The R EdSurvey Package

Analyzing NAEP and TIMSS Data Using R

Presenters: Sinan Yavuz & Paul Bailey

June 2022

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_part2_Script.R

Outline of EdSurvey Workshop - Part 2

1. Summary Statistics
2. Cross Tabulation
3. Achievement Level Analysis
4. Percentile Analysis
5. Linear Regression
6. Gap Analysis

Data Processing

- First, load the `EdSurvey` package and read in the data

```r
# to load the package
library(EdSurvey)
```

NAEP Primer:

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

Summary Statistics

NAEP

NATIONAL ASSESSMENT
OF EDUCATIONAL
PROGRESS

Summary Statistics

`summary2()` produces both weighted and unweighted descriptive statistics for a variable. `summary2()` takes four following arguments in order:

- `data` : an `EdSurvey` object.
- `variable` : name of the variable you want to produce statistics on.
- `weightVar` : name of the weight variable; or `NULL` if users want to produce unweighted statistics.
- `omittedLevels` : if `TRUE`, the function will remove omitted levels for the specified variable before producing descriptive statistics. If `FALSE`, the function will include omitted levels in the output statistics.

Summary Statistics

For a continuous variable (i.e., composite Math score):

- For NAEP data and other datasets with a default weight variable, `summary2` produces weighted statistics by default. If `variable` is a scale or subscale such as `num_oper`, `measurement`, `geometry`, `data_anal_prob`, `algebra`, and `composite` for NAEP Math assessment, the function will produce pooled and weighted summary table.

```r
summary2(sdf, "composite")
```

```
## Estimates are weighted using the weight variable 'origwt'
##    Variable     N Weighted N   Min.  1st Qu.   Median     Mean  3rd Qu.    Max.      SD NA's Zero weights
## 1 composite 16915   16932.46 126.11 251.9626 277.4784 275.8892 301.1827 404.184 36.5713    0            0
```

Summary Statistics

For a continuous variable (i.e., composite Math score):

- By specifying `weightVar = NULL`, the function prints out unweighted descriptive statistics for `variable`, or each plausible value, if `variable` is a scale or subscale.

```
summary2(sdf, "composite", weightVar = NULL)
```

```
## Estimates are not weighted.
##   Variable     N   Min.  1st Qu. Median     Mean  3rd Qu.   Max.       SD NA's
## 1   mrpcm1 16915 130.53 252.0600 277.33 275.8606 300.7200 410.80 35.89864    0
## 2   mrpcm2 16915 124.16 252.2100 277.33 275.6399 300.6900 408.58 36.08483    0
## 3   mrpcm3 16915 115.09 252.0017 277.19 275.6570 300.5600 398.17 36.09278    0
## 4   mrpcm4 16915 137.19 252.4717 277.44 275.7451 300.5767 407.41 35.91078    0
## 5   mrpcm5 16915 123.58 252.4900 277.16 275.6965 300.5000 395.96 36.10905    0
```

Summary Statistics

For a categorical variable (i.e., frequency of students talking about studies at home):

- By default, `omittedLevels` is set to `FALSE`. That is, the function includes omitted levels of the variable `b017451` in the output statistics.

```
summary2(sdf, "b017451")
```

```
## Estimates are weighted using the weight variable 'origwt'
##                b017451    N Weighted N Weighted Percent Weighted Percent SE
## 1 Never or hardly ever 3837  3952.4529      23.34245648           0.4318975
## 2 Once every few weeks 3147  3190.8945      18.84483329           0.3740648
## 3     About once a week 2853  2937.7148      17.34960077           0.3414566
## 4   2 or 3 times a week 3362  3425.8950      20.23270282           0.3156289
## 5            Every day 3132  3223.8074      19.03921080           0.4442216
## 6              Omitted  575   194.3312       1.14768416           0.1272462
## 7             Multiple    9     7.3676       0.04351168           0.0191187
```

Summary Statistics

For a categorical variable (i.e., frequency of students talking about studies at home):

