The R EdSurvey Package

Analyzing NAEP and TIMSS Data Using R: Day 3

Presenters: Eric Buehler, Paul Bailey, Charles Blankenship & Sinan Yavuz

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_day3.R

Outline of EdSurvey Workshop - Day 3

- Summary statistics -edsurveyTable, etc
- Linear regression (mention HLM/WeMix)
- Percentile analysis
- Quantile regression
- Doing your own analyses using NAEP/TIMSS data

Data Processing

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

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

NAEP Primer:

```
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):

```
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
```

- For NAEP data and other datasets that have a default weight variable, `summary2` produces weighted statistics by default. If a specified `variable` is a plausible value and weight option is selected, `summary2` statistics account for both plausible value pooling and weighting.

Summary statistics

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

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

```
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):

```
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
```

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

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
```

Cross tabulation

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

- `formula`: typically written as `a ~ b + c`, in which:
  - `a`: a continuous variable (optional) that the function will return the weighted mean for.
  - `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 studies talk about studies at home
- `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`.

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

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, pctAggregatior
```

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

Self-Reflection - edsurveyTable

Ask yourself: How would you use `edsurveyTable` to create a summary table with these parameters:

- overall math performance across subscales (`composite`)
- a variable that has to do with IEP status
- a variable that has to do with number of books at home

Self-Reflection - edsurveyTable

Scenario Result:

```
edexercise <- edsurveyTable(composite ~ iep + b013801,
                            weightVar = 'origwt', data = sdf)
edexercise
```

```
##
## Formula: composite ~ iep + b013801
##
## Plausible values: 5
## jrrIMax: 1
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## full data n: 17606
## n used: 16351
##
##
## Summary Table:
##  iep b013801    N     WTD_N      PCT   SE(PCT)     MEAN  SE(MEAN)
##  Yes    0-10  304  297.1972 17.33406 1.0388812 226.1623 2.3075125
##  Yes   11-25  430  429.6252 25.05794 1.4034976 231.8103 2.3796081
##  Yes  26-100  517  530.9539 30.96795 1.5297784 249.2306 2.4682667
##  Yes    >100  457  456.7507 26.64004 1.6556494 257.6787 2.8205193
##   No    0-10 1720 1890.3037 12.56502 0.4765198 257.6975 1.2861579
##   No   11-25 2936 3170.9954 21.07789 0.5632689 266.0401 0.9908671
##   No  26-100 5330 5350.4978 35.56524 0.6242526 281.5820 0.8305656
##   No    >100 4657 4632.3807 30.79185 0.8511616 296.2606 1.0533164
```

Linear Regression

NAEP
NATIONAL ASSESSMENT OF EDUCATIONAL PROGRESS

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`: an `EdSurvey` object 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:

```
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|)
```

- Note: we can also use `lm.sdf` for `TIMSS` data.

Regressions - lm.sdf()

Example of multiple regression:

```
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

```
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
```

Self-Reflection - lm.sdf

Ask yourself: How would you use `lm.sdf` to perform a regression with multiple predictors using these parameters:

- overall math performance across subscales (`composite`)
- variable that has to do with computers at home
- variable that has to do with language other than English spoken in home

Self-Reflection - lm.sdf

Scenario Result:

```
lmexercise2 <- lm.sdf(composite ~ b017101 + b018201,
                      weightVar = 'origwt', data = sdf)
summary(lmexercise2)
```

```
##
## Formula: composite ~ b017101 + b018201
##
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## Plausible values: 5
## jrrIMax: 1
## full data n: 17606
## n used: 15884
##
## Coefficients:
##                                coef        se         t    dof  Pr(>|t|)
## (Intercept)                281.95112   0.80871 348.64281 43.827 < 2.2e-16 ***
## b017101No                  -22.44306   1.36521 -16.43932 42.935 < 2.2e-16 ***
## b018201Once in a while       0.63672   0.90717   0.70188 61.423    0.4854
## b018201Half the time        -7.32985   1.58448  -4.62604 50.514 2.624e-05 ***
## b018201All or most of time -12.61417   1.27458  -9.89675 29.860 6.121e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared: 0.0658
```

HLM - WeMix; mixed.sdf()

- `EdSurvey` has the ability to fit ***weighted hierarchical linear models*** using the `WeMix` package that accounts for the sampling strategy as well as random effects.
- When the outcome is not represented with plausible values users may use `mix()` directly
- To fit these models users need information about the weights at every level modeled
- This topic deserves its own series see `?mixed.sdf` and or the WeMix documentation for examples and more details

Even More Plausible Values

- Plausible values only work when the model used to generate them includes all regressors
- When a variable is not included in the *conditioning model* the estimates will be biased
- If you want to include additional variables, such as those from an external data source, in a regression you need to use *marginal maximum likelihood*
- The `Dire` package does this and can be used with `mml.sdf` in `EdSurvey`

