RCT-YES™

Updates to the User's Manual and Statistical Theory Appendix for *RCT-YES* Versions 1.2 and 1.1

Peter Z. Schochet:	Project Lead, Author of Manual
Carlo Caci:	Interface Developer
Mason DeCamillis:	R Software Developer
Matthew Jacobus:	Stata Software Developer

Mathematica Policy Research, Inc.

January 2018

RCT-YES was developed by Mathematica Policy Research, Inc.
with funding from the Institute of Education Sciences
under Contract ED-IES-12-C-0057 with
Decision Information Resources, Inc.

ies NATIONAL CENTER FOR EDUCATION EVALUATION AND REGIONAL ASSISTANCE
Institute of Education Sciences
U.S Department of Education

RCT-YES™

This manual discusses updates to the *RCT-YES* software for new releases since Version 1.0. It serves as a supplement to the more detailed May 2016 *RCT-YES* User's Manual and Statistical Theory Appendix (both found at www.rct-yes.com). The manual discusses new program features and the underlying statistical theory, and provides an updated dictionary of input variables.

Contents

Tables

Introduction

The free *RCT-YES* software (www.rct-yes.com) estimates and reports average treatment effects for evaluations of interventions, programs, and policies using randomized controlled trial designs (RCTs) or quasi-experimental designs (QEDs) with comparison groups. The software is applicable to a wide range of evaluation designs used in social policy and related research. The methods underlying the software are based on a new design-based statistical theory that has important advantages over traditional model-based methods used in social policy research (Schochet, 2016; 2017a; 2017b). The software is user friendly with no knowledge of computer programming required. The software reports study findings in formatted tables and graphs that meet industry standards, and conform to What Works Clearinghouse evidence reviews (Scher and Cole, 2017).

RCT-YES Version 1.0 was released in May 2016 with associated documentation available at www.rct-yes.com. Version 1.1 was released in June 2016 to fix minor program bugs. The most recent Version 1.2 was released in January 2018 with important new features implemented in response to user feedback. The key new feature is that the software can now accommodate designs with more than two research groups (multi-armed designs).

This manual discusses *RCT-YES* updates since Version 1.0 and serves as a supplement to the May 2016 *RCT-YES* User's Manual and Statistical Theory Appendix (both found at www.rct-yes.com). The manual first provides an updated dictionary of input variables and then provides an updated layout of the .csv file containing analysis results that can be used for further analyses and reporting. The manual then discusses updates in Version 1.2, including program inputs and the underlying design-based statistical theory used for impact estimation. The manual concludes with a discussion of changes implemented in Version 1.1.

(This page left intentionally blank for double-sided copying)

Updated Dictionary of Program Input Statements

Table 1. Updated dictionary of input statements for *RCT-YES*

Input variable	Variable definition	Variable format	Additional information
Getting Started: R/Stata and Input Data			
STAT_PACKAGE	Statistical package for the analysis	R Stata	**Required**
DATA_FILE	Name of input data file for the analysis	One record per student, educator, or cluster. The file must be a .rds file for R or a .dta file for Stata.	**Required**
Design Selection and Title			
DESIGN	Type of design	1 = Non-clustered, non-blocked 2 = Non-clustered, blocked 3 = Clustered, non-blocked 4 = Clustered, blocked	**Required**
TITLE	Title for output tables	Character	Optional
Required Design Parameters			
TC _STATUS	Name of variable with codes signifying the research group for each observation	0 = Control group, if one exists 1, 2, … = Treatment groups	**Required for all observations** The codes must be consecutive integers starting at 0 if there is a control group or 1 if not A maximum of 6 research groups is allowed, with 1 control and 5 treatment groups or 6 treatment groups if there is no control group
BLOCK_ID	Name of variable containing the block identification codes	Numeric or character	**Required for Designs 2 and 4 for all observations** For the default finite-population (FP) model, blocks are included in the analysis if they contain at least 2 observations for each research group For the super-population (SP) model or BLOCK_FE=1 FP model, blocks are included with at least 1 observation per research group

Input variable	Variable definition	Variable format	Additional information
MATCHED_PAIR	Indicator for a matched pair design (or matched group design if there are more than two research groups)	0 = Not a matched pair design (default) 1 = Matched pair design	**Required for Designs 2 and 4 for matched pair designs** Pairs (groups) are included only if data are available for all pair (group) members The SP model is used for estimation
CLUSTER_ID	Name of variable containing the cluster identification codes	Numeric or character	**Required for Designs 3 and 4 for all observations** Clusters are included if they have at least one observation with outcome data
TYPE_CLUS_DATA	Indicator for clustered designs as to whether the input file contains individual- or cluster-level data	0 = Cluster-level averages 1 = Individual-level data	**Required for Designs 3 and 4**
CLUSTER_FULL	If TYPE_CLUS_DATA = 0, the name of a binary variable in the input data file indicating whether the cluster-level average pertains to the full sample or a subgroup	0 = Record pertains to a subgroup cluster average 1 = Record pertains to the full sample cluster average	**Required for Designs 3 and 4 if TYPE_CLUS_DATA = 0**
CLUSTER_WGT	Indicator for clustered designs as to whether, by default, clusters or individuals should be weighted equally for the analysis	0 = Clusters are weighted equally for the analysis (default) 1 = Individuals are weighted equally	**Required for Designs 3 and 4** If users input weights for the analysis, these weights will override the default weights
Optional Design and Analysis Parameters			
SUPER_POP	Indicator of preference for the super-population (SP) model	0 = Finite-population (FP) model 1 = SP model	Optional Default is the FP model
CATE_UATE	Indicator for SP designs that the PATE, CATE, or UATE average treatment effect (ATE) parameters should be estimated (see text)	0 = Population average treatment effect (PATE) 1 = Cluster ATE (CATE) 2 = Unit ATE (UATE)	Optional for Designs 2 to 4 if SUPER_POP = 1 Default is the PATE parameter
BLOCK_FE	Indicator for blocked FP and some SP designs that the model should contain main block effects but not block-by-treatment interactions	0 = Model includes interactions and main block effects 1 = Model includes main block effects only	Optional for Designs 2 and 4 Applies to the FP model and the CATE parameter for the SP model Default is the model with interactions

Input variable	Variable definition	Variable format	Additional information
LABEL_RG1, LABEL_RG2, … LABEL_RG6	Labels for the research groups for the output tables (maximum of 6)	Character of length 14 or less	Optional; no quotes needed Defaults are Research 1, Research 2, … , Research 6 "Group" should be omitted from the labels because the program will add it to the end of the labels LABEL_RG1 should refer to the control group if one exists or the first treatment group otherwise
MISSING_COV	Maximum percentage of missing data for a baseline covariate to be included in the regression models. This condition is applied to all research groups.	Numeric: 0 to 75	Optional Default is 30
OBS_COV	Required ratio of the number of observations per covariate for the regression analysis and joint test of baseline equivalence to be performed. The variable pertains to the number of clusters for clustered designs and to the number of blocks for PATE and UATE blocked designs.	Numeric > 1	Optional Default is 5
MIN_NUM	Minimum group size adopted by the state or other entity for reporting outcomes to protect personally identifiable information (PII)	Integer ≥ 3	Optional Default is 10
ALPHA_LEVEL	Significance level for testing the null hypothesis of zero average treatment effects (in percentages)	Integer: 1 to 30	Optional Default is 5
NO_COV_SG	Excludes covariance terms in the statistical tests of differences in impact estimates across subgroup categories (for example, for males and females)	0 = Covariance terms included in the statistical tests 1 = Covariance terms excluded from the statistical tests	Optional for Designs 3 and 4 and the Design 2 PATE and UATE models Default is the inclusion of the covariance terms
LIMIT_PRINT	Suppresses printing of detailed descriptive sample statistics in the output tables	0 = All output tables printed 1 = Printing limited to tables with main impact results only (Tables 1 and 8 to 10)	Optional Default is printing of all tables
NUM_DEC	Number of decimals for the output tables presenting impact findings for continuous variables	Integer: 0 to 3	Optional Default is 2

