Analyzing NAEP/TIMSS Data with Direct Estimation using the R packages EdSurvey and Dire

Presenters: Paul Bailey, Ting Zhang, Michael Lee, & Sinan Yavuz
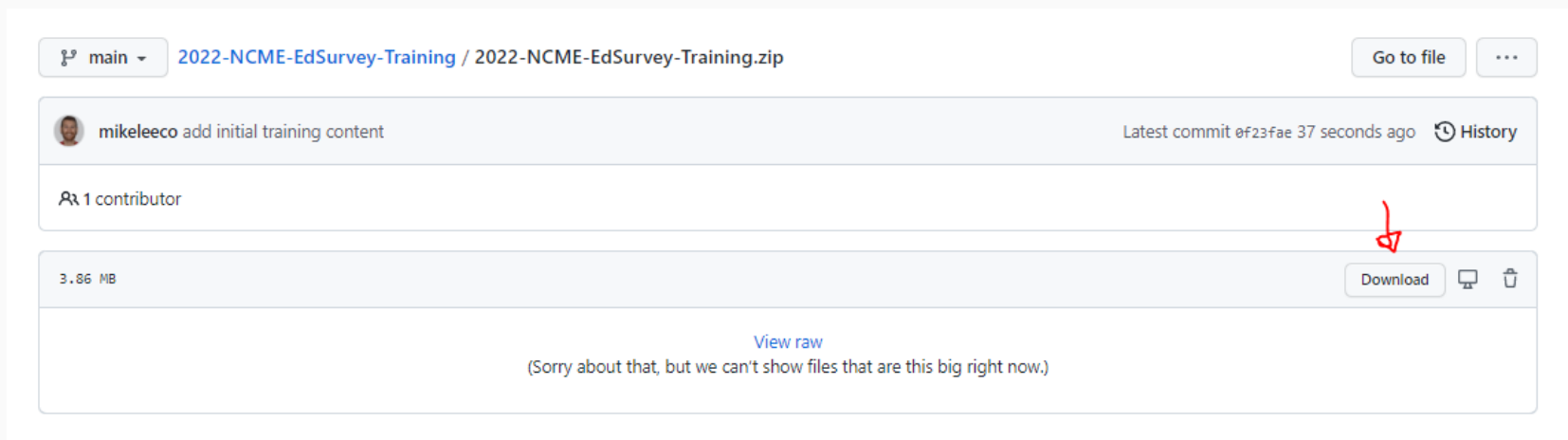
April 2022

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` and `Dire`.

Outline of EdSurvey Workshop

1. Introduction to R, EdSurvey, and Dire

2. Data Processing and Data Manipulation

3. Hands-on practice

4. Descriptive statistics

5. Hands-on practice

6. Direct estimation with EdSurvey and Dire

7. Hands-on practice

Course Materials

Available here: 2022 NCME Training Content

Introduction to R, EdSurvey, and Dire

Why R?

1. **Free:** users can legally use and edit R package code

2. **Extensible:** large variety of contributed packages that expand its functionality

3. **Reproducible:** automated data analysis

4. **Designed by and for researchers:** robust ecosystem to translate data into analyses, visualizations, and summary reports with one software

Why EdSurvey?

1. **One-stop shop** for data downloading, processing, manipulation and analysis of survey data.

2. **Automated**: Weights and complex sampling design calculations are automated following standard OECD methodology.

3. **Simple**: e.g., a regression with 80 replicate weights requires only a few lines of code.

4. **Flexible**: You can use functions that rely on EdSurvey methods or get the data and use traditional R.

5. **Minimizes memory footprint** by only reading in required data.

Why Dire?

1. **Wow**: Assessments with the matrix booklet design require special considerations in data analysis, e.g., IRT and multiple imputation for item responses. `Dire` provides direct estimation functions that handles analyses of these assessment data properly.

2. **Efficient**: Students' latent proficiency distribution, as well as reporting group difference parameters, are estimated on the fly.

3. **Plausible Values Generation**: No need to rely on testing companies. Plausible values can be generated from the user-defined MML model and used for further analysis.

4. **Expanding Research Scope**: Providing the opportunity for researchers to link administrative data, aggregate data about a community from official statistics, or data from multiple surveys to open up new research questions.

Basic R Infrastructure

CRAN stores packages

Basic R Infrastructure

CRAN stores packages

Accessed via

install.packages("ggplot2")

and loaded into R
on your machine

library("ggplot2")

Basic R Infrastructure

CRAN stores packages

Accessed via

`install.packages("ggplot2")`

and loaded into R
on your machine

`library("ggplot2")`

**These package libraries
consist of functions**

ggplot()
geom_point()
...

Basic R Infrastructure

CRAN stores packages

Accessed via

install.packages("ggplot2")

and loaded into R
on your machine

library("ggplot2")

These package libraries consist of functions

ggplot()
geom_point()
...