- By specifying `omittedLevels = TRUE`, the function removes omitted levels out of the output statistics.

```
summary2(sdf, "b017451", omittedLevels = TRUE)
```

```
## Estimates are weighted using the weight variable 'origwt'
##                 b017451    N Weighted N Weighted Percent Weighted Percent SE
## 1 Never or hardly ever 3837   3952.453         23.62386           0.4367548
## 2 Once every few weeks 3147   3190.894         19.07202           0.3749868
## 3    About once a week 2853   2937.715         17.55876           0.3486008
## 4  2 or 3 times a week 3362   3425.895         20.47662           0.3196719
## 5            Every day 3132   3223.807         19.26874           0.4467063
```

- Related Documentation - EdSurvey Book

Cross Tabulation

`edsurveyTable()`: creates a summary table of outcome and categorical variables. There are 3 important arguments as followed:

- `formula`: typically written as `a ~ b + c`, in which:
  - `a`: a continuous variable (optional) that the function will return weighted mean on.
  - `b` and `c`: categorical variable(s) that the function will run cross-tabulation on; multiple crosstab categorical variables can be separated using `+` symbol.
- `data`: an `EdSurvey` object
- `pctAggregationLevel`: a numeric value (i.e., 0, 1, 2) that indicates the level of aggregation in the cross-tabulation result's percentage column.

Cross Tabulation

- Summary table of NAEP composite mathematics performance scale scores (`composite`) of 8th grade students by two student factors:
    - `dsex`: gender
    - `b017451`: frequency of talk about studies at home

```
es1 <- edsurveyTable(composite ~ dsex + b017451, data = sdf)
```

- `pctAggregationLevel` is by default set to `NULL` (or `1`). That is, the `PCT` column adds up to 100 within each level of the first categorical variable `dsex`.

dsex	b017451	N	WTD_N	PCT	SE(PCT)	MEAN	SE(MEAN)
Male	Never or hardly ever	2350	2434.844	29.00978	0.6959418	270.8243	1.057078
Male	Once every few weeks	1603	1638.745	19.52472	0.5020657	275.0807	1.305922
Male	About once a week	1384	1423.312	16.95795	0.5057265	281.5612	1.409587

Cross Tabulation

- By specifying `pctAggregationLevel = 0`, the `PCT` column adds up to 100 across the entire sample.

```
es2 <- edsurveyTable(composite ~ dsex + b017451, data = sdf, pctAggr
```

dsex	b017451	N	WTD_N	PCT	SE(PCT)	MEAN	SE(MEAN)
Male	Never or hardly ever	2350	2434.844	14.553095	0.3738531	270.8243	1.057078
Male	Once every few weeks	1603	1638.745	9.794803	0.2651368	275.0807	1.305922
Male	About once a week	1384	1423.312	8.507154	0.2770233	281.5612	1.409587
Male	2 or 3 times a week	1535	1563.393	9.344421	0.2670298	284.9066	1.546072
Male	Every day	1291	1332.890	7.966700	0.3000579	277.2597	1.795784
Female	Never or hardly ever	1487	1517.609	9.070768	0.2984443	266.7897	1.519020

- Related Documentation - EdSurvey-LaTeXtables.pdf
- Related Documentation - EdSurvey Book

Cross Tabulation - Question

- What percentage of male students talk about studies at home about once a week?

dsex	b017451	N	WTD_N	PCT	SE(PCT)	MEAN	SE(MEAN)
Male	Never or hardly ever	2350	2434.844	14.553095	0.3738531	270.8243	1.057078
Male	Once every few weeks	1603	1638.745	9.794803	0.2651368	275.0807	1.305922
Male	About once a week	1384	1423.312	8.507154	0.2770233	281.5612	1.409587
Male	2 or 3 times a week	1535	1563.393	9.344421	0.2670298	284.9066	1.546072
Male	Every day	1291	1332.890	7.966700	0.3000579	277.2597	1.795784
Female	Never or hardly ever	1487	1517.609	9.070768	0.2984443	266.7897	1.519020
Female	Once every few weeks	1544	1552.149	9.277216	0.2498498	271.2255	1.205528
Female	About once a week	1469	1514.403	9.051606	0.2899668	278.7502	1.719778
Female	2 or 3 times a week	1827	1862.502	11.132198	0.2552321	282.7765	1.404107
Female	Every day	1841	1890.918	11.302039	0.3497982	275.4628	1.219439

Cross Tabulation - Question

- What is the average composite math score of female students who talk about studies at home 2 or 3 times a week?