Input variable	Variable definition	Variable format	Additional information
MULT_COMP	Specifies whether the Benjamini-Hochberg or Bonferroni method should be used for multiple comparisons adjustments	0 = Benjamini-Hochberg 1 = Bonferroni	Optional Default is the Benjamini-Hochberg method For the Bonferroni method, associated confidence intervals can be plotted in *RCT-YES-Graph*
Outcomes, Weights, Covariates, and Subgroups			
OUTCOME_DMN	Title of outcome domain pertaining to a specific class of outcomes for which common analyses are to be conducted	Character	Optional Outcomes with common analyses are grouped to minimize data entry and facilitate reporting and hypothesis testing
OUTCOME	Name of outcome variable	Numeric; all missing data codes are valid based on the language used (Stata or R)	**Required** Cases with missing values for an outcome are excluded from the analysis for that outcome
LABEL	Label for outcome variable	Character Blank	Optional
WEIGHT	Name of the observation-level weight that provides information on how to weight blocks and/or clusters to obtain pooled estimates and to adjust for missing data (nonresponse) or unequal sampling probabilities for other design-related reasons	Numeric Blank	Optional Default is equal weighting of all individuals for non-clustered designs and individuals or clusters for clustered designs (see CLUSTER_WGT) A different weight can be specified for each outcome and subgroup Weights must be positive and nonmissing for cases with outcome data or they are ignored
STD_OUTCOME	Individual-level standard deviation of the outcome variable	Numeric > 0 Blank	**Required for Designs 3 and 4 if TYPE_CLUS_DATA = 0** in order for the program to calculate impacts in effect size units Optional for other designs, where the default is the full sample, pooled standard deviation across all research groups

Input variable	Variable definition	Variable format	Additional information
COVARIATES	List of names of baseline covariates to obtain regression-adjusted impact estimates for full sample or subgroup analyses	Numeric: continuous or binary; all missing data codes are valid based on the language used (Stata or R)	Optional Covariates are excluded if they contain too many missing values (see MISSING_COV above) or if there are too few observations per covariate (see OBS_COV above) A different set of covariates can be specified for each outcome domain and each subgroup
GOT_TREAT	Name of variable indicating the receipt of intervention services for the research groups. The variable should be *binary* for all designs except if TYPE_CLUS_DATA = 1, in which case the variable should be a *numeric service receipt rate* between 0 and 1.	If DESIGN= 1 or 2 or DESIGN = 3 or 4 and TYPE_CLUS_DATA=0: 0 = Treatment not received 1 = Treatment received If TYPE_CLUS_DATA=1: Numeric: ≥ 0 and ≤ 1	Optional for estimating complier average causal effects (CACE) pertaining to those who would receive intervention services as a treatment but not as a control The analysis is conducted only for comparing a treatment group to a control group, but not for comparing treatment groups to each other (see Schochet, 2017) Up to 2 variables are allowed per outcome domain that could pertain to different dimensions of service receipt or dosage. A separate analysis is conducted for each GOT_TREAT and outcome variable combination. Cases with missing GOT_TREAT values are excluded from both the CACE and ATE analyses
SUBGROUP	Name of subgroup variable	Categorical; all missing data codes are valid based on the language used (Stata or R)	Optional Baseline subgroups can pertain to an individual (e.g., student or teacher), a cluster (e.g., school) or other unit and must be large enough to protect data disclosure
Baseline Equivalence Analysis			
BASE_EQUIV	List of names of baseline covariates that are to be used to assess baseline equivalence of the research groups	Numeric: continuous or binary; all missing data codes are valid based on the language used (Stata or R)	Optional Separate analyses are conducted for each outcome and pairwise contrast of the research groups
NO_JNT_TEST	Suppresses the joint test of baseline equivalence	0 = Conduct the joint test 1 = Do not conduct the joint test	Optional Default is to conduct the joint test This option might be desirable if a very large number of baseline variables are specified that could lead to program errors due to matrix size limits in R or Stata

Input variable	Variable definition	Variable format	Additional information
Generate Variable List Window			
BASE_NAME_VL	Base name for the files below. The interface will add a "_VL" suffix to the base name to distinguish these files from other output files.	Character	**Required to produce the files below**
COMP_PROG_VL	Location of the R or Stata program produced by the interface that must be run in a separate step outside the interface to generate the variable list text file	The interface produces a .R file for R or a .do file for Stata with the base name (BASE_NAME_VL) specified above	**Required to produce the file**
FILE_VL	Location of the variable list text file produced by the R or Stata computer program that can then be imported into the interface	The R or Stata computer program produces a .varlist text file with the base name (BASE_NAME_VL) from above	**Required to produce the COMP_PROG_VL file**
IMPORT_VL	Name and location of the variable list text file to import into the interface	The interface will use the .varlist text file to create the variable list window	**Required to produce the variable list window**
Generate Output Files for the Analysis			
BASE_NAME	Common base name for the three files below (that each have different file extensions)	Character	**Required to produce the files below**
INPUT_SPEC_FILE	Location of the interface file containing program inputs that can be opened and edited for future use	The interface produces a file with a .rctyes extension and the base name (BASE_NAME) specified above	**Required to produce the file**
COMPUTER_PROG	Location of the R or Stata program produced by the interface to be run in a separate step to conduct the analysis	The interface produces a .R file for R or a .do file for Stata with the base name (BASE_NAME) specified above	**Required to produce the file**
RESULTS_FILE	Location of the analysis results file produced by the R or Stata computer program that contains formatted output tables	The R or Stata program produces an .html file with the base name (BASE_NAME) specified above and a .log file with estimation results	**Required to produce the COMPUTER_PROG file**

Updated Record Layout of the .CSV File

RCT-YES produces a .csv file of analysis results that users can read into their own computer programs for additional analyses and reporting. Table 2 shows the updated record layout of the .csv file. The format of the .csv file is similar to the format discussed in the *RCT-YES* User's Manual ***except that it now contains stacked information for each pairwise contrast across the research groups***. The specific pairwise contrast can be identified using the columns labeled "group1" and "group2" that specify the data codes for the two research groups being compared. The research group with the smaller code is listed in the group1 column and the research group with the larger code is listed in the group2 column. Several fields have been added corresponding to the new input variables in Version 1.2 (discussed in the next section). The rows listed in Table 2 are repeated for each pairwise contrast across the research groups.