That can be used to analyze data

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()
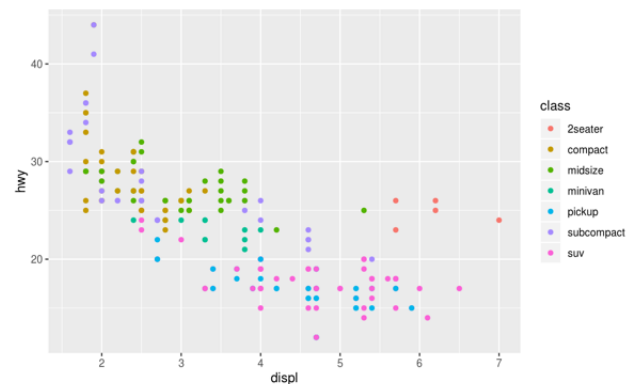
Get to Know the R Environment

Follow Along - R Scripts

Follow along in edsurvey_part1_Script.R

Notes About Using R

- Highlight , `ctrl` + `enter` executes code to console
- Comment character is a hash

```
# this line is not executed
```

- Variables are assigned with an equals or `<-`

```
x <- 12
x
```

```
## [1] 12
```

- In file names on Windows use a forward slash
  - `C:/`
- R is case sensitive!

```
j <- 12
J
```

```
## Error in eval(expr, envir, enclos): object 'J' not found
```

Notes About Using R (cont)

- Any command can be input with a question mark preceding it to open the help guide

```
?mean
```

- Use the up arrow on your keyboard to copy your previous lines of code

- Try tab completion, type, "data" then hit the tab key
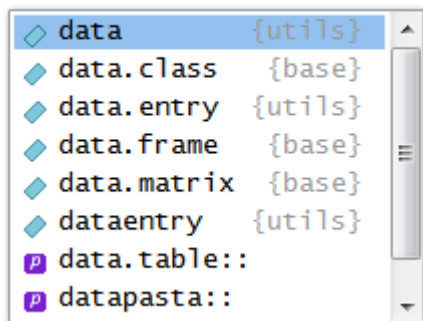
Using R Functions

- `c()` function combines values, separated by a comma, into a vector

```
colors <- c("red", "green", "blue")
colors
```
```
## [1] "red"    "green" "blue"
```
```
numbers <- c(1, 2, 3)
numbers
```
```
## [1] 1 2 3
```

- In the `EdSurvey` package we'll use vectors to combine the names of variables in our analyses

Using R Functions

- Arguments can be explicitly or implicitly named

```
mean(x = numbers)
```
```
## [1] 2
```
```
mean(numbers)
```
```
## [1] 2
```

- Arguments are separated by commas
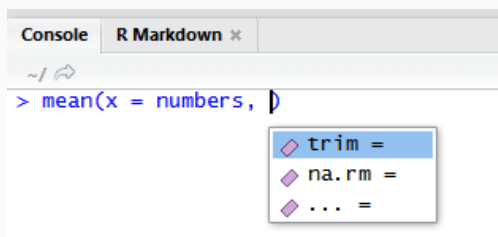
Installing the EdSurvey Package

- After opening up RStudio, run the following scripts in the console to download and initialize the `EdSurvey` package:

```r
# to install the package
install.packages("EdSurvey")

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

Learning EdSurvey

- Reading vignettes provided in training materials

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

- R help

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

- EdSurvey eBook
- EdSurvey Website
- EdSurvey Github
- NAEP Data Training workshop

Self-Reflection - R Functions

Ask yourself: What are the arguments of the function `readNAEP()` ? What are some examples of acceptable values for each argument?

Self-Reflection - R Functions

`readNAEP()` arguments, from R documentation (*type* `?readNAEP` *in the console*)

Usage