```
m1 <- mml.sdf(algebra ~ dsex + b013801, sdf, weightVar='origwt')
```

```
## Warning in mml.sdf(algebra ~ dsex + b013801, sdf, weightVar = "origwt"): Using IRT parameters for National tests by default.
## Warning in mml.sdf(algebra ~ dsex + b013801, sdf, weightVar = "origwt"): These items were in the assessment, but not in your dat
## m140501, m140901, m141501, m141901, m052501, m067001, m051701, m140701, m141601, m0732cl, m092201, m092601, m140601, m141201,
## m141401, m141701, m021001, m020901, m092401, m140401, m140801, m141001, m013331, m073301, m019201, m141101, m141301, m141801,
## m012231, m073001, m073101, m012431, m091901, and m073601
## Warning in getData(data = sdf, varnames = c(polyParamTab$ItemID, dichotParamTab$ItemID, : Updating labels on 'm144901' because
## there are multiples of the label 'Correct'.
## Warning in getData(data = sdf, varnames = c(polyParamTab$ItemID, dichotParamTab$ItemID, : Updating labels on 'm145101' because
## there are multiples of the label 'Correct'.
## Pre-processing Completed.
## Starting MML Procedure.
```

Summary - mml.sdf()

```
summary(m1)
```

```
## Call:
## mml.sdf(formula = algebra ~ dsex + b013801, data = sdf, weightVar = "origwt")
## Summary Call:
## summary.edSurveyMML(object = m1)
##
## Summary:
##                   Estimate    StdErr  t.value
## (Intercept)     255.40723   1.05831 241.3351
## dsexFemale       -1.35625   0.69508  -1.9512
## b01380111-25     11.51585   1.26719   9.0877
## b01380126-100    26.48379   1.16879  22.6592
## b013801>100      40.82529   1.19132  34.2689
## b013801Omitted   14.59416   4.31914   3.3790
## b013801Multiple  10.85895  15.13167   0.7176
##
## Residual Variance Estimate:
##               Estimate    StdErr
```

Percentile Analysis

NAEP
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

Percentile Analysis

`percentile()` - the full range of quantiles

```
# 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

- Related Documentation - EdSurvey-Percentiles.pdf

Self-Reflection - percentile (Breakout Session - 10min)

How to use EdSurvey functions to create a percentile results using `percentile()` with these parameters:

- algebra sub-scale
- by gender (dsex)
- 5th and 95th percentile

  hint: you can use EdSurvey's subset() function

Self-Reflection Solution - percentile

One of many possible solutions:

```
sexes <- levels( sdf$dsex )
sexes #  lapply to pass each sex to subset() using an anon function
```

```
## [1] "Male"   "Female"
```

```
sex_list <- lapply( sexes , function(each_sex) {
                                                subset(sdf, subset=dsex==eac
# create an edsurvey df list, labelling each subset
esdflist <- edsurvey.data.frame.list(sex_list, labels= sexes)

percentile("algebra", percentiles = c(5,95), weightVar = "origwt", esdflist
```

```
## percentileList
## Call: percentile(variable = "algebra", percentiles = c(5, 95), data = esdflist,
##     weightVar = "origwt")
##
##  labels percentile estimate       se       df confInt.ci_lower confInt.ci_upper
##    Male          5  215.227 1.354715 23.63708         211.7716         218.6158
##  Female          5 218.4289 1.760181 67.02682         214.8175         221.6092
##    Male         95 338.8147 1.614728 37.56567         335.4467         342.7369
##  Female         95  336.329 1.436898 24.86614         333.9032         339.4471
```