Table 2. Updated record layout of the .csv file, repeated for each pairwise contrast

Order	Variable Name	Variable Type	Output Table	Description
1	table_id	String	All	Table number (corresponding to the .html output table)
2	group1	Numeric	All	Research group code with the smaller value for the pairwise contrast (hereafter labeled the **"control group"**)
3	group2	Numeric	All	Research group code with the larger value for the pairwise contrast (hereafter labeled the **"treatment group"**)
4	domain	Numeric	All	Outcome domain number (for sorting)
5	domain_name	String	All	Outcome domain name
6	outcome	Numeric	All	Outcome variable number
7	outcome_name	String	All	Outcome variable name
8	outcome_label	String	All	Outcome variable label
9	outcome_std	Numeric	All	Outcome-specific user-specified standard deviation
10	got_treat	Numeric	All	Service receipt indicator variable number
11	got_treat_name	String	All	Service receipt indicator variable name
12	subgroup	Numeric	All	Subgroup number
13	subgroup_name	String	All	Subgroup name
14	sglevel	Numeric	5, 9, 9a, 9b	Subgroup category number
15	sglevel_value	String	5, 9, 9a, 9b	Subgroup category value
16	sglevel_label	String	5, 9, 9a, 9b,	Subgroup category label
17	binary	Numeric	2, 3, 6, 8, 9, 9a, 9b, 10	Variable is binary (1=Yes; 0=No)
18	tc	Numeric	2, 3	1/0 indicator for the pairwise comparison (1=treatment group, 0=control group)

Order	Variable Name	Variable Type	Output Table	Description
19	variable_type	Numeric	2, 3	Variable type (1=OUTCOME, 2=GOT_TREAT, or 3=WEIGHT variable)
20	variable_type_ name	String	2, 3	Variable type name (an OUTCOME, GOT_TREAT, or WEIGHT variable)
21	variable	String	2, 3	Variable name
22	level	Numeric	2, 3, 5	Unit of observation (1=individuals, 2=clusters)
23	level_name	String	2, 3, 5	Unit of observation (individuals or clusters)
24	block	Numeric	4	Block number
25	block_name	String	4	Block name
26	clust	Numeric	4	Cluster number
27	clust_name	String	4	Cluster name
28	bad_block	Numeric	4	Block is invalid (1=Yes; 0=No)
29	bad_clust	Numeric	4	Cluster is invalid (1=Yes; 0=No)
30	covar	Numeric	6	Covariate number
31	covar_name	String	6	Covariate name
32	bequiv	Numeric	8	Baseline equivalency number
33	bequiv_name	String	8	Baseline equivalency name
34	bequiv_valid	Numeric	8	Baseline equivalency variable is valid (1=Yes; 0=No)
35	weight_used	String	8, 9, 9a, 9b, 10	Weight variable used for the analysis (blank if no weight used)
36	covars_used	String	9, 9a, 9b, 10	Covariates used for analyses (blank if no covariates used)
37	any_excl	Numeric	2, 5	Any covariate excluded (1=Yes; 0=No)
38	missing_cov	String	6	Exclusion reason: too many missing values ("X" or missing)
39	zero_sd	String	6	Exclusion reason: not enough variation ("X" or missing)
40	too_few	String	6	Exclusion reason: too few cases/blocks/clusters per covariate ("X" or missing)
41	corr_abs1	String	6	Exclusion reason: the correlation between covariate and outcome is 1.0 or -1.0 ("X" or missing)
42	n_sample	Numeric	2	Number in sample
43	n_avail	Numeric	2	Number with available data
44	n_miss	Numeric	2	Number with missing data
45	pct_avail	Numeric	2	Percentage with available data (0 to 100)
46	mean	Numeric	2	Mean
47	sd	Numeric	2	Standard deviation
48	p5	Numeric	3	5th percentile
49	p25	Numeric	3	25th percentile

Order	Variable Name	Variable Type	Output Table	Description
50	p50	Numeric	3	50th percentile
51	p75	Numeric	3	75th percentile
52	p95	Numeric	3	95th percentile
53	n_avail_t	Numeric	4, 5	Number with available data for treatments
54	n_miss_t	Numeric	4, 5	Number with missing data for treatments
55	n_avail_c	Numeric	4, 5	Number with available data for controls
56	n_miss_c	Numeric	4, 5	Number with missing data for controls
57	swb	Numeric	4	Block or cluster weight
58	r2_t	Numeric	6	Squared partial correlation with other covariates for treatment
59	rho_t	Numeric	6	Correlation with outcome for treatments
60	r2_c	Numeric	6	Squared partial correlation with other covariates for controls
61	rho_c	Numeric	6	Correlation with outcome for controls
62	table_nt	Numeric	8, 9, 9a, 9b	Sample size for treatments (individuals or clusters depending on design)
63	table_nc	Numeric	8, 9, 9a, 9b	Sample size for controls (individuals or clusters depending on design)
64	table_n	Numeric	8, 9, 9a, 9b	Sample size overall (individuals or clusters depending on design)
65	table_indivnt	Numeric	8, 9, 9a, 9b	Number of individuals for treatments (for DESIGN 3 and 4, CLUSTER_DATA=1 only)
66	table_indivnc	Numeric	8, 9, 9a, 9b	Number of individuals for controls (for DESIGN 3 and 4, CLUSTER_DATA=1 only)
67	table_indivn	Numeric	8, 9, 9a, 9b	Number of individuals overall (for DESIGN 3 and 4, CLUSTER_DATA=1 only)
68	ybart	Numeric	8, 9, 9a, 9b	Treatment group mean
69	ybarc	Numeric	8, 9, 9a, 9b	Control group mean
70	impact	Numeric	8, 9, 9a, 9b	Difference (Impact Estimate)
71	effect_size	Numeric	8, 9, 9a, 9b	Effect Size
72	se_impact	Numeric	8, 9, 9a, 9b	Standard error of difference
73	p_impact	Numeric	8, 9, 9a, 9b	p-Value of difference
74	s_impact	String	8, 9, 9a, 9b	Significance marker ("*" if significant at the ALPHA_LEVEL; blank otherwise)

Order	Variable Name	Variable Type	Output Table	Description
75	conf_lower	Numeric	8, 9, 9a, 9b	Lower confidence limit for impact
76	conf_upper	Numeric	8, 9, 9a, 9b	Upper confidence limit for impact
77	conf_lower_adj_all	Numeric	8, 9, 9a, 9b	Lower confidence limit for impact: multiple comparison adjustments for both research groups and domain outcomes using Bonferroni adjustment
78	conf_upper_adj_all	Numeric	8, 9, 9a, 9b	Upper confidence limit for impact: multiple comparison adjustments for both research groups and domain outcomes using Bonferroni adjustment
79	conf_lower_adj_pair	Numeric	8, 9, 9a, 9b	Lower confidence limit for impact: multiple comparison adjustments for domain outcomes only using Bonferroni adjustment
80	conf_upper_adj_pair	Numeric	8, 9, 9a, 9b	Upper confidence limit for impact: multiple comparison adjustments for domain outcomes only using Bonferroni adjustment
81	conf_lower_eff	Numeric	8, 9, 9a, 9b	Lower confidence limit for effect size
82	conf_upper_eff	Numeric	8, 9, 9a, 9b	Upper confidence limit for effect size
83	conf_lower_adj_eff_all	Numeric	8, 9, 9a, 9b	Lower confidence limit for effect size: multiple comparison adjustments for both research groups and domain outcomes using Bonferroni adjustment
84	conf_upper_adj_eff_all	Numeric	8, 9, 9a, 9b	Upper confidence limit for effect size: multiple comparison adjustments for both research groups and domain outcomes using Bonferroni adjustment
85	conf_lower_adj_eff_pair	Numeric	8, 9, 9a, 9b	Lower confidence limit for effect size: multiple comparison adjustments for domain outcomes only using Bonferroni adjustment
86	conf_upper_adj_eff_pair	Numeric	8, 9, 9a, 9b	Upper confidence limit for effect size: multiple comparison adjustments for domain outcomes only using Bonferroni adjustment
87	adj_sig_pair	String	9, 9b	Significance marker after applying the Benjamini-Hochberg or Bonferroni correction for domain outcomes ("^" if significant at the alpha_level; blank if not)
88	adj_sig_all	String	9, 9b	Significance marker after applying the Benjamini-Hochberg or Bonferroni correction for research groups and domain outcomes ("+" if significant at the alpha_level; blank if not)
89	joint_pval	Numeric	8	p-Value for the joint significant test for the baseline equivalence analysis
90	pvalf	Numeric	9, 9a, 9b	p-Values to test for differences in impacts across subgroups
91	sf	String	9, 9a, 9b	p-Values to test for differences in impacts across subgroups, significance marker ("*")
92	r2	Numeric	9	R-squared value (for full sample only for models with covariates)

