



# User's Manual

---

**Peter Z. Schochet:** Author of User's Manual and Project Lead

**Carol Razafindrakoto:** Lead Software Developer

**Carlo Caci:** Lead Interface and Website Developer

**Mason DeCamillis:** R Software Developer

**Matthew Jacobus:** Stata Software Developer

**Mathematica Policy Research, Inc.**

**Version 1.0, May 2016**

*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.

© Mathematica Policy Research, Inc.





## Acknowledgments

The author is very grateful to Dr. Russell Jackson at Decision Information Resources, Inc. and Dr. Joy Lesnick, Dr. Tom Wei, and Dr. Ruth Neild at the Institute for Education Sciences at the U.S. Department of Education for generous assistance throughout the project; Jason DiBartolo who helped design the *RCT-YES* interface screens; Dr. Alex Resch and Dr. Matthew Stagner for general support and guidance; Steve Bates, Hannah Burak, Brigitte Tran, Mark Lafferty, Kenneth Mascaro, Angela Deng, and Brice Overcash for developing the *RCT-YES* website and associated videos and documents; Tamara Barnes for legal counsel; Jennifer McNulty for oversight of the interface development; Brian Daniels for assistance programming the interface; Malik Mubeen for assistance testing the R and Stata software; Andrew McGuirk for programming to translate the output from the computer programs into output tables; Tim Kautz, Charles Tilley, Caroline Francis, and Erin Dillon for software usability testing; Barbara Gelhard for reviewing the .html files for 508 compliance; Margaret Hallisey for 508 compliance production support; and members of the Analytic Technical Assistance and Development project team—Dr. Lauren Scher, Dr. Ronald McCowen, and Dr. Mark Dynarski—for helpful comments.



## Contents

Acknowledgments	iii
Introduction to <i>RCT-YES</i>	1
1. Overview of how to download and use <i>RCT-YES</i>	3
a. System requirements	3
b. How to download <i>RCT-YES</i>	3
c. How to use <i>RCT-YES</i>	4
d. Dictionary of program inputs	7
2. Overview of <i>RCT-YES</i> designs, analyses, and methods	13
a. Defining treatment and control status in RCTs	14
b. Defining outcomes in RCTs	16
c. The four RCT design types considered by <i>RCT-YES</i>	17
d. Impact estimation methods with and without the use of baseline covariates	20
Methods for blocked designs	22
Methods for clustered designs	22
e. Estimating impacts on the offer and receipt of intervention services	23
f. Standard errors and hypothesis testing	23
g. Subgroup analyses	24
h. Baseline equivalence analyses	25
i. Handling missing data	26
Missing outcome data	26
Missing covariate data	27
Missing subgroup data	28
j. Finite-population and super-population models	28
k. Weighting	29
l. Reporting and graphing impact findings	29
m. Summary of designs in <i>RCT-YES</i>	30
3. Minimizing disclosure of personally identifiable information	33
4. Creating the input data file	35

a. Overview of the input data file structure	35
b. Examples	36
Design 1	36
Design 2	39
Design 3	39
Design 4	43
<b>5. Entering inputs, generating output files, and conducting the analysis</b>	<b>45</b>
a. Launching the interface screens and entering program inputs into them	45
Saving program inputs and exiting the program	46
Getting Started Screen	46
Accessing the Variable List Window	49
Design & Analysis Parameters Screen	51
Outcomes, Weights, Covariates, & Subgroup Screens	54
Baseline Equivalence Analysis Screen	60
b. Generating output files using the interface and identifying input errors	61
c. Running the R or Stata computer program and identifying errors	63
<b>6. Interpreting the analysis results</b>	<b>65</b>
a. Design 1 example	65
b. Design 2 example	69
c. Design 3 example	77
d. Design 4 example	77
e. Content of the .csv data file	80
<b>7. Graphing the impact results</b>	<b>85</b>
<b>Appendix A. Downloading and Running R and Stata</b>	<b>91</b>
a. Downloading and Running R	91
Downloading and installing R	91
Converting a .csv file into a .rds file	92
Running the R program produced by the interface	94
b. Downloading and Running Stata	96
Downloading and installing Stata	96
Converting a .csv file into a .dta file	96
Running the Stata program produced by the interface	97

## References 99

### Tables

Table 1. Overview of steps to use <i>RCT-YES</i>	6
Table 2. Dictionary of input statements for <i>RCT-YES</i>	8
Table 3. Glossary of key terms and definitions related to <i>RCT-YES</i> inputs and outputs	15
Table 4. Summary of designs in <i>RCT-YES</i>	31
Table 5. Description of variables used for the Design 1 example	38
Table 6. Hypothetical data file for selected variables for the Design 1 example	38
Table 7. Hypothetical data file for selected variables for the Design 2 example	40
Table 8. Hypothetical data file for selected variables and students for the Design 3 example with individual-level data	41
Table 9. Hypothetical data file for selected variables and schools for the Design 3 example with cluster-level data	42
Table 10. Record layout of the .csv file	81

### Output Tables

Selected output tables for Design 1	70-75
Selected output tables for Design 2	76
Selected output tables for Design 3	78-79

### Figures

Figure 1. Depiction of a non-clustered, non-blocked RCT design where students are randomized in a single school district (Design 1)	18
Figure 2a. Depiction of a clustered, non-blocked RCT design where schools are randomized in a single district (Design 3)	19
Figure 2b. Depiction of a clustered, non-blocked RCT design where classrooms are randomized (Design 3)	19





## Introduction to *RCT-YES*

The *RCT-YES* software package estimates average treatment effects for randomized controlled trials (RCTs) of interventions and policies, where individuals or groups of individuals are randomly assigned to treatment or control groups. The development of *RCT-YES* was funded by the Institute of Education Sciences (IES) at the U.S. Department of Education (ED) to facilitate the conduct of RCTs by state and local education agencies to test promising interventions and policies in their service areas. Student- and teacher-level data from state longitudinal data systems (SLDSs) provide a rich data source for such evaluations, although other data sources could also be used for the analysis. By taking advantage of opportunities to conduct RCTs of new or existing policies, “opportunistic experiments” offer the chance for education agencies and policymakers to generate rigorous evidence about what works in the decisions they make every day.

*RCT-YES* estimates average treatment effects—grounded in rigorous statistical theory—for a wide range of designs used in education research. The program estimates intervention effects by comparing the average outcomes of those randomly assigned to different research conditions for the full sample and for baseline subgroups of students, educators, and schools. The program conducts hypothesis tests to assess the statistical significance of the estimated effects and reports evaluation findings in formatted tables and graphs that conform to the presentation of RCT findings in IES-published reports. While *RCT-YES* was developed for RCTs in the education area, it is also applicable to RCTs in other fields. It can also be used to estimate intervention effects for quasi-experimental designs (QEDs) with comparison groups, although these designs may require supplemental methods.

*RCT-YES* is intended for those with an introductory knowledge of RCT designs and for those with less knowledge and experience who can be trained in its use (for example, as part of researcher-practitioner collaborations). Because *RCT-YES* is intended for users with diverse backgrounds, it was designed to minimize user input and the data required for estimation. Consequently, the program relies on a number of default specifications. Users must be aware that these defaults might not apply in all contexts and should use program options to change the defaults where appropriate.

This *RCT-YES* User’s Manual discusses how to (i) download the program for free from [www.rct-yes.com](http://www.rct-yes.com), (ii) specify program inputs using a Windows-based desktop interface application, and (iii) run the computer program produced by the interface using the R or Stata software to generate an .html file presenting analysis results and a graphing application to plot them. We use examples and screenshots to clarify the program inputs and outputs. We also provide a non-technical discussion of the designs, analyses, and methods in *RCT-YES* to help users understand key program features.

**To run *RCT-YES*, users must have R or Stata installed on their computers** (the free R software can be downloaded from [www.r-project.org](http://www.r-project.org) and Stata from [www.stata.com](http://www.stata.com)). Users do not need to know R or Stata, however, beyond how to run a computer program in the selected language and how to

create a compatible data file for analysis (see Appendix A for a discussion of these topics). The format of the input data file for *RCT-YES* must be a *.rds* file for R or *.dta* file for Stata. If individuals are the unit of random assignment, the data file should contain one record per individual. If groups (such as schools or hospitals) are the unit of random assignment, the data file can contain one record per individual (with a group ID) or one record per group average (for example, average test scores for students in the school). Users can create their data file in any language (for example, Excel) and convert it into an R or Stata file as a last step before running the program (see Appendix A).

Importantly, *RCT-YES* must be considered a *tool* for analyzing RCT data, and is *not* a substitute for researcher experience and judgment. A successful RCT depends on the suitability of the design for addressing well-defined causal research questions with sufficient precision, the successful implementation of the intervention, and high quality study data. Even if these conditions are met, a well-conducted analysis of RCT data requires considerable expertise in a range of methodological areas, such as the construction of outcome variables, impact estimation methods, hypothesis testing, adjustments for missing data, and the interpretation and reporting of evaluation findings. Thus, the policy relevance of the results produced by *RCT-YES* will largely depend on the rigor of the study design, the quality of its data, and user expertise in correctly specifying program inputs and interpreting the program output. Where appropriate, users may want to consult with individuals trained in evaluation methodology to gain the most out of the program.

The remainder of this manual is presented in seven chapters and one appendix:

- Chapter 1 provides an overview of how to download and use *RCT-YES*, including a dictionary of program input statements
- Chapter 2 provides a non-technical discussion of the designs, analyses, and methods in *RCT-YES*, geared towards readers with a basic knowledge of statistical theory. The methods underlying *RCT-YES*, which rely on recent research using design-based theory, are discussed in detail in an accompanying technical appendix (Schochet, 2016).
- Chapter 3 discusses how *RCT-YES* addresses data disclosure issues
- Chapter 4 discusses the structure of the program input file
- Chapter 5 provides details on how to specify program inputs using the *RCT-YES* interface screens, and how to run, in a separate step, the R or Stata computer program file generated by the interface to produce the analysis results
- Chapter 6 discusses the output *.html* tables using examples and the associated *.csv* file
- Chapter 7 discusses how to graph the results using the *RCT-YES-Graph* application
- Appendix A provides step-by-step instructions on how to download the free R software, create an R data file, run an R program, and similarly for Stata

## 1. Overview of how to download and use *RCT-YES*

*RCT-YES* can be downloaded for free from [www.rct-yes.com](http://www.rct-yes.com). Installation of the software will produce an *RCT-YES* icon. Double clicking the icon will yield a series of screens for users to enter program inputs. After entering all required inputs, the interface will produce a computer program in R or Stata (named in the interface), that users will then need to run in a separate step outside the interface to conduct the analysis. To run *RCT-YES*, users must have R or Stata installed on their computers (see Appendix A). The program will produce a .html output file (also named in the interface) displaying formatted tables with analysis results, an associated .csv data file, and a *RCT-YES-Graph* program in R that reads in the .csv file to graph the impact findings. The interface and .html files are accessible (508 compliant) for those with disabilities.

### a. System requirements

*RCT-YES* is a Windows-based application that runs locally on users' computers. The application was tested on a 32-bit Windows 7 Professional platform, and the program files were tested using R Versions 3.1.0 through 3.1.4 and Stata Version 13. The system requirements are as follows:

OS	Windows XP; Windows Server 2003; Windows Vista; Windows Server 2008; Windows 7; Windows 8; Windows 9
CPU	1 GHz
Memory	256 MB
Hard Drive Space	280 MB

### b. How to download *RCT-YES*

*RCT-YES* can be downloaded for free from [www.rct-yes.com](http://www.rct-yes.com). Users can download the program from the website by clicking the **DOWNLOAD** button. A series of installation screens will appear for users to run or save the rct-yes.exe file. Before the executable file is run, users will be asked to check a box indicating that they agree to terms of the *RCT-YES* software license. If the system requires administrative privileges, a system administrator may need to run the executable file to install the software. If users have trouble, they should request help using the support mailbox in the website.

The software installation will produce the following *RCT-YES* desktop icon:



The installation will also produce a *RCT-YES* program link in the **Start/All Programs** menu, **Mathematica Policy Research, Inc** folder. It will also create a directory on a local drive containing program files that should not be opened, moved, or edited, or the program may not run successfully.

## 1. Using *RCT-YES*

The *RCT-YES* website also contains links to the *RCT-YES* User's Manual, a Quick Start Guide, the *RCT-YES* Methods Appendix, a frequently asked questions document, how-to videos, a contact support mailbox, and related websites containing information on how to conduct impact studies.

When launching the program, users will be alerted about new *RCT-YES* versions when they become available. In this case, a dialog box will appear giving the user the option to be directed to the *RCT-YES* website to download the updated software and associated documents.

### c. How to use *RCT-YES*

The program can be run in seven steps (see Table 1 for more details):

- Step 1.** Launch the *RCT-YES* interface using any of the following options by double clicking (i) the *RCT-YES* desktop icon, (ii) the *RCT-YES* program link in the **Start** menu, or (iii) a previously saved input specification file in the directory where it was saved. The interface will provide screens for entering program inputs and generating an R or Stata computer program file to be run outside the interface to conduct the analysis. The interface does not read the input dataset.
- Step 2.** In the **Getting Started** screen, enter whether the analysis is to be conducted in R or Stata and information on the input dataset (a .rds file for R or .dta file for Stata). As an option, navigate to the **Generate Variable List** and **Import Variable List** screens and follow directions to create an interface window displaying a list of variables in the dataset to help with data entry. The variable list window will open automatically, but can also be accessed by (i) clicking the **Variables** command in the toolbar menu or (ii) starting to type variables directly into the input boxes to filter and select the variables.
- Step 3.** Enter remaining program inputs into the interface screens, including key design and analysis parameters, the treatment-control indicator variable, outcome variables, covariates for regression and baseline equivalence analyses, subgroup variables, and weights. If created in Step 2, the variable list window can be used to help enter variables.
- Step 4.** Save the program inputs in a file at any point during the session by clicking the **File** menu and **Save** or **Save As** command. After specifying all inputs, navigate to the **Generate Output Files** screen to provide information on where to save the following files: (i) an R or Stata computer program file, created by the interface, that will need to be run outside the interface in Step 6 below to conduct the analysis; and (ii) several analysis results files, generated by the computer program, including a .html file with formatted tables showing analysis findings, an associated .csv data file, an associated .log file showing detailed regression model results, and a *RCT-YES-Graph* application that can be used for graphing the impact findings in R.

- Step 5.** After entering all information in the **Generate Output Files** screen, click the **Generate Files** button at the bottom of that screen to produce the output files. A window will display information on the output files that were successfully generated and file generation errors. After generating all files, minimize or exit the interface.
- Step 6.** Run the R or Stata program file outside the interface using procedures typically used to run such programs. The program will produce a .html file containing tables with analysis results, a .csv data file containing information in the output tables, a .log file displaying estimation results from the R or Stata regression functions, and a .R program that accesses the *RCT-YES-Graph* application. These files will all have the same base name.
- Step 7.** Graph the impact results by running in R the *RCT-YES-Graph* program. A dashboard will appear for entering inputs and creating graphs. **The graphics app requires that the free R software has been downloaded and the free “shiny”, “shinydashboard”, and “ggplot2” R packages have been installed (see Chapter 7 and Appendix A).**

### Identifying and Fixing Program Errors

Program errors will be displayed in several places:

- **The interface’s File Generation Summary window that appears when users attempt to generate the output files.** The interface will check that all required program inputs have been entered, variables do not have unallowable symbols, key design variables are not specified more than once, and all specified directories and files exist. If errors are present, the interface will not produce the R or Stata computer program file for the analysis until the errors have been fixed. Note that the interface does not read the data file, so it does not conduct data checks.
- **Table 1 of the .html file produced by the R or Stata program.** This table will indicate data problems and how they are handled. The table will also display critical program errors, such as invalid input directories and data files.
- **The R or Stata program screen and .log file.** If the inputs have been specified correctly and the data file is in the correct format, errors that crash the program without a reason provided in Table 1 of the .html file should be rare, but examples of how this could occur are:
  1. The analysis uses a dataset that contains a much larger number of observations or a system with less memory than was used for program testing
  2. The dataset contains a very large number of blocks (see page 22 for a discussion of this issue and possible solutions)
  3. A very large number of baseline covariates are specified for the baseline equivalence analysis (see page 26 for a discussion of this issue and solutions)

**If users encounter errors that they cannot resolve, they should use the contact support mailbox in the *RCT-YES* website.**

## 1. Using *RCT-YES*

**Table 1. Overview of steps to use *RCT-YES***

Step	Purpose	Actions
<b>Step 1</b>	Launch the <i>RCT-YES</i> interface screens to enter or edit the program inputs	Double click the <i>RCT-YES</i> icon, the <i>RCT-YES</i> link in the <b>Start</b> menu, or a previously saved input specification file [.rctyes extension] in the directory where it was saved. For the first two options, either enter inputs for the first time or open and edit a previously saved input file using (i) the <b>File</b> menu and <b>Open</b> or <b>Open Recent</b> command or (ii) the <b>Recently Used Files</b> list in the <b>Getting Started</b> screen.
<b>Step 2</b>	In the <b>Getting Started</b> subscreens, enter required information on (i) the desired language for the analysis (R or Stata) and (ii) the input dataset (a .rds file for R or a .dta file for Stata). As an option, create a variable list window that can be used to help enter variables into the interface.	To create the variable list window, proceed as follows: <ul style="list-style-type: none"> <li>• Navigate to the <b>Generate Variable List</b> screen</li> <li>• Enter information on the base and path names for the R or Stata computer program file [.R or .do extension], to be generated by the interface, that will need to be run outside the interface to create a variable list text file [.varlist extension]. The interface will add a “_VL” suffix to the base name to distinguish it from other output files discussed below. Click the <b>Generate Variable List File</b> button.</li> <li>• Minimize (or exit) the interface, and run the R or Stata computer file outside the interface to create the variable list text file [.varlist extension]</li> <li>• Return to the interface, navigate to the <b>Import Variable List</b> screen, input information on the variable list file to import, and click the <b>Import Variable List</b> button. The variable list window will open automatically. If closed, it can be re-opened by clicking the <b>Variables</b> command in the toolbar. It can also be accessed using the autocomplete feature by starting to type variables directly into the input boxes to filter and select the desired variables.</li> </ul>
<b>Step 3</b>	Enter the remaining program inputs using the interface screens. The inputs can be saved in a file at any point during the session (as discussed in Steps 4 and 5).	Specify the following required or optional program inputs: <ul style="list-style-type: none"> <li>• Treatment-control status indicator variable (required)</li> <li>• Design and analysis parameters (some required, some optional)</li> <li>• Outcome variables (required)</li> <li>• Weights, covariates for regression analyses, intervention service receipt indicator variables, baseline subgroups, and covariates for baseline equivalence analyses (optional)</li> </ul>
<b>Step 4</b>	Enter information on where to save the interface file for future use, and, after specifying all inputs, enter information on where to save an R or Stata file to run in Step 6 and associated analysis results files	Save program inputs in a file at any point during the session by clicking the <b>File</b> menu and <b>Save</b> or <b>Save As</b> command. If all inputs have been entered, navigate to the <b>Generate Output Files</b> screen by clicking (i) the <b>Generate</b> tab or (ii) the <b>Generate</b> menu and <b>Generate Files</b> command. The screen will request information on the base name and locations of several files. Proceed as follows: <ol style="list-style-type: none"> <li>1. Enter the common base name for all files (which will be pre-filled if the input specification file was previously saved)</li> <li>2. Use the <b>Browse</b> button to locate and click the folder location for saving the input specification file [.rctyes extension]</li> <li>3. Use the <b>Browse</b> button to click the folder locations for saving the following two categories of output files: <ol style="list-style-type: none"> <li>i. An R or Stata computer program file [.R or .do extension], created by the interface, to be run in Step 6</li> <li>ii. Several analysis results files, generated by the R or Stata computer program, including a .html file with tables of analysis results, a .csv file, a .log file, and a .R graphics program</li> </ol> </li> </ol>

Step	Purpose	Actions
<b>Step 5</b>	Generate the output files, check for errors, and then either (i) exit the interface using the <b>File</b> menu and <b>Exit</b> command or (ii) minimize the interface	Click the <b>Generate File</b> button at the bottom of the <b>Generate Output Files</b> screen to save the files. A pop-up window will provide information on the files that were successfully generated and file generation errors. If needed, fix errors, generate the files, and exit or minimize the interface.
<b>Step 6</b>	Run the R or Stata program file generated in Step 5 to conduct the analysis and produce the analysis results files	Run the R or Stata program file <u>outside</u> the interface as you would normally run such programs. This program will read the input dataset named in Step 4 and conduct the analysis using the information specified in the interface. The program will create the analysis results files specified in Step 4, including a .html file with tables of impact results, a .log file with detailed model estimation results, a .csv data file containing information from the output tables, and a .R computer program that accesses the <i>RCT-YES-Graph</i> application (see Step 7).
<b>Step 7</b>	Graph the impact results and save them to a file	<b>Make sure that R and the free shiny, shinydashboard, and ggplot2 R packages have been installed on your computer (see Chapter 7 and Appendix A).</b> In R, use the <b>File/Source R code</b> command to locate and select the .R graphics program produced in Step 6. The base name and location of this graphics program will be the same as for the .html file except the base name will contain the “_graph” suffix. A dashboard will appear that requests information on the location of the .csv file with the impact results; the domains, outcomes, and subgroups to chart; the graph type; graph options (such as labels and y-axis ranges); and the file to save the graphs. Press <b>Submit</b> at the bottom of the dashboard to display the graph and any subsequent changes to it. Press <b>Download Plot</b> to save each graph to a file. When you are done, exit the app to return to the R prompt (if not, you will need to hit the esc key or the Stop command in the toolbar to return to R).

#### d. Dictionary of program inputs

Table 2 displays a dictionary of program inputs used in the interface, including the names, definitions, formats, and default values for each input, and page numbers where the inputs are discussed in this manual. The ordering of the inputs in Table 2 follows the ordering of the inputs in the *RCT-YES* interface screens (except for the inputs in the **Generate Variable List** and **Import Variable List** screens which are shown near the end of the dictionary). As indicated in the table, some inputs are required for all designs, others are required for some designs only, while others are optional.

The next two chapters provide an overview of the designs, analyses, and methods in *RCT-YES*. These topics are critical for understanding the dictionary of inputs in Table 2 and how to use the software effectively. Chapters 4 to 6 provide more details on program inputs and outputs using examples, and how to use the *RCT-YES-Graph* application. Finally, for users unfamiliar with R and Stata, Appendix A provides information on how to download R and Stata, how to run R and Stata programs, and how to install the free shiny, shinydashboard, and ggplot2 R packages that are needed to use *RCT-YES-Graph*.