Quantile Regression Analysis

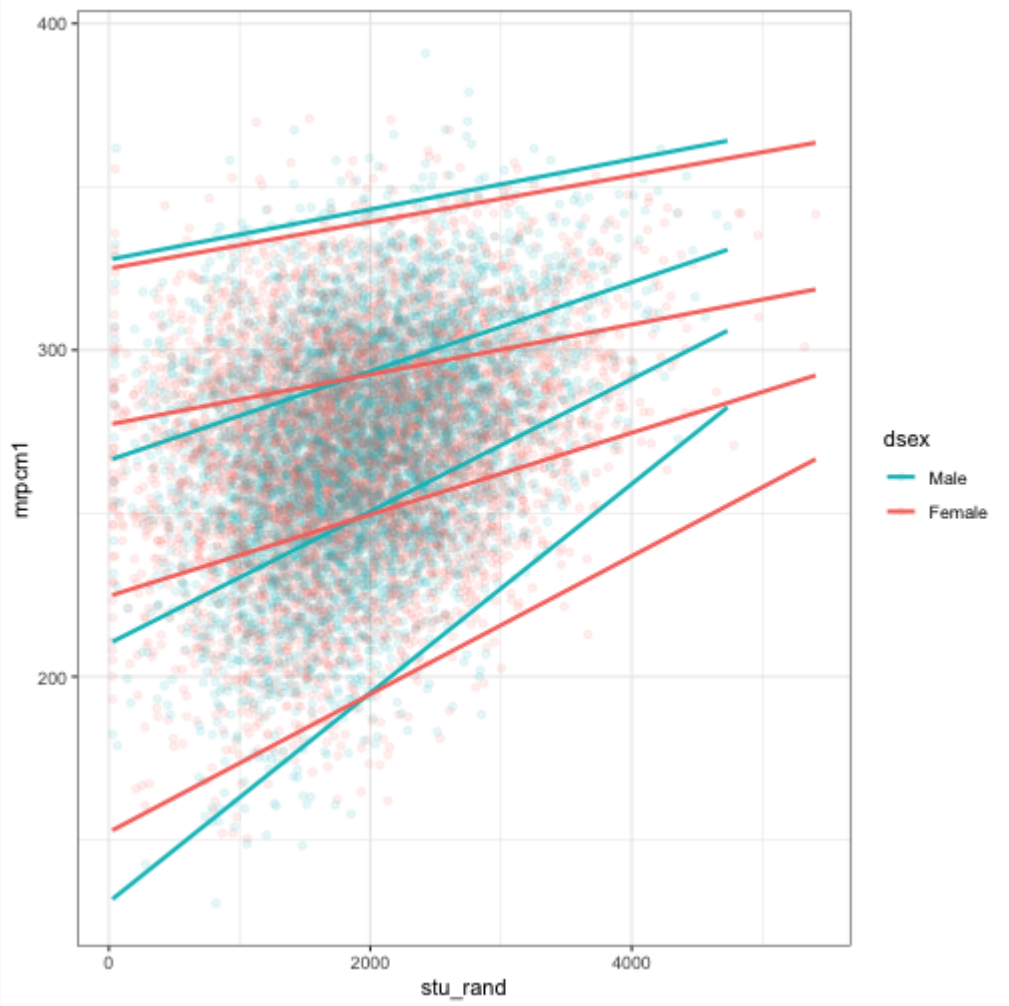
NAEP
NATIONAL ASSESSMENT OF EDUCATIONAL PROGRESS

Quantile Regression Analysis

- Quantile Regression; a visual example

```
invisible(lapply(c("EdSurvey", "ggplot2", "RColorBrewer"), library, charact
sdf <- readNAEP(system.file("extdata/data", "M36NT2PM.dat", package = "NAEP
# make a regular data frame, with some columns we may be interested in
allvars <- colnames(sdf)
somevars <- allvars[c(1:10, grep(sdf$stratumVar , allvars),grep(sdf$psuVar
df <- getData(data=sdf, varnames=somevars)
# for demo purposes, add a random continuous IV that varies by sex
rand_vals <- list(df$mrpcm1 * pmax(.01, rnorm(1:nrow(df),mean=7,sd=2.5)), c
df$stu_rand[df$dsex==1] <- df$mrpcm1[df$dsex==1]/7 + rand_vals[[1]][df$dsex
df$stu_rand[df$dsex==2] <- df$mrpcm1[df$dsex==2]/8 + rand_vals[[2]][df$dsex
# ggplot will plot quantile regression lines. see ggplot2::geom_quantile
ggplot(df, aes(y=mrpcm1, x=stu_rand, color=dsex)) + geom_point(alpha=.1) +
  geom_quantile(aes(y=mrpcm1, x=stu_rand, color=dsex ), quantiles= c(.01,
```

Quantile Regression Analysis



Quantile Regression Analysis

`rq.sdf()` - computes an estimate on the tau-th conditional quantile function of the response

- assumes a linear specification of the quantile regression function
- jackknife is the only available variance estimator