Order	Variable Name	Variable Type	Output Table	Description
93	icc	Numeric	9	Intraclass correlation coefficient (for DESIGN 3 and 4, CLUSTER_DATA=1 only)
94	n_blocks	Numeric	10	Number of blocks
95	sd_impact	Numeric	10	Standard deviation of impact
96	pct_positive	Numeric	10	Proportion positive (0 to 100)
97	range	Numeric	10	Difference between the largest and smallest block-specific impact estimate
98	block_pvalf	Numeric	10	p-Value from joint test of differences across blocks
99	block_sf	String	10	p-Value from joint test of differences across blocks, significance marker ("*")
100	Input	String	Appendix	.rctyes input specification file field name
101	specification	String	Appendix	.rctyes input specification file field value

Note: For simplicity in the descriptions above, the ”control group” refers to the research group code with the smaller value for a given pairwise contrast, and the “treatment group” refers to the research group code with the larger value. In practice, analyses with multiple treatment groups will include some pairwise contrasts where the “control group” is actually a treatment group that is being compared to another treatment group.

(This page left intentionally blank for double-sided copying)

Version 1.2 Updates: January 2018

RCT-YES Version 1.2 was released in January 2018 and updates Version 1.1 which was released in June 2016.

Key Updates

- ***RCT-YES*** **can now accommodate designs where individuals or groups are randomized to more than two research groups (multi-armed designs).**
- **For clustered designs, the regression models are now estimated using individual-level data (if provided) rather than data averaged to the cluster level. This means that baseline covariates can now explain within-cluster variation in the outcomes to improve precision.**
- **For clustered designs, an option has been added that allows users to specify whether clusters or individuals should be weighted equally for the analysis.**
- **To correct for the multiple comparisons problem, users can now specify the Bonferroni method as an alternative to the Benjamini-Hochberg method.**
- **F-tests rather than chi-squared tests are now used to test for differences in impacts across subgroups and blocks, because they perform better for designs with small sample sizes.**

Below, we describe Version 1.2 updates to the program inputs and the underlying statistical theory, and then describe updates to the output tables and *RCT-YES-Graph*.

A. Updates to Program Inputs

1. The TC_STATUS variable can now contain codes for multiple research groups

Previously, the required treatment status variable could take on two values: 0 for those randomly assigned to the control group and 1 for those randomly assigned to the treatment group. The program can now accommodate codes for up to 6 research groups. The codes must be consecutive integers starting at 0 for designs with a control group or starting at 1 for designs with only treatment groups. Specifically, the treatment status variable must now be coded in one of two ways, where we use the symbol R to represent the total number of research groups in the study:

i. **Consecutive integer codes from 0 to (R-1) for studies with a control group, where 0 is the code for the control group, or**

ii. **Consecutive integer codes from 1 to R for studies with only treatment groups**

For example, for a design with one control group and two treatment groups, the codes would be 0 for the control group and 1 and 2 for the two treatment groups. If the study instead contained 3 treatment groups but no control group, the codes would be 1, 2, and 3. Codes must be available for all observations, and each research group must be sufficiently large or the program will issue an error message and abort (see Section C below).

2. The LABEL_RG1, LABEL_RG2, …, LABEL_RG6 inputs replace the LABEL_T and LABEL_C inputs for labeling the research groups

Users can input optional labels in the Optional Design and Analysis screen that are used to identify the research groups in the output tables. The default labels are "Research 1", "Research 2", …, "Research 6", where the program automatically adds "Group" to the end of the labels. The number of entered labels should align with the number of research groups in the study. LABEL_RG1 should refer to the control group if one exists or the first treatment group otherwise.

3. The MULT_COMP option has been added to select between the Benjamini-Hochberg or Bonferroni methods to adjust for multiple comparisons

MULT_COMP has been added to the Optional Design and Analysis screen to allow users to specify the Bonferroni method rather than the Benjamini-Hochberg (BH) method to adjust for the multiple comparisons problem when conducting full sample hypothesis tests across pairwise contrasts and outcomes in the same domain. The default value is MULT_COMP = 0 for the BH procedure, but MULT_COMP can be set to 1 for the Bonferroni method, where associated confidence intervals can be plotted using *RCT-YES-Graph*.

The Bonferroni procedure was added as an option in response to user comments because it controls the familywise error rate (FWER) rather than the false discovery rate (FDR) as for the BH procedure. The FWER is the probability that at least one null hypothesis will be rejected when all null hypotheses are true. The false discovery rate (FDR), however, is the expected fraction of significant test statistics that are false discoveries. The BH approach—which is similar to the Bonferroni approach in that it is based only on adjusting p-values for each test in isolation—can lead to power gains if there are many contrasts that truly differ. However, the FDR uses a preponderance-of-evidence standard that allows for extra false positives if many contrasts are found to be statistically significant. Thus, the FDR evidence standard is less stringent than the FWER standard, and may not always be appropriate for RCTs that require a high bar of evidence to identify treatment effects.

The Bonferroni method was selected for the program because it aligns well with the design-based framework in that it does not require assumptions on the distributions of the potential outcomes or the model structure. A disadvantage of this approach, however, is that by ignoring the correlational structure across tests, the method yields conservative bounds on Type I errors and, hence, sacrifices statistical power. Nonetheless, Schochet (2009) shows that relative to other more powerful approaches (such as resampling methods), precision losses under the Bonferroni method are modest except if the test statistics are highly correlated with each other and there are many test statistics.

If the Bonferroni method is selected, the associated confidence intervals are written to the output .csv file and can be plotted in *RCT-YES-Graph*. Confidence intervals are not computed for the BH adjustment method.

4. For clustered designs, the CLUSTER_WGT option has been added to select the default weights for aggregating clusters for the analysis

An important issue for clustered designs is how to weight (aggregate) clusters to estimate average treatment effects. By default, *RCT-YES* weights clusters equally (CLUSTER_WGT=0), but users can now set CLUSTER_WGT=1 to weight individuals equally in the Optional Design and Analysis screen. The choice of weighting scheme will depend on whether interest lies in estimating impacts for the average cluster or individual, as well as practical concerns about whether a few very large clusters can drive the impact findings. Other weighting schemes can be implemented by inputting weight variables into the program.