```
readNAEP(path, defaultWeight = "origwt", defaultPvs = "composite",
   omittedLevels = c("Multiple", NA, "Omitted"), frPath = NULL)
```

Arguments

`path`	a character value indicating the full filepath location and name of the (.dat) data file
`defaultWeight`	a character value that indicates the default weight specified in the resulting `edsurvey.data.frame`. Default value is `origwt` if not specified.
`defaultPvs`	a character value that indicates the default plausible value specified in the resulting `edsurvey.data.frame`. Default value is `composite` if not specified.
`omittedLevels`	a character vector indicating which factor levels/labels should be excluded. When set to the default value of `c('Multiple',NA,'Omitted')`, adds the vector to the `edsurvey.data.frame`.
`frPath`	a character value indicating the location of the `fr2` parameter layout file included with the data companion to parse the specified `filepath` data file

Data Processing

Data Processing

- First, read in the publicly available NAEP data from `NAEPprimer`

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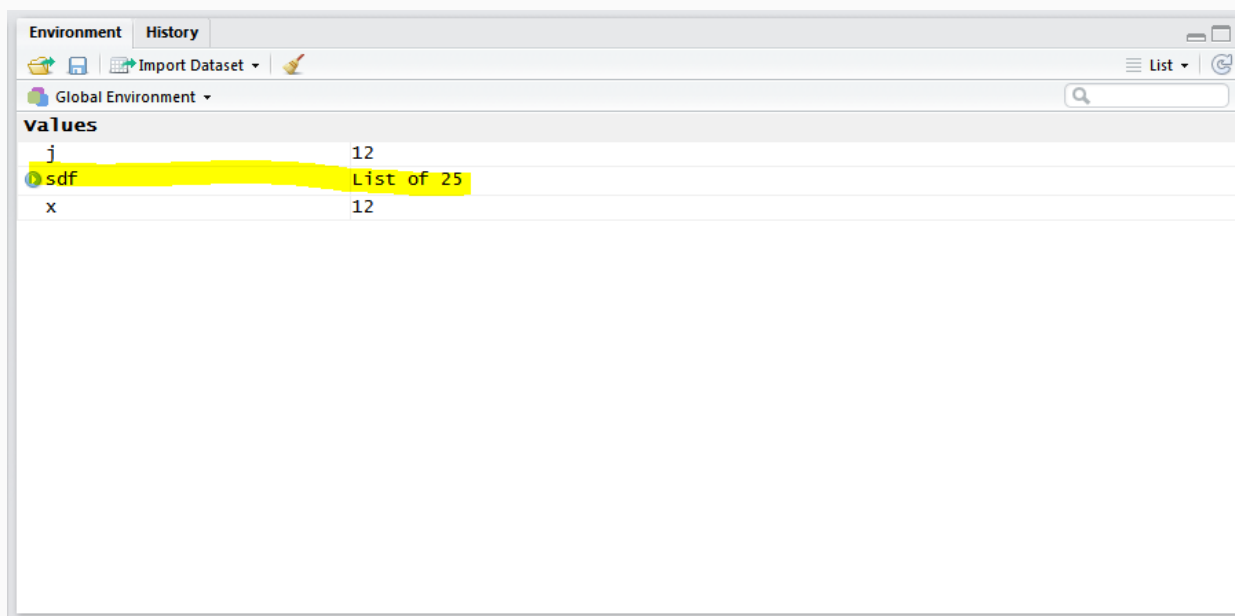

- How long did that take?

Data Processing
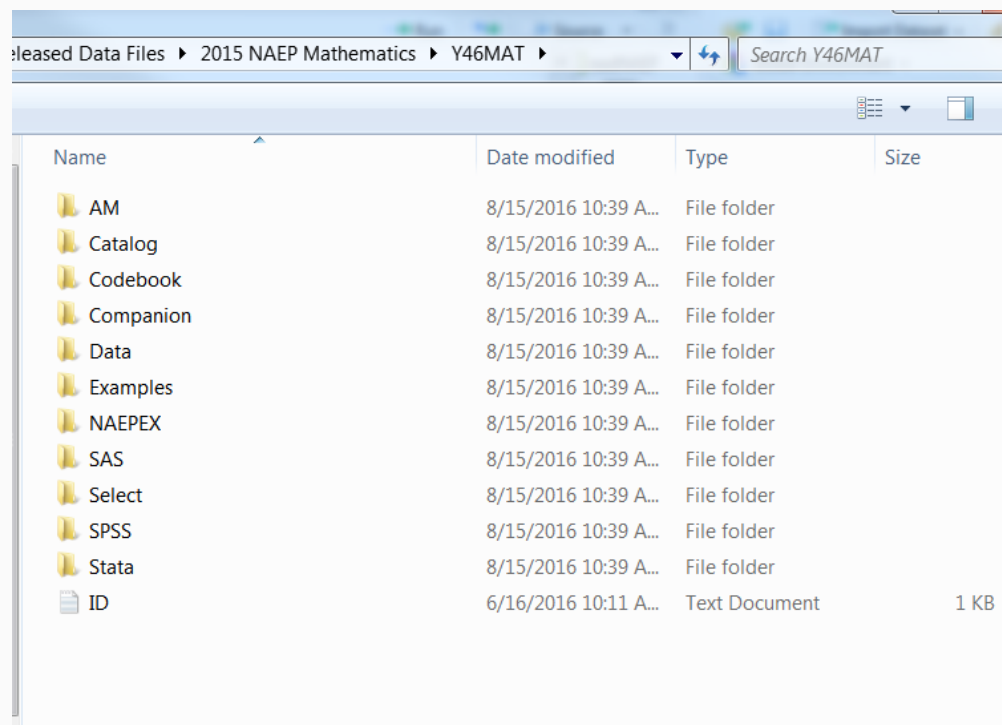
- Can also read in other restricted-use NAEP data by naming the file location