- at present only accepts a single tau value (median is the default)
- based on quantreg package (Roger Koenker, 2021)
- ?rq.sdf
- Related Documentation - quantreg.pdf

Quantile Regression Analysis

```
sdf <- readNAEP(system.file("extdata/data", "M36NT2PM.dat", package = "NAEP
# conduct quantile regression at a given tau value (by default, tau is set
rq1 <- rq.sdf(composite ~ dsex + b017451, data=sdf, tau = .75)
summary(rq1)
```

```
##
## Formula: composite ~ dsex + b017451
##
## tau: 0.75
## jrrIMax: 1
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## full data n: 17606
## n used: 16331
##
## Coefficients:
##                                 coef        se         t    dof  Pr(>|t|)
## (Intercept)                294.12400   1.36145 216.03674 31.146 < 2.2e-16 ***
## dsexFemale                  -4.33000   0.99955  -4.33195 28.455 0.0001668 ***
## b017451Once every few weeks  6.26800   1.42437   4.40053 62.544 4.289e-05 ***
## b017451About once a week    12.71800   1.77125   7.18022 38.021 1.400e-08 ***
## b0174512 or 3 times a week  15.77400   1.52721  10.32867 32.974 7.224e-12 ***
## b017451Every day            12.20000   1.69426   7.20078 65.993 7.115e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Quantile Regression Analysis

- Often we want multiple quantiles. First, some setup.