dsex	b017451	N	WTD_N	PCT	SE(PCT)	MEAN	SE(MEAN)
Male	Never or hardly ever	2350	2434.844	14.553095	0.3738531	270.8243	1.057078
Male	Once every few weeks	1603	1638.745	9.794803	0.2651368	275.0807	1.305922
Male	About once a week	1384	1423.312	8.507154	0.2770233	281.5612	1.409587
Male	2 or 3 times a week	1535	1563.393	9.344421	0.2670298	284.9066	1.546072
Male	Every day	1291	1332.890	7.966700	0.3000579	277.2597	1.795784
Female	Never or hardly ever	1487	1517.609	9.070768	0.2984443	266.7897	1.519020
Female	Once every few weeks	1544	1552.149	9.277216	0.2498498	271.2255	1.205528
Female	About once a week	1469	1514.403	9.051606	0.2899668	278.7502	1.719778
Female	2 or 3 times a week	1827	1862.502	11.132198	0.2552321	282.7765	1.404107
Female	Every day	1841	1890.918	11.302039	0.3497982	275.4628	1.219439

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
- Related Documentation - EdSurvey Book

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

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

# store the data in this folder, you may want to update this
downloadTIMSS(years = c(2019), root = "~/")

TIMSS19 <- readTIMSS("~/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
```

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
```

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
```

Question

- What percentage of Female students are at or above proficient level on composite math score?

```
Q1 <- achievementLevels(c("composite", "dsex"), aggregateBy = "dsex"
                        data = sdf, returnCumulative = TRUE)
Q1$cumulative
```

```
##          composite_Level   dsex      N      wtdN   Percent StandardError
## 1            Below Basic   Male 2880.8 2865.6455 33.666050     1.0951825
## 2      At or Above Basic   Male 5605.2 5646.3287 66.333950     1.0951825
## 3 At or Above Proficient   Male 2339.0 2409.9253 28.312178     0.8635866
## 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 or Above Basic Female 5578.6 5506.6295 65.395601     1.1154848
## 7 At or Above Proficient Female 2149.2 2162.8149 25.685145     1.0073379
## 8            At Advanced Female  360.4  378.8444  4.499079     0.3888590
```

Question 2

- What percentage of students at proficient level on composite math score are Male?

```
Q2 <- achievementLevels(c("composite", "dsex"),
                        aggregateBy = "composite", data = sdf)
Q2$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
```

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()`

Percentile Analysis

NATIONAL ASSESSMENT OF EDUCATIONAL PROGRESS

Percentile Analysis

`percentile()` - calculates the percentiles of a numeric variable

- typically a subject scale or subscale (`"composite"`)
- numeric vector of percentiles in the range 0 to 100 (`c(25,50,75)`)

```
# 25th, 50th and 75th percentiles
per <- percentile("composite", percentiles = c(25,50,75), data = sdf
per
```

```
## Percentile
## Call: percentile(variable = "composite", percentiles = c(25, 50, 75),
##     data = sdf)
## full data n: 17606
## n used: 16915
##
##  percentile estimate        se       df confInt.ci_lower confInt.ci_upper
##          25 251.9626 1.0179363 42.53475         249.7120         254.0142
##          50 277.4784 1.1375443 51.15378         275.7035         279.1926
##          75 301.1827 0.9141083 70.56403         299.4265         302.8973
```

- Related Documentation - EdSurvey-Percentiles.pdf
- Related Documentation - EdSurvey Book

Percentile Analysis

`percentile()` - the full range of quantiles

```r
# note df/se at 0 and 100. We would not report these.
per <- percentile("composite", percentiles = c(0:100), data = sdf)
```

	percentile	estimate	se	df	confInt.ci_lower	confInt.ci_upper
P0	0	126.1100	13.7363161	2.80821	126.1100	143.0444
P1	1	185.9546	3.3733809	44.17526	179.4243	190.7146
P2	2	196.7552	1.6605381	39.21315	192.5669	200.4021
P3	3	203.8506	1.6674788	58.18808	200.2818	206.6049
P4	4	208.7937	1.3931131	33.14432	205.8532	211.2580
P5	5	212.9238	1.0713696	23.75427	210.0002	215.7760
P6	6	216.6267	1.4713185	44.29842	213.8604	219.3234
P7	7	219.9586	1.1825717	47.74121	217.1187	222.3583

Linear Regression

Linear Regression - lm.sdf()

`lm.sdf()`: fits a linear model formula using sampling weights and a variance estimation method. The format is:

```
myfit <- lm.sdf(formula, data, weightVar, varMethod,
relevels)
```

- `formula`: model to be fit.
- `data`: data frame containing the data to be used in fitting the model.
- `weightVar`: indicates the weight variable to use.
- `varMethod`: the variance estimation method (Jackknife or Taylor series) with the Jackknife as the default.
- `relevels`: is used when the user wants to change the reference level of a categorical variable.

Linear Regression - lm.sdf()

The resulting object (`myfit` in this case) is a list containing extensive information about the fitted model.

Formula notation is typically written as:

```
Y ~ X1 + X2 + ... + Xk
```

- The `~` separates the response variable on the left from the predictor variables on the right.
- The `+` sign separates the predictor variables.

Regressions - lm.sdf()

Example of bivariate regression:

$$\texttt{Composite} = \beta_0 +$$

$$\beta_1 \texttt{Freq. of talk about studies at home} + \epsilon$$

```
lm1 <- lm.sdf(composite ~ b017451,
              weightVar = 'origwt', data = sdf)
summary(lm1)
```

```
##
## Formula: composite ~ b017451
##
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## Plausible values: 5
## jrrIMax: 1
## full data n: 17606
## n used: 16331
##
## Coefficients:
##                                  coef       se         t    dof  Pr(>|t|)
```

Regressions - lm.sdf()

Example of multiple regression:

$$\texttt{Composite} = \beta_0 + \beta_1\texttt{Gender}+$$

$$\beta_2\texttt{Freq. of talk about studies at home} + \epsilon$$

```
lm2 <- lm.sdf(composite ~ dsex + b017451,
              weightVar = 'origwt', data = sdf)
```

- the sampling weight for this regression: `origwt`

Regressions - lm.sdf()

```
summary(lm2)
```

```
##
## Formula: composite ~ dsex + b017451
##
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## Plausible values: 5
## jrrIMax: 1
## full data n: 17606
## n used: 16331
##
## Coefficients:
##                                  coef        se        t    dof  Pr(>|t|)
## (Intercept)                 270.41112   1.02443 263.9615 54.670 < 2.2e-16 ***
## dsexFemale                   -2.95858   0.60423  -4.8965 54.991 8.947e-06 ***
## b017451Once every few weeks   4.23341   1.18327   3.5777 57.316 0.0007131 ***
## b017451About once a week     11.22612   1.25854   8.9200 54.683 2.983e-12 ***
## b0174512 or 3 times a week   14.94591   1.18665  12.5951 72.582 < 2.2e-16 ***
## b017451Every day              7.52998   1.30846   5.7549 48.470 5.755e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared: 0.0224
```

Regressions - lm.sdf()

Adding `src = TRUE` displays standardized regression coefficients

```r
summary(lm2, src = TRUE)
```

```
##
## Formula: composite ~ dsex + b017451
##
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## Plausible values: 5
## jrrIMax: 1
## full data n: 17606
## n used: 16331
##
## Coefficients:
##                                    coef         se        t    dof   Pr(>|t|) stdCoef    stdSE
## (Intercept)                 270.4111210  1.0244340 263.9615 54.670 0.0000e+00      NA       NA
## dsexFemale                   -2.9585783  0.6042285  -4.8965 54.991 8.9474e-06 -0.0407 0.008313 **
## b017451Once every few weeks   4.2334144  1.1832671   3.5777 57.316 7.1311e-04  0.0458 0.012791 *
## b017451About once a week     11.2261232  1.2585369   8.9200 54.683 2.9834e-12  0.1175 0.013175 *
## b0174512 or 3 times a week   14.9459085  1.1866461  12.5951 72.582 0.0000e+00  0.1659 0.013175 *
## b017451Every day              7.5299837  1.3084558   5.7549 48.470 5.7550e-07  0.0817 0.014200 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared: 0.0224
```

Regressions - lm.sdf()

Use `relevels` to set omitted / reference level of `dsex` to "Female":