5. The NUM_DEC option has been added to select the number of digits following the decimal point to report for the impact findings

This option has been added to the Optional Design and Analysis screen and applies to continuous variables in Tables 8 and 9 of the output tables. The default value is 2, but can range from 0 to 3. This option applies to the research group means, impact estimates, and standard errors but not to other statistics (for example, *p*-values are always reported using three decimal places).

6. The CSV_FILE option was removed

In previous software versions, this option allowed users to suppress the R or Stata computer program from producing a .csv file of analysis results. It was removed because, based on program updates, there is now no reason to suppress the creation of the .csv file which can be used to plot the impact findings using *RCT-YES-Graph* and to conduct additional analyses and reporting.

B. Updates to Running the Program

1. Stata users can now run multiple Stata .do programs generated by *RCT-YES* in the same Stata window

Previously, Stata users needed to close and re-open the Stata window for each new run of an *RCT-YES* analysis program. Otherwise, an error message would be issued saying that the .log file is open. This issue has been fixed.

C. Updates to Statistical Theory

1. The software can now handle designs with multiple treatment groups

Impact evaluations with multiple research groups can simultaneously examine the effects of multiple interventions in a single study, thereby increasing the amount that researchers and policymakers can learn from evaluations. In social policy research, these designs are particularly relevant for interventions that are relatively easy to implement—for example, an RCT or QED testing several texting initiatives to improve student engagement and achievement. Relatedly, multi-armed designs are useful for rapid-cycle or opportunistic experiments aimed at continuous program improvement, for example, using behavioral-based interventions and encouragement designs.

Based on user feedback, *RCT-YES* can now estimate average treatment effects for designs with multiple research groups, where individuals or clusters (groups) are randomized to either a control group (if one exists) or to one of several treatment groups. The program estimates intervention effects for these designs by comparing pairs of research groups to each other. For example, if there are four research groups, *RCT-YES* will sequentially estimate impacts for the 6 possible pairwise contrasts and report impact findings for each one.

As discussed in Schochet (2017b), the design-based theory for the simple treatment-control design presented in the *RCT-YES* Statistical Theory Appendix (Schochet, 2016) largely applies to multi-armed designs where pairs of research groups are compared to each other. Thus, for each pairwise contrast, *RCT-YES* creates an indicator variable signifying the two research groups being compared, and then applies very similar methods as for the two-group design. However, some program modifications are required for multi-armed designs to account for both statistical and analysis issues.

The key modifications to *RCT-YES* to accommodate multiple research groups are as follows:

- **The same data checks are applied to each research group so that consistent analyses are conducted across the pairwise contrasts.** The program uses the same rules as for the simple two-group design to determine which outcome variables, covariates, blocks, weights, and baseline equivalence analyses should be included in all pairwise analyses (see the 2016 User's Manual and Statistical Theory Appendix). For example, if a block has insufficient sample sizes for any research group, that block will be excluded from all analyses. This process will ensure a consistent set of impact findings across the contrasts.

- **For models with covariates, separate regression models are run for each pairwise contrast using a common set of covariates.** Estimating a single pooled regression model for all research groups together complicates the design-based theory without adding statistical rigor (Schochet, 2017b).

- **Simple modifications were made to the variance formulas and weights.** For the reasons discussed in Schochet (2017b), for finite-population models, the variance formulas for each pairwise contrast now reflect the generalization of the impact estimates to all randomized research groups, not just to the two groups being compared. For similar reasons, for blocked designs, the weights for each pairwise contrast are now scaled to represent the full randomized sample for each block.

- **The program applies multiple comparisons corrections across pairwise comparisons for full sample analyses.** To account for the inflation of Type 1 errors due to repeated hypothesis testing across the pairwise contrasts, the program uses the Benjamini-Hochberg or Bonferroni method to adjust the *p*-values from the individual *t*-tests. These corrections are made for full sample analyses but not for subgroup analyses. Multiple comparisons corrections are not applied for baseline equivalence analyses.

- **Impacts in effect size units are calculated using standard deviations of the outcome variables for the control group if one exists, or across all treatment groups otherwise.** For all designs, the same standard deviation is used across all pairwise contrasts so that effect sizes can be consistently compared across contrasts.

- **The complier average treatment effect (CACE) parameter is estimated only for contrasts comparing a treatment and control group.** As discussed in Schochet (2017b), CACE analyses become very complex in multi-armed trials due to the increase in the number of compliance parameters that need to be estimated and the complex assumptions required to identify them. In some settings, CACE analyses can be justified for pairwise contrasts of a treatment group to the control group (if one exists). Accordingly, *RCT-YES* estimates CACE effects for pairwise contrasts that include a control group, but not for analyses contrasting two treatment groups. Users must interpret these CACE results carefully, because the analyses are based on strong assumptions that may not always hold (see Schochet, 2017b). In particular, the approach assumes *no crossovers* (that is, that those in a particular research group do not receive treatment services slated for other research groups), and that complier populations are the same across the treatment groups (which could be violated, for example, if compliance rates differ markedly across the treatment groups).

  For CACE analyses, the variable in the data file indicating the receipt of intervention services (the GOT_TREAT input variable) should be coded as 1 for treatment group members who received intervention services offered to their research group, and 0 for everyone else, including all control group members (recall that CACE analyses for multi-armed trials should only be conducted if crossover rates are small). For example, if there are two treatment groups (T1 and T2) and a control group (C), the GOT_TREAT variable should be set to 1 for those in T1 who received T1 services and those in T2 who received T2 services, and 0 otherwise. The software allows for two GOT_TREAT variables capturing different dimensions of service receipt. If the input data are cluster-level averages (for clustered designs), as in earlier versions, the GOT_TREAT variables should be numeric service receipt rates between 0 and 1 (see the *RCT-YES* User's Manual for details).

2. For clustered designs (Designs 3 and 4), the models are now estimated using individual-level data (if provided) rather than data averaged to the cluster level

For clustered designs, *RCT-YES* users have the option of inputting individual-level data or aggregate data (cluster-level averages) for the analysis. In previous software versions, if users provided individual-level data, *RCT-YES* estimated impacts by first averaging the data to the cluster level. Therefore, the estimation models could only include covariates aggregated to the cluster level to explain variation in mean outcomes between clusters, but not covariates at the individual level that could also explain variation in outcomes within clusters. Based on user feedback, the software now estimates the regression models using individual-level data, so that the models can now accommodate within-cluster covariates to help improve precision of the estimated impacts. These updates do not involve changes to the program inputs. The updates apply only if the input data file contains individual-level data (TYPE_CLUS_DATA=1), but not if the input data file contains clustered-level data (TYPE_CLUS_DATA=0).

The design-based theory underlying this approach uses results in Schochet (2013), where impacts are estimated using weighted least squares on the individual-level data, and standard errors are estimated using the model residuals. To demonstrate the approach, consider the finite-population model for the clustered, non-blocked design (Design 3), where the pairwise contrast involves comparing two research groups labeled as the "treatment" and "control" groups. The data generating process for the observed outcome variable, y_{ij}, for individual i in cluster j is

$$(1) \quad y_{ij} = T_j Y_{ij}(1) + (1 - T_j) Y_{ij}(0),$$

where T_j equals 1 for clusters randomized to the treatment group and 0 for control group clusters; $Y_{ij}(1)$ is the individual's potential outcome in the treatment condition; and $Y_{ij}(0)$ is the individual's potential outcome in the control condition. Note that Equation (1) differs from the approach used in Schochet (2016) where the data generating process is specified at the cluster level.