## 1. Using *RCT-YES*

**Table 2. Dictionary of input statements for *RCT-YES***

Input variable (Page references)	Variable definition	Variable format	Additional information
<b>Getting Started: R/Stata and Input Data</b>			
<b>STAT_PACKAGE</b> (1; 46; 91-97)	Statistical package for the analysis	R Stata	<b>Required</b>
<b>DATA_FILE</b> (35-43; 46)	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.	<b>Required</b>
<b>Design Selection and Title</b>			
<b>DESIGN</b> (15; 17-20; 51)	Type of design	1 = Non-clustered, non-blocked 2 = Non-clustered, blocked 3 = Clustered, non-blocked 4 = Clustered, blocked	<b>Required</b>
<b>TITLE</b> (51)	Title for output tables	Character	Optional
<b>Required Design Parameters</b>			
<b>TC_STATUS</b> (14; 35-36; 50-52)	Name of treatment or control status indicator variable	0 = Control 1 = Treatment	<b>Required for all observations</b>
<b>BLOCK_ID</b> (15; 17-20; 22; 35; 39-40; 51-52)	Name of variable containing the block identification codes	Numeric or character	<b>Required for Designs 2 and 4 for all observations</b>  For the default finite-population (FP) model, blocks are included in the analysis if they contain at least 2 treatment and at least 2 control group observations  For the super-population (SP) model or BLOCK_FE=1 FP model, blocks are included with at least 1 treatment and at least 1 control
<b>MATCHED_PAIR</b> (15; 20; 51-52)	Indicator for a matched pair design	0 = Not a matched pair design (default)  1 = Matched pair design	<b>Required for Designs 2 and 4 for matched pair designs</b>  Pairs are included only if data are available for both pair members  The SP model is used for estimation
<b>CLUSTER_ID</b> (15; 17-20; 22; 35; 39-43; 51-52)	Name of variable containing the cluster identification codes	Numeric or character	<b>Required for Designs 3 and 4 for all observations</b>  Clusters are included if they have at least one observation with outcome data
<b>TYPE_CLUS_DATA</b> (35; 39-43; 51-52)	Indicator for clustered designs as to whether the input file contains individual- or cluster-level data	0 = Cluster-level averages 1 = Individual-level data	<b>Required for Designs 3 and 4</b>



Input variable (Page references)	Variable definition	Variable format	Additional information
<b>CLUSTER_FULL</b> (41-43; 51-52)	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	<b>Required for Designs 3 and 4 if TYPE_CLUS_DATA = 0</b>
<b>Optional Design and Analysis Parameters</b>			
<b>SUPER_POP</b> (15; 28-29; 52-53)	Indicator of preference for the super-population (SP) model	0 = Finite-population (FP) model 1 = SP model	Optional Default is the FP model
<b>CATE_UATE</b> (28-29; 52-53)	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
<b>BLOCK_FE</b> (22; 52-53)	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
<b>LABEL_T</b> <b>LABEL_C</b> (14; 52-53)	Labels for the treatment and control groups, respectively	Character of length 14 or less	Optional; no quotes needed Defaults are Treatment and Control "Group" should be omitted from the label because the program will add it to the end of the label
<b>MISSING_COV</b> (27-28; 52-53; 67)	Maximum percentage of missing data for a baseline covariate to be included in the regression models. This condition is applied to both the treatment and control groups.	Numeric: 0 to 75	Optional Default is 30
<b>OBS_COV</b> (20; 52-53; 66)	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
<b>MIN_NUM</b> (33; 52-53; 66)	Minimum group size adopted by the state or other entity for reporting outcomes to protect personally identifiable information (PII)	Integer $\geq 3$	Optional Default is 10
<b>ALPHA_LEVEL</b> (16; 23-24; 52-53)	Significance level for testing the null hypothesis of zero average treatment effects (in percentages)	Integer: 1 to 30	Optional Default is 5

## 1. Using *RCT-YES*

Input variable (Page references)	Variable definition	Variable format	Additional information
<b>NO_COV_SG</b> (24-25; 52-53)	Excludes covariance terms in the statistical tests of differences in impact estimates across subgroup categories (for example, for males and females)	0 = Include covariance terms in the statistical tests 1 = Exclude covariance terms in 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
<b>LIMIT_PRINT</b> (29; 52-53; 65)	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
<b>CSV_FILE</b> (30; 52-53; 65; 80-83)	Specifies that the computer program should produce a .csv data file containing information from the output tables for further analyses and reporting	0 = .csv file not produced 1 = .csv file produced	Optional  Default is the production of the .csv file
<b>Outcomes, Weights, Covariates, and Subgroups</b>			
<b>OUTCOME_DMN</b> (16; 54-56)	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
<b>OUTCOME</b> (15; 16; 26; 35-43; 54-56)	Name of outcome variable	Numeric; all missing data codes are valid based on the language used (Stata or R)	<b>Required</b>  Cases with missing values for an outcome are excluded from the analysis for that outcome
<b>LABEL</b> (55-56)	Label for outcome variable	Character Blank	Optional
<b>WEIGHT</b> (26; 29; 36; 55-56; 57-58)	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 clusters for clustered designs  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

Input variable (Page references)	Variable definition	Variable format	Additional information
<b>STD_OUTCOME</b> (30; 55-56)	Individual-level standard deviation of the outcome variable	Numeric > 0  Blank	<b>Required for Designs 3 and 4 if TYPE_CLUS_DATA = 0</b> in order for the program to calculate impacts in effect size units  Optional for other designs, where the default is the full sample standard deviation for the control group in the data
<b>COVARIATES</b> (15; 20-21; 27-28; 35-43; 55-56; 57-59)	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
<b>GOT_TREAT</b> (16; 23; 55-56)	Name of variable indicating the receipt of intervention services for the treatment and control groups. The variable should be <i>binary</i> for all designs except if TYPE_CLUS_DATA = 1, in which case the variable should be a <i>numeric service receipt rate</i> 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: $\geq 0$ and $\leq 1$	Optional for estimating complier average causal effects (CACE) pertaining to those who would receive intervention services as a treatment but not as a control (see Chapter 2e)  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 values are excluded from both the CACE and ATE analyses
<b>SUBGROUP</b> (15; 24-25; 28; 35-43; 56-59)	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 student, teacher, school, or other characteristics and must be large enough to protect data disclosure
<b>Baseline Equivalence Analysis</b>			
<b>BASE_EQUIV</b> (16; 25-26; 28; 60)	List of names of baseline covariates that are to be used to assess baseline equivalence for treatments and controls	Numeric: continuous or binary; all missing data codes are valid based on the language used (Stata or R)	Optional

## 1. Using *RCT-YES*

Input variable (Page references)	Variable definition	Variable format	Additional information
<b>NO_JNT_TEST</b> (25-26; 60)	Suppresses the joint test of baseline equivalence	0 = Conduct the joint test  1 = Do not conduct the joint test	Optional  Default is the conduct of 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

### Generate Variable List Window

<b>BASE_NAME_VL</b> (47-48)	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	<b>Required to produce the files below</b>
<b>COMP_PROG_VL</b> (47-48)	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	<b>Required to produce the file</b>
<b>FILE_VL</b> (47-48)	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	<b>Required to produce the COMP_PROG_VL file</b>
<b>IMPORT_VL</b> (47-49)	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	<b>Required to produce the variable list window</b>

### Generate Output Files for the Analysis

<b>BASE_NAME</b> (61-63)	Common base name for the three files below (that each have different file extensions)	Character	<b>Required to produce the files below</b>
<b>INPUT_SPEC_FILE</b> (61-63)	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	<b>Required to produce the file</b>
<b>COMPUTER_PROG</b> (61-63)	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	<b>Required to produce the file</b>
<b>RESULTS_FILE</b> (61-63)	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	<b>Required to produce the COMPUTER_PROG file</b>

## 2. Overview of *RCT-YES* designs, analyses, and methods

A randomized controlled trial (RCT) is a design where study participants—such as students, teachers, schools, or other entities—are randomly assigned to research conditions (for example, using a random number generator). In an RCT with a single treatment and control group, the treatment group is offered the intervention or policy under investigation whereas the control group is not (but is offered existing services). In RCTs with multiple treatment groups, each treatment group is offered a different intervention or a variation of an intervention. RCTs have been broadly accepted as the best design for providing convincing impact estimates, because random assignment ensures that the research groups are *balanced* on both observable and unobservable characteristics. Thus, differences in follow-up outcomes across the randomized groups can be attributed to the tested interventions. Stated differently, RCTs provide estimates of the “*causal*” connection between interventions and outcomes rather than associations (correlations).<sup>1</sup>

*RCT-YES* estimates intervention effects for commonly used education RCT designs that address the following causal research questions:

1. What are average effects of the intervention on student or educator outcomes for the full sample?
2. Do intervention effects differ for key subgroups of students, educators, and schools defined by their pre-randomization (baseline) characteristics?

*RCT-YES* addresses these research questions by comparing the average outcomes of those randomly assigned to different research conditions for the full sample and for baseline subgroups. For brevity, in this document, we often refer to these average treatment effects as “*impacts*” or “*intervention effects*.”

*RCT-YES* estimates intervention effects using data for *two* research groups. For designs with more than two research groups, pairs of research conditions can be compared to each other in separate runs of *RCT-YES*. For clarity, we assume hereafter a design with a single treatment and control group, but the discussion pertains fully to an analysis comparing two treatment groups.

An RCT contrasts one or more approaches, policies, or interventions. Thus, to correctly interpret the results produced by *RCT-YES*, researchers should, to the extent possible, obtain information on the intervention-related services received by the randomized groups. For example, this information could include the extent to which the interventions were implemented as planned, the nature and amount of intervention services received by the treatment group, and “business as usual” services

---

<sup>1</sup>The focus of this chapter is on “gold-standard” RCT designs. Most of the key concepts, however, apply also to quasi-experimental designs that can be analyzed using *RCT-YES*.

## 2. Designs, analyses, and methods

received by the control group. Information on services received by the control group is critical for understanding how the intervention being tested differs from what the treatment group otherwise would have experienced.

*RCT-YES* does not conduct additional analyses that researchers sometimes employ to help understand the variation in treatment effects in RCTs (see Schochet et al., 2014). For example, the program does not conduct analyses to identify mediating factors that account for intervention effects on longer-term outcomes, examine the variation in treatment effects for subgroups defined by their post-baseline experiences, or estimate quantile treatment effects to assess how intervention effects vary along the distribution of an outcome measure. Rather the focus of *RCT-YES* is on impact estimation for the full sample and for baseline subgroups.

Next, we provide an overview of key RCT concepts, designs, and statistical analyses that underlie *RCT-YES*. We illustrate the issues in the education context, but the issues pertain more broadly to other fields. For example, for a hospital-based RCT, “students” could be replaced by “patients” and “schools” could be replaced by “hospitals” or “practices.” For reference, Table 3 provides a glossary of key terms and definitions that link directly to the program input statements in the interface screens and the dictionary of program inputs in Table 2 from Chapter 1.

### a. Defining treatment and control status in RCTs

An RCT requires that treatment and control group designations be defined at the time of random assignment and *should never change*. Furthermore, *all* treatment and control group members should be included in the follow-up data collection and impact analysis. If a treatment group member does not receive intervention services (for whatever reason), that individual must remain in the treatment group for the analysis. Similarly, if a control group member receives barred intervention services (for whatever reason), that individual must remain in the control group for the analysis. Random assignment ensures that the full treatment and control groups are balanced at the time of randomization; changing treatment status designations or excluding sample members from the analysis based on events that occur *after* random assignment undermines the validity of an RCT design and must be avoided.

*The input data file for RCT-YES must contain an indicator variable that equals 1 for those randomly assigned to the treatment group and 0 for those randomly assigned to the control group.* This treatment status indicator variable must be available (nonmissing) for *all* observations in the data file or the program does not proceed. This requirement helps ensure that the data file does not include individuals who were not randomly assigned to a research condition, and hence, who are not part of the study sample.

In *RCT-YES*, users can label the treatment and control groups using the LABEL\_T and LABEL\_C options. The labels should not contain quotes. Importantly, the program will add “Group” to the end of the labels in the output tables. The default labels are “Treatment” and “Control”.

**Table 3. Glossary of key terms and definitions related to RCT-YES inputs and outputs**

Term	Definition
Randomized controlled trial (RCT)	A method for estimating average effects of interventions and policies where study participants are randomly assigned to research conditions, such as to a single treatment or control group or to multiple treatment groups
<b>Outcomes, impacts, covariates, and subgroups</b>	
Outcome	A result of a study participant that occurs <i>after</i> random assignment. In RCT-YES, outcome variables can be continuous or binary variables.
Impacts in nominal and effect size units	Impacts in nominal units are differences between the average outcomes of treatment and control group members. In RCT-YES, impacts in effect size units are impacts measured relative to the standard deviation of the outcome for control students.
Baseline covariate	A characteristic of a study participant that pertains to the period <i>prior</i> to or at the point of random assignment. In RCT-YES, covariates can be continuous or binary variables.
Subgroup	Study participants who share a common set of baseline characteristics. In RCT-YES, subgroups must be categorical.
Types of variables: continuous, categorical, and binary	Continuous variables can take on any value in a certain range (for example, achievement test scores). Categorical variables can take on values that are labels or names, and where there is no intrinsic ordering to the categories (for example, race/ethnicity categories). Binary variables are categorical variables that can take on two possible values, coded as 1 and 0 in RCT-YES.
<b>RCT designs and estimators</b>	
Non-blocked design	An RCT design where random assignment is conducted within a single population
Blocked design	An RCT design where random assignment is conducted separately within partitions of the entire sample (such as school districts or grades)
Matched pair design	A blocked RCT with one treatment and one control unit per block. In these designs, similar study units are paired using baseline covariates and one unit in each pair is then randomly assigned to the treatment group and the other to the control group.
Non-clustered design	An RCT where individuals are randomly assigned to a research condition
Clustered design	An RCT where groups—such as classrooms, schools, or districts—rather than individuals are randomly assigned to a research condition
Simple differences-in-means impact estimator	An estimate of the average intervention effect that is calculated as the difference between the average outcomes of the treatment and control groups
Regression-adjusted impact estimator	An impact estimator that controls for baseline covariates to improve the precision of the impact estimates and to adjust for baseline differences between the research groups due to missing data or random chance
Finite- and super-population models	The finite-population (FP) model assumes that the impact findings pertain to the study sample only, whereas the super-population (SP) model assumes that the impact findings generalize to a broader population of students and sites

## 2. Designs, analyses, and methods

Term	Definition
Causal average complier effect (CACE) or treatment-on-the-treated (TOT) estimator	An impact estimator that adjusts for treatment group members who did not receive intervention services and control group members who did receive barred intervention services (for whatever reason). These estimates pertain to intervention effects for those who “complied” with their treatment assignments.
<b>Standard errors, hypothesis testing, and baseline equivalence analyses</b>	
Standard error	A measure of uncertainty in the impact estimates due to random assignment and the actual or perceived sampling of study participants and sites from broader populations
t-test, p-value, statistical significance, and confidence interval	A t-test uses the estimated impacts and standard errors to statistically test the null hypothesis (the <i>default</i> position) that the average treatment effect is zero against the alternative that it differs from zero. The p-value evaluates how compatible the impact findings are with the null hypothesis of zero effects and can take on any value between 0 and 1. By default, <i>RCT-YES</i> labels impact estimates as statistically significant (not compatible with the null hypothesis) if the p-value is less than or equal to 0.05 (which can be changed using program options). A 95 percent confidence interval (for example) is a range of values that will have a 95 percent chance of containing the true impact.
Baseline equivalence analysis	A statistical analysis using t-tests and joint chi-squared or F-tests to examine whether the treatment and control groups are balanced using observable pre-intervention (baseline) covariates

### b. Defining outcomes in RCTs

In RCTs, outcomes are events that occur *after random assignment*. In the education context, outcomes can pertain to students, parents, educators, schools, or other entities. *RCT-YES* can estimate impacts on *continuous* outcomes (such as student achievement test scores or days absent) and *binary* (1 or 0) outcomes (such as whether or not a student graduated high school or was proficient in math). For any RCT, outcomes should be measured *the same way* for the treatment and control groups to avoid spurious findings that could arise from measuring outcomes differently for the two groups.

Importantly, in *RCT-YES*, outcomes are entered separately for each “outcome domain” pertaining to a specific class of outcomes for which common analyses are to be conducted. For example, an outcome domain could consist of test score outcomes or socio-emotional outcomes. This grouping helps minimize data entry, organizes the reporting of the impact findings, and facilitates hypothesis testing to adjust for the multiple testing problem (see Section 1f).

We strongly recommend that, if possible, the input data file should contain records for *all treatment and control group members, including those with missing outcome data*. *RCT-YES* can then calculate data response rates for each research group, which can provide important information on the extent to which the estimated impacts are biased due to missing data. Missing outcome data can be coded using standard missing data codes for the statistical package used for estimation (R or Stata). Organizations that conduct systematic reviews of the quality of evidence from impact evaluations typically require information on missing data rates.



### c. The four RCT design types considered by *RCT-YES*

*RCT-YES* estimates impacts for four types of RCT designs commonly used in education and other social policy research. These designs are defined by combining two key features. First, the designs are defined by the unit of randomization:

- *Non-clustered designs*, where individual students are randomly assigned to a treatment or control condition
- *Clustered designs*, where groups, such as classrooms, schools, or districts, are randomized to a research condition rather than individuals. Under these designs, all students within a group are assigned to the treatment or control status of their group. Clustered designs are common in education RCTs, because these studies often test interventions that are targeted to the group (for example, a school re-structuring initiative or professional development services for all teachers in a school). Clustered designs are also sometimes used to minimize the “spillover” of intervention effects from the treatment to control groups due to their interactions, which could lead to impact estimates that are biased.

Second, the designs in *RCT-YES* are defined by whether random assignment is conducted separately within blocks (strata):

- *Non-blocked designs*, where random assignment is conducted for a single population (for example, within a single school or school district)
- *Blocked designs*, where random assignment is conducted separately within non-overlapping subpopulations that comprise the entire sample. An example of a blocked design is a multidistrict RCT where students are randomly assigned separately within each district. Another example is where random assignment is conducted separately within demographic subgroups (for example, for girls and boys) to ensure treatment-control group balance for each subgroup. Another example is a longitudinal design where random assignment is conducted separately at different points in time (for example, incoming third graders in two separate years or student cohorts entering a program at different times of the year).

The four designs considered by *RCT-YES* combine these two key design features:

1. **Design 1:** Non-clustered, non-blocked designs
2. **Design 2:** Non-clustered, blocked designs
3. **Design 3:** Clustered, non-blocked designs
4. **Design 4:** Clustered, blocked designs

## 2. Designs, analyses, and methods

It is critical to understand the differences between these designs so that *RCT-YES* can produce correct impact findings. Users will likely generate incorrect results, for example, if they specify a non-clustered design when they actually have a clustered design.

To help clarify these designs, see Figure 1 which depicts Design 1, and Figures 2a and 2b which depict Design 3. In Figure 1, *students* are randomly assigned to a treatment or control group within a single school or school district. This design is *non-clustered* because students are the unit of randomization. It is *non-blocked* because random assignment is conducted for a single population. In contrast, in Figure 2a, *schools* are randomly assigned to a treatment or control group within a single school district. This design is *clustered*, because schools are randomized rather than students and is *non-blocked*. Figure 2b also shows a clustered, non-blocked design where *classrooms* are randomized.

In a *blocked* design (Design 2 or 4), the sample is partitioned and random assignment is conducted *separately* within each partition. Thus, to depict a *non-clustered, blocked* design (Design 2)—for example, a multisite RCT where students are randomized within each study site—we could imagine repeating Figure 1 for each site. Similarly, to depict a *clustered, blocked* design (Design 4), we could imagine repeating Figure 2a or 2b for each site. Thus, in blocked designs, a “mini-RCT” is conducted within each block and impact findings within each block are aggregated to yield overall results.

**Figure 1.** Depiction of a non-clustered, non-blocked RCT design where students are randomized in a single school district (Design 1)

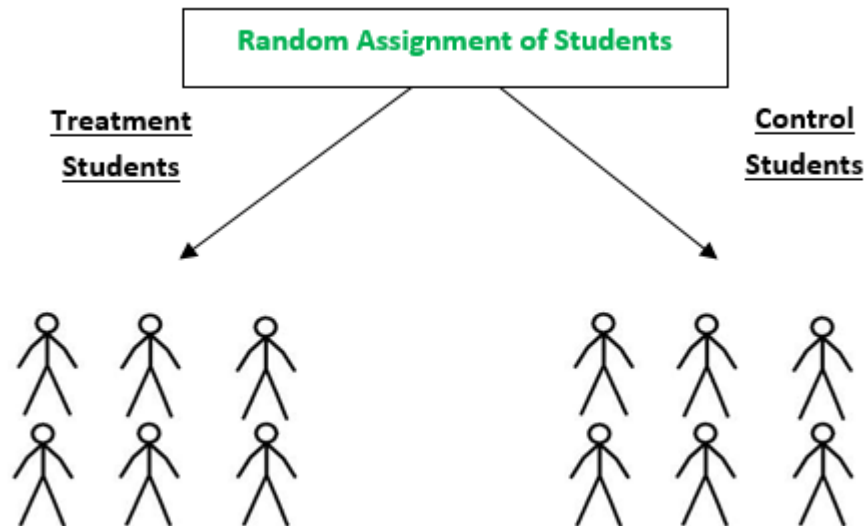


Figure 2a. Depiction of a clustered, non-blocked RCT design where schools are randomized in a single district (Design 3)

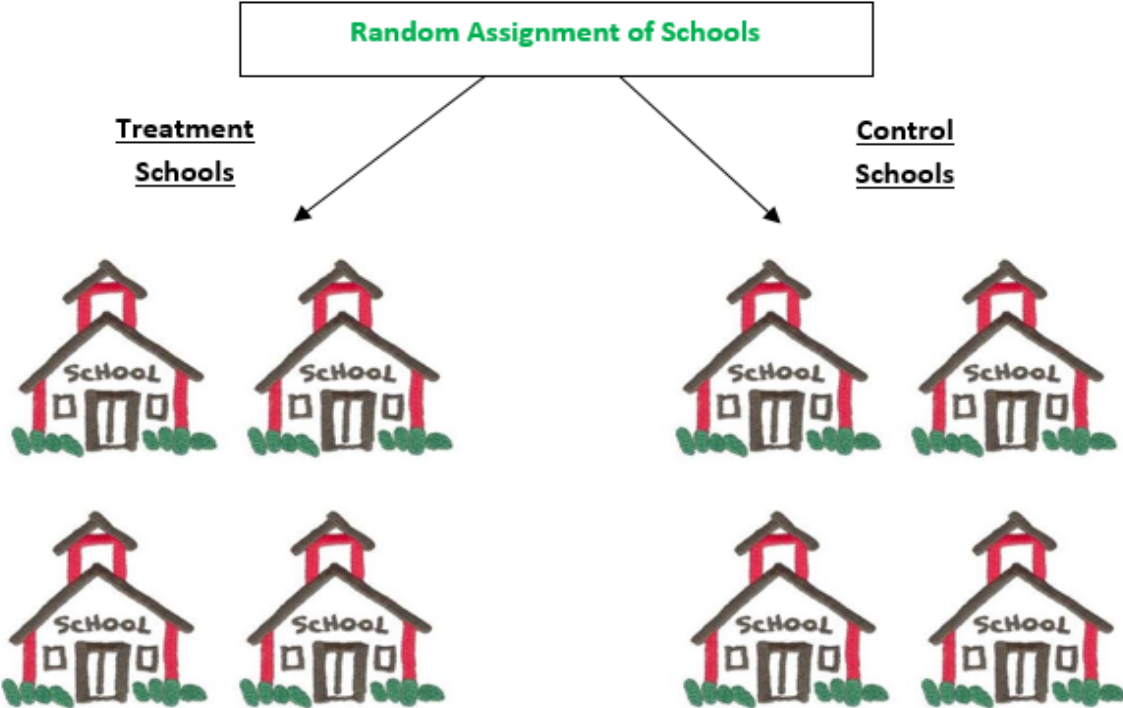
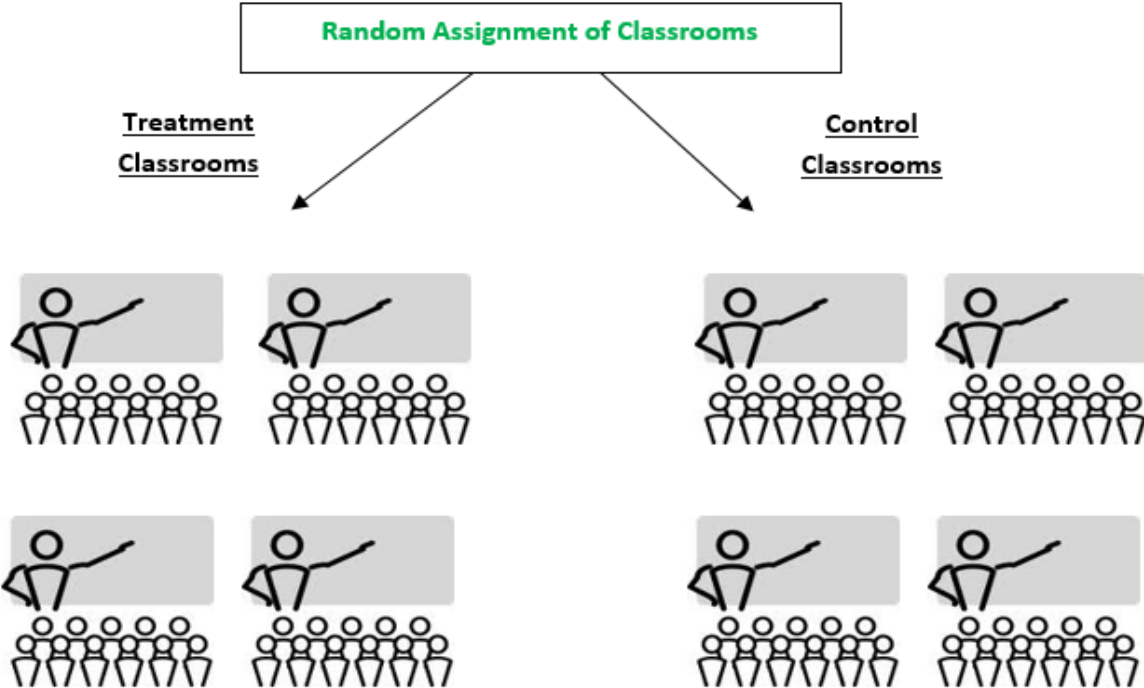


Figure 2b. Depiction of a clustered, non-blocked RCT design where classrooms are randomized (Design 3)



## 2. Designs, analyses, and methods

An important type of blocked RCT design is a *matched pair design* with one treatment and one control unit per block. For these designs, similar study units are paired using baseline data and one unit in each pair is randomly assigned to the treatment group and the other to the control group. Matched pair designs are common in education research (especially for clustered designs) when sample sizes are small. These designs help avoid the possibility of a “bad draw” where, for example, students with higher ability (as measured by prior year test scores) are, by chance, disproportionately assigned to one research condition. *RCT-YES* does not perform pairwise matching (or any type of randomization), but it can accommodate matched pair designs.