```
math17 <- readNAEP("//path_to_directory/Data/M48NT2AT.dat")
```

- The first character indicates the subject - M (Math)

- The second and third characters indicate the NAEP year - 48 (2017 - 1969 = 48)

- The fourth character indicates the component - N (National)

- The fifth character indicates the type of data - T (Student)

- The sixth character indicates the grade cohort - 2 (8th)

- The seventh and eighth characters indicate the sample - AT (Main NAEP)

Data Processing

NOTE: the dat file requires its intact data folder directory in order to be read in correctly; containing both the student and school level files to merge data

Meet Your Data

Quick Terminology Notes

The `edsurvey.data.frame` class stores information about survey data via a data connection, which allows for:

- Correct calculation of relevant statistics.

- Limited working memory usage.

SDF

The Edsurvey package uses the acronym `SDF` in the names of several functions to signify their relationship to **S**urvey **D**ata **F**rames.

Quick Terminology Notes

The `edsurvey.data.frame.list` class stores a list of `edsurvey.data.frame` objects.

- The list can be passed to the analysis functions, and a result list will be returned.

- A list can store both `cov` (covariants) and `labels` arguments. For example, the 'year' and 'country' might vary across the `edsurvey.data.frame`s in the list.

- `edsurvey.data.frame.lists` can also be constructed manually. See the `?edsurvey.data.frame.list` documentation.

One-stop Shop for NCES Survey Analysis

00 Initialize your machine

01 Understand the data

02 Manipulate the data

in EdSurvey · in R

03 Run analyses
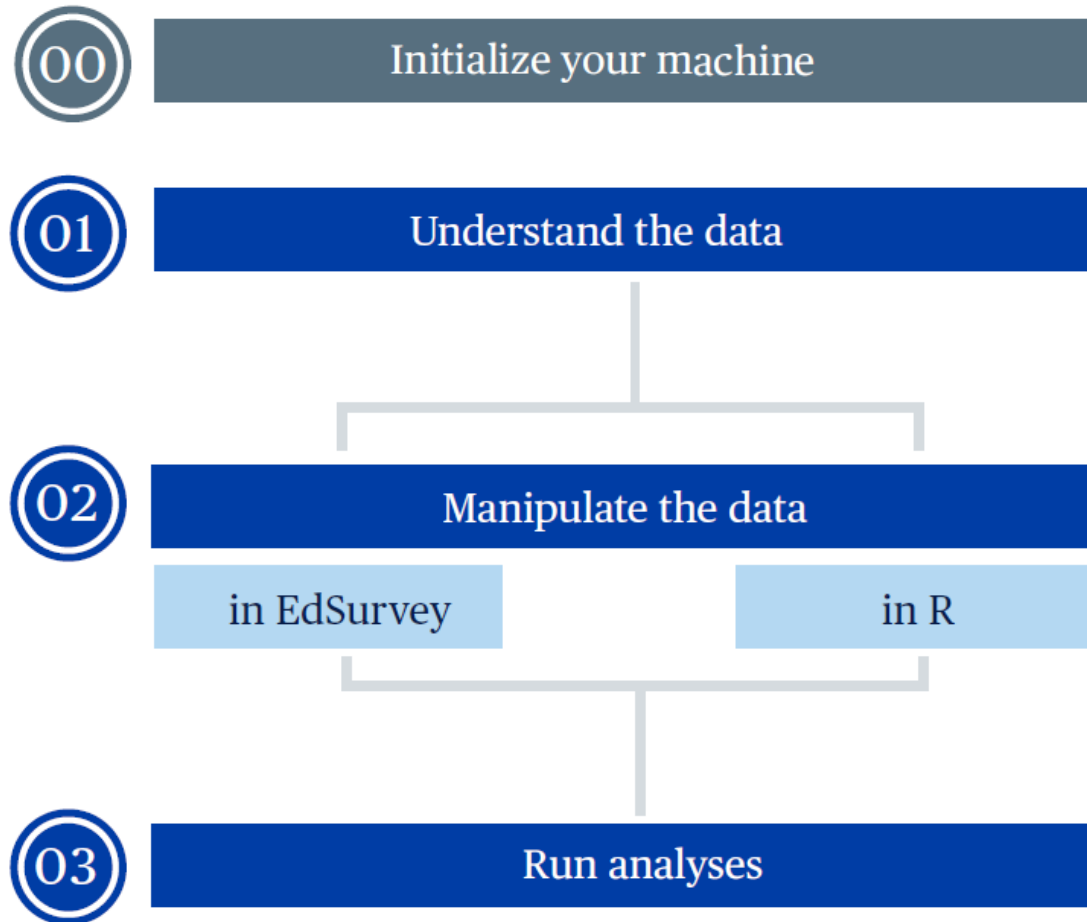
One-stop Shop for NCES Survey Analysis

Initialize your machine — 00

- Install R and EdSurvey
- Download and read-in data

Example functions:

- `readTIMSS`
- `downloadTIMSS`

One-stop Shop for NCES Survey Analysis

01	Understand the data

- *Explore*: explore the codebook, see the variables with plausible values, see weights

- *Search*: search variables

- *Expand*: see variable levels, tabulate response percentages, see assessment scores by response category, summarize continuous variables

Example functions:

- `showCodebook`, `showPlausibleValues`, `showWeights`

- `searchSDF`, `levelsSDF`

- `summary2`, `edsurveyTable`

One-stop Shop for NCES Survey Analysis
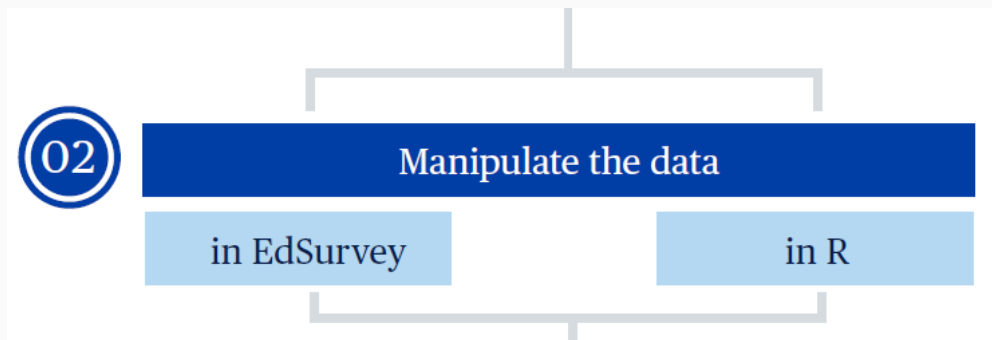
In *EdSurvey*: Clean and manipulate data with built-in subset and recode features.

In *R*: Extract and manipulate data as a data frame (for experienced users)

Example functions:

- `subset`

- `rename.sdf`, `recode.sdf`

- `getData`, `rebindAttributes`

One-stop Shop for NCES Survey Analysis

03   **Run analyses**

- *Run analyses*: such as regression analysis, logit analysis, mixed models, show gaps, calculate achievement levels, correlate variables, calculate percentiles.

Example functions:

- `achievementLevels`, `percentile`
- `cor.sdf`
- `gap`
- `lm.sdf`, `glm.sdf`
- `mvrlm.sdf`, `mixed.sdf`

One-stop Shop for NCES Survey Analysis
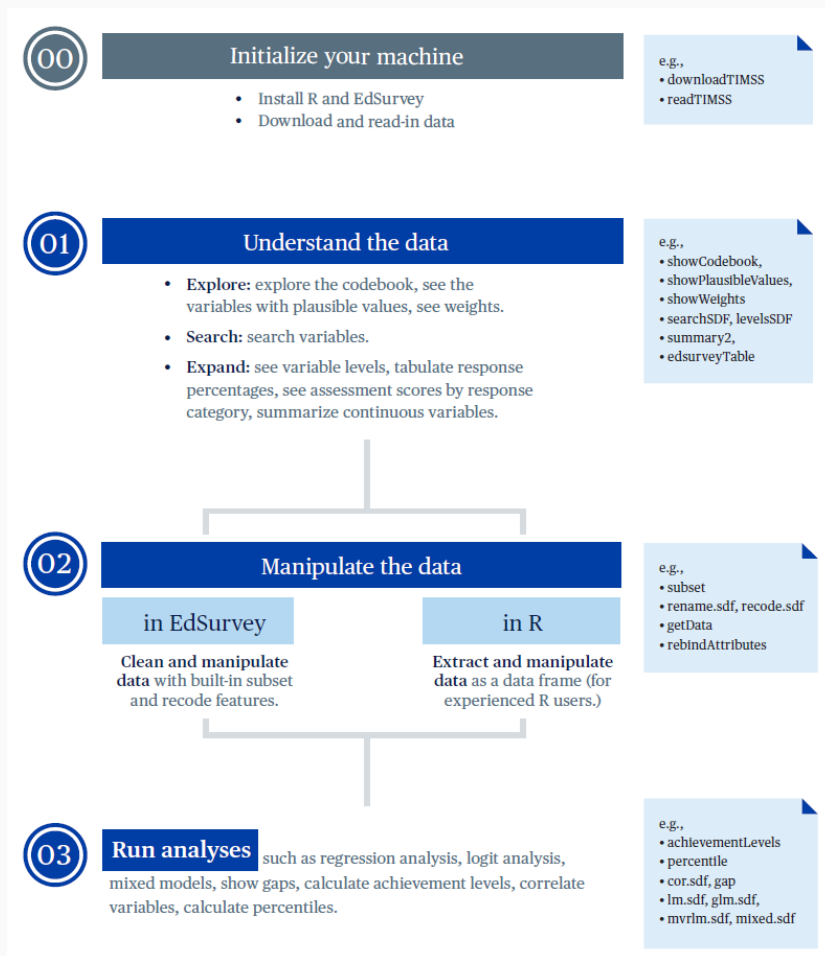
- Related Documentation - EdSurvey.pdf, Chap 3, EdSurvey Book

Meet Your Data - print

`print()`

- Print returns detailed data file information:

