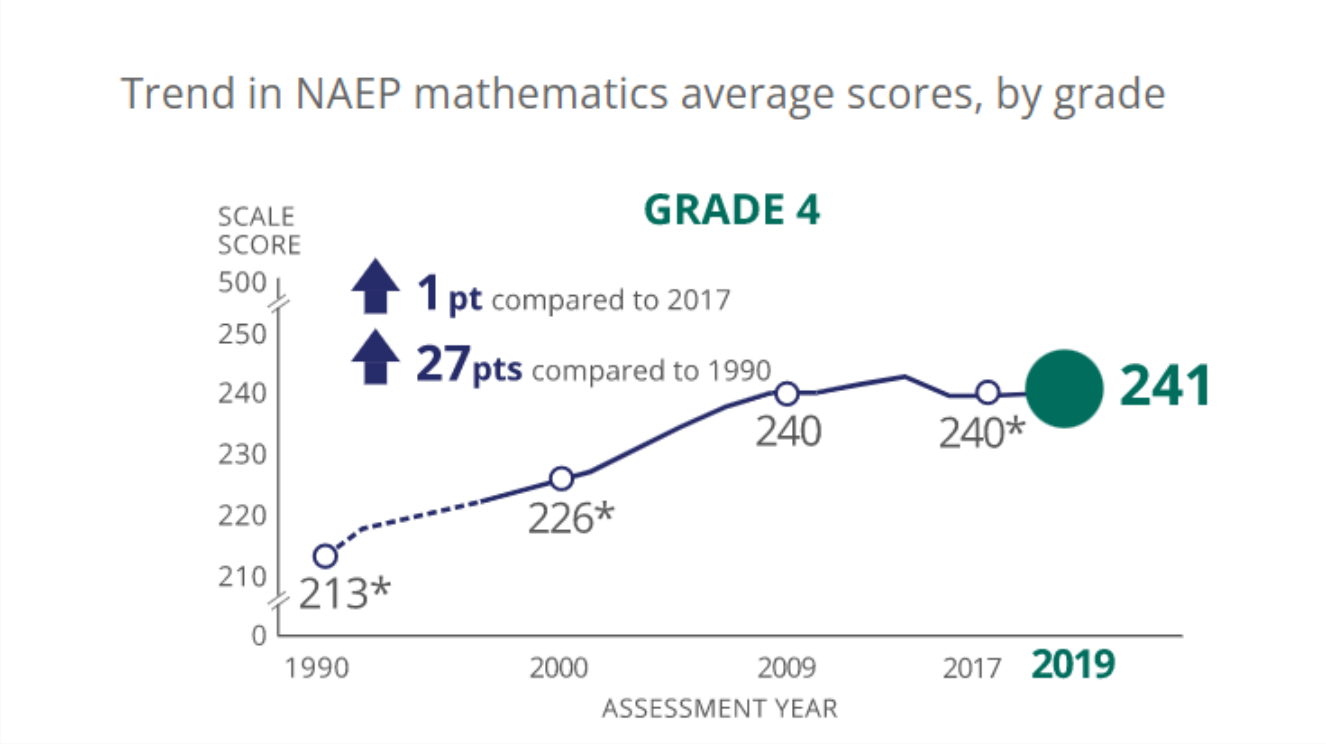
- Comparisons of different groups/jurisdictions within years
  - Female in 2019 to Male in 2019
  - USA in 2019 to Singapore in 2019
  - USA in 2019 to the international average in 2019 (part/whole)



Source: 2019 TIMSS report

Typical Gap Comparisons (cont.)