```
# convert the data frame back to an edsurvey data frame using rebindAttribu
new_sdf <- rebindAttributes(df, sdf)
# typically, one is interested in multiple quantiles, which can be achieved
taus <- c(.01, .25, .50, .75, .99)
# set color palette (blue for boys, red for girls), go slightly darker for
shades <- c(RColorBrewer::brewer.pal(length(taus)+2, "Blues")[1:length(taus
            RColorBrewer::brewer.pal(length(taus)+2, "Reds")[1:length(taus)
```

Quantile Regression Analysis

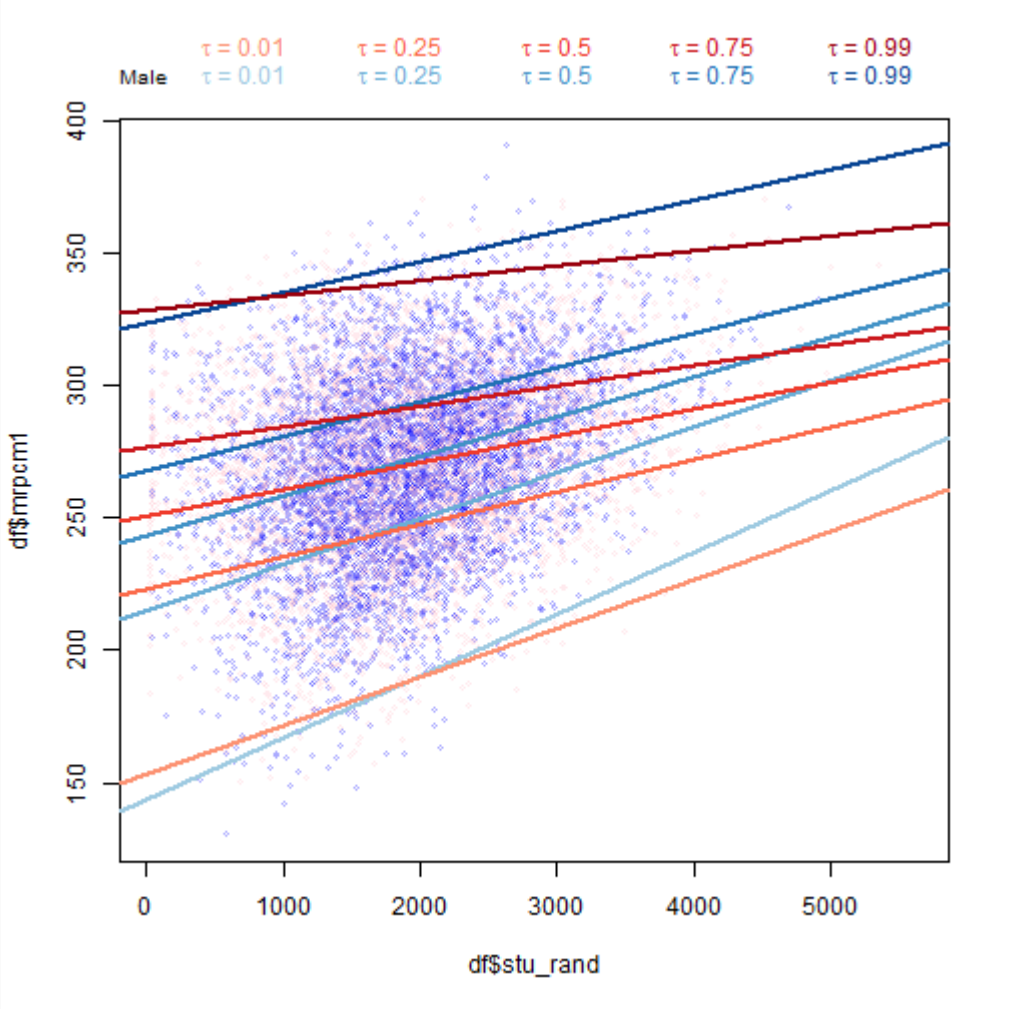
- Run quantile regressions and plot lines for multiple quantiles.

```
plot(df$stu_rand, df$mrpcm1, col= alpha(c("blue","pink")[df$dsex], .33), ce
for( tau_n in 1:length(taus)) {                    # tau_n = 1,2,3... to num
  for (sex_n in 1:length(levels(df$dsex))) {       # sex_n = 1 for males (ds

    rq <- rq.sdf(composite ~ stu_rand,  data=new_sdf[new_sdf$dsex==sex_n,],
    abline(rq$coef, lwd=2.5, col=shades[((sex_n-1)*5) + tau_n])

    if (tau_n==1 & sex_n==1) {  # add text to margins and stack output
      mtext(line=sex_n, adj=0, text=paste(levels(df$dsex)[sex_n]," "), col=
      mtext(line=sex_n, adj=tau_n*.21-.1, text=bquote(tau ~ "=" ~ .(taus[ta
      models <- data.frame("sex"=levels(df$dsex)[sex_n], "tau"=taus[tau_n],
    } else {
      mtext( line=sex_n, adj=tau_n*.21-.1, text=bquote(tau ~ "=" ~ .(taus[t
      models <- rbind(models, data.frame("sex"=levels(df$dsex)[sex_n], "tau
  }
}
```

Quantile Regression Analysis



Quantile Regression Analysis

- View quantile regression output.