```
lm3 <- lm.sdf(composite ~ dsex + b017451,
              weightVar = 'origwt',
              relevels = list(dsex = "Female"), data = sdf)
```

Regressions - lm.sdf()

```
summary(lm3)
```

```
##
## Formula: composite ~ dsex + b017451
##
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## Plausible values: 5
## jrrIMax: 1
## full data n: 17606
## n used: 16331
##
## Coefficients:
##                                coef       se        t    dof  Pr(>|t|)
## (Intercept)               267.45254  1.13187 236.2919 76.454 < 2.2e-16 ***
## dsexMale                    2.95858  0.60423   4.8965 54.991 8.947e-06 ***
## b017451Once every few weeks   4.23341  1.18327   3.5777 57.316 0.0007131 ***
## b017451About once a week     11.22612  1.25854   8.9200 54.683 2.983e-12 ***
## b0174512 or 3 times a week   14.94591  1.18665  12.5951 72.582 < 2.2e-16 ***
## b017451Every day              7.52998  1.30846   5.7549 48.470 5.755e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared: 0.0224
```

Poll - lm.sdf

Poll: `lm.sdf` shows standard errors, t-statistics, and *p*-values

- These statistics account for the sampling variance only
- These statistics account for the imputation variance (uncertainty associated with the student-level imprecision of the test)
- These statistics account for both sampling and imputation variance

Gap Analysis: Estimating the Difference in Two Statistics

NAEP

NATIONAL ASSESSMENT OF EDUCATIONAL PROGRESS

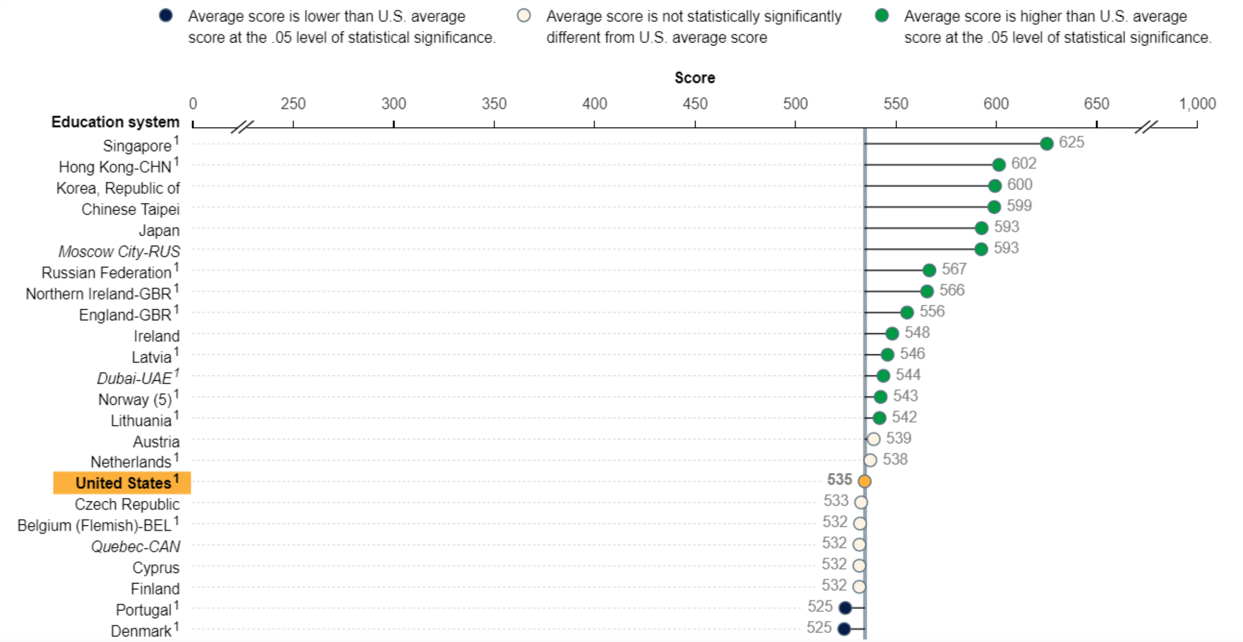
Gap Analysis

`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 $$ {\rm Var}(\theta_A-\theta_B)= {\rm Var}(\theta_A) + {\rm Var}(\theta_B) - 2 {\rm Cov}(\theta_A,\theta_B)$$
- Related Documentation - e-book section on gap analysis
- Related Documentation - EdSurvey-TIMSS.pdf

Typical Gap Comparisons

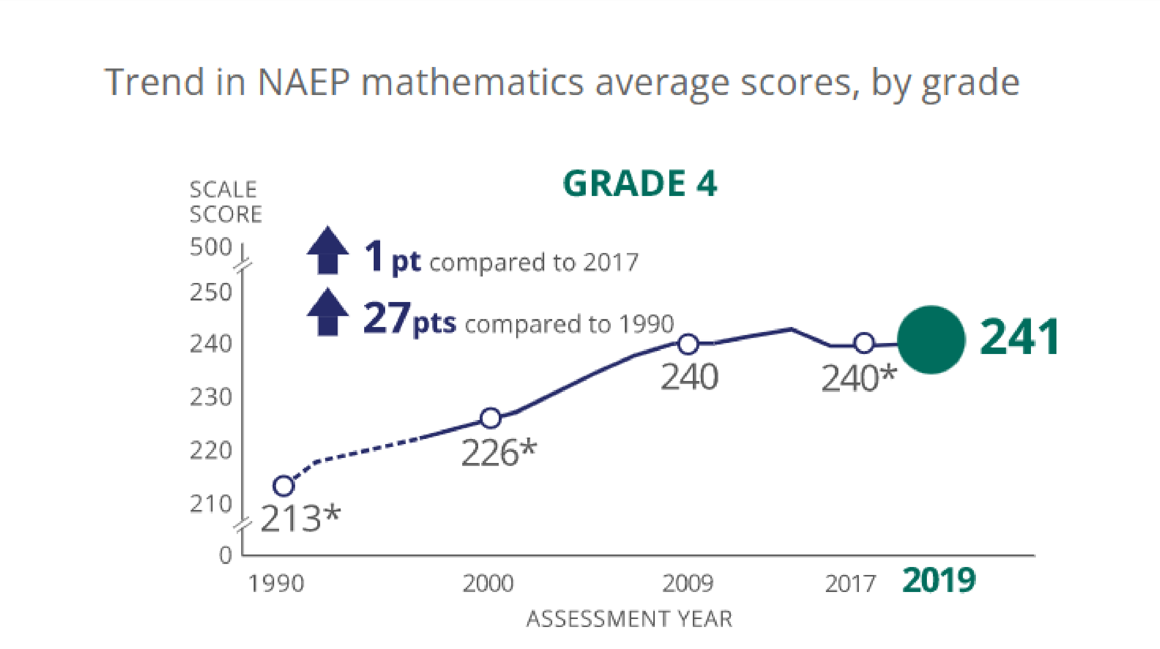
- Comparisons of different groups/jurisdictions within years
  - Female in 2019 to Male in 2019
  - USA in 2019 to Singapore in 2019
  - USA in 2019 to the international average in 2019 (part/whole)



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

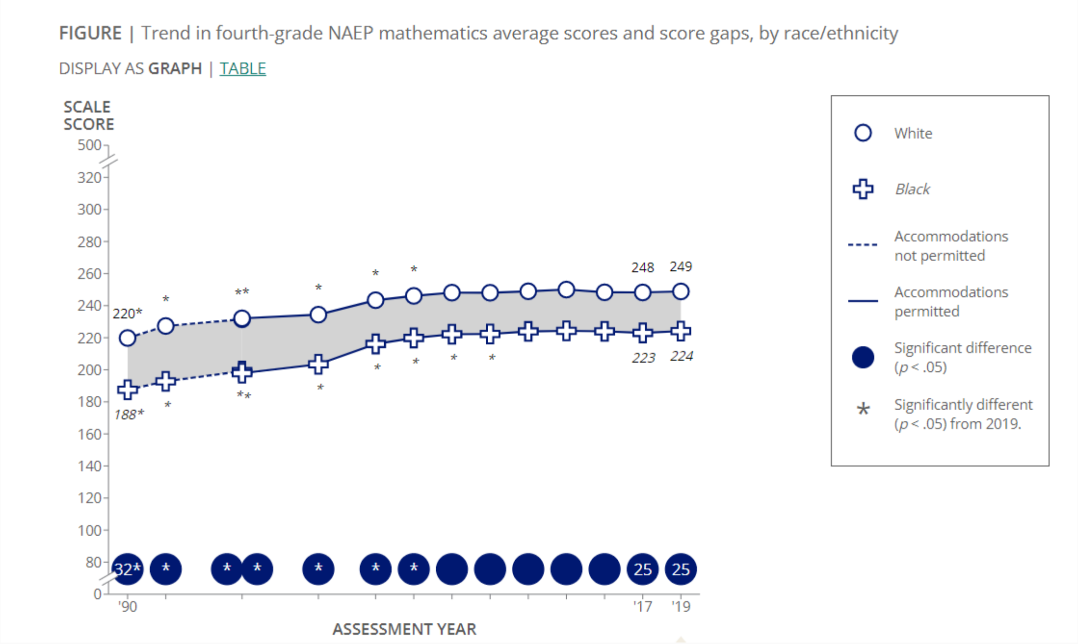


FIGURE | Trend in fourth-grade NAEP mathematics average scores and score gaps, by race/ethnicity

DISPLAY AS **GRAPH** | TABLE

Source: NAEP Math Report 2019

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"`