### d. Impact estimation methods with and without the use of baseline covariates

*RCT-YES* estimates impacts using two basic approaches.<sup>2</sup> First, by default, the program uses a “*simple differences-in-means*” impact estimator, which is calculated as the difference between the average outcomes of treatment and control group members.

Second, as an option, *RCT-YES* uses a “*regression-adjusted*” impact estimator that controls for baseline variables pertaining to the pre-randomization period. The use of baseline measures or “*covariates*” can improve the precision of the estimated impacts by explaining some of the variation in the outcome variables across the sample; they can also adjust for observable baseline differences between the treatment and control groups due to random chance or missing outcome data.

For an RCT, *covariates in the regression models must pertain to the period before random assignment*. In *RCT-YES*, the covariates can be continuous variables (for example, the student’s achievement math test score in the year prior to random assignment) or binary (1 or 0) variables (for example, a female indicator or a set of indicators for race/ethnicity categories). The covariates can pertain to students, classrooms, schools, or other entities. As with the outcomes, baseline covariates should be measured similarly for the treatment and control groups. Missing values for covariates can be coded using standard missing data codes for the statistical package used for estimation (R or Stata).

For an RCT, it is sound research practice to include only a *small number of highly predictive baseline covariates* in the regression models (that are specified prior to the analysis). The literature has shown that models that include pre-intervention measures of outcomes as covariates are often strong predictors. These designs are known as “pretest-posttest” designs. By default, to help limit the number of specified covariates, *RCT-YES* requires that the data file contain at least 5 students per covariate for non-clustered designs and 5 clusters per covariate for clustered designs or all covariates are excluded from the analysis; these defaults can be changed using the `OBS_COV` program option.

---

<sup>2</sup>These statistical methods are discussed in detail in the *RCT-YES* technical report (Schochet, 2015), which includes all mathematical formulas and proofs derived from design-based theory.

We recommend that if *RCT-YES* users estimate impacts using regression methods with covariates, they should also estimate impacts using simple differences-in-means methods. The two sets of estimates should be carefully compared and large differences should be resolved.

### Covariates to Avoid

- **Block indicator variables for blocked designs (Designs 2 and 4).** The R and Stata computer programs will automatically include block indicators in the models, so *RCT-YES* will drop block indicators if they are specified as covariates.
- **Cluster indicator variables for clustered designs (Designs 3 and 4).** It is incorrect to adjust for clustering by including indicators of cluster membership as covariates in the models. *RCT-YES* will drop these indicators if they are included as covariates.
- **Subgroup variables for the associated subgroup analysis.** For example, if a subgroup analysis examines impacts by gender, users should not include a binary gender covariate that equals 1 for females and 0 for males, because the program includes these indicators.
- **Variables formed by interacting (multiplying) the treatment status indicator variable with other covariates.** The inclusion of these interactions could lead to incorrect impact estimates and should be avoided.
- **Binary variables for each level of a categorical variable.** For example, if a categorical variable for the age of the teacher has three levels (1 = younger than 30, 2 = 30 to 55, and 3 = older than 55), users should not include in the model three binary variables corresponding to each teacher age category. Instead, users should only include two of the three binary variables to avoid collinearity issues (that is, the “dummy variable trap”), although *RCT-YES* will drop one variable if needed.

Importantly, users should examine the .log files presenting the regression results to see which covariates were included in the models and those that were excluded due to collinearity or other reasons. Tables 6 and 7 in the .html files also provide summary information on the covariates that entered the models, but not collinear covariates that were subsequently dropped from the models.

**If needed, users should respecify the model covariates and rerun the models.**

## 2. Designs, analyses, and methods

### Methods for blocked designs

For blocked designs (Designs 2 and 4), by default, *RCT-YES* first estimates impacts within each block and then averages these impacts to obtain overall (pooled) impact estimates. Either simple differences-in-means or regression methods can be used for estimation.

*RCT-YES* does not report impact estimates for each block due to data disclosure concerns that could arise for small blocks (see Chapter 3). Instead, for the default model, the program reports summary statistics on the block impact estimates so that users can examine the extent to which they vary. Understanding this variation is important: study findings could have different policy implications if impact estimates are similar across blocks or vary considerably across blocks.

By default, *RCT-YES* only includes blocks in the analysis with at least 2 treatment and 2 control group observations (where observations are individuals for Design 2 and clusters for Design 4) so that standard errors can be calculated in each block. If users are concerned that this requirement will exclude too many blocks from the analysis, *RCT-YES* has a program option (`BLOCK_FE=1`) where the estimation model controls for main block effects but does not control for “block-by-treatment status” interaction terms (see Schochet, 2016). This approach does not fully conform with the statistical theory underlying blocked RCT designs, but has the advantage that it requires only 1 treatment and 1 control group record per block and may be a parsimonious specification for studies with many small blocks. Alternatively, users can specify the `SUPER_POP=1` default option (see Section 1j below) where the study blocks are considered to be random samples from a larger population of blocks rather than fixed for the study; this specification also requires only 1 treatment and 1 control group member per block (and is the default specification for matched pair designs).

Importantly, if the data contain many blocks, *RCT-YES* will invoke the `SUPER_POP=1` default option. This procedure overcomes potential size constraints in R and Stata (and user operating systems) on the number of right-hand side variables that can be included in the estimation models. *RCT-YES* will invoke the `SUPER_POP=1` option for a particular analysis if the total number of model covariates ( $2bs + x$ ) is determined to be too large (more than 200), where  $b$  is the number of blocks,  $s$  is the number of subgroup levels for a subgroup analysis (and 1 for the full sample analysis), and  $x$  is the number of baseline covariates. ***There may be instances, however, where *RCT-YES* may not run successfully if the data contain many blocks and subgroups with many categories. In these cases, users can reduce potential problems by (i) specifying the `SUPER_POP=1` option and (ii) excluding baseline covariates from the estimation models.***

### Methods for clustered designs

For clustered designs (Designs 3 and 4), *RCT-YES* can accommodate data in two formats. First, the input data file can contain records for individual students. In this case, clusters are included in the analysis if they contain at least 1 student with available outcome data. Second, the program can use

data that have already been averaged to the cluster level (for example, average student test scores in the school). For either format, the program calculates impact estimates using cluster-level averages using either simple differences-in-means methods (the default) or regression methods (an option).

### e. Estimating impacts on the offer and receipt of intervention services

By default, *RCT-YES* provides impact estimates on the *offer* of intervention services, which is known in the literature as an “intention-to-treat (ITT)” analysis. The offer of services, however, can differ from the receipt of services. This can occur because some treatment group members might not receive intervention services either by choice (for example, a student decides not to attend the after-school program) or due to problems with intervention implementation (for example, some teachers might never receive the new professional development training). Similarly, some controls might receive intervention services due to inadvertent errors or for other reasons. Thus, by comparing the outcomes of all treatments to all controls, the ITT estimator combines the outcomes of treatments who received intervention services with treatments who did not, and similarly for controls.

If data are available on the receipt of intervention services for sample members, *RCT-YES* can also estimate intervention effects that statistically adjust for treatments who did not receive intervention services and controls who did receive barred intervention services. This estimator—known in the literature as the “complier average casual effect (CACE)” estimator—pertains to “compliers” who would receive intervention services as a treatment but not as a control. To request a CACE analysis, *RCT-YES* users can specify up to two intervention receipt variables per domain—that could measure different dimensions of service receipt or dosage—using the *GOT\_TREAT* input variable. A separate analysis is conducted for each one. The variables can differ by domain, but will often be the same.

It is important to recognize that the ITT parameter pertains to real-world intervention effects, because non-exposure or partial exposure to treatment services is likely to occur for most interventions if they were to be rolled out more broadly. The CACE parameter, however, is important for understanding the “pure” effects of the intervention for those who received a meaningful dose of intervention services, especially for efficacy studies that aim to assess whether the studied intervention *can* work. Decision makers may also be interested in the CACE parameter if they believe that intervention implementation could be improved in their sites. Furthermore, the CACE parameter can be critical for drawing policy lessons from ITT effects; for instance, the CACE parameter can distinguish whether a small ITT effect is due to low rates of service receipt or due to small impacts among those who received intervention services.

### f. Standard errors and hypothesis testing

For any RCT, impact estimates are *measured with sampling error*. This occurs because the treatment and control groups are both random samples created from the combined groups. There are many possible allocations of study participants to the treatment and control groups. If it were possible to

## 2. Designs, analyses, and methods

carry out the experiment for each allocation, the estimated impacts would almost surely differ for each one. Of course, in reality, we can only carry out the experiment for a single treatment-control group allocation, and thus, we can only obtain one estimate of the intervention effect. *RCT-YES* calculates *standard errors* to reflect this uncertainty in the impact estimates.

It is critical that *RCT-YES* users correctly specify their design type to obtain correct standard errors. Standard errors typically are *larger* for clustered designs than non-clustered designs (for a given student sample size). This inflation occurs because students within the same cluster tend to have similar outcomes due to shared experiences and environments and share the same treatment status (Schochet, 2016). In contrast, standard errors typically are *smaller* for blocked designs than non-blocked designs if the blocking is based on characteristics associated with the outcomes of interest.

*RCT-YES* uses the estimated impacts and standard errors to conduct “t-tests” of the null hypothesis (default position) that the average treatment effect is zero against the alternative hypothesis that it differs from zero. *RCT-YES* applies a two-tailed test for hypothesis testing (it is agnostic about whether the intervention will change outcomes in a positive or negative direction), and, by default, it uses a 5 percent significance level (which can be changed using the ALPHA\_LEVEL option). *RCT-YES* reports p-values from the hypothesis tests and attaches the symbol \* to the p-value for impact estimates that are “*statistically significant*” (that is, when p-values are less than or equal to the alpha level cutoff, which is 5 percent by default). Confidence intervals are not reported in the output .html tables, but are provided in the .csv file, and thus, can be plotted using the *RCT-YES-Graph* app.

If users specify multiple outcomes, *RCT-YES* also indicates whether findings remain statistically significant after applying the Benjamini and Hochberg (1995) procedure, which adjusts p-values to account for the multiple hypothesis tests that are being conducted for the full sample analysis. This adjustment reduces the likelihood of finding a spurious significant result for one or more outcomes by chance, which is the “multiple comparisons” problem (see Schochet, 2016 for a detailed discussion of the problem and adjustment methods used in *RCT-YES*). These adjustments are made for all outcomes within an outcome domain, but not across outcome domains.

### **g. Subgroup analyses**

In education RCTs, researchers often estimate impacts for subgroups defined by *student, teacher, school, and local area characteristics measured prior to random assignment*. For instance, researchers may be interested in assessing whether intervention effects differ by gender, test score levels in the prior year, student’s eligibility status for free and reduced-priced meals, teacher credentials, or rural/urban status. These analyses can be used to assess the extent to which intervention effects vary across policy-relevant groups. Results from subgroup analyses can help inform decisions about how to best target specific interventions, and possibly to suggest ways to improve the design or implementation of the tested interventions.



*RCT-YES* conducts subgroup analyses for *categorical* subgroups in which each sample member is allocated to a discrete, mutually exclusive category (for example, 1=student was not proficient in math in the prior year; 2=student was proficient in math; and 3=student was highly proficient in math). Subgroup variables for *RCT-YES* can have numeric or character codes. Subgroups for *RCT-YES* cannot be continuous variables (for example, prior year test scores), but users can redefine such variables as categorical subgroup variables for the analysis. Subgroup variables should be measured in a similar way for the treatment and control groups.

*RCT-YES* estimates impacts for subgroups using similar methods that are used to estimate impacts for the full sample. For example, to estimate impacts for girls, the program compares the average outcomes of girls in the treatment and control groups. The program also conducts statistical tests of whether impact estimates *differ* across subgroups (for example, for boys and girls). It is sound research practice to downplay significant findings for individual subgroups if there is no evidence that subgroup estimates differ from each other. This is especially important if policymakers aim to use evaluation findings to target services to those who can most benefit from them.

For Designs 3 and 4 and some Design 2 specifications, by default, *RCT-YES* conducts tests of differences in impact estimates across subgroups accounting for the potential correlations of outcomes or impacts for observations in the same cluster or block. If the number of clusters or blocks is small, the covariance terms and test statistics can become unstable, yielding implausibly large or small values. In these cases, users can specify the `NO_COV_SG=1` option to exclude the covariance terms, which will likely lead to conservative test statistics (that is, upper bounds on p-values).

It is prudent to define, prior to the analysis, only a small number of key, policy-relevant subgroups that align with the study's conceptual model, and to avoid ex post "fishing" for positive subgroup findings. Conducting the analysis on many subgroups will likely generate statistically significant findings for some subgroups, but these findings could be spurious and nonreplicable.

### **h. Baseline equivalence analyses**

An RCT design ensures that the treatment and control groups are balanced (in expectation) at the time of random assignment. This feature allows researchers to learn whether an intervention *causes* improvements in outcomes. While treatment-control group balance is ensured in theory, it is not ensured in practice. For example, missing data or problems with the procedures used to conduct random assignment could lead to imbalances.

To strengthen the credibility of RCT findings, it is good research practice to demonstrate baseline equivalence of the treatment and control groups. As an option, *RCT-YES* conducts such analyses using t-tests for each specified baseline variable (which can be continuous or binary). The analysis is conducted separately for each outcome variable using the sample that has data for that outcome.

## 2. Designs, analyses, and methods

RCT-YES also tests the hypothesis that the baseline variables are *jointly* equivalent for the treatment and control groups. The joint test will only be conducted if the number of observations is large relative to the number of covariates (based on the OBS\_COV value). ***But even if this condition is met, the joint test of baseline equivalence can lead to program errors if the number of baseline covariates is very large. If users encounter errors, they should reduce the number of baseline covariates or specify the NO\_JNT\_TEST=1 option to suppress the joint test.***

### i. Handling missing data

An important analytic issue for any RCT is how to handle missing data. In this section, we discuss RCT-YES procedures for treating missing data for outcomes, covariates, and subgroups.

#### Missing outcome data

For an RCT, researchers should attempt to collect outcome data for all those randomized. In education RCTs, typical sources of outcome data are administrative records, surveys, and assessments. In practice, however, data might not be available for some sample members—that is, there may be “*data nonresponse*.” For example, test scores might not be available for students absent on the day of the test. Similarly, survey data might be missing for students who move out of the study districts and cannot be located or for those who refuse to answer specific survey items. Missing outcome data can bias the impact estimates if the factors generating the missing data are related to intervention effects.

By default, RCT-YES estimates impacts ***using only observations with nonmissing values for the outcome under investigation***. This method is known as “*case deletion*.” The program does not impute (fill in) outcomes for those with missing data. Case deletion yields unbiased impact estimates if the missing data mechanisms are random for both the treatment and control groups.

RCT-YES uses the case deletion approach for several reasons. First, case deletion is easy to apply and understand. Second, data response rates and mechanisms are typically similar for treatments and controls. Finally, Puma, Olsen, Bell, and Price (2009), using simulated test score data, found that case deletion performs reasonably well relative to other missing data methods for education RCTs.

RCT-YES users can include *weight variables* in the input data file to adjust for data nonresponse. For instance, researchers often create nonresponse weights by estimating statistical models in which an indicator of data nonresponse is regressed on baseline covariates, and weights are then constructed using predicted probabilities from these models. This approach assigns large weights to sample members with baseline characteristics that are associated with high nonresponse rates. The use of these weights in the analysis will yield unbiased impact estimates if the missing data process is random for those with similar baseline covariate values. In RCT-YES, users can specify separate weights for different outcomes and subgroups to adjust for differences in patterns of missing data.

Missing data can threaten the validity of RCT findings. Thus, for any RCT, it is important to conduct “nonresponse analyses.” These analyses could include examining response rates for the full sample and for key subgroups, and comparing the baseline characteristics of respondents and nonrespondents.<sup>3</sup> In addition, it is important to conduct “sensitivity analyses” to examine changes in the impact findings using alternative methods for handling missing outcome data. For instance, RCT-YES users can compare impact findings using simple differences-in-means and regression methods, and adopting various approaches for imputing outcome data in the input data file (see, for example, the methods discussed in Puma et al, 2009; Rubin, 1987; and Schafer, 1997). Differences in impact findings using alternative methods for handling missing data could indicate the presence of data nonresponse biases.

Finally, for CACE analyses, RCT-YES excludes observations that have missing values for the intervention service receipt variables specified in the GOT\_TREAT input variables. If CACE analyses are specified, both the ITT and CACE analyses are conducted using observations that have nonmissing data for *both* the outcome and intervention receipt variables under investigation to align the impact findings from the two sets of analyses.

### Missing covariate data

The RCT-YES approach to adjust for missing covariates for the regression models depends on the prevalence of missing data:

1. ***The covariate is missing for 30 percent or fewer cases for both the treatment and control groups.*** In this case, the program replaces missing values of covariates with the covariate average calculated from the sample with nonmissing covariate values, separately for the treatment and control groups. The replacements are done separately for each specified outcome variable (which may have different percentages and patterns of missing data) and for both continuous and binary variables (including binaries that comprise levels of a categorical variable). If pertinent, nonresponse weights are used for the imputations.
2. ***The covariate is missing for more than 30 percent of cases for either research group.*** In this case, the covariate is dropped from the regression model.

The 30 percent missing data cutoff rule is consistent with results from the data nonresponse analysis conducted by IES’s What Works Clearinghouse (2014). The 30 percent cutoff can be changed using the MISSING\_COV program option.

---

<sup>3</sup>RCT-YES users can compare the baseline characteristics of respondents and nonrespondents by creating a variable equal to 1 for respondents and 0 for nonrespondents and specifying this variable as the treatment indicator in the program inputs. This analysis can be conducted separately for the treatment and control groups.

## 2. Designs, analyses, and methods

If desired, users can instead adopt an approach where (i) missing covariate indicator variables are included as model covariates and (ii) missing covariate values are replaced with a constant (for example, 0). Jones (1996) discusses limitations of this approach and how it can yield biased estimates.

A somewhat different approach is used for the baseline equivalence analysis. To analyze baseline equivalence for a specific covariate, *RCT-YES* drops observations that have missing data for that covariate. It does not use averages to replace missing covariate values.

### Missing subgroup data

For subgroup analyses, *RCT-YES* excludes cases that have missing values for the categorical subgroup variables. For example, if gender is missing for an individual, *RCT-YES* will exclude this observation from the analysis when estimating impacts for boys and girls (even if that observation has available outcome data). Thus, samples for the full sample and subgroup analyses could differ due to missing subgroup data (that is, impacts for boys and girls might not average to the full sample impacts).

### j. Finite-population and super-population models

A key but subtle issue for the statistical theory underlying *RCT-YES* is the distinction between the *finite-population (FP)* and *super-population (SP) models*, which can affect the impact estimates and their standard errors. Under the FP model, impact findings are interpreted to apply only to the students, schools, and districts in the study and not to the broader population. This interpretation has merit when study samples are *purposively selected* for RCTs, which could occur, for example, if site participation depends on a site's willingness to participate or its suitability for the study based on its population and context, or if the student sample includes only those whose parents or guardians provided written consent for their children to participate in the study.

In contrast, under the SP model, the sample is assumed to be a *random sample* from a larger population. In this scenario, impact findings are interpreted to be estimates that apply to a broader population of students, schools, and sites "similar" to those in the study. The SP framework is appropriate, for instance, if study sites are actually randomly sampled from a larger population or are deemed to be representative of a larger population (perhaps because they are geographically dispersed or serve a range of students who could potentially be targeted for the intervention). The SP framework could also be justified if the study provides a primary basis for deciding whether to implement the tested interventions more broadly. Hierarchical linear model (HLM) methods (Raudenbush and Bryk, 2002), which are commonly used in education research to analyze RCT data, are SP models.

***The default specification in RCT-YES is the FP model for all designs except matched pair designs.***

The SUPER\_POP=1 program option, however, can be specified to estimate the SP model for all designs. In addition, there are multiple SP specifications for Designs 2 to 4 that can be requested

using the CATE\_UATE option. These specifications depend on researcher assumptions about the sampling of blocks, clusters, and/or students from broader populations. The default SP specification assumes random sampling at all stages (this is known as the population average treatment effect [PATE] parameter). For example, for Design 2, random sampling at all stages means that both blocks and students within blocks are assumed to be random samples. Alternatively, it is possible to allow random sampling at some stages but not others. For example, for Design 2, researchers can assume the random sampling of blocks but not students (the unit average treatment effect [UATE] parameter; CATE\_UATE=2) or vice versa (the cluster average treatment effect [CATE] parameter; CATE\_UATE=1). Schochet (2016) discusses these various SP parameters in detail.

### k. Weighting

RCT-YES can accommodate weights to adjust for data nonresponse or other design-related factors. If users are interested in using weights, they must include weight variables in the input data file and specify them using the WEIGHT option. A separate set of weight variables can be specified for each outcome measure. Weights specified for full sample analyses are pre-filled for subgroup analyses, but users can override these weights with subgroup-specific ones. *If weights are specified, RCT-YES requires weights to be nonmissing for all observations with available outcome data* or the weights are ignored.

A potentially important use of weights arises in clustered or block designs. By default, RCT-YES weights individuals equally for non-clustered designs (Design 2) and weights clusters equally for clustered designs (Designs 3 and 4). Similarly, RCT-YES weights blocks by their total numbers of students for Design 2 and by their total numbers of clusters for Design 4.