Equation (1) can be used to generate a regression model where the error term has both between- and within-cluster components (Schochet, 2013). Estimating this regression model using weighted least squares (with or without baseline covariates) yields a consistent estimator of the average treatment effect that is asymptotically normal. A consistent (upper bound) variance estimator is similar to Equation (7.22) in Schochet (2016) except that the mean square error terms, MSE_{TW} and MSE_{CW}, are now calculated using estimated model residuals for *individuals* rather than for clusters:

$$(7.22a)\quad As\hat{y}Var_R(\hat{\beta}_{clus,MR,FP,W}) = \frac{MSE_{TW}}{\overline{w}_T^2 mp} + \frac{MSE_{CW}}{\overline{w}_C^2 m(1-p)} - \frac{1}{m}(\frac{\sqrt{MSE_{TW}}}{\overline{w}_T} - \frac{\sqrt{MSE_{CW}}}{\overline{w}_C})^2 \text{, where}$$

$$MSE_{TW} = \frac{1}{(m-v)p-1}\sum_{j:T_j=1}^{m_T}\sum_{i=1}^{n_j}\sum_{i'=1}^{n_j} w_{ij}w_{i'j}\hat{e}_{ij}\hat{e}_{i'j} \text{ and}$$

$$MSE_{CW} = \frac{1}{(m-v)(1-p)-1}\sum_{j:T_j=0}^{m_C}\sum_{i=1}^{n_j}\sum_{i'=1}^{n_j} w_{ij}w_{i'j}\hat{e}_{ij}\hat{e}_{i'j} .$$

In this expression, $\hat{e}_{ij}$ and $\hat{e}_{i'j}$ are estimated model residuals for individuals i and i' in cluster j; m_T and m_C are the number of treatment and control clusters, respectively; $m = m_T + m_C$ is the total number of clusters; v is the number of covariates; $p = (m_T / m)$ is the proportion of clusters assigned to the treatment group; n_j is the number of individuals in cluster j; w_{ij} and $w_{i'j}$ are individual-level weights; $\overline{w}_T = (\sum_{j:T_j=1}^{m_T}\sum_{i=1}^{n_j} w_{ij} / m_T)$ is the average cluster-level weight for the treatment group; and similarly for $\overline{w}_C$.

In (7.22a), the MSE terms are obtained by summing cross-products of the residuals for all individuals within the same cluster, and then summing these values across clusters. This approach shares features with the generalized estimating equations (GEE) approach of Liang and Zeger (1986) based on the sandwich variance estimator. Perhaps more intuitively, the MSE terms in (7.22a) can be calculated by averaging the individual-level residuals to the cluster level and then using the same aggregate-level formulas as in Schochet (2016) for clustered designs. Even though the analysis is now conducted at the individual level, by default, *RCT-YES* continues to weight clusters equally in the analysis, although the CLUSTER_WGT option can be used to weight individuals equally.

Because the weighted least square estimator is asymptotically normal, the hypothesis testing strategy is the same using the individual- or aggregate-level data for estimation (where the degrees of freedom are based on the number of clusters in the sample). A parallel approach is used for blocked designs, subgroup analyses, and complier average causal effects (CACE) analyses by updating the corresponding estimators in Schochet (2016).

Finally, in Table 6 of the .html output files (that presents information on the model covariates), the correlations among the covariates and between the covariates and outcome variables are now calculated using the individual-level data rather than the cluster-level data as was done before.

3. F-tests rather than chi-squared tests are now used to test for differences in treatment effects across subgroups

The F-statistics for the subgroup interaction tests are now calculated by dividing the chi-squared statistics discussed in Schochet (2016) by $df = s - 1,$ where s is the number of categories of the subgroup under investigation (for example, $s = 2$ for testing differences in impacts for males and females). These F-statistics have an approximate $F(df, ddf)$ distribution, where ddf is the denominator degrees of freedom that depends on the sample size (individuals or clusters depending on the design) and the number of covariates and blocks. More specifically, ddf equals the degrees of freedom that the program uses to conduct *t*-tests for the full sample analysis associated with the subgroup analysis under investigation (see Schochet, 2016).

The program was revised to use F-tests rather chi-squared tests (even though variances are allowed to differ across both subgroups and treatment conditions), because simulation evidence shown in Table 3 below suggests that Type 1 error rates for the F-tests are closer to the nominal (5 percent) significance level for designs with small numbers of clusters. This occurs because the F-tests adjust for degrees of freedom losses for designs with small sample sizes whereas chi-squared tests do not. These results are consistent with the literature on small sample adjustments for generalized estimating equations (GEE) estimators that share features with the design-based approach (see, for example, Bell and McCaffery, 2002; Guo and Pan, 2002; Mancl and DeRouen, 2001; Pan and Wall, 2002; and Pustejovsky and Tipton, 2016).

The simulations in Table 3 were conducted using the following assumptions: (1) a clustered design with the randomization of clusters to a treatment or control group; (2) sample sizes of 8, 12, or 16 clusters split evenly between the two research groups; (3) the number of individuals per cluster ranging randomly between 5 and 30; (4) an individual-level subgroup variable with 3 categories, with subgroup proportions varying randomly across clusters; (5) a 5 percent significance level for a two-tailed test; and (6) two weighting schemes where clusters are weighted by their sample size or equally. Individual-level data were generated for the simulations using the following model:

$$(1) \quad y_{ij} = .1s_{ij} + z_{ij} + u_j + e_{ij},$$

where y_{ij} is the outcome variable for individual i in cluster j; s_{ij} equals 1, 2, or 3 corresponding to the subgroup category; z_{ij} are independent and identically distributed (*iid*) $N(0,1)$ model covariates; u_j are *iid* $N(0,1)$ cluster-specific random effects, and e_{ij} are *iid* $N(0,1)$ individual-level errors. We conducted 5,000 simulations for each specification, and calculated Type 1 error rates using F-tests and chi-squared tests based on the design-based formulas in Schochet (2016) and the revised covariance estimates described in the next subsection. Table 3 also displays Type 1 error rates for the GEE approach using Proc Genmod in SAS, where we report results for the subgroup interaction tests using the default score test as well as the Wald (chi-squared) statistic option.