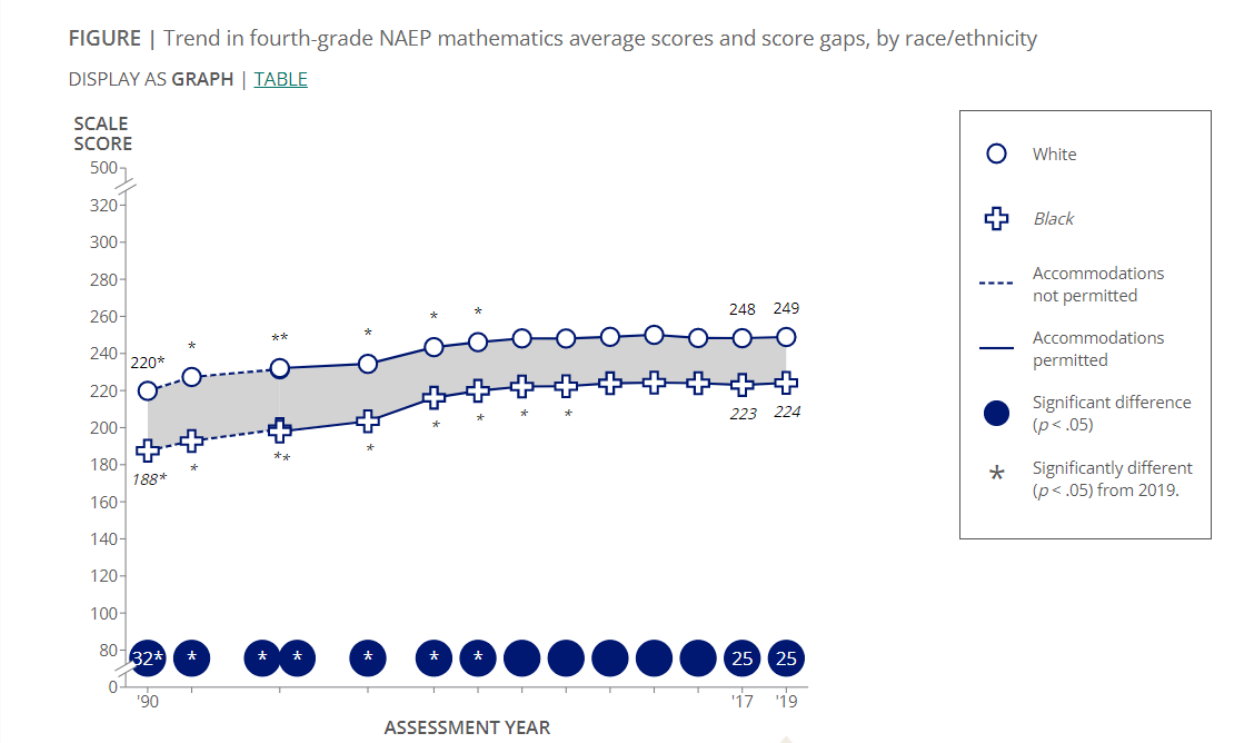
- Comparisons of the same group/jurisdiction between years
  - Female in 2019 to Female in 2017
  - USA in 2019 to USA in 2017



Source: NAEP Math Report 2019

Typical Gap Comparisons (cont.)

- Comparisons of the gap of different groups/jurisdictions between years
  - The BW gap in USA in 2019 compared to the gap in USA in 2017
  - The difference between USA and Canada in 2019, compared to the same gap in 2015



Within Year Comparisons

Comparison between students groups

- `groupA`: defines a condition to subset data
  - `dsex %in% "Male"`
- `groupB`: defines a condition to subset data to compare to `groupA`
  - `dsex %in% "Female"`

```
mathGap <- gap(variable = "composite", data = sdf,
               groupA = dsex %in% "Male", groupB = dsex %in% "Female")
```

Within Year Comparisons (cont.)