Weight variables can be included in the data file to override these defaults. For example, users may want to weight clusters by the number of students in the cluster rather than equally. Similarly, users may want to weight blocks equally in designs where blocks are sites, especially if some sites are much larger than other sites and heavily influence the impact estimates. In addition, users may want to input a different set of weight variables for the optional SP model than the default FP model. For instance, weights for the SP model might reflect the size of the broader student population in the blocks or clusters included in the study.

### l. Reporting and graphing impact findings

When users run the R or Stata computer program file, an .html file (named in the interface input screens) will be produced that presents analysis results in formatted tables. The output file will report three types of information: (i) program errors; (ii) summary statistics on the outcomes, subgroups, covariates, and weights to help users assess data quality; and (iii) findings from the baseline equivalence and impact analyses. These tables can be printed. The LIMIT\_PRINT option can be specified to suppress the printing of some output tables presenting detailed summary statistics on the study samples which can be long for some designs.

## 2. Designs, analyses, and methods

*RCT-YES* was designed to produce only limited information on data quality to avoid flooding users with a large amount of output. Thus, it is imperative that users examine the quality of the input data before running the program.

The output file will report findings from the baseline equivalence and impact analyses for each specified outcome and subgroup. For these analyses, *RCT-YES* presents the *control group mean for the sample* and the *adjusted treatment group mean* calculated as the sum of the unadjusted control group mean and the impact estimate. For clustered designs, the control group mean is calculated using cluster-level averages. All calculations are conducted using the specified or default weights.

*RCT-YES* will report treatment-control differences in both nominal (unadjusted) and “*effect size*” units. An effect size is a measure of the size of the intervention effect relative to a benchmark. Different benchmarks can be used. The benchmark used in *RCT-YES* is the standard deviation of the outcome calculated using the sample.<sup>4</sup> However, the `STD_OUTCOME` option can be invoked if users want to specify their own standard deviation. The reporting of impacts in effect size units is becoming increasingly popular in evaluation research to facilitate the comparison of impact findings across outcomes that are measured on different scales.

If CACE analyses are specified, *RCT-YES* reports separate tables for the ITT and CACE analyses. Both analyses are conducted using the sample with nonmissing data for both the outcome and service receipt variables under investigation so that the two analyses can be compared.

The R or Stata computer program will produce a .csv data file containing information from the output tables (unless the `CSV_FILE` option is set to 0). This .csv file can be read in by computer programs for further analyses and reporting. The R or Stata computer program will also produce a .log file with detailed results from the impact estimation models.

Finally, the impact findings can be graphed using the *RCT-YES-Graph* application (see Chapter 7). The app reads in the .csv data file to create the graphs.

### **m. Summary of designs in *RCT-YES***

It is important that *RCT-YES* users correctly specify their design type and understand the associated input data requirements and default specifications for their design. Table 4 summarizes key features of the four RCT designs in *RCT-YES* based on the topics discussed in this chapter. For each design, the table provides information on the unit of random assignment, blocking information, data requirements, and key program default specifications for the impact analysis.

---

<sup>4</sup> The standard deviation is calculated using control group individuals for the impact analysis and using the combined treatment and control group individuals for the baseline equivalence analysis.

**Table 4. Summary of designs in RCT-YES**

Design	Unit of random assignment	Blocking	Data requirements and key default <i>RCT-YES</i> specifications for ATE estimation
<b>Design 1:</b> Non-clustered, non-blocked	Students or other individuals	None	<p>Input data requires one record per observation, and outcome data for at least 10 treatment (T) and 10 control observations (Cs)</p> <p>Deletion of cases with missing values for the considered outcome</p> <p>Simple differences-in-means estimator</p> <p>Finite population (FP) model</p>
<b>Design 2:</b> Non-clustered, blocked	Students or other individuals	Districts, schools, classrooms, matched pairs, demographic groups, cohorts	<p>Input data requires one record per observation and sample size requirements the same as for Design 1 after removing small blocks</p> <p>Blocks are included if they contain at least 2 T and 2 C observations with outcome data; at least 1 T and 1 C observation is required for the super-population (SP) model option (with CATE_UATE=0 or 2) and the FP model with the BLOCK_FE=1 option</p> <p>Deletion of cases with missing values for the considered outcome</p> <p>Simple differences-in-means estimator within each block; blocks are weighted by their student sample sizes to obtain overall impact estimates</p> <p>FP model, except for matched pair designs</p>
<b>Design 3:</b> Clustered, non-blocked	Districts, schools, classrooms, etc.	None	<p>Input data requires one record per observation or one record per cluster average, with Design 1 sample size requirements and at least 2 T and 2 C clusters</p> <p>Clusters are included if they contain at least 1 observation with outcome data</p> <p>Deletion of cases with missing values for the considered outcome</p> <p>Simple differences-in-means estimator using cluster averages; clusters are weighted equally to obtain overall impact estimates</p> <p>FP model</p>
<b>Design 4:</b> Clustered, blocked	Districts, schools, classrooms, etc.	Districts, schools, matched pairs, demographic groups, cohorts	<p>Input data requirements combine those from Design 2 for blocks and Design 3 for clusters</p> <p>Simple differences-in-means estimator using cluster averages; clusters are weighted equally to obtain block estimates, and blocks are weighted by their number of clusters to obtain overall impact estimates</p> <p>FP model, except for matched pair designs</p>





## 3. Minimizing disclosure of personally identifiable information

In reporting results from education RCTs, researchers must consider the protection of personally identifiable information (PII) on students and educators. For some data sources, this protection is mandated by law. For example, the Family Educational Rights and Privacy Act (FERPA) legally requires PII protection for student education records. In general, RCT findings should only be reported for subgroups that are sufficiently large and for outcomes that have sufficient variation across the sample so that it is not possible for someone to infer sensitive information for an individual student (such as an achievement test score). Two Technical Briefs published by the National Center for Education Statistics (NCES) provide a detailed discussion of data disclosure issues for the reporting of statistics using state longitudinal data system (SLDS) data (NCES 2011-601, November 2010; NCES 2011-603, December 2010).

It is very difficult to develop a computer program that can prevent PII disclosure in all instances. Thus, *RCT-YES* users will need to carefully assess which impact findings can be reported in their own contexts. *RCT-YES*, however, employs several key features to help minimize data disclosure risks. First, the program provides descriptive statistics on all outcomes, subgroups, covariates, and weights that are listed as inputs into the program, and provides formatted tables that indicate data problems (for example, outcomes or subgroups with small sample sizes). Users can use this information to update the input data files and program specifications.

Second, the program uses several criteria for excluding outcomes, subgroups, and baseline covariates from the analysis and for reporting specific impact findings. These criteria follow some of the best reporting practices specified in a Technical Brief published by NCES on statistical methods for PII protection in the aggregate reporting of SLDS data (NCES 2011-603, December 2010). These criteria include:

- ***Omitting outcomes, subgroups, and baseline covariates that have small numbers of students with available data.*** Individual states have adopted minimum group size rules for reporting SLDS outcomes to prevent PII disclosure. Most states have set this minimum group size to be 10 students (the default in *RCT-YES*), but in 2010, the minimum number ranged from 5 to 30. This threshold value can be set using the `MIN_NUM` input variable in *RCT-YES* (it must be at least 3). The program checks that the minimum size threshold holds for the treatment group and separately for the control group. Stated differently, by default, the program checks that there are at least 10 treatment and at least 10 control group members with available data.
- ***Omitting the entire subgroup if any subgroup category is too small.*** If any subgroup category has fewer than the minimum number of students from above, the entire subgroup is omitted from the analysis. For instance, to examine impacts for race/ethnicity categories,

### 3. Minimizing disclosure

if one category has too few sample members (for example, Pacific Islanders), the program omits *all* race/ethnicity categories from the analysis. This procedure is used because knowledge of the outcomes from the larger subgroups and for the full sample can be used to calculate the outcomes of students in the small subgroups. In these cases, users should combine small subgroup categories into larger ones.

- ***Omitting outcomes and baseline covariates that do not have sufficient variation.*** RCT-YES conducts analyses using only outcomes and covariates whose values vary across the sample; this condition must hold for both the treatment and control groups. The program excludes variables that have zero variance (this removes outcomes that all have the same value). In addition, RCT-YES excludes binary outcomes or covariates where there are fewer than 5 observations with a value of 0 or fewer than 5 observations with a value of 1 for either the treatment or control group.
- ***Not reporting impact findings for individual blocks (for example, sites) or mean outcomes for individual clusters (for example, schools).*** The concern is that student sample sizes in some blocks or clusters might be small, which could lead to data disclosure issues. RCT-YES, however, produces summary statistics on impact estimates across blocks so that users can examine the variation in the block-specific impact findings.
- ***Reporting findings for binary outcomes by multiplying them by 100 and reporting them as whole numbers without decimals.*** This procedure can help guard against data disclosure for binary variables with means near 0 or 100 percent.

The program does not mask variables (by hiding original data with random numbers or characters) or top- or bottom-code continuous variables (by setting maximum or minimum data values), because the goal of the program is to generate impact estimates that are transparent and replicable.

Finally, to address potential PII concerns, while the RCT-YES interface requires that users provide the name and location of the input data file, the interface never reads the file. Rather, the input dataset is only read when the user runs the R or Stata program produced by the interface.

## 4. Creating the input data file

This chapter first provides an overview of the structure of the input data file for Designs 1 to 4 and then provides examples. We use the terminology and concepts for RCTs presented in Chapter 2.

### a. Overview of the input data file structure

The format of the input dataset for the statistical analysis must conform with the statistical package used for estimation: **a .rds file for R or a .dta file for Stata**. All missing data should be coded using standard missing data codes for the language used. Users can create their data file in any language and convert it to an R or Stata file before running the program. Appendix A discusses how .rds files in R and .dta files in Stata can be created from text files saved as .csv files in Microsoft Excel.

For *non-clustered* designs (Designs 1 and 2), RCT-YES requires individual-level data with one record per individual. Individuals will typically be students, but they could also be teachers or principals if the intervention targets educators and their outcomes (for example, an intervention that provides mentors for new teachers to help improve their teaching practices and retention).

For *clustered* designs (Designs 3 and 4), RCT-YES can accommodate data in two formats: (i) individual-level data or (ii) data averaged to the cluster level (for example, school test scores for students in the sample). For the latter format, the input data file should contain a separate set of stacked cluster-level averages for the full sample analysis and for each specified subgroup analysis (see pages 41-43).

For clustered designs, it is preferable that the data file contain individual-level records so that the full complement of analyses can be conducted. However, the program allows data to be provided at the cluster level for several reasons. First, for studies relying on administrative SLDS data, it may be easier for agency staff to provide data in this format. Second, requesting SLDS data as cluster-level averages might help minimize the disclosure of PII, thereby facilitating data requests (see Chapter 3). Finally, education researchers sometimes collect outcome measures from data sources that are available only at the cluster level, such as the Common Core of Data (CCD).

The input data file does not need to include student identifiers (such as name, address, or date of birth). However, the data file must contain block and/or cluster identifiers for Designs 2, 3, and 4 for all student observations or the program does not proceed. Importantly, these identifiers could be *masked* so as not to reveal the specific names or locations of blocks or clusters in the sample.

The input data file for RCT-YES must contain an indicator variable that equals 1 for those randomly assigned to the treatment group and 0 for those randomly assigned to the control group. This treatment status indicator variable must be available (nonmissing) for *all* observations in the data file or the program does not proceed. For designs with more than two research groups, pairs of research conditions can be compared to each other in separate runs of RCT-YES. For example, to compare

#### 4. The input data file

the outcomes of two treatment groups in a multi-group RCT (say, T1 and T2), users could create a treatment group indicator that equals 1 for those assigned to T1 and 0 for those assigned to T2, and run *RCT-YES* using a dataset that includes only T1 and T2 cases (excluding other research groups).

The data file should contain data on each specified outcome variable. It is recommended that, if possible, the data file should contain records on *all* treatment and control group members, including those with missing outcome data. The availability of data on all sample members allows the program to calculate data nonresponse rates to help users assess potential biases of the impact findings due to missing data. To estimate impacts for a particular outcome, *RCT-YES* excludes from the analysis observations with missing values for that outcome.

The input data file can include weights to adjust for data nonresponse or other design-related factors. If weights are specified, *RCT-YES* requires that weights be nonmissing for all those with available outcome data or the program ignores the weights.

If users are interested in conducting subgroup analyses, the file must contain *categorical* subgroup variables where each sample member is assigned to a discrete, mutually exclusive category (for example, 1=not proficient in math in the prior year; 2=proficient; and 3=highly proficient). The subgroup variables can have numeric or character codes. These subgroup variables should pertain to the period before random assignment and should be measured similarly for treatments and controls. Individuals with missing data on subgroup membership are excluded from the subgroup analysis.

If users are interested in obtaining regression-adjusted impact estimates or conducting baseline equivalence analyses, the data file must contain data on each specified baseline covariate. The baseline covariates can be continuous or binary and should pertain to the pre-randomization period. The covariates can differ for the regression and baseline equivalence analyses. Methods for treating missing baseline covariates were discussed in Chapter 1i.

#### **b. Examples**

This section provides examples of the structure of the input dataset for Designs 1 to 4 using simulated data from a hypothetical RCT of an after-school program with an academic focus. These same examples are used throughout this manual. Users can replicate the examples by downloading the *AFTER\_SCHOOL\_RCT\_DATA* data files (in R or Stata format) from the *RCT-YES* website.

##### **Design 1**

Design 1 pertains to an RCT where individuals are randomly assigned to a treatment or control group within a single population. To demonstrate the data file for this design, we consider a hypothetical example with a sample of 2,256 students, about half of whom are randomly assigned to a treatment group and the remainder to a control group at the start of the school year. The 1,073

students in the treatment group are offered the opportunity to attend an after-school program where they receive academic instruction and other services, whereas the 1,183 students in the control group cannot attend the program (although they can enroll in alternative programs in their communities).

To assess the effects of the after-school program on student outcomes, we assume that in the following spring, researchers collect math and reading achievement test scores from school records data for all treatment and control students. We assume also that the researchers collect school records data on each student's (i) gender and (ii) math and reading test scores and proficiency levels in the year before randomization. These baseline data will be used to conduct subgroup and baseline equivalence analyses and to construct covariates to obtain regression-adjusted impact estimates.

After collecting the data, we assume that the researchers create a Stata rectangular data file for the analysis (but could also create an R file). Table 5 provides information on the data items in the file and their purposes for the analysis. Table 6 displays the structure of the file for selected variables and students in the sample.

The data file contains one record per student and includes the indicator variable, `TREATMENT`, which equals 1 for treatment group students and 0 for control group students. This variable must be nonmissing for all students in the data file or `RCT-YES` will not proceed. Note that the data file does not include student identifiers, which are not needed for the analysis.

The outcome variables for the analysis are `MATH_SCORE` and `READ_SCORE`. As shown in Table 6, some students have missing values for the outcomes, which are coded using the Stata missing data code `.e` (any valid Stata missing data code can be used). It is good research practice to include all observations in the data file—including those with missing outcome data—so that the program can compute data nonresponse rates for the treatment and control groups.

We assume that the researchers aim to conduct optional subgroup analyses to assess whether the effects of the after-school intervention differ by the student's gender and test score proficiency levels in the year before random assignment. Subgroup variables must be *categorical* variables with numeric or character codes. The variables in the data file for this subgroup analysis are `GENDER` (coded as 0 and 1), `SG_MATH_PROF` (coded as 1, 2, and 3), and `SG_READ_PROF` (coded as 1, 2, and 3). `RCT-YES` excludes students with missing subgroup data from the subgroup analysis, but not from the full sample analysis if the student has available outcome data.

The baseline covariates in the data file for the optional regression analysis are `GENDER` (a binary variable) and `PRIOR_MATH_SCORE` and `PRIOR_READ_SCORE` (both continuous variables). These covariates are appropriate for the regression analysis because they pertain to the period before random assignment. `RCT-YES` replaces missing values for covariates for the regression analysis as long as they are nonmissing for most sample members; otherwise, the covariates are omitted.

#### 4. The input data file

**Table 5. Description of variables used for the Design 1 example**

Name	Description	Type	Values	Purpose
TREATMENT	Indicator variable for treatment or control group status	Binary	1 for treatment group 0 for control group	Identifies the research groups
MATH_SCORE	Spring math test score	Numeric	0 to 100	Outcome
READ_SCORE	Spring reading test score	Numeric	0 to 100	Outcome
GENDER	Indicator variable for whether the student is a girl or boy	Numeric	1 = Girls 0 = Boys	Subgroup analysis and baseline covariate
PRIOR_MATH_SCORE	Math test score in the year prior to random assignment	Numeric	0 to 100	Baseline covariate
PRIOR_READ_SCORE	Reading test score in the year prior to random assignment	Numeric	0 to 100	Baseline covariate
SG_MATH_PROF	Math proficiency level in the year prior to random assignment	Categorical	1 = Not proficient 2 = Proficient 3 = Highly proficient	Subgroup analysis
SG_READ_PROF	Reading proficiency level in the year prior to random assignment	Categorical	1 = Not proficient 2 = Proficient 3 = Highly proficient	Subgroup analysis

**Table 6. Hypothetical data file for selected variables for the Design 1 example**

TREATMENT	MATH_SCORE	READ_SCORE	GENDER	PRIOR_MATH_SCORE	SG_MATH_PROF
1	82	88	1	83	2
1	59	66	0	59	1
1	e	74	0	e	e
0	53	58	0	69	2
0	73	85	0	71	2
0	76	e	0	94	3
...	...	...	...	...	...

## Design 2

Design 2 pertains to an RCT where individuals are randomly assigned to a treatment or control group *separately* within blocks. To demonstrate the file structure for this design, we continue with our hypothetical evaluation discussed in the last section for Design 1, where we now assume that random assignment is conducted separately within each of ten school districts (blocks). Within each school district, we assume that students are randomly assigned to a treatment group (who can attend a district-sponsored after-school program) and or to a control group (who cannot attend the after-school program). We use the same simulated dataset for the analysis as for Design 1 (AFTER\_SCHOOL\_RCT\_DATA).

The data file for the analysis contains the variable DISTRICT with codes 1 to 10, where each value uniquely identifies one of the ten study school districts. ***This block identifier must be available for each student or the program does not continue.*** The identifier can be masked to hide the true identities of the school districts, and can be coded using any valid nonmissing numeric or character value. We assume that student sample sizes in the districts range from 118 students in District 5 to 326 students in District 5.

We assume that the researchers collect the same data items as discussed for Design 1 from each school district and are interested in conducting similar analyses. In addition, we assume that, if necessary, the test score outcomes in each district have been appropriately scaled so that the test scores can be combined across districts.<sup>5</sup>

Table 7 displays the structure of the input data file for our hypothetical RCT for selected variables and observations. As we can see, the data structure for Design 2 essentially *repeats* the data structure for Design 1 for each district (block). Note that the TREATMENT indicator varies within each district, because random assignment was conducted separately by district.

## Design 3

Design 3 pertains to a clustered RCT design where *groups rather than individuals* are randomly assigned to the treatment and control groups. To demonstrate the input data file for this design, we use the hypothetical evaluation discussed for Designs 1 and 2, except we now assume that 39 schools containing the 2,256 students are randomly assigned to the treatment and control groups. It is assumed that 18 schools are randomly assigned to the treatment group and 21 to the control group. Under this design, all students in the treatment schools can attend an after-school program in the district, whereas students in the control schools cannot.

---

<sup>5</sup> For example, one popular scaling approach is to convert the test scores to “z-scores” by subtracting the statewide (or sample) test score mean from the individual test scores and dividing by the statewide (or sample) test score standard deviation (see May et al., 2009 for a discussion of scaling methods).

#### 4. The input data file

**Table 7. Hypothetical data file for selected variables for the Design 2 example**

DISTRICT	TREATMENT	MATH_SCORE	READ_SCORE	PRIOR_MATH_SCORE	SG_MATH_PROF
1	1	65	88	74	2
1	1	e	64	57	1
1	1	73	90	94	3
1	0	80	83	75	2
1	0	50	e	65	1
1	0	60	77	74	2
...	...	...	...	...	...
2	1	74	77	73	2
2	1	58	79	e	e
2	1	58	68	68	2
2	0	40	57	54	1
2	0	54	75	63	1
2	0	68	70	70	2
...	...	...	...	...	...

For clustered designs, *RCT-YES* can accommodate data in two formats: (i) individual-level data or (ii) data averaged to the cluster level (for example, average school test scores for students in the study). For the latter format, the input data file must contain a separate set of stacked cluster-level averages for the full sample analysis and for each subgroup analysis. The program allows data to be provided at the cluster level, because this format might better accommodate the collection of administrative records data and might help minimize PII disclosure for evaluations that rely on these data. *Users must specify the data format using the TYPE\_CLUS\_DATA input variable (0 = cluster-level data, 1 = individual-level data).* The data file for our analysis, AFTER\_SCHOOL\_RCT\_DATA, contains data at the individual level.

In what follows, we use our hypothetical RCT to discuss in more detail the structure of the data file for the two possible data formats for Design 3. We assume that the researchers collect the same data items as discussed for Designs 1 and 2 for students in each of the 39 study schools and are interested in conducting similar analyses. In some RCTs, the sample of students for data collection might include students in the study schools at the time of random assignment, whereas in other studies the sample might include students in the study schools at the time of follow-up data collection.

**Data with individual-level records.** Table 8 displays the structure of our data file (AFTER\_SCHOOL\_RCT\_DATA) that contains individual-level records for students in the 39 study schools. The table displays selected variables for selected observations.



**Table 8. Hypothetical data file for selected variables and students for the Design 3 example with individual-level data**

SCHOOL	TREATMENT	MATH_SCORE	READ_SCORE	PRIOR_MATH_SCORE	SG_MATH_PROF
3	1	69	76	52	1
3	1	81	e	72	2
3	1	61	66	76	2
3	1	80	82	70	2
3	1	67	91	e	e
...	...	...	...	...	...
17	0	65	73	78	2
17	0	55	62	66	2
17	0	52	59	e	e
17	0	67	80	69	2
17	0	54	58	44	1
...	...	...	...	...	...

As can be seen from the table, the data file is very similar to the data files for Designs 1 and 2 with the following differences:

- The file contains the cluster identifier, SCHOOL, with codes 1 to 39, where each value uniquely identifies one of the 39 schools in the sample. *This identifier must be nonmissing for all individuals or the program does not proceed.*
- The TREATMENT indicator has the same value for all students in the same school because schools were randomly assigned to the treatment and control groups

**Data with cluster-level records.** The structure of the input data file is more complex if it contains cluster-level data rather than individual-level data. With cluster-level records, *the file must contain separate cluster-level observations for the full sample analysis and for each optional subgroup analysis.* In addition, an indicator variable must be included in the data file that identifies observations for the full sample analysis or a subgroup analysis.

Table 9 provides an example of the structure of the data file with cluster-level records for our hypothetical RCT. Each record is a school, not a student, and each school is identified using the SCHOOL variable. Accordingly, the MATH\_SCORE outcome pertains to *average test scores* in the school (calculated using all students or specific subgroups of students in the school sample).