```
print(sdf)
```

```
## edsurvey.data.frame for 2005 NAEP (Mathematics) in USA
## Dimensions: 17606 rows and 302 columns.
##
## There is 1 full sample weight in this edsurvey.data.frame:
##   'origwt' with 62 JK replicate weights (the default).
##
##
## There are 6 subject scale(s) or subscale(s) in this edsurvey.data.frame:
## 'num_oper' subject scale or subscale with 5 plausible values.
##
## 'measurement' subject scale or subscale with 5 plausible values.
##
## 'geometry' subject scale or subscale with 5 plausible values.
##
## 'data_anal_prob' subject scale or subscale with 5 plausible values.
##
```

Meet Your Data - dim

`dim()`

- Returns the dimensions of the student data set:

```
dim(sdf)
```
```
## [1] 17606   302
```

Meet Your Data - colnames

colnames()

- Prints the names of all variables in the student and school data sets:

```
colnames(sdf)
```

```
##    [1] "year"     "cohort"   "scrpsu"   "dsex"     "iep"      "lep"      "ell3"     "sdracem"  "pared"    "b003501"  "b003601"  "b013801"
##   [13] "b017001"  "b017101"  "b018101"  "b018201"  "b017451"  "m815401"  "m815501"  "m815601"  "m815801"  "m815701"  "rptsamp"  "repgrp1"
##   [25] "repgrp2"  "jkunit"   "origwt"   "srwt01"   "srwt02"   "srwt03"   "srwt04"   "srwt05"   "srwt06"   "srwt07"   "srwt08"   "srwt09"
##   [37] "srwt10"   "srwt11"   "srwt12"   "srwt13"   "srwt14"   "srwt15"   "srwt16"   "srwt17"   "srwt18"   "srwt19"   "srwt20"   "srwt21"
##   [49] "srwt22"   "srwt23"   "srwt24"   "srwt25"   "srwt26"   "srwt27"   "srwt28"   "srwt29"   "srwt30"   "srwt31"   "srwt32"   "srwt33"
##   [61] "srwt34"   "srwt35"   "srwt36"   "srwt37"   "srwt38"   "srwt39"   "srwt40"   "srwt41"   "srwt42"   "srwt43"   "srwt44"   "srwt45"
##   [73] "srwt46"   "srwt47"   "srwt48"   "srwt49"   "srwt50"   "srwt51"   "srwt52"   "srwt53"   "srwt54"   "srwt55"   "srwt56"   "srwt57"
##   [85] "srwt58"   "srwt59"   "srwt60"   "srwt61"   "srwt62"   "smsrswt"  "mrps11"   "mrps12"   "mrps13"   "mrps14"   "mrps15"   "mrps21"
##   [97] "mrps22"   "mrps23"   "mrps24"   "mrps25"   "mrps31"   "mrps32"   "mrps33"   "mrps34"   "mrps35"   "mrps41"   "mrps42"   "mrps43"
##  [109] "mrps44"   "mrps45"   "mrps51"   "mrps52"   "mrps53"   "mrps54"   "mrps55"   "mrpcm1"   "mrpcm2"   "mrpcm3"   "mrpcm4"   "mrpcm5"
##  [121] "m075201"  "m075401"  "m075601"  "m019901"  "m066201"  "m047301"  "m046201"  "m066401"  "m020101"  "m067401"  "m086101"  "m047701"
##  [133] "m067301"  "m048001"  "m093701"  "m086001"  "m051901"  "m076001"  "m046001"  "m046101"  "m067701"  "m046701"  "m046901"  "m047201"
##  [145] "m046601"  "m046801"  "m067801"  "m066601"  "m067201"  "m068003"  "m068005"  "m068008"  "m068007"  "m068006"  "m093601"  "m053001"
##  [157] "m047801"  "m086301"  "m085701"  "m085901"  "m085601"  "m085501"  "m085801"  "m019701"  "m020001"  "m046301"  "m047001"  "m046501"
##  [169] "m066501"  "m047101"  "m066301"  "m067901"  "m019601"  "m051501"  "m047901"  "m053101"  "m143601"  "m143701"  "m143801"  "m143901"
##  [181] "m144001"  "m144101"  "m144201"  "m144301"  "m144401"  "m144501"  "m144601"  "m144701"  "m144801"  "m144901"  "m145001"  "m145101"
```

Meet Your Data - searchSDF

searchSDF() - Search the survey data frame by character strings