- Mean score results returned with `mathGap$results`

```
mathGap$results
```

```
##   estimateA estimateAse estimateB estimateBse   diffAB     covAB  diffABse diffABpValue    dofAB
## 1  276.7235   0.8207151  275.0458   0.9402535 1.677756 0.5676583 0.6498719   0.01259479 53.70969
```

- `estimateA`/`estimateB` - Value of estimate
- `estimateAse`/`estimateBse` - Standard error of estimates
- `diffAB/diffABse` - Difference between estimateA and estimateB and standard error of the difference in group estimates
- `covAB` - The covariance used in calculating diffABse
- `diffABpValue` - The p-value associated with the t-test used for the hypothesis test that diffAB is zero
- `dofAB` - The degrees of freedom used in calculating diffABpValue

Within Year Comparisons (cont.)

- Percentage results returned with `mathGap$percentage`

```
mathGap$percentage
```

```
##       pctA    pctAse     pctB    pctBse    diffAB      covAB diffABse diffABpValue    dofAB
## 1 50.27015 0.5016796 49.72985 0.5016796 0.5402935 -0.2516824 1.003359    0.5924778 53.45667
```

Same as `mathGap$results` except:

- `pctA` / `pctB` - The percent of respondents in groups
- `pctAse` / `pctBse` - Standard errors of the percent of respondents in groups

Within Year, Benchamrks/Achievement levels

Comparison by achievement level

```
mathGap2 <- gap(variable = "composite", data = sdf,
                groupA = dsex %in% "Male", groupB = dsex %in% "Female",
                achievementLevel = c("Basic", "Proficient", "Advanced"))

mathGap2$results
```

```
##         achievementLevel estimateA estimateAse estimateB estimateBse    diffAB     covAB  diffABse diffABpValue    dofAB
## 1      At or Above Basic 66.333950   1.0951825 65.395601    1.115485 0.9383491 0.6757857 1.0450644  0.373988103 45.35383
## 2 At or Above Proficient 28.312178   0.8635866 25.685145    1.007338 2.6270329 0.4581070 0.9188566  0.005776729 62.16756
## 3            At Advanced  5.863965   0.5081607  4.499079    0.388859 1.3648866 0.1156585 0.4220445  0.002272814 45.59694
```

Within Year, Percentiles

Comparison by percentile

```
mathGap3 <- gap(variable = "composite", data = sdf,
                groupA = dsex %in% "Male", groupB = dsex %in% "Female",
                percentiles = c(25, 50, 75))

mathGap3$results
```

```
##   percentiles estimateA estimateAse estimateB estimateBse   diffAB     covAB  diffABse diffABpValue    dofAB
## 1          25  252.6120   1.1653238  251.2700    1.167151 1.341967 0.6013256 1.2318965   0.28215615 42.37247
## 2          50  278.0204   0.8174635  276.9481    1.055243 1.072283 0.4379389 0.9517912   0.26406360 64.87830
## 3          75  302.4865   0.8225389  299.7701    1.283120 2.716347 0.4558281 1.1879866   0.02572763 60.68734
```

Between Datasets Comparisons

Workflow for conducting between datasets comparisons, including between years or between jurisdictions/educational system comparisons:

- Load the data into R
- Rename variables or recode values for consistency across datasets
- Create an `edsurvey.data.frame.list` with all the datafiles in it
- Recode the variable values as necessary
- Run your analysis

- Related Documentation - EdSurvey-Trend.pdf
- Related Documentation - EdSurvey-TIMSS.pdf

Between Year Comparisons

- Download your TIMSS datasets

```
downloadTIMSS(year=c(2019, 2015, 2011), root = "C:/")
```

- Read in datasets from multiple years

```
TIMSS11<- readTIMSS("C:/TIMSS/2011",
                    countries = c("usa"), gradeLvl = "4")
TIMSS15<- readTIMSS("C:/TIMSS/2015",
                    countries = c("usa"), gradeLvl = "4")
TIMSS19<- readTIMSS("C:/TIMSS/2019",
                    countries = c("usa"), gradeLvl = "4")
```

This operation takes several minutes to run the first time and then runs nearly instantly after that. Subsequent calls to `readTIMSS` are stored on the user's drive for easy access.

Between Year Comparisons (cont.)