```r
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

```
Gap2 <- gap(variable = "composite", data = sdf,
            groupA = dsex %in% "Male", groupB = dsex %in% "Female",
            achievementLevel = c("Basic", "Proficient", "Advanced"))

Gap2$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

```
Gap3 <- gap(variable = "composite", data = sdf,
            groupA = dsex %in% "Male", groupB = dsex %in% "Female",
            percentiles = c(25, 50, 75))

Gap3$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(2015, 2011), root = "~/")
```

- Read in datasets from multiple years

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

```
trend <- edsurvey.data.frame.list(list(TIMSS19, TIMSS15, TIMSS11))
```

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 2019 2015 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$itsex <- ifelse(TIMSS11$itsex == "GIRL", "FEMALE", "MALE")
```

- Update the trend datasets

```
trend2 <- edsurvey.data.frame.list(list(TIMSS19, TIMSS15, TIMSS11))
```

Between Year Comparisons (cont.)

- Run gap analysis between years

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

```
##   year estimateA estimateAse    diffAA covAA diffAAse diffAApValue    dofAA sameSurvey
## 1 2019  534.7324    2.550249        NA    NA       NA           NA       NA         NA
## 2 2015  539.1556    2.231920 -4.423178     0 3.388988   0.19303604 250.3516      FALSE
## 3 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 year. For example, set `referenceDataIndex` argument = 2 to make the second row the reference.

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

```
##   year estimateA estimateAse    diffAA covAA diffAAse diffAApValue    dofAA sameSurvey
## 1 2019  534.7324    2.550249  4.423178     0 3.388988    0.1930360 250.3516      FALSE
## 2 2015  539.1556    2.231920        NA    NA       NA           NA       NA         NA
## 3 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
```

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

Poll - Gap Analysis

the gap function shows standard errors and *p*-values

- These statistics account for the sampling variance only
- These statistics account for the imputation variance (uncertainty associated with the student-level imprecision of the test)
- These statistics account for both sampling and imputation variance
- These statistics treat the data as independent and as measured without variance

Poll - Gap Analysis 2

the `gap` function shows a lot of data, what is your favorite place to remind yourself of the meaning of a statistic

- The slides from this presentation
- The AIR EdSurvey home page
- The help for gap
- Other [comment in the chat]

Poll - Gap Analysis 3

the gap function shows covariances for contrasts between some statistics

- Those need to account for covariance within years, such as Male vs Female; students in the same school are more similar than randomly drawn students
- Those need to account for covariance across years, such as 2017 v 2019; the same school may have been sampled in both years
- Those need to account for covariance both within year and across years; when the students are different, they do not covary
- Those statistics do not need to account for covariance

Wrap Up

Learning EdSurvey

- Reading vignettes provided in training materials

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

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

- R help

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

- EdSurvey Website
- EdSurvey e-book
- EdSurvey Github
- NAEP Data Training workshop

Under development

- Package is still under development
- Your feedback is important to us!

Contact Information

EdSurvey Package Help

- https://github.com/American-Institutes-for-Research/EdSurvey/issues

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