```
searchSDF("education", sdf)
```

```
##    variableName                              Labels
## 1        pared Parental education level (from 2 questions)
## 2      b003501                  Mother's education level
## 3      b003601                  Father's education level
## 4      c044007                  Percent in special education
```

- Add argument `levels = TRUE` to return variable levels.

```
searchSDF("b003501", sdf, levels = TRUE)
```

```
## Variable: b003501
## Label: Mother's education level
## Levels (Lowest level first):
##      1. Did not finish H.S.
##      2. Graduated H.S.
##      3. Some ed after H.S.
##      5. I don't know
```

- What occurs with an empty string?

```
searchSDF("", sdf)
```

Meet Your Data - levelsSDF

levelsSDF()

- Show the levels of a variable

```
levelsSDF("b018201", sdf)
```

```
## Levels for Variable 'b018201' (Lowest level first):
##     1. Never (n = 9524)
##     2. Once in a while (n = 3328)
##     3. Half the time (n = 1178)
##     4. All or most of time (n = 2133)
##     8. Omitted* (n = 741)
##     0. Multiple* (n = 11)
##     NOTE: * indicates an omitted level.
```

Meet Your Data - showCodebook

showCodebook()

- Show the levels of a variable

```
showCodebook(sdf)
```

```
##      variableName                                          Labels
## 1           year                                 Assessment year
## 2         cohort                                    All students
## 3         scrpsu                     Scrambled PSU and school code
## 4           dsex                                          Gender
## 5            iep     Student classified as having a disability (504)
## 6            lep           Student classified as ELL (2 categories)
## 7           ell3       Student classified Eng lang learner (3 categ)
## 8        sdracem               Race/ethnicity (from school records)
## 9          pared       Parental education level (from 2 questions)
## 10       b003501                         Mother's education level
## 11       b003601                         Father's education level
## 12       b013801                                   Books in home
## 13       b017001                               Newspaper in home
## 14       b017101                                 Computer at home
```

- **View()** shows a preview of a selected data set

```
View(showCodebook(sdf))
```

Meet Your Data - showPlausibleValues

showPlausibleValues() - Prints all plausible values

```
showPlausibleValues(sdf)

## There are 6 subject scale(s) or subscale(s) in this edsurvey.data.frame:
## 'num_oper' subject scale or subscale with 5 plausible values.
##
## 'measurement' subject scale or subscale with 5 plausible values.
##
## 'geometry' subject scale or subscale with 5 plausible values.
##
## 'data_anal_prob' subject scale or subscale with 5 plausible values.
##
## 'algebra' subject scale or subscale with 5 plausible values.
##
```

- add **verbose = TRUE**

```
showPlausibleValues(sdf, verbose = TRUE)

## There are 6 subject scale(s) or subscale(s) in this edsurvey.data.frame:
## 'num_oper' subject scale or subscale with 5 plausible values.
##    The plausible value variables are 'mrps11', 'mrps12', 'mrps13', 'mrps14', and 'mrps15'
##
## 'measurement' subject scale or subscale with 5 plausible values.
##    The plausible value variables are 'mrps21', 'mrps22', 'mrps23', 'mrps24', and 'mrps25'
##
```

Meet Your Data - showWeights

showWeights() - Prints all weights:

```
showWeights(sdf)
```

```
## There is 1 full sample weight in this edsurvey.data.frame:
##   'origwt' with 62 JK replicate weights (the default).
```

- add **verbose = TRUE** to print the complete list of jackknife replicate weights associated with each full sample weight.

```
showWeights(sdf, verbose = TRUE)
```

```
## There is 1 full sample weight in this edsurvey.data.frame:
##   'origwt' with 62 JK replicate weights (the default).
##     Jackknife replicate weight variables associated with the full sample weight 'origwt':
##     'srwt01', 'srwt02', 'srwt03', 'srwt04', 'srwt05', 'srwt06', 'srwt07', 'srwt08', 'srwt09', 'srwt10', 'srwt11',
##     'srwt12', 'srwt13', 'srwt14', 'srwt15', 'srwt16', 'srwt17', 'srwt18', 'srwt19', 'srwt20', 'srwt21', 'srwt22',
##     'srwt23', 'srwt24', 'srwt25', 'srwt26', 'srwt27', 'srwt28', 'srwt29', 'srwt30', 'srwt31', 'srwt32', 'srwt33',
##     'srwt34', 'srwt35', 'srwt36', 'srwt37', 'srwt38', 'srwt39', 'srwt40', 'srwt41', 'srwt42', 'srwt43', 'srwt44',
##     'srwt45', 'srwt46', 'srwt47', 'srwt48', 'srwt49', 'srwt50', 'srwt51', 'srwt52', 'srwt53', 'srwt54', 'srwt55',
##     'srwt56', 'srwt57', 'srwt58', 'srwt59', 'srwt60', 'srwt61', and 'srwt62'
```

Meet Your Data - Omitted Levels

- Levels of the variables that will be omitted by default from the `edsurvey.data.frame`

```
> sdf
edsurvey.data.frame with 17606 rows and 302 columns.