The simulations show that for the design-based approach, the F-test yields Type 1 error rates that are slightly inflated when individuals are weighted equally for the analysis, whereas the chi-squared test yields highly inflated Type 1 error rates. When clusters are instead weighted equally, the F-test is conservative, whereas the chi-squared test still yields inflated Type 1 errors. The score test in SAS Proc Genmod performs slightly better than the F-test, but the Wald test option in SAS Proc Genmod yields highly inflated Type 1 errors (similar to the results found in the literature cited above). These results suggest that for design-based estimators, the F-test typically performs better than the chi-squared test for designs with small sample sizes.[1]

Table 3. Simulated Type 1 error rates for F-tests and chi-squared tests of subgroup interactions

	Design-based approach		Generalized Estimating Equations (GEE) approach	
Number of treatment / control group clusters	F-test	Chi-squared test	Score test	Wald test
Equal weighting of individuals				
4 / 4	0.049	0.157	0.025	0.342
6 / 6	0.063	0.137	0.051	0.238
8 / 8	0.065	0.113	0.049	0.174
Equal weighting of clusters				
4 / 4	0.012	0.070	0.029	0.255
6 / 6	0.024	0.068	0.039	0.165
8 / 8	0.034	0.063	0.043	0.128

4. Covariances for the F-tests to test for differences in treatment effects across subgroups use a revised weighting scheme for clustered and random blocked designs

For clustered, non-blocked designs (Design 3), the mean outcomes of subgroups of individuals (for example, girls and boys) within the same cluster could be correlated due to common teachers, school

[1] Future versions of *RCT-YES* might incorporate bias-reducing corrections found in the literature (such as those in Bell and McCaffery, 2002). However, more research needs to be conducted to apply these methods to the design-based context rather than the GEE one.

staff, and school environments. These covariances should be incorporated into the F-tests to assess differences in intervention effects across subgroups.

RCT-YES now estimates these covariances by updating Equation (7.16a) in Schochet (2016). For models without covariates, the program now estimates the covariances between the outcomes of individuals in subgroups g and g' as follows:

$$(7.16a1) \quad C\hat{o}v_{IRS}(\hat{\beta}_{clus,g,FP},\hat{\beta}_{clus,g',FP}) = \frac{\Delta_{TW}(g,g')}{m_T\bar{w}^*_{Tg}\bar{w}^*_{Tg'}} + \frac{\Delta_{CW}(g,g')}{m_C\bar{w}^*_{Cg}\bar{w}^*_{Cg'}}, \text{ where}$$

$$\Delta_{TW}(g,g') = \frac{1}{m_T-1}\sum_{j:T_j=1}^{m_T} w^*_{jg}w^*_{jg'}(\bar{y}_{jg}-\bar{\bar{y}}_{TgW})(\bar{y}_{jg'}-\bar{\bar{y}}_{Tg'W}) \text{ and}$$

$$\Delta_{CW}(g,g') = \frac{1}{m_C-1}\sum_{j:T_j=0}^{m_C} w^*_{jg}w^*_{jg'}(\bar{y}_{jg}-\bar{\bar{y}}_{CgW})(\bar{y}_{jg'}-\bar{\bar{y}}_{Cg'W}).$$

In this expression, w^*_{jg} is the cluster-level weight (sum of individual-level weights) for cluster j and subgroup g ; $\bar{w}^*_{Tg} = \sum_{j:T_j=1}^{m_T} w^*_{Tg} / m_T$ and $\bar{w}^*_{Cg} = \sum_{j:T_j=0}^{m_C} w^*_{Cg} / m_C$ are associated mean values for the treatment and control groups (or two contrasted treatment groups); m_T and m_C are the number of treatment and control clusters; T_j is the treatment-control indicator variable; $\bar{y}_{jg}$ is the mean outcome in the cluster, and $\bar{\bar{y}}_{TgW}$ and $\bar{\bar{y}}_{CgW}$ are mean outcomes across all clusters. If a cluster does not contain subgroup g, then w^*_{jg} is set to 0. A similar weighting scheme is used for models with covariates based on regression residuals.

There are several key differences between Equation (7.16a1) and the old approach using Equation (7.16a) in Schochet (2016). First, the old approach used the full-sample cluster-level weight, $w^*_j = \sum_g w^*_{jg}$, for *all* covariance calculations rather than w^*_{jg} and $w^*_{jg'}$. The new approach accounts for the possibility that even though a cluster might be large (so that w^*_j is large), some subgroups within the cluster may be small.

A second key difference between the new and old approaches is that for models with baseline covariates, the new approach uses the divisor (m_T-1) for $\Delta_{TW}(g,g')$ and (m_C-1) for $\Delta_{CW}(g,g')$ without subtracting out the number of covariates (v) as was done before. The reason for this is that the inclusion of baseline covariates in the regression models should have little effect on the subgroup covariance estimates, because *RCT-YES* includes a single set of baseline covariates without terms formed by interacting the baseline covariates with subgroup and treatment status indicator variables.

The new approach yields much more stable F-statistics for tests of subgroup interaction effects than before. In simulations comparing the new and old approaches using the setup in Table 3 above, nearly 25 percent of F-statistics were invalid under the old approach due to non-invertible variance-covariance matrices, whereas this never occurred under the new approach.

The same updates apply to the subgroup interaction tests for random blocked designs (Designs 2 and 4 for the population average treatment effect [PATE] parameter). For the Design 2 PATE analysis, *RCT-YES* now uses the following revised version of Equation (6.25b) in Schochet (2106) for calculating the covariances between the impact estimates for those in subgroups g and g':

$$(6.25b1)\,\frac{1}{(h-1)h\bar{w}_g^*\bar{w}_{g'}^*}\sum_{b=1}^{h}(w_{gb}^*\hat{\beta}_{nclus,g,b,PATE}-\bar{w}_g^*\hat{\beta}_{nclus,g,blocked,PATE})(w_{g'b}^*\hat{\beta}_{nclus,b,g',PATE}-\bar{w}_{g'}^*\hat{\beta}_{nclus,g',blocked,PATE}),$$

where w_{gb}^* is the sum of the weights for individuals in subgroup g and block b, $\bar{w}_g^* = \sum_{b=1}^{h} w_{gb}^* / h$ is the associated mean weight, h is the number of blocks, $\hat{\beta}_{nclus,g,b,PATE}$ is the block-specific impact estimate for the subgroup, and $\hat{\beta}_{nclus,g,blocked,PATE}$ is the pooled impact estimate across all blocks. This approach differs from the previous *RCT-YES* approach which used the *aggregated* w_b^* and $\bar{w}^*$ weights for *all* covariance calculations. In addition, unlike the old approach, for models with covariates, the divisors in (6.25b1) do not account for the number of covariates included in the models for the reasons described above. The revised approach for the Design 4 PATE parameter is identical.

5. For blocked designs, F-tests rather than chi-squared tests are now used to test for differences in treatment effects across blocks

For blocked designs under the finite-population model (Designs 2 and 4), the F-statistics for the block interaction tests for full sample analyses are calculated by dividing the chi-squared statistics by $df = h-1$, where h is the number of blocks. These F-statistics have an approximate $F(df, ddf)$ distribution, where ddf is the denominator degrees of freedom that depends on the sample size (individuals or clusters depending on the design) and the number of covariates and blocks. The program now uses F-tests rather than chi-squared tests for similar reasons as discussed above for the subgroup interaction tests. These results are presented in Table 10 in the .html output tables for the finite-population (default) specification.

D. Updates to Output Tables and *RCT-YES-Graph*

1. The study results are reported using the same .html tables as in earlier versions, adapted for multi-armed designs

The .html tables are now created separately for each pairwise contrast. A new table called “Table 9: Summary” reports the full sample results for all pairwise contrasts and outcomes in a single table.