```
models
```

```
##                sex  tau        term         coef           se          t      dof     Pr...t..
## (Intercept)    Male 0.01 (Intercept) 1.439302e+02 7.3255806489  19.647619 18.81585 5.373479e-14
## stu_rand       Male 0.01    stu_rand 2.336310e-02 0.0044089530   5.299012 10.79391 2.693801e-04
## (Intercept)1 Female 0.01 (Intercept) 1.531052e+02 9.7508684675  15.701701 17.16792 1.279998e-11
## stu_rand1    Female 0.01    stu_rand 1.838698e-02 0.0050855234   3.615553 15.09292 2.520913e-03
## (Intercept)2   Male 0.25 (Intercept) 2.152974e+02 3.4572767906  62.273683 51.56615 0.000000e+00
## stu_rand2      Male 0.25    stu_rand 1.731027e-02 0.0017095966  10.125354 46.81865 2.235989e-13
## (Intercept)3 Female 0.25 (Intercept) 2.232113e+02 3.9453209479  56.576211 41.73574 0.000000e+00
## stu_rand3    Female 0.25    stu_rand 1.221357e-02 0.0016536342   7.385899 35.55651 1.110474e-08
## (Intercept)4   Male 0.50 (Intercept) 2.432040e+02 3.0384548214  80.042000 37.73396 0.000000e+00
## stu_rand4      Male 0.50    stu_rand 1.496582e-02 0.0013745894  10.887483 45.34113 3.064216e-14
## (Intercept)5 Female 0.50 (Intercept) 2.508590e+02 2.1895191024 114.572659 21.14549 0.000000e+00
## stu_rand5    Female 0.50    stu_rand 1.004908e-02 0.0010340882   9.717820 21.04905 3.119162e-09
## (Intercept)6   Male 0.75 (Intercept) 2.682928e+02 3.6891581779  72.724657 53.43964 0.000000e+00
## stu_rand6      Male 0.75    stu_rand 1.299375e-02 0.0019008634   6.835711 74.28693 1.956488e-09
## (Intercept)7 Female 0.75 (Intercept) 2.764803e+02 1.9794796143 139.673236 32.39708 0.000000e+00
## stu_rand7    Female 0.75    stu_rand 7.776635e-03 0.0009840611   7.902594 37.76808 1.596942e-09
## (Intercept)8   Male 0.99 (Intercept) 3.236108e+02 9.3685652509  34.542199 42.95047 0.000000e+00
## stu_rand8      Male 0.99    stu_rand 1.166189e-02 0.0057078521   2.043130 41.61915 4.740545e-02
## (Intercept)9 Female 0.99 (Intercept) 3.286164e+02 6.5120946731  50.462477 29.95339 0.000000e+00
## stu_rand9    Female 0.99    stu_rand 5.655736e-03 0.0023601722   2.396324 24.83438 2.441819e-02
```

Self-Reflection - Quantile Regression (Breakout Session 10min)

Ask yourself: Use EdSurvey functions to perform quantile regression using `rq.sdf()` with these parameters:

- algebra performance by sex and race
- 5th and 95th percentile
- compare run times of Frisch-Newton vs default method qr.fit method

Self-Reflection Solution - Quantile Regression

```
qfit5 <- rq.sdf(composite ~ dsex + sdracem , tau=.05, data = sdf)
qfit95 <- rq.sdf(composite ~ dsex + sdracem , tau=.95, data = sdf)
summary(qfit5)
```

```
##
## Formula: composite ~ dsex + sdracem
##
## tau: 0.05
## jrrIMax: 1
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## full data n: 17606
## n used: 16915
##
## Coefficients:
##                                 coef       se         t     dof  Pr(>|t|)
## (Intercept)                  230.02800   2.96635  77.54584 51.3270 < 2.2e-16 ***
## dsexFemale                     1.95800   2.89235   0.67696 33.2492   0.50312
## sdracemBlack                 -33.15800   3.30823 -10.02289 32.9875 1.535e-11 ***
## sdracemHispanic              -28.69800   2.98598  -9.61090 72.0220 1.533e-14 ***
## sdracemAsian/Pacific Island   -7.73200   3.46678  -2.23031 55.2389   0.02981 *
## sdracemAmer Ind/Alaska Natv  -22.15800   8.43002  -2.62846  3.6187   0.06458 .
## sdracemOther                 -10.41800  11.38321  -0.91521 55.2939   0.36406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Self-Reflection Solution - Quantile Regression

```
#install.packages("rbenchmark")
rbenchmark::benchmark(replications = 3, columns = c("test", "replications",
 "Barrodale & Roberts" ={v1<-rq.sdf(composite ~ dsex , tau=.95, data = sdf,
 "Frisch-Newton"       ={v2<-rq.sdf(composite ~ dsex , tau=.95, data = sdf,
)
```

```
##                 test replications elapsed relative user.self sys.self
## 1 Barrodale & Roberts            3  37.379    3.794    35.086    1.136
## 2       Frisch-Newton            3   9.851    1.000     8.425    0.853
```

```
v2$coefmat
```

```
##                coef       se          t      dof   Pr(>|t|)
## (Intercept) 335.558 2.149052 156.142345 57.04900 0.00000000
## dsexFemale   -4.162 1.785129  -2.331485 60.48926 0.02307631
```

```
v1$coefmat
```

```
##                coef       se          t      dof   Pr(>|t|)
## (Intercept) 335.558 2.149052 156.142345 57.04900 0.00000000
## dsexFemale   -4.162 1.785129  -2.331485 60.48926 0.02307631
```