There are 1 full sample weight(s) in this edsurvey.data.frame
  'origwt' with 62 JK replicate weights (the default).

There are 6 subject scale(s) or subscale(s) in this edsurvey.data.frame
  'num_oper' subject scale or subscale with 5 plausible values.
  'measurement' subject scale or subscale with 5 plausible values.
  'geometry' subject scale or subscale with 5 plausible values.
  'data_anal_prob' subject scale or subscale with 5 plausible values.
  'algebra' subject scale or subscale with 5 plausible values.
  'composite' subject scale or subscale with 5 plausible values (the default).

Omitted Levels: 'Multiple', 'NA', 'Omitted'

Default Conditions:
  tolower(rptsamp) == "reporting sample"

Achievement Levels:
  Basic:      262
  Proficient: 299
  Advanced:   333

Survey: NAEP
```

Meet Your Data - Default Conditions

- Special considerations for a particular `edsurvey.data.frame`

```
> sdf
edsurvey.data.frame with 17606 rows and 302 columns.

There are 1 full sample weight(s) in this edsurvey.data.frame
  'origwt' with 62 JK replicate weights (the default).

There are 6 subject scale(s) or subscale(s) in this edsurvey.data.frame
   'num_oper' subject scale or subscale with 5 plausible values.
   'measurement' subject scale or subscale with 5 plausible values.
   'geometry' subject scale or subscale with 5 plausible values.
   'data_anal_prob' subject scale or subscale with 5 plausible values.
   'algebra' subject scale or subscale with 5 plausible values.
   'composite' subject scale or subscale with 5 plausible values (the default).

Omitted Levels: 'Multiple', 'NA', 'Omitted'

Default Conditions:
   tolower(rptsamp) == "reporting sample"

Achievement Levels:
   Basic:       262
   Proficient: 299
   Advanced:    333

Survey: NAEP
```

Data Manipulation

EdSurvey Calls Network Connection

Small Memory Footprint
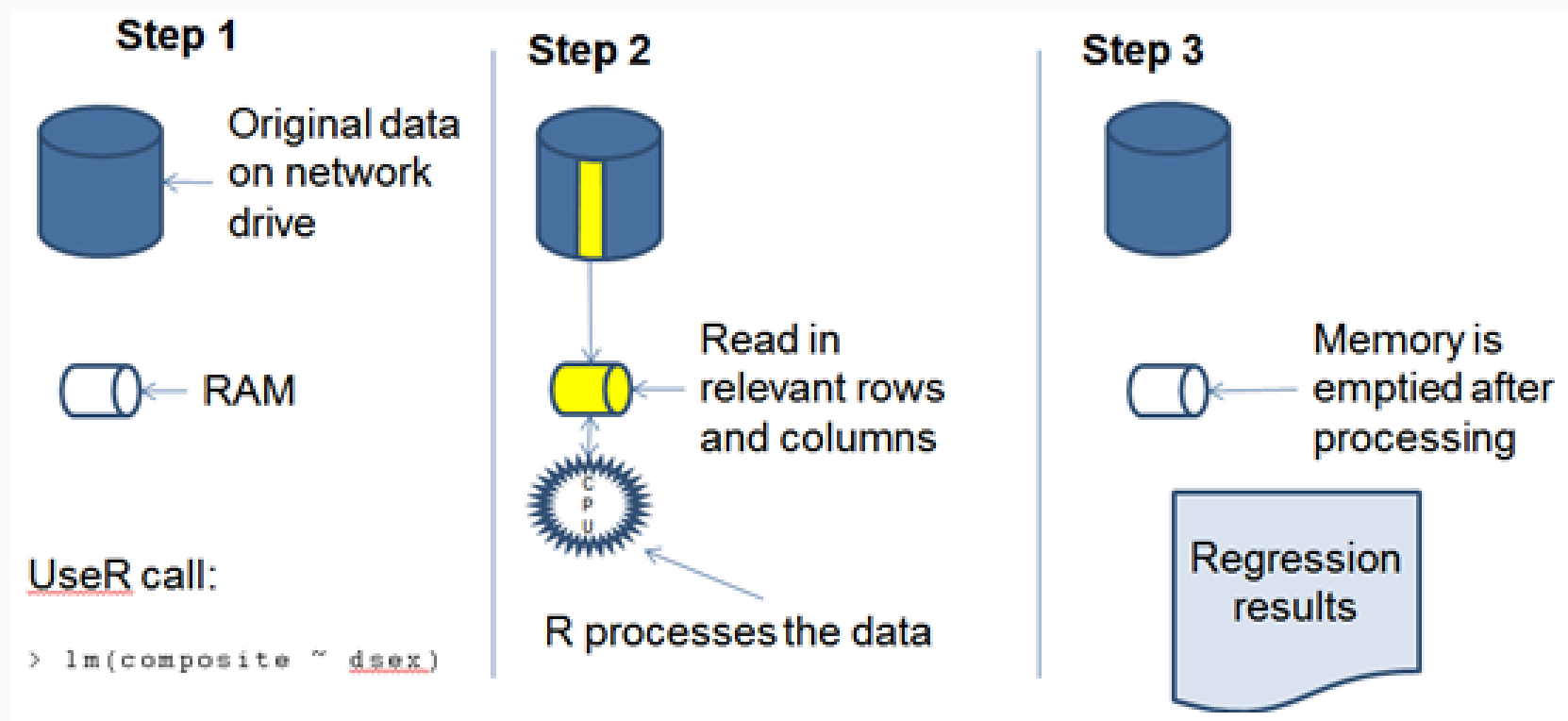
Data Manipulation - getData

`getData()`: reads in selected variables and sampling weights from the EdSurvey database and returns a `light.EdSurvey.data.frame` (a data frame like object) into the Global environment.

Functionality

- Retrieve variables by call.

- Manipulate the resulting `light.EdSurvey.data.frame`:

  - Subset.

  - Recode.

  - Drop levels.

- Use `EdSurvey` package functions on `light.EdSurvey.data.frames`.

- Related Documentation - getData.pdf, Chap 9, EdSurvey Book

Data Manipulation - getData

getData()