- Combine data from each year into an `edsurvey.data.frame.list` and assign a label to each year.

```
trend <- edsurvey.data.frame.list(list(TIMSS19, TIMSS15, TIMSS11),
                  labels=c("TIMSS 2019", "TIMSS 2015", "TIMSS 2011"))
```

Between Year Comparisons (cont.)

- Check for data consistency across datasets.

```
#check the consistency of the gender variable
searchSDF("itsex", trend, level=TRUE)
```

```
##   variableName           Labels                                    Levels TIMSS 2019 TIMSS 2015 TIMSS 2011
## 1       itsex *SEX OF STUDENTS*    1. GIRL; 2. BOY; 9. OMITTED OR INVALID                              *
## 2       itsex   Sex of Students 1. FEMALE; 2. MALE; 9. OMITTED OR INVALID         *         *
```

- Recode or rename if inconsistencies identified.

```
TIMSS11_recode <- recode.sdf(TIMSS11,
                      recode = list(itsex = list(from = "GIRL",
                                                 to = "FEMALE"),
                                    itsex = list(from = "BOY",
                                                 to = "MALE")))
```

- Update the trend datasets

```
trend2 <- edsurvey.data.frame.list(list(TIMSS19, TIMSS15, TIMSS11_recode),
                      labels=c("TIMSS 2019", "TIMSS 2015", "TIMSS 2011"))
```

Between Year Comparisons (cont.)

- Run gap analysis between years

```
mathGap4 <- gap(variable = 'mmat', data = trend2)
mathGap4$results
```

```
##       labels estimateA estimateAse    diffAA covAA diffAAse diffAApValue    dofAA sameSurvey
## 1 TIMSS 2019  534.7324    2.550249        NA    NA       NA           NA       NA         NA
## 2 TIMSS 2015  539.1556    2.231920 -4.423178     0 3.388988   0.19303604 250.3516      FALSE
## 3 TIMSS 2011  540.6493    1.816651 -5.916884     0 3.131133   0.05996098 248.8887      FALSE
```

- `estimateA` - Value of estimate for each year
- `estimateAse` - Standard error of estimates for each year
- `diffAA` and `diffAAse` - Difference between two years and standard error of the difference.
- `covAA` - The covariance used in calculating diffAAse
- `diffAApValue` - The p-value associated with the t-test used for the hypothesis test that diffAA is zero
- `dofAA` - The degrees of freedom used in calculating diffABpValue

Between Year Comparisons (cont.)

- Change the reference group
  - By default, the `gap` function treats the first data in an `edsurvey.data.frame.list` as the reference data.
  - We can use the `referenceDataIndex` argument to change the reference to another data. For example, set `referenceDataIndex` argument = 2 to make the second data as the reference data.

```
mathGap5 <- gap(variable = 'mmat', data = trend,
                referenceDataIndex = 2)
mathGap5$results
```

```
##       labels estimateA estimateAse    diffAA covAA diffAAse diffAApValue    dofAA sameSurvey
## 1 TIMSS 2019  534.7324    2.550249  4.423178     0 3.388988    0.1930360 250.3516      FALSE
## 2 TIMSS 2015  539.1556    2.231920        NA    NA       NA           NA       NA         NA
## 3 TIMSS 2011  540.6493    1.816651 -1.493707     0 2.877792    0.6042789 208.5829      FALSE
```