Logistic Regression

`logit.sdf()` and `probit.sdf()` - predict binary outcomes from a set of continuous predictor variables (sampling weights and variance estimates)

- `I()` used to specify the outcome level of the `b013801` variable (Books in home)

```
logit1 <- logit.sdf(I(b013801 %in% ">100") ~ dsex,
                    weightVar = 'origwt', data = sdf)
```

Logistic Regression

```
summary(logit1)
```

```
##
## Formula: b013801 ~ dsex
## Family: binomial (logit)
##
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## full data n: 17606
## n used: 16359
##
## Coefficients:
##                   coef         se          t    dof  Pr(>|t|)
## (Intercept)  -0.920421   0.046355 -19.855835 60.636 < 2.2e-16 ***
## dsexFemale    0.178274   0.050129   3.556331 54.578 0.0007863 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The *log* odds of having more than 100 books in home (versus less than or equal to 100 books) increases by 0.178274 for female students, compared with male students.

Logistic Regression

```
oddsRatio(logit1)
```

```
##                    OR      2.5%     97.5%
## (Intercept) 0.3983511 0.3630823 0.4370459
## dsexFemale  1.1951531 1.0809029 1.3214796
```

Alternatively, the odds of having more than 100 books in home (versus less than or equal to 100 books) increases by 1.1951531 for female students, compared with male students.

Bonus point: The Wald test is available for `logit.sdf` and `lm.sdf` models. See details in `?waldTest`.

- Related Documentation - EdSurvey-Wald Test.pdf

Self-Reflection - Logistic Regression

Ask yourself: Use EdSurvey functions to perform a logistic regression using `logit.sdf` using these parameters:

- use an outcome variable that has to do with English Langauge Learners (ELL)
  - code it to 1 when `%in% "Yes"`
- use a predictor variable that has to do with language spoken at home

Self-Reflection - Logistic Regression

Scenario Result:

```
logitexercise1 <- logit.sdf(I(lep %in% "Yes") ~ b018201,
                            weightVar = 'origwt', data = sdf)
summary(logitexercise1)
```

```
##
## Formula: lep ~ b018201
## Family: binomial (logit)
##
## Weight variable: 'origwt'
## Variance method: jackknife
## JK replicates: 62
## full data n: 17606
## n used: 16159
##
## Coefficients:
##                               coef        se         t     dof  Pr(>|t|)
## (Intercept)                -4.78197   0.19709 -24.26309  9.8977 3.796e-10 ***
## b018201Once in a while      1.94536   0.20702   9.39713 26.0267 7.539e-10 ***
## b018201Half the time        3.13919   0.15354  20.44573 38.9521 < 2.2e-16 ***
## b018201All or most of time  3.63098   0.17657  20.56339 26.7407 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Homework - 1

- Get prepared for the next week

```
#Read in multiple USA G4 students' data
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")
```

Homework - 2

```
##Read in 2019 TIMSS G4 student data from multiple education systems.
## Pick the education systems that you like. Below are examples
TIMSS19USA<- readTIMSS(path = "C:/TIMSS/2019/",
                       countries = c("usa"), gradeLvl = "4")
TIMSS19FIN<- readTIMSS(path = "C:/TIMSS/2019/",
                       countries = c("fin"), gradeLvl = "4")
TIMSS19HKG<- readTIMSS(path = "C:/TIMSS/2019/",
                       countries = c("hkg"), gradeLvl = "4")
```

- Additionally, read one more country data of your choice for 4th grade TIMSS data.