```
gddat <- getData(sdf, varnames = c('dsex', 'sdracem', 'b018201', 'b017451',
                                   'composite', 'geometry', 'origwt'),
                 addAttributes = TRUE, omittedLevels = FALSE)
```

NAEP mathematics composite scale scores of 8th grade students

- A vector of variable names, including `dsex` (Gender), `sdracem` (Race/ethnicity), `b018201` (Language other than English spoken in home) and `b017451` (Frequency of talk about studies at home)

- Overall math performance across subscales (`composite`) and five others associated with `geometry`

- The sampling weight for this dataframe: `origwt`

Data Manipulation - getData

Output:

```
# Note: head returns the first 6 rows of a data frame

head(gddat)
```

```
##      dsex sdracem           b018201             b017451 mrpcm1 mrpcm2 mrpcm3 mrpcm4 mrpcm5 mrps31 mrps32 mrps33 mrps34 mrps35
## 1   Male   White             Never             Every day 318.01 303.68 296.61 328.97 315.70 294.79 286.84 264.39 311.77 304.62
## 2 Female   White             Never    About once a week 288.43 283.93 280.45 290.03 286.23 277.26 266.43 261.98 286.23 264.76
## 3 Female   White             Never             Every day 342.72 338.03 329.48 352.46 342.26 354.18 320.11 331.88 354.47 365.00
## 4   Male   White             Never             Every day 348.76 321.79 327.87 333.35 327.32 326.91 302.79 321.28 333.43 318.45
## 6 Female   White Once in a while Once every few weeks 278.44 245.08 263.00 277.50 285.04 263.22 232.62 260.05 280.10 278.96
## 7   Male   White Once in a while  2 or 3 times a week 327.95 338.59 328.07 334.07 320.02 309.38 317.19 328.37 331.75 309.70
##   srwt01 srwt02 srwt03 srwt04 srwt05 srwt06 srwt07 srwt08 srwt09 srwt10 srwt11 srwt12 srwt13 srwt14 srwt15 srwt16 srwt17 srwt18
## 1 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004
## 2 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004
## 3 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004
## 4 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004
## 6 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004
## 7 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004
##   srwt19 srwt20 srwt21 srwt22 srwt23 srwt24 srwt25 srwt26 srwt27 srwt28 srwt29 srwt30 srwt31 srwt32 srwt33 srwt34 srwt35 srwt36
## 1 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 2.2008 1.1004 1.1004 1.1004 1.1004
## 2 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 2.2008 1.1004 1.1004 1.1004 1.1004
## 3 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 2.2008 1.1004 1.1004 1.1004 1.1004
## 4 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 1.1004 2.2008 1.1004 1.1004 1.1004 1.1004
```

Data Manipulation - getData

getData()

```r
gddat <- getData(sdf, varnames = c('dsex', 'sdracem', 'b018201', 'b017451',
                                   'composite', 'geometry', 'origwt'),
             addAttributes = TRUE, omittedLevels = FALSE)
```

- A few important things to note:
  - "`addAttributes = TRUE`" allows the data.frame to be passed to `EdSurvey` package functions

  - all of the jackknife replicates are automatically returned (`srwt01` to `srwt62`)

  - "`omittedLevels = FALSE`" returns variables with special values (such as multiple entries or NA's) and manipulated by the user

Data Manipulation - subset

subset() : Returns only the data matching elements from a variable

- Subset the connection to the data for all analyses:

```
subsetSDF <- subset(sdf, dsex %in% c("Male"))
```

- As expected the `subsetSDF` contains about half of the rows as the original:

```
dim(sdf)
```
```
## [1] 17606   302
```
```
dim(subsetSDF)
```
```
## [1] 8905  302
```

Data Manipulation - recode.sdf

recode.sdf() is used to recode the levels of a variable

- collapse or rename values

```
sdf2 <- recode.sdf(sdf, recode =
                    list(b017451 = list(from = c("Never or hardly ever", "
                                        to = c("Infrequently")),
                         b017451 = list(from = c("Every day"),
                                        to = c("Frequently")))
                    )
searchSDF("b017451", sdf2, levels = TRUE)
```

```
## Variable: b017451
## Label: Talk about studies at home
## Levels (Lowest level first):
##      3. About once a week
##      4. 2 or 3 times a week
##      8. Omitted
##      0. Multiple
##      9. Infrequently
##      10. Frequently
```

Data Manipulation - rename.sdf

`rename.sdf()` is used to rename variables

```
sdf2 <- rename.sdf(sdf2, oldnames = "b017451",
                         newnames = "studytalkfrequency")
searchSDF("studytalkfrequency", sdf2, levels = TRUE)
```

```
## Variable: studytalkfrequency
## Label: Talk about studies at home
## Levels (Lowest level first):
##      3. About once a week
##      4. 2 or 3 times a week
##      8. Omitted
##      0. Multiple
##      9. Infrequently
##      10. Frequently
```