The first panel in Table 9 contains school-level observations for the full sample analysis where MATH\_SCORE is calculated as the average test score for all study students in the school. This panel contains 39 records, one per school. The remaining panels contain school-level observations for subgroup analyses. For example, consider the second and third panels that pertain to the subgroup

4. The input data file

**Table 9. Hypothetical data file for selected variables and schools for the Design 3 example with cluster-level data**

SCHOOL	TREATMENT	FULL_SUBGR	GENDER	SG_MATH_PROF	MATH_SCORE
1	0	1	d	d	58
2	1	1	d	d	61
3	1	1	d	d	65
...	...	...	...	...	...
39	1	1	d	d	64
1	0	0	1	d	59
2	1	0	1	d	59
3	1	0	1	d	65
...	...	...	...	...	...
39	1	0	1	d	63
1	0	0	0	d	60
2	1	0	0	d	64
3	1	0	0	d	65
...	...	...	...	...	...
39	0	0	0	d	64
1	0	0	d	1	50
2	1	0	d	1	57
3	1	0	d	1	59
...	...	...	...	...	...
39	1	0	d	1	49
1	0	0	d	2	65
2	1	0	d	2	62
3	1	0	d	2	65
...	...	...	...	...	...
39	1	0	d		68
1	0	0	d	3	68
2	1	0	d	3	67
3	1	0	d	3	77
...	...	...	...	...	...
39	1	0	d	3	74

Panel 1: School averages for all students in the sample

Panel 2: School averages for girls

Panel 3: School averages for boys

Panel 4: School averages for students not proficient in math

Panel 5: School averages for students proficient in math

Panel 6: School averages for students highly proficient in math

analysis for boys and girls based on the GENDER binary variable. Panel 2 contains school-level observations for girls (GENDER=1), where MATH\_SCORE is calculated using the test scores only of girls in the school. Schools without girls do not contribute records to this panel; it is preferable (although not required) that these schools be included in the data file with a missing value code for MATH\_SCORE. Similarly, Panel 3 contains school-level observations for boys (GENDER=0). Importantly, the GENDER indicator is set to .d (not applicable) for all records outside Panels 2 and

3 (but could be set to any valid Stata missing data code). A similar data structure holds for Panels 4 to 6 for the subgroup analysis to examine impacts by baseline math proficiency level using the categorical variable SG\_MATH\_PROF. Although not shown, there are six schools that have missing values for the MATH\_SCORE variable for Panel 6 because they do not contain students who were highly proficient in math in the prior year.

Table 9 shows that the file contains a required binary variable that indicates whether the record corresponds to a full sample or subgroup analysis. The name of this variable in our example is FULL\_SUBGR and equals 1 for records in Panel 1 and 0 for records in all other panels. ***This variable must be specified in the CLUSTER\_FULL input and must contain codes of 1 for the full sample analysis and 0 for an optional subgroup analysis or the program terminates.*** The variable must be specified even if no subgroup analyses are conducted.

Additional outcome variables (for example, READ\_SCORE) can be included as separate panels or as extra columns in existing panels. For the former specification, READ\_SCORE must be set to missing for the MATH\_SCORE observations and vice versa. The same outcome variable (for example, MATH\_SCORE) cannot be repeated across panels. The CLUSTER\_FULL variable must be coded for all panels.

Finally, covariates should be aggregated to the cluster level in a similar way as for the outcome variables and entered as separate variables in the data file. The program will impute missing covariates using rules similar to those discussed in Chapter 2i.

#### Design 4

Design 4 pertains to a clustered, blocked RCT design where *groups* are randomly assigned to the treatment and control groups separately within blocks. In our after-school program evaluation, this design would occur if schools are randomly assigned to the treatment and control groups within school districts. Thus, Design 4 combines features of both Designs 2 and 3. Thus, we only highlight several key features of the data file structure for this design and do not present examples:

- The data file for Design 4 must contain *both block and cluster identifiers*. Block identifiers must be unique across blocks, although cluster identifiers can repeat across blocks (although this is not required).
- For Design 4 as with Design 3, *RCT-YES* can accommodate both individual- and cluster-level data. In essence, the data file for Design 4 repeats the data structure for Design 3 for each block. If the data file contains cluster-level records, the data file must contain a binary variable that indicates whether the observation pertains to a full sample or subgroup analysis.



## 5. Entering Inputs, generating output files, and conducting the analysis

This chapter discusses how to (i) launch the *RCT-YES* interface screens and enter program inputs into them, (ii) save program inputs into a file for future use, and (iii) generate an R or Stata computer program file using the interface that users will then need to run outside the interface to conduct the analysis and obtain study results.

### a. Launching the interface screens and entering program inputs into them

The *RCT-YES* interface screens can be launched in several ways:

1. Double click the *RCT-YES* desktop icon



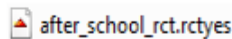
Click the desktop icon to launch the program

2. Double click the *RCT-YES* program link in the Start/All Programs menu, Mathematica Policy Research, Inc folder



Locate and click the *RCT-YES* program in the Start menu

3. Double click a previously saved input specification file in the directory where it is located. This process will pre-fill the interface screens with the inputs previously specified.



Click a saved input file to launch the program

1/29/2016 4:03 PM

RCTYES File

4 KB

The interface contains five green tabs displayed at the top of the screens that can be navigated by clicking on them or using the **Prev** / **Next** buttons on the bottom of the screens. The first four tabs pertain to program inputs, whereas the final tab pertains to the generation of output files to conduct the analysis. Some tabs have associated subtabs. Users can resize or minimize the screens at any time.

To demonstrate how to enter the inputs into the interface screens, we use the Design 4 example from Chapter 4 for an RCT of an after-school intervention where schools are randomly assigned to a treatment or control group within school districts (blocks), and where the input dataset, AFTER\_SCHOOL\_RCT\_DATA, contains individual-level data. We assume that the analysis will be conducted using Stata and that interest lies in conducting the optional subgroup, regression, and baseline equivalence analyses discussed in Chapter 4.

## 5. Program inputs and outputs

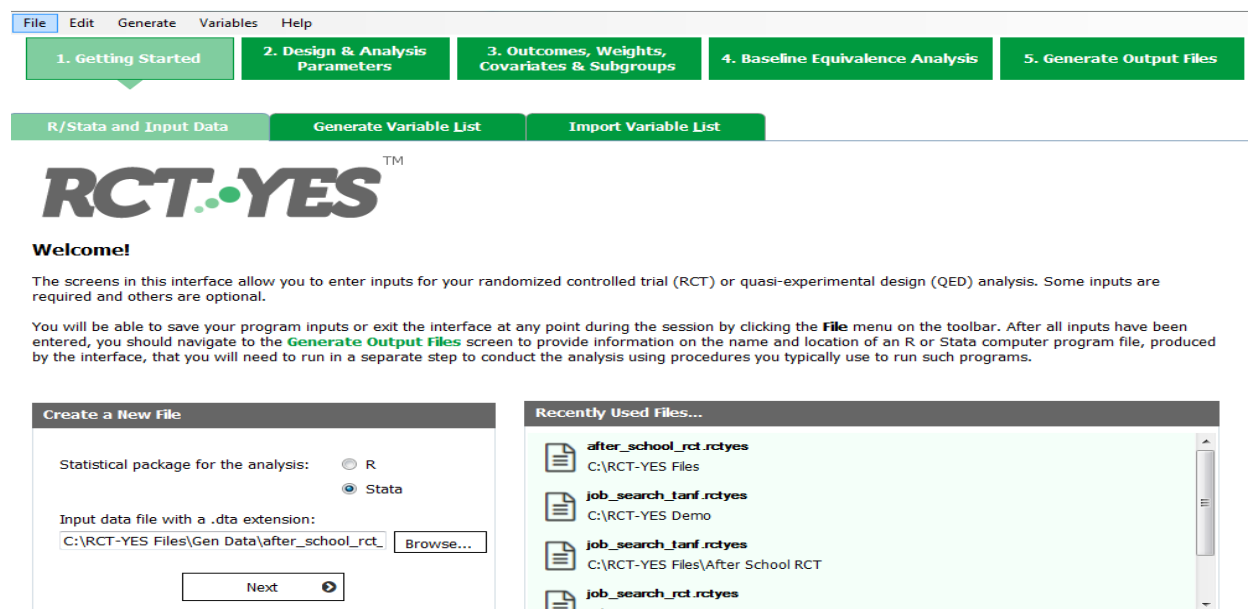
### Saving program inputs and exiting the program

Program inputs can be saved to a file at any point during an *RCT-YES* input session by clicking the **File** menu and **Save** or **Save As** command. Program inputs will also be saved when generating the output files. Users can exit the program by clicking the **File** menu and **Exit** command or clicking “x” in the upper right corner of the screen. If users try to exit the program before saving changes to the program inputs, they will be prompted to save the changes before exiting.

### Getting Started Screen

When the interface is launched, users will be directed to the **Getting Started** screen which has three subscreens (i) the **R/Stata and Input Data** screen (see Screenshot 5.1), (ii) the **Generate Variable List** screen, and (iii) the **Import Variable List** screen. If the interface was launched using the *RCT-YES* icon or **Start** menu, users will be able to enter program inputs for the first time (as is the case for our example). Alternatively, users can open a previously saved input specification file [.rctyes extension] using the **File** menu and **Open** or **Open Recent** command or the **Recently Used Files** list displayed at the bottom right corner of the screen. In these cases, the screens will pre-fill with the inputs entered previously. The same situation will occur if users launch the interface by directly opening a previously saved input specification file in the directory where it was saved.

### Screenshot 5.1: R/Stata and Input Data Screen



#### i. R/Stata and Input Data Screen

The **R/Stata and Input Data** screen provides a brief program description. It then asks users (i) to specify, by clicking the appropriate circle, whether the analysis should be conducted in R or Stata (Stata in our case) and (ii) to specify, using the **Browse** button, the name and location of the input

data file (a .rds file for R or a .dta file for Stata). In our example, the name of the input data file is AFTER\_SCHOOL\_RCT\_DATA.DTA. The two inputs in this screen are required or the interface will not generate the computer program file needed to conduct the analysis, as will be the case for any required input that is not specified.

Users interested in creating a variable list window to help with the entry of variables into the screens should now click **Next** at the bottom of the screen, which will direct them to the **Generate Variable List** screen. Or, users can navigate to other screens using the green tabs at the top of the screens.

### ii. Generate Variable List and Import Variable List Screens

To describe the process for creating a variable list window, it is critical that users understand that *the interface does not read the input data file*, and thus, does not directly create the variable list window. Rather, the interface will create an R or Stata computer program that users will then need to run outside the interface to produce a text file containing the list of variables in the data file. Users will then be able to import this file into the interface to create the variable list window. Users who do not create a variable list window will need to type in variables directly into the interface.

The variable list window can be created using the **Generate Variable List** and **Import Variable List** screens (see Screenshots 5.2 and 5.3) using the following steps:

- In the **Generate Variable List** screen, enter information on the base name and path name for the R or Stata computer program file [.R or .do extension], to be generated by the interface, that will need to be run outside the interface to create a variable list text file [.varlist extension]. *The interface will add a “\_VL” suffix to the base name to distinguish it from other output files discussed later.* In our example, we specify the base name as AFTER\_SCHOOL\_RCT\_DATA, so the interface will produce a Stata computer program file called AFTER\_SCHOOL\_RCT\_DATA\_VL.DO, which, when run, will produce a .varlist text file with the same base name.
- Click the **Generate Program File** button to create the R or Stata computer program file. A dialog box will indicate if the file was created successfully and, if not, file generation errors.
- Minimize (or exit) the interface, and run the R or Stata computer file outside the interface to create the variable list file (AFTER\_SCHOOL\_RCT\_DATA\_VL.VARLIST in our example). Note that this file will only need to be created once per dataset.
- Navigate to the **Import Variable List** screen to input information on the variable list file. The import box will pre-fill with the .varlist file generated above, but a different .varlist file can be specified using the **Browse** button to locate the file.
- Click the **Import Variable List** button to create the variable list window. A dialog box will indicate whether the window was created successfully.

## 5. Program inputs and outputs

### Screenshot 5.2: Generate Variable List Screen

The screenshot shows the RCT-YES software interface. At the top, there is a menu bar with 'File', 'Edit', 'Generate', 'Variables', and 'Help'. Below the menu bar are five green navigation buttons: '1. Getting Started', '2. Design & Analysis Parameters', '3. Outcomes, Weights, Covariates & Subgroups', '4. Baseline Equivalence Analysis', and '5. Generate Output Files'. Below these are three sub-navigation buttons: 'R/Stata and Input Data', 'Generate Variable List' (which is highlighted), and 'Import Variable List'. The main area features the RCT-YES logo and a section titled 'File Information to Create the Variable List Window (required)'. This section contains a form with the following fields: 'Common base name for both files:' with the value 'after\_school\_rct\_VL'; a note stating 'Importantly, the interface will add a \"\_VL\" suffix to the specified base name for the files.'; 'Folder location (path name) for each file...'; 'Stata program file for you to run:' with the value 'C:\RCT-YES Files'; and 'Variable list text file produced by the Stata program:' with the value 'C:\RCT-YES Files'. Below the form is a 'Generate Program File' button. A text instruction reads: 'After specifying information for all files, click the Generate Program File button below, minimize the interface, and run the computer program:'. Below this is a 'Previous' button and a 'Next' button.

File Edit Generate Variables Help

1. Getting Started 2. Design & Analysis Parameters 3. Outcomes, Weights, Covariates & Subgroups 4. Baseline Equivalence Analysis 5. Generate Output Files

R/Stata and Input Data Generate Variable List Import Variable List

# RCT-YES™

This screen and the next one can be used to create an optional interface window displaying a list of variables in your input data file, after\_school\_rct\_data.dta, that can be directly. Instead, you should use this screen to produce a Stata computer program file [.do extension] that you can run outside the interface as you normally run such program variables in your input data file that you can then import in the next screen to create the variable list window.

**File Information to Create the Variable List Window (required)**

**Common base name for both files:** after\_school\_rct\_VL

Importantly, the interface will add a "\_VL" suffix to the specified base name for the files.

**Folder location (path name) for each file...**

Stata program file for you to run: C:\RCT-YES Files

Variable list text file produced by the Stata program: C:\RCT-YES Files

**After specifying information for all files, click the Generate Program File button below, minimize the interface, and run the computer program:**

Generate Program File

**After you have run the computer program, you should navigate to the next screen to input the .varlist text file to create the variable list window:**

Previous Next

### Screenshot 5.3: Import Variable List Screen

The screenshot shows the RCT-YES software interface. At the top, there is a menu bar with 'File', 'Edit', 'Generate', 'Variables', and 'Help'. Below the menu bar are five green navigation buttons: '1. Getting Started', '2. Design & Analysis Parameters', '3. Outcomes, Weights, Covariates & Subgroups', '4. Baseline Equivalence Analysis', and '5. Generate Output Files'. Below these are three sub-navigation buttons: 'R/Stata and Input Data', 'Generate Variable List', and 'Import Variable List' (which is highlighted). The main area features the RCT-YES logo and a section titled 'Below, provide information on the .varlist text file to import into the interface to create the variable list window:'. This section contains a text input field with the value 'C:\RCT-YES Files\after\_school\_rct\_VL.varlist' and a 'Browse' button. Below this is an 'Import Variable List' button. A text instruction reads: 'After specifying information for the file, click the Import Variable List button below:'. To the right of the main interface, a 'Variable List Window' is open, showing a list of variables with checkboxes: DISTRICT, GENDER, MATH\_SCORE, PARTIC, PRIOR\_MATH\_SCORE, PRIOR\_READ\_SCORE, READ\_SCORE, SCHOOL, SG\_MATH\_PROF, SG\_READ\_PROF, and TREATMENT. The window has 'Close' and 'Insert Variable' buttons. A red arrow points from the 'Import Variable List' button to the 'Variable List Window'.

File Edit Generate Variables Help

1. Getting Started 2. Design & Analysis Parameters 3. Outcomes, Weights, Covariates & Subgroups 4. Baseline Equivalence Analysis 5. Generate Output Files

R/Stata and Input Data Generate Variable List Import Variable List

# RCT-YES

Below, provide information on the .varlist text file to import into the interface to create the variable list window:

C:\RCT-YES Files\after\_school\_rct\_VL.varlist Browse

**After specifying information for the file, click the Import Variable List button below:**

Import Variable List

The variable list window will open automatically. If you close it, you can re-open it by clicking the **Variables** command in the toolbar menu. You can also use the autocomplete feature by starting to type variables directly into the input boxes to filter and select the desired variables.

Variable List Window

Variable Filter:

Variables:

- DISTRICT
- GENDER
- MATH\_SCORE
- PARTIC
- PRIOR\_MATH\_SCORE
- PRIOR\_READ\_SCORE
- READ\_SCORE
- SCHOOL
- SG\_MATH\_PROF
- SG\_READ\_PROF
- TREATMENT

Close Insert Variable

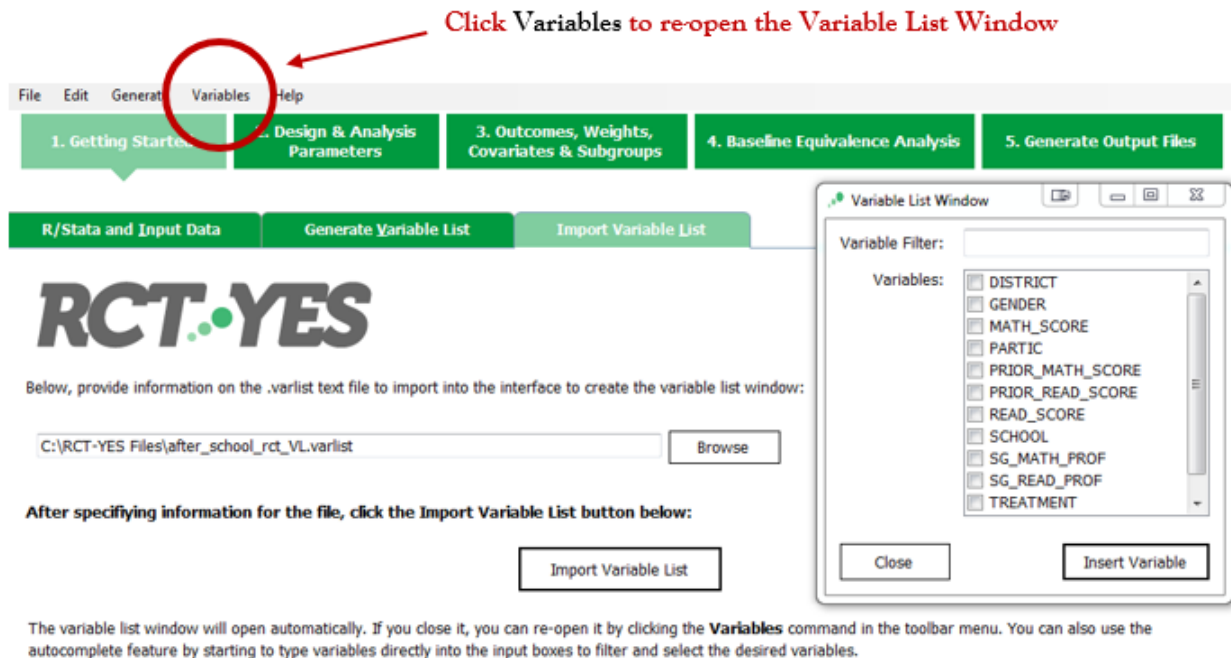
Click to import and open the Variable List Window



## Accessing the Variable List Window

The variable list window will open automatically when it is created (see Screenshot 5.3 above). The window can be resized or moved at any time. Users can filter variables in the window by starting to type variable names in the **Variable Filter** box at the top of the window. The window will stay open unless a user decides to close it using the **Close** button; in this case, the window can be re-opened by clicking the **Variables** command in the toolbar menu (see Screenshot 5.4).

**Screenshot 5.4: Re-opening the Variable List Window if it was closed**



Once the variable list window is open, users can select a variable to insert into an input box by (i) clicking the input box (which will then be highlighted), (ii) clicking the small square box to the left of the desired variable in the variable list window, and (iii) clicking the **Insert Variable** button at the bottom of the window. For example, in Screenshot 5.5 below, to insert the treatment status indicator variable, TREATMENT, into the TC\_STATUS input box, we would first click the TC\_STATUS input box, locate TREATMENT in the variable list window, click the associated box, and click **Insert Variable**.

## 5. Program inputs and outputs

### Screenshot 5.5: Inserting variables using the Variable List Window

Click the **TREATMENT** box and then click the **Insert Variable** button

The screenshot displays the software's main interface. At the top, there is a menu bar with 'File', 'Edit', 'Generate', 'Variables', and 'Help'. Below this is a progress bar with five steps: '1. Getting Started', '2. Design & Analysis Parameters', '3. Outcomes, Weights, Covariates & Subgroups', '4. Baseline Equivalence Analysis', and '5. Generate Output Files'. The 'Design Selection' section is active, showing 'Required Design Parameters' and 'Optional Design & Analysis Parameters'. The 'Input Specifications' table is visible, with columns for 'Input Variables', 'Input Specifications', and 'Input Definitions'. The 'TREATMENT' variable is highlighted in the 'Input Specifications' column. A 'Variable List Window' is open, showing a list of variables with 'TREATMENT' selected. A red arrow points from the 'TREATMENT' box in the table to the 'TREATMENT' variable in the list window. Below the table are 'Previous' and 'Next' navigation buttons.

Input Variables	Input Specifications	Input Definitions
TC_STATUS	TREATMENT	Name of the treatment file
BLOCK_ID	DISTRICT	Name of the block
MATCHED_PAIR	<input checked="" type="radio"/> 0 = Not a matched pair, blocked design (default) <input type="radio"/> 1 = Matched pair, blocked design	A matched pair design with matched control users
CLUSTER_ID	SCHOOL	Name of the cluster
TYPE_CLUS_DATA	<input type="radio"/> 0 = Cluster-level data <input checked="" type="radio"/> 1 = Individual-level data	Indicator for cluster-level or individual-level data
CLUSTER_FULL		If TYPE_CLUS_DATA = 1, signifying whether

Variable List Window

Variable Filter:

Variables:

- DISTRICT
- GENDER
- MATH\_SCORE
- PARTIC
- PRIOR\_MATH\_SCORE
- PRIOR\_READ\_SCORE
- READ\_SCORE
- SCHOOL
- SG\_MATH\_PROF
- SG\_READ\_PROF
- TREATMENT

Close Insert Variable

Previous Next

Most input boxes request the name of a single variable only. If these boxes contain an existing variable, inserting a different variable from the variable list window will replace the existing variable.

The input boxes pertaining to baseline covariates for the regression and baseline equivalency analyses, however, allow for multiple variables. In these cases, users can select multiple variables from the variable list window, and inserting them into the input boxes will supplement existing variables.

The second way to access the variable list is to start typing the name of a variable into an input box. In this case, an “autocomplete” window will appear that filters the list of variables that match the inserted text (ignoring whether the text is lower or upper case). Users can then click the desired variable to insert into the input box.

## Design & Analysis Parameters Screen

After completing the **Getting Started** screen, users should navigate to the **Design & Analysis Parameters** screen. As discussed next, this screen is in three parts.

### i. Design Selection Screen

The **Design Selection** screen (Screenshot 5.6) requests that users specify whether their design is clustered and/or blocked (Designs 1, 2, 3, or 4), so that *RCT-YES* can determine the appropriate methods to use for impact estimation (see Chapter 2). This input is required. Users can also enter an optional title for the output tables.

#### Screenshot 5.6: Design Selection Screen

Input Variables	Input Specifications	Input Definitions
DESIGN (required)	<input type="radio"/> 1 = Non-clustered, non-blocked <input type="radio"/> 2 = Non-clustered, blocked <input type="radio"/> 3 = Clustered, non-blocked <input checked="" type="radio"/> 4 = Clustered, blocked	Non-clustered RCT designs are those where individuals are randomized, whereas clustered designs are those where groups (such as schools or hospitals) are randomized. Non-blocked designs are those where randomization is conducted within a single population, whereas blocked designs are those where randomization is conducted separately within partitions of the entire sample (such as school districts, grades, counties, or demographic subgroups).

TITLE (optional)  Title for outcome tables presenting analysis results

### ii. Required Design Parameters Screen

The **Required Design Parameters** screen (Screenshot 5.7) allows users to enter key information for impact estimation. The input, `TC_STATUS`, is required for all designs, whereas the other inputs are required for certain designs only. For instance, the inputs, `BLOCK_ID` and `MATCHED_PAIR`, pertain only to blocked designs (Designs 2 and 4), `CLUSTER_ID` and `TYPE_CLUS_DATA` pertain only to clustered designs (Designs 3 and 4), and `CLUSTER_FULL` pertains only to designs where `TYPE_CLUS_DATA` is set to 0. Users will only be able to enter information for inputs that pertain to their particular design; other inputs will be disabled.