Comparisons of the Gap Between Years

- Gap results

```
trendGap <- gap(variable = "mmat",
                data = trend2,
                groupA = itsex %in% "MALE",
                groupB = itsex %in% "FEMALE")
trendGap$results
```

```
##       labels estimateA estimateAse estimateB estimateBse    diffAB    covAB diffABse diffABpValue     dofAB    diffAA covAA
## 1 TIMSS 2019  540.1785    2.902865  529.0473    2.969877 11.131165 4.638792 2.822980 1.763565e-04  76.77150        NA    NA
## 2 TIMSS 2015  542.6617    2.501978  535.7482    2.312608  6.913450 4.068738 1.862948 3.199990e-04 114.68273 -2.483171     0
## 3 TIMSS 2011  545.0321    1.888004  536.3852    2.091729  8.646838 2.586252 1.663546 1.459974e-06  81.63345 -4.853573     0
##   diffAAse diffAApValue    dofAA    diffBB covBB diffBBse diffBBpValue    dofBB diffABAB covABAB diffABABse diffABABpValue
## 1       NA           NA       NA        NA    NA       NA           NA       NA       NA      NA         NA             NA
## 2 3.832300    0.5181810 127.0670 -6.700886     0 3.764084   0.07620343 261.0728 4.217715       0   3.382276      0.2144723
## 3 3.462829    0.1639786 104.9637 -7.337900     0 3.632561   0.04456383 225.2671 2.484327       0   3.276675      0.4497654
##    dofABAB sameSurvey
## 1       NA         NA
## 2 140.3769      FALSE
## 3 125.1551      FALSE
```

Gap Analysis - Summary

Analyses:

- mean scores
- student group percentages
- achievement level percentages
- percentiles

Comparison Types:

- within year
  - between variable levels (uses `edsurvey.data.frame`)
  - between education systems (uses `edsurvey.data.frame.list`)
- between years (uses `edsurvey.data.frame.list`)

Think Pair Share - Gap Analysis

A. Select Your Own Variables: Use EdSurvey functions to select variables of interest to calculate between education systems results using `gap`

```
# for reference
help(package = "EdSurvey")
searchSDF("text", sdf)
levelsSDF("myvar", sdf)
```

B. Replicate This Scenario: Use EdSurvey functions to calculate between year results using `gap` with these parameters:

- overall TIMSS 2019 4th grade math performance
- across three education systems (e.g., usa, fin, hkg)
- between-country comparison by gender (itsex %in% "MALE and itsex %in% "FEMALE")

Think Pair Share - Gap Analysis

Scenario Result:

- Reading multiple datasets and combine them into an `edsurvey.data.frame.list`.

```
TIMSS19USA<- readTIMSS(path = "C:/TIMSS/2019/", countries = c("usa"), grade
TIMSS19FIN<- readTIMSS(path = "C:/TIMSS/2019/", countries = c("fin"), grade
TIMSS19HKG<- readTIMSS(path = "C:/TIMSS/2019/", countries = c("hkg"), grade
```

- Form an "edsurvey.data.frame.list"

```
trend3 <- edsurvey.data.frame.list(list(TIMSS19USA, TIMSS19FIN, TIMSS19HKG)
```

Think Pair Share - Gap Analysis (cont.)

- Gap results

```
trendGap <- gap(variable = "mmat",
                data = trend3,
                groupA = itsex %in% "MALE",
                groupB = itsex %in% "FEMALE")
trendGap$results
```

Wrap Up

NAEP
NATIONAL ASSESSMENT OF EDUCATIONAL PROGRESS

Learning EdSurvey

- Reading vignettes provided in training materials

```
vignette("introduction", package="EdSurvey")

# There are additional functions that we couldn't cover!
gap() #gap analysis
cor.sdf() # Bivariate correlations using "Pearson", "Spearman", "polychoric
edsurveyTable2pdf() # creating production ready summary tables
cbind(), rbind(), append(), merge() # useful functions in processing data
```

- R help

```
help(package = "EdSurvey")
```

- EdSurvey Website
- EdSurvey Github
- NAEP Data Training workshop

Under development

- Package is still under development
  - Subsequent releases of the EdSurvey package will provide additional functionality
- Your feedback is important to us!

Contact Information

EdSurvey Package Help

- EdSurvey.help@air.org

EdSurvey Package Help on NCES.ed.gov

- http://nces.ed.gov/nationsreportcard/contactus.aspx

Ting Zhang

- tzhang@air.org

Paul Bailey

- pbailey@air.org

Emmanuel Sikali

- Emmanuel.Sikali@ed.gov