In reporting the impact findings for models with baseline covariates, *RCT-YES* reports the regression-adjusted mean outcome for the first contrasted group and the unadjusted (raw) mean outcome for the second contrasted group. Thus, for models with covariates and some blocked designs, the reported means for a particular research group might differ depending on whether that research group is considered to be the first or second group in the contrast. *RCT-YES* considers the research group with the larger code to be the first group and the research group with the smaller code to be the second group. For example, for a design with three research groups coded as 0, 1, and 2, the output will display the pairwise contrasts in the following order: (i) Group 1 versus Group 0, (ii) Group 2 versus Group 0, and (iii) Group 2 versus Group 1. In this example, if baseline covariates are included in the estimation models, the reported means for Group 1 will differ for the first and third contrasts.

For multi-armed designs, *RCT-YES* reports statistical significance of the estimated impacts in three ways regarding adjustments for multiple testing: (1) no adjustments using the * symbol, (2) adjustments across outcomes within the same domain but not across pairwise contrasts using the ^ symbol, and (3) adjustments across both domain outcomes and pairwise contrasts using the + symbol. For example, if there are 4 research groups (with 6 possible pairwise contrasts) and 2 domain outcome variables, the program will use the ^ symbol when adjusting the p-values for the 2 domain outcomes only (for each pairwise contrast in isolation), and the + symbol when adjusting the p-values for the 12 hypothesis tests. Only the * and ^ symbols apply for designs with two research groups.

2. The .csv file produced by the program now contains the impact findings for each pairwise contrast

The revised .csv file, discussed earlier in this manual, contains the study findings stacked for each pairwise contrast. The pairwise contrasts in the file can be identified using the “group1” and “group2” variables. Users can read in the .csv file into their own computer programs for additional analyses or reporting.

3. *RCT-YES-Graph* was updated to allow for multiple research groups

Users can select which pairwise comparisons they want to plot and for which outcomes and subgroups. Users can also select whether the graphs for full sample analyses should display multiple comparison corrections for both multiple treatment groups and multiple domain outcomes, for only one of these reasons but not the other, or for neither of them. Note that the graphs will display multiple comparisons corrections using only the Benjamini-Hochberg method or the Bonferroni method, depending on which method was pre-specified for the analysis (Benjamini-Hochberg is the default), but not both.

(This page left intentionally blank for double-sided copying)

Version 1.1 Updates: June 2016

Version 1.1 of the *RCT-YES* software was released in June 2016 and updated Version 1.0 (released in May 2016) in several ways:

1. The maximum number of blocks that can be included in the analysis was increased for BLOCK_FE=1 designs

For blocked designs, *RCT-YES* automatically invokes the super-population model (SUPER_POP=1) if the model contains too many blocks. This design feature was implemented to minimize the chances that the R or Stata programs will crash because of too many model covariates. In Program Version 1.0, for finite-population specifications, the super-population model was invoked if (2bs + x) > 200, where b is the number of blocks, s is the number of categories of the subgroup variable under investigation for a given subgroup analysis (and equals 1 for the full sample analysis), and x is the number of baseline covariates. This same rule was used for the default BLOCK_FE=0 specification and for the optional BLOCK_FE = 1 specification where the model includes block indicators but not block-by-treatment interaction terms. In Version 1.1, the rule for invoking the super-population model for the BLOCK_FE=1 specification is now (b + s + x) > 200 instead of (2bs + x) > 200, reflecting the smaller number of model covariates for this specification.

2. Bugs were fixed that caused the program to crash for numeric subgroup variables when the user specified both numeric and non-numeric codes in the interface

Users may inadvertently specify both numeric and character codes for a particular subgroup variable (for example, a code of 0 for males and F for females for the numeric subgroup variable GENDER). The program can now handle these input errors and will exclude the subgroup from the analysis in these instances. In addition, Table 1 in the .html file will now note the reason for excluding the subgroup in order to help users identify and fix the problem.

3. Bugs were fixed that caused the program to crash for some cases where the same variable is specified more than once in the interface

The program can now handle instances where the same variable is specified as an outcome, covariate, weight, and/or subgroup. Although some of these combinations will be highly unusual (for example, specifying an outcome variable as a weight variable), the program allows them.

4. All potential scalar/variable conflicts were fixed

In Stata, there are scalar variables and data variables. Scalar variables are similar to macro variables and are used throughout the Stata *RCT-YES* program to store numeric constants. Recent testing revealed the potential for overlap between these two types of variables, where the data variable would be used instead of the scalar variable. The code was modified to add the scalar() function to all scalar variable references to avoid any scalar/data variable naming collisions.

5. Error messages were improved for some subgroup analyses in Table 1 of the .html file

Table 1 now lists the outcome variable associated with certain subgroup exclusion error messages so that these errors can be more easily identified and fixed.

6. Footnotes are now printed for the "Treatment-control means" chart for the "Line graph" option in *RCT-YES-Graph*

These footnotes describe the symbols used in the graph to signify statistical significance of the impact estimates.

7. A blank line has been added between the graph title and the graph in *RCT-YES-Graph*

The spacing of the title is now parallel to the spacing of the footnotes.

8. A bug was fixed to allow *RCT-YES* to be launched by opening a previously-saved input specification (.rctyes) file from the directory where it was saved

Previously, clicking on the .rctyes file would not launch the interface if there was a space in the .rctyes file's path name. This problem is now resolved.

References

Bell, R. and D. McCaffrey (2002). Bias reduction in standard errors for linear regression with multi-stage samples. *Survey methodology, 28(2)*, 169–181.

Guo, X. and W. Pan (2002). Small-sample performance of the score test in GEE. University of Minnesota Division of Biostatistics: Working Paper.

Liang, K. and S. Zeger (1986). Longitudinal data analysis using generalized linear models. *Biometrika, 73*, 13–22.

Mancl, L.A. and T.A. DeRouen (2001). A covariance estimator for GEE with improved small sample properties. *Biometrics, 57*, 126–134.

Pan, W. and M. Wall (2002). Small sample adjustments in using the sandwich variance estimator in generalizing estimating equations. *Statistics in Medicine, 21*, 1429–1441.

Pustejovsky, J. and E. Tipton (2016). Small sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models. *Journal of Business and Economic Statistics, online*.

Scher, L. and R. Cole (2017). *Evidence review standards considerations when using RCT-YES.* https://www.rct-yes.com/Content/PDF/Evidence%20Standards%20When%20Using%20RCT-YES.pdf.

Schochet, P. Z. (2009). An approach for addressing the multiple testing problem in social policy impact evaluations. *Evaluation Review, 33*(6), 539–567.

Schochet, P. Z. (2013). Estimators for clustered education RCTs using the Neyman model for causal inference. *Journal of Educational and Behavioral Statistics, 38*(3), 219–238.

Schochet, P. Z. (2016). *Statistical theory for the RCT-YES software: Design-based causal inference for RCTs, Second Edition* (NCEE 2015–4011). Washington, DC: U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance. https://ies.ed.gov/ncee/pubs/20154011/pdf/20154011.pdf

Schochet, P. Z. (2017a). *What is design-based causal inference for RCTs and why should I use it?* (NCEE 2017–4025). Washington, DC: U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance. https://ies.ed.gov/ncee/pubs/20174025/pdf/20174025.pdf

Schochet, P. Z. (2017b). *Multi-armed RCTs: A design-based framework* (NCEE 2017–4027). Washington, DC: U.S. Department of Education, Institute of Education Sciences, National Center for Education Evaluation and Regional Assistance. https://ies.ed.gov/ncee/pubs/20174027/pdf/20174027.pdf