In our example, `TC_STATUS` is set to the name of the treatment-control indicator variable in the input data file (`TREATMENT`), `BLOCK_ID` is set to the school district identifier (`DISTRICT`), and `CLUSTER_ID` is set to the school identifier (`SCHOOL`). The input variable, `TYPE_CLUS_DATA`, is set to 1 because the input file contains student-level records (not school-level records), and thus, `CLUSTER_FULL` is not pertinent for the design and is disabled. The variable list window can be used to enter variable names into the input boxes using the methods discussed in the last section.

## 5. Program inputs and outputs

### Screenshot 5.7: Required Design Parameters Screen

File Edit Generate Variables Help

1. Getting Started 2. Design & Analysis Parameters 3. Outcomes, Weights, Covariates & Subgroups 4. Baseline Equivalence Analysis 5. Generate Output Files

Design Selection Required Design Parameters Optional Design & Analysis Parameters

Input Variables	Input Specifications	Input Definitions
TC_STATUS	<input type="text" value="TREATMENT"/>	Name of the treatment or control status indicator (1/0) variable in the data file
BLOCK_ID	<input type="text" value="DISTRICT"/>	Name of the block identifier variable in the data file
MATCHED_PAIR	<input checked="" type="radio"/> 0 = Not a matched pair, blocked design (default) <input type="radio"/> 1 = Matched pair, blocked design	A matched pair design is a blocked RCT with one matched treatment and one matched control unit per block, where similar study units are paired prior t...
CLUSTER_ID	<input type="text" value="SCHOOL"/>	Name of the cluster identifier variable in the data file
TYPE_CLUS_DATA	<input type="radio"/> 0 = Cluster-level data <input checked="" type="radio"/> 1 = Individual-level data	Indicator for clustered designs of whether the data file contains individual-level records or cluster-level averages (for example, school-level ...
CLUSTER_FULL	<input type="text"/>	If TYPE_CLUS_DATA = 0, name of the indicator variable in the data file signifying whether the cluster-level average record pertains to the full sam...

### iii. Optional Design & Analysis Parameters Screen

The final design-related screen (Screenshot 5.8) allows users to override pre-filled program default values for several design and analysis parameters. All inputs in this screen are optional. Some inputs pertain to all designs (SUPER\_POP, LABEL\_T, LABEL\_C, MISSING\_COV, OBS\_COV, MIN\_NUM, ALPHA\_LEVEL, LIMIT\_PRINT, and CSV\_FILE), whereas others (BLOCK\_FE, CATE\_UATE, and NO\_SG\_COV) pertain only to certain designs and model specifications. We refer users to Chapter 2 for a discussion of these options (as well as to Schochet, 2016 for further details). Screenshot 5.8 displays program default values for our Design 4 example which we do not change.

Screenshot 5.8: Optional Design & Analysis Parameters Screen

File Edit Generate Variables Help

1. Getting Started 2. Design & Analysis Parameters 3. Outcomes, Weights, Covariates & Subgroups 4. Baseline Equivalence Analysis 5. Generate Output Files

Design Selection Required Design Parameters **Optional Design & Analysis Parameters**

Input Variables	Input Specifications (default values are filled in)	Input Definitions
SUPER_POP	<input checked="" type="radio"/> 0 = Finite-population (FP) model <input type="radio"/> 1 = Super-population (SP) model	Indicator of preference for the FP or SP model. The FP model assumes that the study findings pertain to the study sample only, whereas the SP model assumes tha...
CATE_UATE	<input type="radio"/> 0 = PATE parameter <input type="radio"/> 1 = CATE parameter <input type="radio"/> 2 = UATE parameter	Different SP parameters that assume random sampling of units at all hierarchical levels (PATE), at the highest level only (UATE), or at the lowest level only (CATE) (see the RCT-YES User's Manual for details)
BLOCK_FE	<input checked="" type="radio"/> 0 = Model should contain both main block effects and block-by-treatment interactions (default) <input type="radio"/> 1 = Model should contain main block effects only	If this option is set to 1, the estimation model includes block indicator variables, but not terms formed by interacting treatment status and block indicator variables. This option might be attractive if many blocks contain small numbers of individuals (for non-clustered designs...
LABEL_T	Treatment	Label, up to 14 characters, for the treatment group (TC_STATUS=1): the program will add "Group" to the ...
LABEL_C	Control	Label, up to 14 characters, for the control group (TC_STATUS=0): the program will add "Group" to the ...
MISSING_COV	Number between 0 to 75 for regression analyses 30	Maximum percentage of missing data for a baseline covariate to be included in the regression models
OBS_COV	Number greater than 1.0 for regression analyses 5.0	Required ratio of the number of observations (or clusters) per covariate for the regression analysis to be...
MIN_NUM	Integer greater than or equal to 3 10	Minimum group size adopted by the state or other entity for reporting outcomes to protect personally...
ALPHA_LEVEL	Integer greater than 0 and less than 30 5	Significance level used for hypothesis testing (in percents)
NO_COV_SG	<input checked="" type="radio"/> 0 = Subgroup interaction tests should include covariance terms (default) <input type="radio"/> 1 = Subgroup interaction tests should exclude covariance terms	If this option is set to 1, the statistical tests to gauge differences in impact estimates across subgroups ignores the potential covariances between subgroup estimates within the same cluster or block. This option might be attractive if the number of cluster...
LIMIT_PRINT	<input checked="" type="radio"/> 0 = All output tables printed (default) <input type="radio"/> 1 = Printing limited to tables with main impact results only (Tables 1 to 3 and 8 to 10)	Suppresses printing of detailed descriptive sample statistics in the output tables
CSV_FILE	<input type="radio"/> 0 = .csv data files not produced (default) <input checked="" type="radio"/> 1 = .csv data files produced	If this option is set to 1, the computer program will produce a .csv file containing data from the output tables. This .csv file can be used for further analyses and reporting.

Previous Next

## 5. Program inputs and outputs

### Outcomes, Weights, Covariates, & Subgroup Screens

In *RCT-YES*, outcomes must be entered separately for each “outcome domain” pertaining to a specific class of outcomes in which common analyses are to be conducted. This grouping helps minimize data entry and organize the reporting of the impact findings. For each outcome domain, users can specify optional covariates and subgroups that pertain to *all* outcomes in the domain.

The **Outcomes, Weights, Covariates & Subgroup** screen consists of three nested subscreens:

1. An **Outcome Domain** screen used to create new outcome domains or edit existing ones
2. An **Outcome Details** screen used to enter outcomes and associated weights and covariates for the full sample analysis, and to request subgroup analyses
3. A **Subgroup Analysis** screen to enter subgroup analysis information, including the subgroup variable name, subgroup category codes and labels, and associated weights and covariates

Next, we describe each of these nested subscreens in turn.

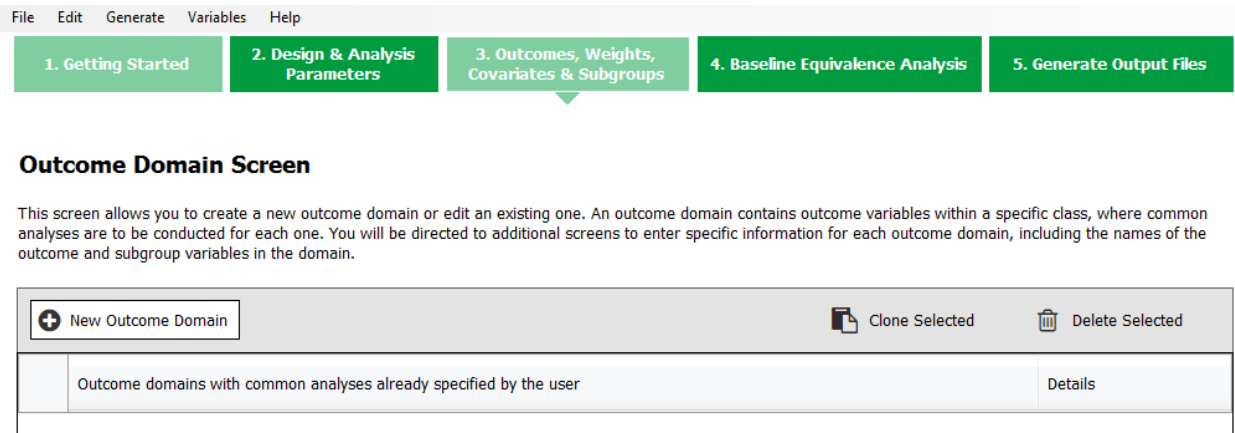
#### i. Outcome Domain Screen

The **Outcome Domain** screen shown in Screenshot 5.9 allows the user to:

- Create a new outcome domain by clicking the **New Outcome Domain** button
- Open and edit an outcome domain already specified, by first clicking the title of the outcome domain and then clicking the **Edit/View** button. This option is not available in Screenshot 5.9 because we have not yet entered information on an outcome domain.
- Clone (copying) an existing outcome domain into a new outcome domain by first clicking the title of the outcome domain and then clicking the **Clone Selected** button. This option can be useful for running different model specifications that involve small changes to the inputs (for example, estimating impacts for models with and without weights or covariates).
- Delete an existing outcome domain

For our example, we click the **New Outcome Domain** button, which will send us to the **Outcome Details** screen discussed next.

## Screenshot 5.9: Outcome Domain Screen



### ii. Outcome Details Screen

Screenshot 5.10 displays the interface screen for entering the following detailed information for each outcome domain (moving from the top to the bottom of the screen):

1. **Title for the outcome domain (recommended)**, which identifies the outcome domain. In our example, the title for the outcome domain is “**Achievement Test Scores.**” *RCT-YES* will provide a generic name if this title is not entered.
2. **The name of each outcome variable, where each one is entered in a separate row in the table (required)**. In our example, the outcomes are `MATH_SCORE` and `READ_SCORE`.
3. **Optional labels, weights, and individual-level standard deviations for the effect size calculations (see Chapter 2)**. The weights pertain to the full sample analysis and can differ across outcomes, and similarly for the standard deviations.
4. **Optional baseline covariates for the full sample analysis, entered into the COVARIATES input box, to obtain regression-adjusted impact estimates**. The list of covariates should be separated by spaces and can be entered all at the same time or in batches using the variable list window (if created). In our example, the baseline covariates are `PRIOR_MATH_SCORE`, `PRIOR_READ_SCORE`, and `GENDER`. These same covariates will be used in the full sample analysis for all outcomes in the domain.
5. **Names of up to two optional variables, entered into the GOT\_TREAT input boxes, indicating whether the sample member *received* intervention services**. These variables allow users to obtain CACE impact estimates that adjust for treatment group members who did not receive intervention services and control group members who did (see Chapter 2). This analysis will be conducted for all specified outcomes and subgroups in the outcome domain. In our example, we do not specify intervention service receipt variables.

## 5. Program inputs and outputs

### Screenshot 5.10: Outcome Details Screen

Save & Return to the Outcome Domain ScreenCancel

#### Outcome Details Screen

This screen allows you to enter the title for the outcome domain, the names of the outcome variables in the outcome domain, optional weights and covariates for the full sample analysis, and optional intervention receipt variables. It also allows you to request subgroup analyses.

**Title for outcome domain (recommended)**

In the table below, enter information for each outcome variable in the outcome domain:

Name of outcome variable (required)	Label for outcome variable (optional)	Name of weight variable for the full sample analysis (optional)	Individual-level standard deviation (optional)
OUTCOME	LABEL	WEIGHT	STD_OUTCOME
MATH_SCORE	Math Test Scores		
READ_SCORE	Reading Test Scores		
*			

COVARIATES (optional)  List of names of baseline covariates to obtain regression-adjusted impact estimates for the full sample analysis

---

GOT\_TREAT (optional)   List of names of up to two variables indicating intervention receipt used to estimate complier average causal effects (see the *RCT-YES* User's Manual for details)

---

Use the following section to specify a new subgroup analysis or edit an existing one. You will be directed to a separate screen to enter specific information on the subgroup variable name, category codes, and labels. Once you have finished entering subgroup information, you should return to the **Outcome Domain Screen** to specify more outcome domains or navigate to other input screens.

**Subgroup analyses (optional)**

+ New Subgroup AnalysisClone SelectedDelete Selected

At this point, if users are not interested in conducting optional subgroup analyses, they can click the **Save & Return** button at the bottom or top left corner of the screen to return to the **Outcome Domain** screen.

Alternatively, if users are interested in conducting subgroup analyses for the outcome domain, they should navigate to the bottom of screen to the **Subgroup analyses** section. This section will allow users to (i) enter information on a new subgroup by clicking the **New Subgroup Analysis** button or (ii) edit, clone, or delete a previously entered subgroup.



For our example, we are interested in conducting subgroup analyses, so we click the **New Subgroup Analysis** button. This will bring us to the **Subgroup Analysis** screen, discussed in the next section, which is contained within the **Outcome Details** screen, which in turn, is contained within the **Outcome Domain** screen. In other words, the subgroup information will apply to both the MATH\_SCORE and READ\_SCORE outcomes in the “Achievement Test Scores” domain.

### iii. Subgroup Analysis Screen

In *RCT-YES*, information for each subgroup analysis is entered using the screen shown in Screenshot 5.11, where a separate screen must be filled in for each subgroup. Variables pertaining to subgroups and associated covariates and weights can be entered into the screen using the variable list window (if created).

Key features of this screen (moving from the top to bottom of the screen) are as follows:

- Users are **required** to enter the name of the (categorical) subgroup variable in the **SUBGROUP input box**. Only one subgroup name can be provided; separate subgroup screens are required to enter information for each subgroup (the same subgroup can be repeated across screens). In our example, the subgroup of interest is SG\_MATH\_PROF, and separate screens are filled in for SG\_READ\_PROF and GENDER (not shown).
- Users are **required** to enter codes and labels for each subgroup category in a separate table row. Not all possible subgroup categories need to be listed. For example, if a subgroup variable takes on the values 1 to 5, users could enter codes of 3 and 5 if interest lies in these two subgroups only. In this case, *RCT-YES* will conduct the analysis using data for these two categories only. In our example, we seek to obtain impact estimates for those not proficient in math in the prior year (SG\_MATH\_PROF = 1), those proficient in math (SG\_MATH\_PROF = 2), and those highly proficient in math (SG\_MATH\_PROF = 3).
- Users can specify **optional** weights for the subgroup analysis for each outcome specified in the **Outcome Details** screen. These weights will **default** to the weights specified in the **Outcome Details** screen for the full sample analysis whenever those weights are entered or changed. However, users can override these defaults by directly entering weights for the subgroup analysis. We do not specify weights in our example.
- **Optional baseline covariates can be specified for the subgroup analysis using the COVARIATES input box**. The list of covariates must be separated by spaces and can all be entered at the same time or in batches using the variable list window (if created). The covariates can differ for the subgroup and full sample analyses. For instance, in our example, the covariate, GENDER, is included for the full sample analysis, but is excluded from the GENDER subgroup analysis (not shown).

## 5. Program inputs and outputs

### Screenshot 5.11: Subgroup Analysis Screen

#### Subgroup Analysis Screen - Achievement Test Scores

This screen allows you to enter information for a subgroup analysis. Subgroup variables must be categorical variables with numeric or character codes (e.g., GENDER=1 for girls and GENDER=0 for boys). **Only a single subgroup variable can be listed** (a separate screen should be used for each subgroup variable). The subgroup analysis will be conducted for all outcome variables in the outcome domain.

Name of SUBGROUP variable in the data file (e.g., GENDER) (required)

In the table below, enter the codes and labels for each SUBGROUP category (e.g., codes of 1 and 0 for GENDER, with labels Girls and Boys, respectively):

Code for SUBGROUP category (required)	Label for category (required)
1	Not Proficient in Math
2	Proficient in Math
3	Highly Proficient in Math

In the table below, specify optional weights for the subgroup analysis, by outcome. These weights default to the weights specified in the **Outcome Details Screen** whenever those weights are entered or changed.

OUTCOME variable (Name of outcome for the subgroup analysis)	WEIGHT variable (Name of weight for the subgroup analysis; optional)
MATH_SCORE	
READ_SCORE	

Below, list the names of optional baseline covariates to obtain regression-adjusted impact estimates for the subgroup analysis:

COVARIATES (optional)

After entering inputs for a particular subgroup analysis, users should now click the **Save Subgroup Analysis** button to return to the **Outcome Details** screen (see Screenshot 5.12). Note that the SG\_MATH\_PROF, SG\_READ\_PROF, and GENDER subgroups are now listed as subgroups and can be edited, cloned, or deleted. Users can now create new subgroups or edit existing ones using the same procedures discussed above.

Screenshot 5.12: Outcome Details Screen after entering subgroup information

Save & Return to the Outcome Domain Screen
Cancel

### Outcome Details Screen

This screen allows you to enter the title for the outcome domain, the names of the outcome variables in the outcome domain, optional weights and covariates for the full sample analysis, and optional intervention receipt variables. It also allows you to request subgroup analyses.

**Title for outcome domain (recommended)**

In the table below, enter information for each outcome variable in the outcome domain:

	Name of outcome variable (required)	Label for outcome variable (optional)	Name of weight variable for the full sample analysis (optional)	Individual-level standard deviation (optional)
	OUTCOME	LABEL	WEIGHT	STD_OUTCOME
▶	MATH_SCORE	Math Test Scores		
	READ_SCORE	Reading Test Scores		
*				

COVARIATES (optional)  List of names of baseline covariates to obtain regression-adjusted impact estimates for the full sample analysis

---

GOT\_TREAT (optional)   List of names of up to two variables indicating intervention receipt used to estimate complier average causal effects (see the *RCT-YES* User's Manual for details)

Use the following section to specify a new subgroup analysis or edit an existing one. You will be directed to a separate screen to enter specific information on the subgroup variable name, category codes, and labels. Once you have finished entering subgroup information, you should return to the **Outcome Domain Screen** to specify more outcome domains or navigate to other input screens.

**Subgroup analyses (optional)**

+ New Subgroup Analysis
Clone Selected
Delete Selected

Subgroup Variable Name	
SG_MATH_PROF	✎ Edit/View
SG_READ_PROF	✎ Edit/View
GENDER	✎ Edit/View

Once users have finished entering information on subgroups, they should click the **Save & Return** button on the **Outcome Details** screen to return to the **Outcome Domain** screen (Screenshot 5.13). Information on additional outcome domains can now be entered. Note that the outcome domain entitled “Achievement Test Scores” now appears and can be edited, cloned, or deleted.

## 5. Program inputs and outputs

### Screenshot 5.13: Outcome Domain Screen after entering outcome and subgroup information

**Outcome Domain Screen**

This screen allows you to create a new outcome domain or edit an existing one. An outcome domain contains outcome variables within a specific class, where common analyses are to be conducted for each one. You will be directed to additional screens to enter specific information for each outcome domain, including the names of the outcome and subgroup variables in the domain.

Outcome domains with common analyses already specified by the user	Details
Achievement Test Scores	Edit/View

### Baseline Equivalence Analysis Screen

If users are interested in conducting an optional analysis to assess the baseline equivalence of the treatment and control groups, they should navigate to the **Baseline Equivalence Analysis** screen to enter a list of baseline covariates (see Screenshot 5.14). These covariates can be entered using the variable list window (if created). The covariates can differ for the baseline equivalence and regression analyses (although this is not the case in our example). The baseline equivalence analysis is conducted for all specified outcomes and is conducted for the full sample analysis (with associated weights), but not for subgroups. The `NO_JNT_TEST` option can be set to 1 to suppress the joint test of baseline equivalence if users specify a very large number of baseline covariates for the analysis and encounter program errors due to matrix size limitations in R or Stata.

### Screenshot 5.14: Baseline Equivalence Screen

**BASE\_EQUIV (optional)**

GENDER PRIOR\_MATH\_SCORE PRIOR\_READ\_SCORE

List of names of baseline covariates in the data file to assess baseline equivalence of the treatment and control groups

---

**NO\_JNT\_TEST**

0 = Joint tests of baseline equivalence should be conducted (default)

1 = Joint tests should not be conducted

If this option is set to 1, the joint tests of baseline equivalence will not be conducted. This option might be specified if there are a large number of baseline covariates, which could cause program errors when the joint tests are conducted.

## b. Generating output files using the interface and identifying input errors

After entering all inputs, users can generate output files by navigating to the **Generate Files** screen by clicking (i) the **Generate Files** tab or (ii) the **Generate** menu and **Generate Files** command. Users will then be directed to the following screen:

### Screenshot 5.15: Generate Output Files Screen

**This screen will allow you to provide information for several types of output files:**

1. An input specification file [.rctyes extension], produced by the interface, that saves your inputs for future use
2. A computer program in Stata [.do extension], produced by the interface, that you will need to run in a separate step to conduct the analysis using procedures you typically employ to run such programs
3. Several analysis results files produced by the computer program: a .html file containing formatted tables displaying results of the impact analysis, a .csv data file containing information from the output tables, a .log file, and a .R computer program that accesses the *RCT-YES-Graph* application for plotting the impact results

**Output Specifications (required)**

**Common base name for all files:**

**Folder location (path name) for each file...**

Input specification file:

Stata program file:

Analysis results files:

**After specifying information for all files, click the Generate Files button below:**

Generate Files

Close

Click to generate files

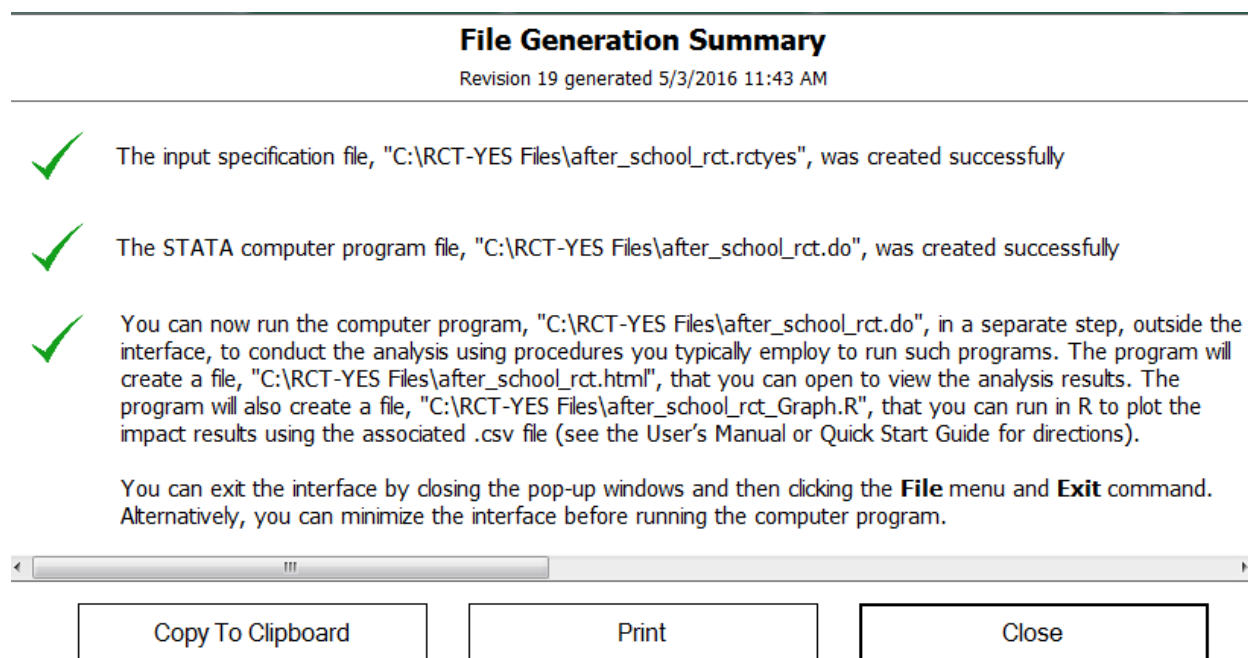
The screen will first request a common base name for all files. This name will be pre-filled if the user previously provided this information when saving the input specification file (but can be changed). The screen will then request path names, located using the **Browse** button, for three types of files:

## 5. Program inputs and outputs

1. The input specification file for future use
2. An R or Stata computer program file [.R or .do extension], to be generated by the interface, that users will need to run in a separate step to conduct the analysis
3. Several analysis results files produced by the R or Stata program:
  - a. A .html file containing formatted tables displaying results of the impact analysis
  - b. A .csv file containing information from the output tables that can be read in for further analyses and that can be used to plot the impact results using *RCT-YES-Graph*
  - c. A .log file containing detailed regression results produced by the R or Stata regression routines used for estimation, as well as R or Stata program errors
  - d. A .R computer program that accesses the *RCT-YES-Graph* application for plotting the impact results (whose base name will have a “\_graph” suffix)

Users should now press the **Generate Files** button at the bottom of the screen to produce the output files. A dialog box and **File Generation Summary** pop-up window will provide information on whether the input specification file and the computer program file were successfully generated and reasons why some files may not have been generated (see Screenshot 5.16).

### Screenshot 5.16: File Generation Summary





The interface will check that:

- All required items in the **Getting Started** and **Required Design Parameters** screens have been entered and that the input data file exists
- Each outcome domain contains at least one outcome variable. The interface will remove domains with no specified outcomes.
- Each subgroup variable is named and contains at least one subgroup category that is specified and labeled. The interface will remove problem subgroup variables or categories.
- No variable names or codes contain unallowable symbols for the language used for estimation (R or Stata). The **File Generation Summary** window will list the problem variables and users will need to fix them before generating the Stata or R computer program.
- The specified treatment status indicator, block identifier, cluster identifier, and intervention service receipt variables are not specified elsewhere. If duplicates are found, they will be listed in the **File Generation Summary** window and will need to be fixed.
- The output file information in the **Generate Output Files** screen is complete

Note that the interface does not identify data problems, such as misspelled outcomes or covariates, because the *interface does not read the input dataset*. These checks will be conducted when running the R or Stata computer program file (as discussed in the next section).

If pertinent, users should fix errors and return to the **Generate Output Files** screen to generate the output files. When finished, users should exit or minimize the interface and run the computer program file outside the interface.

### Screenshot 5.17: Files produced by the interface

Name	Date modified	Type	Size
 after_school_rct.do	3/7/2016 5:53 PM	DO File	1 KB
 after_school_rct.rctyes	3/7/2016 5:53 PM	RCTYES File	4 KB







### c. Running the R or Stata computer program and identifying errors

To conduct the analysis, users should locate the R or Stata computer program file produced by the interface, and run this program using the same procedures that they typically use to run such programs. Appendix A discusses how to download and run R and Stata.

## 5. Program inputs and outputs

The R or Stata computer program file will read the input dataset and conduct the analysis using the design and analysis inputs specified in the interface. The program will then create an analysis results (.html) file that was named in the interface containing formatted tables with analysis findings. The content of these tables will be discussed in Chapter 6 using examples. The program will also create several other files with the same base name (Screenshot 5.17). First, it will create a .log file containing regression model results (including parameter estimates for all covariates). Second, it will create a .csv file containing information in the output tables. Users can develop computer programs to read in the .csv files to conduct further analyses. Finally, it will create a .R computer program to access the *RCT-YES-Graph* application for plotting the impact findings using the .csv file (see Chapter 7).

### Screenshot 5.17: Files after running the R or Stata computer program file

Name	Date modified	Type	Size
 after_school_rct.csv	3/19/2016 12:05 PM	Microsoft Excel C...	65 KB
 after_school_rct.do	3/19/2016 12:05 PM	DO File	1 KB
 after_school_rct.html	1/6/2016 9:15 PM	HTML Document	138 KB
 after_school_rct.log	3/7/2016 5:54 PM	LOG File	37 KB
 after_school_rct.rctyes	3/19/2016 12:06 PM	RCTYES File	4 KB
 after_school_rct_graph.R	3/19/2016 12:06 PM	R File	1 KB

*RCT-YES* will run successfully only if the variables in the input data file are specified and formatted correctly. The computer program will check for data problems and indicate reasons for program errors in Table 1 of the .html file and how these problems were handled. For instance, the table will indicate if the treatment status indicator is missing for some records or contains values other than 0 and 1. The table will also list input variables that are not recognized, which could occur, for example, due to misspellings or syntax errors.

*RCT-YES* was designed not to conduct the analysis if critical errors are found rather than to use algorithms to guess how to fix the problems. For example, if the treatment status indicator is missing for a student, it is unclear whether that student is not in the study and should be omitted from the file or if the student is part of the sample but has a data problem. In these cases, users should fix the problems and rerun the computer program (and, if needed, update the interface screens and regenerate the output files). If data errors are not critical (for example, weights are missing for some cases), the program will typically “fix” the errors (for example, by ignoring the weights in the above example) and continue running. The .html file will describe the issues and how they were handled.

The .log file produced by the R or Stata computer program will display system errors. If the inputs have been specified correctly and the data file is in the correct format, errors that crash the program without a reason provided in the .html file should be rare. Users who encounter errors that they cannot resolve should use the contact support mailbox in the *RCT-YES* website.



## 6. Interpreting the analysis results

The R or Stata computer program file will produce an .html file named in the interface containing formatted tables with analysis findings and associated .csv and .log files. The .html tables will contain five types of information: (i) program errors; (ii) summary statistics on the specified outcomes, weights, covariates, and subgroups; (iii) findings from the baseline equivalence analysis; (iv) impact analysis results, and (v) an appendix table displaying key inputs specified in the interface. Where pertinent, table notes describe the calculations and notation. The LIMIT\_PRINT option can be used to suppress the printing of some output tables presenting sample statistics that can be long for some designs. The program will also create a .csv data file containing information from the output tables for further analyses or reporting (unless the CSV\_FILE option is set to 0) as well as a .log file that contains complete results from the impact estimation models. Finally, the impact findings can be graphed in R using the *RCT-YES-Graph* application that reads in the .csv file (see Chapter 7).

The .html files can be printed in hardcopy form, where each table will start on a different page. There are several options for importing the output tables into Microsoft Word or other word processing languages, and we provide a few examples. One option is to cut and paste the desired tables from the .html files into Excel (or a similar spreadsheet application) and then import the text from there into a word processor. Another option is to use a Windows snipping tool to insert the selected tables as screenshots into the desired documents. Another more flexible option is to create computer programs to read in and manipulate the .csv data file to achieve the desired tables.

This chapter first discusses the content of the output tables for our Design 1 to 4 examples from Chapters 4 and 5. To fix concepts, we focus on Design 1 because the output tables are similar for each *RCT-YES* design. For Designs 2 to 4, we focus on differences in the tables from Design 1. For simplicity, we ignore the SG\_READ\_PROF analysis. We then discuss the content of the .csv file.

### a. Design 1 example

To demonstrate the program output, in what follows, we describe the 9 formatted tables for our Design 1 example where students are the unit of random assignment; we refer to these tables as “Output Tables,” and provide excerpts from selected tables at the end of this section.

**Output Table 1. *Input specification errors (not shown).*** This information allows users to identify and fix errors in the input data file or interface specifications.

**Output Table 2. *Sample sizes and key summary statistics for study outcomes and weights for the full sample, by treatment-control status.*** The output includes sample sizes, missing data rates, variable means, and variable standard deviations for the full sample analysis. In our example, we find that MATH\_SCORE is missing for 10 percent of treatment and control group students and

## 6. Analysis results

has a standard deviation of 12.6 scale points for treatment group students and 13.1 scale points for control group students.

Although not germane to our example, values of “-” in the table indicate that the outcome was excluded from the analysis to help protect the disclosure of personally identifiable information (PII). As discussed in Chapter 2, *RCT-YES* will exclude outcomes (and covariates and subgroups) if sample sizes are too small, using rules adopted by individual states for reporting SLDS outcomes. Most states have set a minimum group size for reporting SLDS data of 10 students—the default in *RCT-YES*—but this threshold can be changed using the `MIN_NUM` input (that must be at least 3). The program checks that the minimum size threshold holds for *both* the treatment and control groups. The program also excludes outcomes not found in the dataset, those with insufficient variation, and binary outcomes that pertain to very rare or common events (those with fewer than 5 observations with a value of 0 or fewer than 5 observations with a value of 1 for either research group).

***Output Table 3. Percentiles of the distributions of study outcomes and weights, by treatment-control status.*** The table displays variable distributions (5th, 25th, 50th, 75th, and 95th percentiles) so that users can assess data quality and the presence of outliers. The  $p$ th percentile of the distribution of a variable is a number such that  $p$  percent of the sample has a variable value less than or equal to that number. In our example, we see that the 75<sup>th</sup> percentile of `MATH_SCORE` is 69.2 for the treatment group, which means that 75 percent of students in the treatment group with nonmissing test score data have a test score value less than or equal to 69.2; the corresponding value for the control group is 65.

***Output Table 5. Sample sizes for baseline subgroups.*** Sample size and missing data information for each specified subgroup is presented separately by outcome and treatment-control status. In our example, `SG_MATH_PROF` is missing for 89 treatment group and 96 control group students (among the sample with available `MATH_SCORE` data).

Although not pertinent to our example, the table also indicates whether a subgroup is excluded from the analysis because of small sample sizes (using the same rules as discussed above for Output Table 3). Importantly, if any subgroup category is too small for either research group, the *entire* subgroup is excluded from the analysis. In these cases, users should combine small subgroups into larger ones.

***Output Table 6. Baseline covariates for the full sample regression analysis.*** This table displays three types of information for optional regression analyses. First, the table indicates whether a covariate is excluded from the analysis because it is not found in the data file or for any of the following reasons (none of which apply for our example):

- There are too few sample members per covariate. The default ratio is 5, but the ratio can be changed using the `OBS_COV` option.

- There are too many missing covariate values for either the treatment or control group. The default cutoff is 30 percent missing data, but this cutoff can be changed using the `MISSING_COV` option.
- The covariate has a zero standard deviation or, for binary outcomes, a mean very close to 0 or 100 percent for either research group (using the same rules as for binary outcomes above)
- The correlation between the covariate and outcome is 1.0 or -1.0. This could occur, for example, if the outcome variable is inadvertently included as a covariate.

Second, the table reports the regression  $R^2$  value (squared partial correlation coefficient) when each covariate is regressed on the others. If a covariate is highly collinear with the others (for example, an  $R^2$  value of greater than 0.90), users might consider omitting the covariate from the analysis to avoid needless losses in the degrees of freedom for hypothesis testing. In our example, the  $R^2$  value when `PRIOR MATH_SCORE` is regressed on `PRIOR_READ_SCORE` and `GENDER` is 0.58 for the treatment group and 0.53 for the control group; a similar  $R^2$  value is found if `PRIOR_READ_SCORE` is regressed on `PRIOR_MATH_SCORE` and `GENDER`.

Finally, the last two columns of the table display correlation coefficients between the covariates and outcomes to help users identify covariates that can most improve the precision of the impact estimates. These correlations are calculated separately for the treatment and control groups. In our example, the correlations between prior year test scores and follow-up test scores range from about 0.44 to 0.69; there is little correlation between gender and the test score outcomes.

**Output Table 7. Baseline covariates used in the full sample and subgroup regression analyses.**

This table lists the covariates included in the regression models for the impact analysis. Some covariates included in the inputs may be omitted from these lists for the reasons discussed in connection with Output Table 6, but this does not occur for our example.

**Output Table 8. Assessing baseline equivalence of the treatment and control groups in the analysis sample.** The baseline equivalence analysis is conducted using the full sample. For each specified covariate and outcome with nonmissing data, `RCT-YES` displays treatment and control group means, the difference between the two means, the difference in effect size units, the standard error of the difference, and the p-value of the difference with an attached symbol \* indicating statistical significance at the 5 percent level. All figures are calculated using sample weights if specified (see Chapter 2k).

The table also displays p-values from hypothesis tests to assess whether covariate means are *jointly* similar. This test accounts for potential dependencies among the covariates and multiple testing issues and is a good summary statistic to assess baseline equivalence. This test is conducted using the sample that has available data for all covariates and is performed if the sample size is large relative to the number of covariates.

## 6. Analysis results

Using our example, we see in Output Table 8 that among students in the analysis sample with available data for MATH\_SCORE, we find no statistically significant differences between the treatment and control groups in their mean values for the three considered baseline characteristics. Of particular importance, the treatment-control difference between the prior year math test score (0.34 scale points) is not statistically significant at the 5 percent level (p-value of 0.579), and the p-value for the joint test of significance across all three baseline covariates is 0.733. A similar pattern holds using the sample with available data for READ\_SCORE, although there is some evidence of a treatment-control difference in the PRIOR\_READ\_SCORE variable.

**Output Table 9. *Impacts on study outcomes for the full sample and baseline subgroups.*** RCT-YES reports estimated impacts using regression methods if baseline covariates are specified for the analysis; otherwise, the program reports estimated impacts using simple differences-in-means methods. For both specifications, the impact findings are reported using a similar format as for the baseline equivalence analysis.

To report impact estimates, RCT-YES presents the *unadjusted control group mean* and the *adjusted treatment group mean* calculated as the sum of the unadjusted control group mean and the (regression-adjusted) impact estimate. All figures are calculated using specified weights or the default weights (for blocked and clustered designs) discussed in Chapter 2k.<sup>6</sup>

For full sample analyses, the output indicates that an impact estimate remains statistically significant after applying a multiple comparisons correction using the symbol  $\wedge$  after the \* symbol attached to the p-value. For regression analyses,  $R^2$  values are presented in table notes.

Referring to Output Table 9, a summary of key impact findings for our example is as follows:

- ***The estimated impact of the after-school intervention on math test scores for the full sample is about 3.4 scale points, which is statistically significant at the 5 percent level (p-value of 0.001).*** The mean math score is 59.54 for the treatment group, compared to 56.17 for the control group. The estimated impact of 3.4 scale points translates into an estimated impact in *effect size units* of 0.26 standard deviations. The estimated impact remains statistically significant after applying a statistical correction to adjust the p-values for the two hypothesis tests that are being conducted for the full sample analysis using the math and reading test score outcomes.
- ***The intervention had no overall effect on reading test scores.*** The mean reading test score value is about 70 for both the treatment and control groups. The impact estimate of 0.10 scale points is not statistically significant (the p-value is 0.822).

---

<sup>6</sup> For the CACE analysis, RCT-YES reports the control group mean for compliers.

- ***Intervention effects on math test scores differed by math proficiency level in the year prior to the intervention but not by gender.*** The p-value of 0.011 in the header row for SG\_MATH\_PROF suggests that intervention effects differed based on the student's math proficiency level in the prior year, with evidence that the intervention was more effective for those with lower than higher baseline math scores. The p-value of 0.850 in the header row for GENDER means that the difference in the estimated impacts for boys and girls is not statistically significant: the intervention improved math scores for both girls and boys.
- ***The intervention had no effect on reading test scores for any of the considered subgroups.*** The estimated impacts are not statistically significant for boys or girls or for any subgroup defined by a student's reading proficiency level at baseline.

**Appendix Table. Summary of key input specifications.** This table displays information on key inputs specified in the interface, including key design and analysis parameters and the input data and output files. Users can reference this information to help interpret the analysis findings.

## b. Design 2 example

For our Design 2 example, students are randomized separately within school districts (blocks). The program output for Design 2 is very similar to the output for Design 1. The key differences are that the program produces the following additional information for Design 2:

- ***Summary statistics on block sample sizes and weights.*** This information can be used to determine the number of blocks excluded from the analysis and how RCT-YES weights blocks to obtain overall impact findings. Output Table 4a for Design 2 displays these summary statistics for our hypothetical RCT for Design 2.
- ***Summary statistics on the impact estimates across blocks.*** RCT-YES does not report impact estimates for each block due to PII concerns for small blocks, but it provides the following summary statistics if the sample contains at least three blocks: (i) the standard deviation, (ii) the range, and (iii) the percentage of block impact estimates with a positive sign. This information is important to help interpret the impact findings, because the study results could have different policy implications if the block impact estimates are similar or vary considerably across blocks. RCT-YES also reports the p-value from a hypothesis test to gauge the statistical significance of differences in the block impact estimates.

Output Table 10 for Design 2 displays summary statistics on the impact estimates across blocks using our Design 2 example. We see that there is some variation in the district-level impact estimates for math test scores. The joint test of variation in the impact estimates is statistically significant and the range is greater than 10 scale points. Furthermore, the standard deviation is 2.9, so that the coefficient of variation (the standard deviation divided by the mean) is 0.86, indicating a distribution

## 6. Analysis results

with modest variation. Nonetheless, about 90 percent of the impact estimates have a positive sign, suggesting consistency in effects across districts. For reading test scores, the joint test of variation is statistically significant, suggesting that the intervention produced beneficial effects in some sites but not others, and that more exploratory analyses should be conducted to examine this variation.

### Selected Output Tables for Design 1

Table 2. Sample sizes and key summary statistics for study outcomes for the full sample, by treatment-control status

Outcome	Number in Sample	Number with Available Data	Number with Missing Data	Percentage with Available Data	Mean	Standard Deviation
<b>Achievement Test Scores</b>						
<b>Individual-level data</b>						
<b>Treatment group</b>						
MATH_SCORE	1,073	971	102	90	60.05	12.58
READ_SCORE	1,073	977	96	91	71.37	14.05
<b>Control group</b>						
MATH_SCORE	1,183	1,067	116	90	56.17	13.14
READ_SCORE	1,183	1,062	121	90	70.13	13.29

Notes: Values of "-" indicate that the variable is excluded from the analysis due to the potential for the disclosure of personally identifiable information (PII). The main reasons for exclusion of an outcome variable are small sample sizes, insufficient variation across the outcome, very rare or common events for binary outcomes, or the input name for the outcome is not found in the data file. The means and standard deviations are unweighted, and are presented for binary (0 or 1) outcomes as percentages and without decimals to help minimize the disclosure of PII.

Table 3. Percentiles of the distributions of study outcomes for the full sample, by treatment-control status

Outcome	Percentiles of the distribution				
	5 <sup>th</sup>	25 <sup>th</sup>	50 <sup>th</sup>	75 <sup>th</sup>	95 <sup>th</sup>
<b>Achievement Test Scores</b>					
<b>Individual-level data</b>					
<b>Treatment group</b>					
MATH_SCORE	38.20	52.21	60.72	69.20	79.34
READ_SCORE	46.31	61.30	72.83	81.93	91.49
<b>Control group</b>					
MATH_SCORE	34.06	48.14	56.83	65.04	76.50
READ_SCORE	47.80	61.19	71.04	79.63	90.11

Notes: The  $p$ th percentile of the distribution of a variable is a number such that  $p$  percent of the sample has a variable value less than or equal to that number. For example, if 112 is the 75th percentile of an outcome variable, 75 percent of the sample with nonmissing outcome data has an outcome value less than or equal to 112. These figures can be used to assess unusually large or small values (outliers) or other data problems.

Table 5. Sample sizes for baseline subgroups

Outcome and Subgroup	Subgroup Excluded from the Analysis	Treatment Group		Control Group	
		Number with Available Data	Number with Missing Data	Number with Available Data	Number with Missing Data
<b>Achievement Test Scores</b>					
<b>Individual-level data</b>					
<b>MATH_SCORE: Math Test Scores</b>					
Full Sample		971	102	1,067	116
SG_MATH_PROF		853	89	933	96
1 = Not Proficient in Math		307	34	341	36
2 = Proficient in Math		472	51	528	57
3 = Highly Proficient in Math		74	4	64	3
GENDER		971	102	1,067	116
1 = Girls		516	56	551	63
0 = Boys		455	46	516	53
<b>READ_SCORE: Reading Test Scores</b>					
Full Sample		977	96	1,062	121
SG_MATH_PROF		853	89	920	109
1 = Not Proficient in Math		300	41	335	42
2 = Proficient in Math		483	40	530	55
3 = Highly Proficient in Math		70	8	55	12
GENDER		977	96	1,062	121
1 = Girls		522	50	552	62
0 = Boys		455	46	510	59

Notes: Subgroups with small sample sizes are excluded from the analysis to help guard against the disclosure of personally identifiable information (PII). If any subgroup category is too small for either the treatment or control group, the entire subgroup is excluded from the analysis.

## 6. Analysis results

**Table 6. Baseline covariates for the full sample regression analysis: reasons for exclusion, collinearities among the covariates, and correlations with outcomes**

Outcome and Covariate	Reasons for Exclusion			Squared Partial Correlation ( $R^2$ ) with Other Covariates		Correlation with Outcome	
	Too Few Cases per Covariate	Too Many Missing Values	Not Enough Variation	Treatment Group	Control Group	Treatment Group	Control Group
<b>Achievement Test Scores</b>							
<b>MATH_SCORE: Math Test Scores</b>							
GENDER				0.00	0.01	-0.01	-0.02
PRIOR_MATH_SCORE				0.58	0.53	0.44	0.48
PRIOR_READ_SCORE				0.58	0.53	0.51	0.57
<b>READ_SCORE: Reading Test Scores</b>							
GENDER				0.00	0.00	-0.01	0.00
PRIOR_MATH_SCORE				0.57	0.53	0.58	0.55
PRIOR_READ_SCORE				0.57	0.53	0.64	0.69

Notes: Covariates are excluded from the regression analysis for several possible reasons: (1) there are too few cases per covariate; (2) there are too many missing covariate values for either the treatment or control group; (3) the covariate has a zero standard deviation or has a mean very close to 0 or 1 for binary covariates; (4) the correlation between the covariate and outcome is 1.0 or -1.0; or (5) the name of the covariate is not found in the data file. If a covariate is highly collinear with the others (for example with a squared partial correlation of greater than .90), users might consider omitting the covariate from the analysis to avoid needless losses in the degrees of freedom for hypothesis testing. The table displays bivariate correlation coefficients between the covariates and outcomes to help users identify covariates that can most improve the precision of the impact estimates. Some covariates may subsequently be dropped because they are collinear with each other or with covariates generated by the program for blocked designs (see the .log file produced by the program).

**Table 7. Baseline covariates used in the full sample and baseline subgroup regression analyses**

Outcome and Subgroup	List of Covariates Included in the Regression Analysis
<b>Achievement Test Scores</b>	
<b>MATH_SCORE: Math Test Scores</b>	
Full Sample	GENDER PRIOR_MATH_SCORE PRIOR_READ_SCORE
SG_MATH_PROF	GENDER PRIOR_READ_SCORE
GENDER	PRIOR_MATH_SCORE PRIOR_READ_SCORE
<b>READ_SCORE: Reading Test Scores</b>	
Full Sample	GENDER PRIOR_MATH_SCORE PRIOR_READ_SCORE
SG_MATH_PROF	GENDER PRIOR_READ_SCORE
GENDER	PRIOR_MATH_SCORE PRIOR_READ_SCORE

Notes: Some covariates specified in the program inputs might be omitted from the list of covariates for the reasons provided in Table 6. Some covariates may subsequently be dropped because they are collinear with each other or with covariates generated by the program for blocked designs and subgroup analyses (see the .log file produced by the program).



**Table 8. Assessing baseline equivalence of the treatment and control groups in the analysis sample**

Outcome and Covariate	Treatment Group Mean	Control Group Mean	Difference	Effect Size	Standard Error of Difference	p-Value of Difference
<b>Achievement Test Scores</b>						
<b>MATH_SCORE: Math Test Scores</b>						
GENDER	53	52	2	0.03	2	0.498
PRIOR_MATH_SCORE	69.81	69.48	0.34	0.03	0.60	0.579
PRIOR_READ_SCORE	71.53	70.42	1.11	0.07	0.71	0.118
<b>Joint test</b>						<b>0.733</b>
<b>READ_SCORE: Reading Test Scores</b>						
GENDER	53	52	1	0.03	2	0.512
PRIOR_MATH_SCORE	70.01	69.27	0.74	0.06	0.60	0.217
PRIOR_READ_SCORE	71.86	70.05	1.81	0.12	0.70	0.010*
<b>Joint test</b>						<b>0.315</b>
<b>Sample Size</b>	<b>977</b>	<b>1,067</b>	<b>2,044</b>			

Notes: The analysis is conducted using the full sample where cases with missing data for the baseline covariate and outcome under investigation are excluded from the analysis. The effect size is the treatment-control difference divided by the standard deviation of the covariate for individuals in the treatment and control groups. The findings for binary (0 or 1) outcomes are presented as percentages.

Values of "\*" indicate that the baseline covariate is excluded from the analysis because the input name for the covariate is not found in the data file or due to the potential for the disclosure of personally identifiable information (PII). The reasons for exclusion due to PII-related reasons are small sample sizes, insufficient variation across the outcome, or very rare or common events for binary outcomes.

The sample size is the maximum sample size across the baseline covariates.

\* Difference is statistically significant at the 0.05 level, two-tailed test.

## 6. Analysis results

**Table 9. Impacts on study outcomes for the full sample and baseline subgroups**

Outcome and Subgroup	Treatment Group Mean	Control Group Mean	Difference (Impact Estimate)	Effect Size	Standard Error of Difference	p-Value of Difference
<b>Achievement Test Scores</b>						
<b>MATH_SCORE: Math Test Scores</b>						
Full Sample	59.54	56.17	3.37	0.26	0.48	0.000* <sup>^</sup>
SG_MATH_PROF						0.011 <sup>^</sup> *
1=Not Proficient in Math	54.17	49.18	4.99	0.38	0.87	0.000*
2=Proficient in Math	62.39	59.28	3.11	0.23	0.66	0.000*
3=Highly Proficient in Math	70.76	70.99	-0.23	-0.02	1.54	0.883
GENDER						0.850 <sup>^</sup>
1=Girls	59.36	55.91	3.45	0.26	0.66	0.000*
0=Boys	59.72	56.45	3.27	0.25	0.69	0.000*
<b>READ_SCORE: Reading Test Scores</b>						
Full Sample	70.23	70.13	0.10	0.01	0.45	0.822
SG_MATH_PROF						0.500 <sup>^</sup>
1=Not Proficient in Math	61.88	62.26	-0.38	-0.03	0.82	0.646
2=Proficient in Math	74.51	73.94	0.57	0.04	0.61	0.349
3=Highly Proficient in Math	86.65	87.57	-0.91	-0.07	1.50	0.543
GENDER						0.600 <sup>^</sup>
1=Girls	70.06	70.18	-0.12	-0.01	0.62	0.843
0=Boys	70.43	70.08	0.35	0.03	0.66	0.595
<b>Sample Size</b>	<b>977</b>	<b>1,067</b>	<b>2,044</b>			

Notes: The impact estimates are calculated using simple differences-in-means methods or regression models that control for baseline covariates if specified. Cases with missing data for the outcome and subgroup under investigation are excluded from the analysis. The control group means are sample means, and the treatment group means are calculated by summing the control group means and the impact estimates. The effect size is the treatment-control difference divided by the standard deviation of the outcome for individuals in the control group. All estimates are obtained using weights if specified.

The sample size is the maximum sample size across the outcomes for the full sample analysis.

Regression  $R^2$  values for the full sample analysis are: 0.32 for MATH\_SCORE; 0.46 for READ\_SCORE.

\* Difference is statistically significant at the 0.05 level, two-tailed test.

<sup>^</sup> Difference remains statistically significant at the 0.05 level, two-tailed test after applying the Benjamini-Hochberg correction for multiple hypothesis testing across all full sample analyses in the same domain.

<sup>^</sup> Indicates p-values to test for differences in impacts across subgroup categories.

## Appendix. Summary of key input specifications

Input	Specification
<b>Key Design and Analysis Inputs</b>	
Statistical package	STATA
Input data file	
Name	after_school_rot_data1.dta
Folder location	C:/RCT-YES Files/Gen Data
<b>Required parameters</b>	
DESIGN	1 = Non-clustered, non-blocked
TC_STATUS	TREATMENT
<b>Optional parameters</b>	
SUPER_POP	0 = Finite-population (FP) model
MISSING_COV	30 = Maximum percentage of missing data for a baseline covariate to be included in the regression models
OBS_COV	5 = Required ratio of the number of observations (or clusters) per covariate for the regression analysis to be performed
MIN_NUM	10 = Minimum group size adopted by the state or other entity for reporting outcomes to protect personally identifiable information (PII)
ALPHA_LEVEL	5 = Significance level used for hypothesis testing (in percents)
LIMIT_PRINT	0 = All output tables printed
CSV_FILE	1 = .csv data file produced
<b>Outcomes, Weights, Covariates, and Subgroups</b>	
<b>Achievement Test Scores</b>	
OUTCOMES	MATH_SCORE, READ_SCORE
WEIGHTS (Full sample)	0, 0
COVARIATES (Full sample)	GENDER, PRIOR_MATH_SCORE, PRIOR_READ_SCORE
GOT_TREAT	
SUBGROUPS	SG_MATH_PROF, GENDER
<b>Baseline equivalence analysis</b>	
List of variables	GENDER, PRIOR_MATH_SCORE, PRIOR_READ_SCORE
NO_JNT_TEST	0 = Joint test of baseline equivalence should be conducted
<b>Files generated by the interface</b>	
Common base name	after_school_rot
Folder locations	
Input specification file (.rotyes)	C:/RCT-YES Files
Stata program file (.do)	C:/RCT-YES Files
Analysis results file (.html, .csv)	C:/RCT-YES Files

## 6. Analysis results

### Selected Output Tables for Design 2

Table 4a. Block sample sizes for the full sample analysis

Outcome and Block	Block Excluded from the Analysis	Treatment Group		Control Group		Block Weight <sup>a</sup>
		Number with Available Data	Number with Missing Data	Number with Available Data	Number with Missing Data	
<b>Achievement Test Scores</b>						
<b>MATH_SCORE: Math Test Scores</b>						
Full Sample		971	102	1,067	116	
DISTRICT						
1		147	16	148	15	295
2		94	11	69	9	163
3		127	15	114	12	241
4		50	7	150	13	200
5		47	7	59	5	106
6		91	9	56	13	147
7		107	6	165	17	272
8		96	9	159	20	255
9		111	13	62	4	173
10		101	9	85	8	186
<b>READ_SCORE: Reading Test Scores</b>						
Full Sample		977	96	1,062	121	
...						

Notes: Blocks with small sample sizes are excluded from the analysis so that proper variance estimates can be calculated.

<sup>a</sup> The block weight is used to aggregate the block impact estimates to calculate overall impact estimates. If no weights are specified, the block weight is the total number of treatment and control group individuals in the block.

Table 10. Variation in impact estimates across blocks for the full sample analysis

Outcome and Subgroup	Standard Deviation	Range <sup>a</sup>	Proportion Positive	p-Value from Joint Test of Differences
<b>Achievement Test Scores</b>				
MATH_SCORE	2.9	10.3	90	0.004*
READ_SCORE	3.0	9.0	70	0.000*
<b>Number of Blocks</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>

\* The difference in impact estimates across blocks is statistically significant at the 0.05 level, two-tailed test.

<sup>a</sup> The range is the difference between the largest and smallest block-specific impact estimate.

### c. Design 3 example

Our Design 3 example is a clustered design where schools rather than students are randomized to the treatment and control groups, and where the input data file contains student-level data. The program output for Design 3 is very similar to the program output for Design 1. The key differences are that the program produces the following additional information for Design 3:

- ***Summary statistics on cluster sample sizes and weights.*** This information can be used to determine how RCT-YES weights clusters to obtain overall impact estimates and how much the weighting scheme might matter. Output Table 4b for Design 3 below displays this information for our RCT example.
- ***Estimates of intraclass correlations (ICCs).*** “Design effects” for a clustered RCT design are typically defined as the inflation in the variance estimates due to clustering relative to a non-clustered design of the same size (see Schochet, 2016). The calculation of design effects is often expressed in terms of the intraclass correlation coefficient (ICC), which is the proportion of variance in the outcome that lies between clusters. The ICC is an important parameter to help interpret variance estimates for clustered designs (and to calculate statistical power to assess appropriate sample sizes when designing clustered RCTs in the future). Thus, the program prints out ICCs in table notes in Table 9 for each specified outcome measure using the full sample.

Output Table 9 displays impact findings for our hypothetical RCT for Design 3 with individual-level data. The impact findings are very similar to those discussed for Design 1: participation in the tested after-school programs improved student math test scores for the full sample and broadly across the considered baseline subgroups, but participation did not improve student reading test scores. As expected, standard errors for Design 3 are larger than for Design 1 because of the clustering (nesting) of students within schools.

### d. Design 4 example

Our Design 4 example pertains to a clustered, blocked RCT design where schools are randomly assigned within school districts. Thus, the program output for Design 4 combines the outputs for Designs 2 and 3. The output tables provide summary information on block and cluster sample sizes, findings from the baseline equivalency analysis, and results from the impact analyses for the full sample and for specified subgroups, including summary statistics on the variation in impact estimates across blocks. We do not provide examples of these tables.

## 6. Analysis results

### Selected Output Tables for Design 3

Table 4b. Cluster sample sizes for the full sample analysis

Outcome and Cluster	Cluster Excluded from Analysis	Treatment (1) or Control (0) Group Status	Number of Individuals with Available Data	Number of Individuals with Missing Data	Cluster Weight <sup>a</sup>
<b>Achievement Test Scores</b>					
<b>MATH_SCORE: Math Test Scores</b>					
Full Sample			2,038	218	
SCHOOL					
1		0	75	8	1
2		1	90	9	1
3		1	57	7	1
4		0	73	7	1
5		1	57	7	1
6		0	28	1	1
7		1	37	4	1
8		0	41	8	1
9		0	59	7	1
10		1	90	12	1
11		0	55	5	1
12		1	37	3	1
13		0	53	5	1
14		0	48	3	1
15		1	50	7	1
...					

Notes: Clusters are excluded from the analysis if they do not contain any individuals with available outcome data.

<sup>a</sup> The cluster weight is used to aggregate cluster-level mean outcomes to calculate overall impact estimates. If no weights are specified, the cluster weight is 1.

**Table 9. Impacts on study outcomes for the full sample and baseline subgroups**

Outcome and Subgroup	Treatment Group Mean	Control Group Mean	Difference (Impact Estimate)	Effect Size	Standard Error of Difference	p-Value of Difference
<b>Achievement Test Scores</b>						
<b>MATH_SCORE: Math Test Scores</b>						
Full Sample	58.50	55.88	2.62	0.20	0.74	0.001* <sup>^</sup>
SG_MATH_PROF						0.006 <sup>§</sup> *
1=Not Proficient in Math	54.14	49.20	4.94	0.36	1.17	0.000*
2=Proficient in Math	61.61	58.84	2.77	0.20	0.90	0.004*
3=Highly Proficient in Math	69.95	70.71	-0.76	-0.05	1.63	0.648
GENDER						0.994 <sup>§</sup>
1=Girls	58.41	55.56	2.84	0.22	0.87	0.002*
0=Boys	59.19	56.36	2.84	0.22	0.86	0.002*
<b>READ_SCORE: Reading Test Scores</b>						
Full Sample	69.39	69.81	-0.42	-0.03	0.80	0.600
SG_MATH_PROF						0.454 <sup>§</sup>
1=Not Proficient in Math	62.09	62.56	-0.47	-0.03	1.30	0.720
2=Proficient in Math	74.27	73.35	0.93	0.07	1.06	0.386
3=Highly Proficient in Math	83.61	86.77	-3.16	-0.22	3.62	0.390
GENDER						0.450 <sup>§</sup>
1=Girls	69.34	69.89	-0.55	-0.04	0.86	0.525
0=Boys	69.97	69.86	0.11	0.01	0.96	0.910
<b>Sample Size</b>						
<b>Clusters</b>	<b>18</b>	<b>21</b>	<b>39</b>			
<b>Individuals</b>	<b>977</b>	<b>1,067</b>	<b>2,044</b>			

Notes: The impact estimates are calculated using simple differences-in-means methods or regression models that control for baseline covariates if specified. Cases with missing data for the outcome and subgroup under investigation are excluded from the analysis. The control group means are sample means, and the treatment group means are calculated by summing the control group means and the impact estimates. The effect size is the treatment-control difference divided by the standard deviation of the outcome for individuals in the control group. All estimates are obtained using weights if specified.

The sample size is the maximum sample size across the outcomes for the full sample analysis.

Regression  $R^2$  values for the full sample analysis are: 0.77 for MATH\_SCORE; 0.80 for READ\_SCORE.

Intraclass correlation coefficient (ICC) values for the full sample analysis are: 0.10 for MATH\_SCORE; 0.14 for READ\_SCORE.

\* Difference is statistically significant at the 0.05 level, two-tailed test.

<sup>^</sup> Difference remains statistically significant at the 0.05 level, two-tailed test after applying the Benjamini-Hochberg correction for multiple hypothesis testing across all full sample analyses in the same domain.

<sup>§</sup> Indicates p-values to test for differences in impacts across subgroup categories.

## 6. Analysis results

### e. Content of the .csv data file

By default, the R or Stata computer program will create a .csv file containing two types of information: (i) data from the .html output tables and (ii) the full set of inputs entered into the interface. The created .csv file will have the same base name and folder location as the .html file. Users can write computer programs to read in the .csv file for further analyses and reporting, and can also import the .csv file into the *RCT-YES-Graph* app to plot the impact findings.

Table 10 displays the record layout of the .csv file. The rows of the file correspond directly to the tables in the .html file that can be identified using the “table\_id” variable. The file is also organized by outcome domain, outcome variable, subgroup variable, and other fields varying by the design and table number. The file layout should be apparent by comparing the .csv and .html files. All variable names are in lowercase letters.

The .csv file will always contain 90 variables (columns) where each one corresponds to a specific table entry. Thus, each variable will typically contain many blank values. For example, Output Table 10 is produced only for certain blocked designs, but the .csv file will always contain variables for that table even if they are always blank. Some variables are provided to help users sort the data for their analyses (for example, to help order outcomes and subgroups). The final two columns and final rows of the file provide information from the .rctyes file on all program inputs that users entered into the interface.

The box below shows computer code in R, Stata, and SAS to read in the .csv file containing the variables listed in Table 10. Users can then write computer programs to manipulate these data keeping only the appropriate observations and variables for the desired analyses.

#### Computer code to read in the .csv file

##### R code on command line:

```
data <- read.csv("csvfilename.csv", header=TRUE, stringsAsFactors=FALSE)
```

##### Stata code in command window:

```
insheet using "csvfilename.csv", comma
```

##### SAS code:

```
proc import datafile = " csvfilename.csv"  
  out = rct_yes_output dbms = csv replace;  
  getnames = yes;  
  guessingrows=100000;  
run;
```



**Table 10. Record layout of the .csv file**

Order	Variable Name	Variable Type	Output Table	Description
1	table_id	String	All	Table number (corresponding to the .html output table)
2	domain	Numeric	All	Outcome domain number (for sorting)
3	domain_name	String	All	Outcome domain name
4	outcome	Numeric	All	Outcome variable number
5	outcome_name	String	All	Outcome variable name
6	outcome_label	String	All	Outcome variable label
7	outcome_std	Numeric	All	Outcome-specific user-specified standard deviation
8	got_treat	Numeric	All	Service receipt indicator variable number
9	got_treat_name	String	All	Service receipt indicator variable name
10	subgroup	Numeric	All	Subgroup number
11	subgroup_name	String	All	Subgroup name
12	sglevel	Numeric	5, 9, 9a, 9b	Subgroup category number
13	sglevel_value	String	5, 9, 9a, 9b	Subgroup category value
14	sglevel_label	String	5, 9, 9a, 9b,	Subgroup category label
15	binary	Numeric	2, 3, 6, 8, 9, 9a, 9b, 10	Variable is binary (1=Yes; 0=No)
16	tc	Numeric	2, 3	Treatment indicator (1=Treatment; 0=Control)
17	variable_type	Numeric	2, 3	Variable type (1=OUTCOME, 2=GOT_TREAT, or 3=WEIGHT variable)
18	variable_type_name	String	2, 3	Variable type name (an OUTCOME, GOT_TREAT, or WEIGHT variable)
19	variable	String	2, 3	Variable name
20	level	Numeric	2, 3, 5	Unit of observation (1=individuals, 2=clusters)
21	level_name	String	2, 3, 5	Unit of observation (individuals or clusters)
22	block	Numeric	4	Block number
23	block_name	String	4	Block name
24	clust	Numeric	4	Cluster number
25	clust_name	String	4	Cluster name
26	bad_block	Numeric	4	Block is invalid (1=Yes; 0=No)
27	bad_clust	Numeric	4	Cluster is invalid (1=Yes; 0=No)
28	covar	Numeric	6	Covariate number
29	covar_name	String	6	Covariate name
30	bequiv	Numeric	8	Baseline equivalency number
31	bequiv_name	String	8	Baseline equivalency name
32	bequiv_valid	Numeric	8	Baseline equivalency variable is valid (1=Yes; 0=No)

## 6. Analysis results

Order	Variable Name	Variable Type	Output Table	Description
33	weight_used	String	8, 9, 9a, 9b, 10	Weight variable used for the analysis (blank if no weight used)
34	covars_used	String	9, 9a, 9b, 10	Covariates used for analyses (blank if no covariates used)
35	any_excl	Numeric	2, 5	Any covariate excluded (1=Yes; 0=No)
36	missing_cov	String	6	Exclusion reason: too many missing values ("X" or missing)
37	zero_sd	String	6	Exclusion reason: not enough variation ("X" or missing)
38	too_few	String	6	Exclusion reason: too few cases/blocks/clusters per covariate ("X" or missing)
39	corr_abs1	String	6	Exclusion reason: the correlation between covariate and outcome is 1.0 or -1.0 ("X" or missing)
40	n_sample	Numeric	2	Number in sample
41	n_avail	Numeric	2	Number with available data
42	n_miss	Numeric	2	Number with missing data
43	pct_avail	Numeric	2	Percentage with available data (0 to 100)
44	mean	Numeric	2	Mean
45	sd	Numeric	2	Standard deviation
46	p5	Numeric	3	5th percentile
47	p25	Numeric	3	25th percentile
48	p50	Numeric	3	50th percentile
49	p75	Numeric	3	75th percentile
50	p95	Numeric	3	95th percentile
51	n_avail_t	Numeric	4, 5	Number with available data for treatments
52	n_miss_t	Numeric	4, 5	Number with missing data for treatments
53	n_avail_c	Numeric	4, 5	Number with available data for controls
54	n_miss_c	Numeric	4, 5	Number with missing data for controls
55	swb	Numeric	4	Block or cluster weight
56	r2_t	Numeric	6	Squared partial correlation with other covariates for treatment
57	rho_t	Numeric	6	Correlation with outcome for treatments
58	r2_c	Numeric	6	Squared partial correlation with other covariates for controls
59	rho_c	Numeric	6	Correlation with outcome for controls
60	table_nt	Numeric	8, 9, 9a, 9b	Sample size for treatments (individuals or clusters depending on design)
61	table_nc	Numeric	8, 9, 9a, 9b	Sample size for controls (individuals or clusters depending on design)
62	table_n	Numeric	8, 9, 9a, 9b	Sample size overall (individuals or clusters depending on design)

Order	Variable Name	Variable Type	Output Table	Description
63	table_indivnt	Numeric	8, 9, 9a, 9b	Number of individuals for treatments (for DESIGN 3 and 4, CLUSTER_DATA=1 only)
64	table_indivnc	Numeric	8, 9, 9a, 9b	Number of individuals for controls (for DESIGN 3 and 4, CLUSTER_DATA=1 only)
65	table_indivn	Numeric	8, 9, 9a, 9b	Number of individuals overall (for DESIGN 3 and 4, CLUSTER_DATA=1 only)
66	ybart	Numeric	8, 9, 9a, 9b	Treatment group mean
67	ybarc	Numeric	8, 9, 9a, 9b	Control group mean
68	impact	Numeric	8, 9, 9a, 9b	Difference (Impact Estimate)
69	effect_size	Numeric	8, 9, 9a, 9b	Effect Size
70	se_impact	Numeric	8, 9, 9a, 9b	Standard error of difference
71	p_impact	Numeric	8, 9, 9a, 9b	p-Value of difference
72	s_impact	String	8, 9, 9a, 9b	Significance marker ("*" if significant at the ALPHA_LEVEL; blank otherwise)
73	conf_lower	Numeric	8, 9, 9a, 9b	Lower confidence limit for impact
74	conf_upper	Numeric	8, 9, 9a, 9b	Upper confidence limit for impact
75	conf_lower	Numeric	8, 9, 9a, 9b	Lower confidence limit for effect_size
76	conf_upper	Numeric	8, 9, 9a, 9b	Upper confidence bound for effect_size
77	bh_sig	String	9, 9b	Significance marker after applying the Benjamini-Hochberg correction ("^" if significant at the alpha_level; blank if not)
78	joint_pval	Numeric	8	p-Value for the joint significant test for the baseline equivalence analysis
79	pvalchi	Numeric	9, 9a, 9b	p-Values to test for differences in impacts across subgroups
80	schi	String	9, 9a, 9b	p-Values to test for differences in impacts across subgroups, significance marker ("*")
81	r2	Numeric	9	R-squared value (for full sample only for models with covariates)
82	icc	Numeric	9	Intraclass correlation coefficient (for DESIGN 3 and 4, CLUSTER_DATA=1 only)
83	n_blocks	Numeric	10	Number of blocks
84	sd_impact	Numeric	10	Standard deviation of impact
85	pct_positive	Numeric	10	Proportion positive (0 to 100)
86	range	Numeric	10	Difference between the largest and smallest block-specific impact estimate
87	block_pvalchi	Numeric	10	p-Value from joint test of differences across blocks
88	block_schi	String	10	p-Value from joint test of differences across blocks, significance marker ("*")
89	input	String	Appendix	.rctyes input specification file field name
90	specification	String	Appendix	.rctyes input specification file field value



## 7. Graphing the impact results

The impact findings from Table 9 (or Tables 9a, 9b, or 9c for analyses that include CACE analyses) can be plotted in R using the *RCT-YES-Graph* application. Importantly, the application can only be run if **the free R software and the free shiny, shinydashboard, and ggplot2 R packages have been installed (see below and Appendix A)**. To produce graphs, users must run in R the graphics program created by the R or Stata computer program that conducted the analysis. The base name and location of the graphics program will be the same as for the .html file except the name will contain the “\_graph” suffix and will have a .R extension. *RCT-YES-Graph* creates graphs using the .csv file produced by the analysis program.

After installing or updating R (see Appendix A), users will need to install the free shiny, shinydashboard, and ggplot2 packages using the following steps:

- Open the R interactive workspace by double-clicking the R icon on the desktop or in the **Start** menu
- At the interactive prompt, type the following command to install the shiny, shinydashboard, and ggplot2 packages (and then hit enter):

```
install.packages(c('shiny', 'shinydashboard', 'ggplot2'), repos='https://cran.rstudio.com')
```

Note that `repos='https://cran.rstudio.com'` in the command is intended to bypass selecting a download mirror location, solely for simplicity. Experienced users may want to select their download mirror, in which case the `repos` argument can be excluded, and a prompt for selection will appear. The command would then be:

```
install.packages(c('shiny', 'shinydashboard', 'ggplot2'))
```

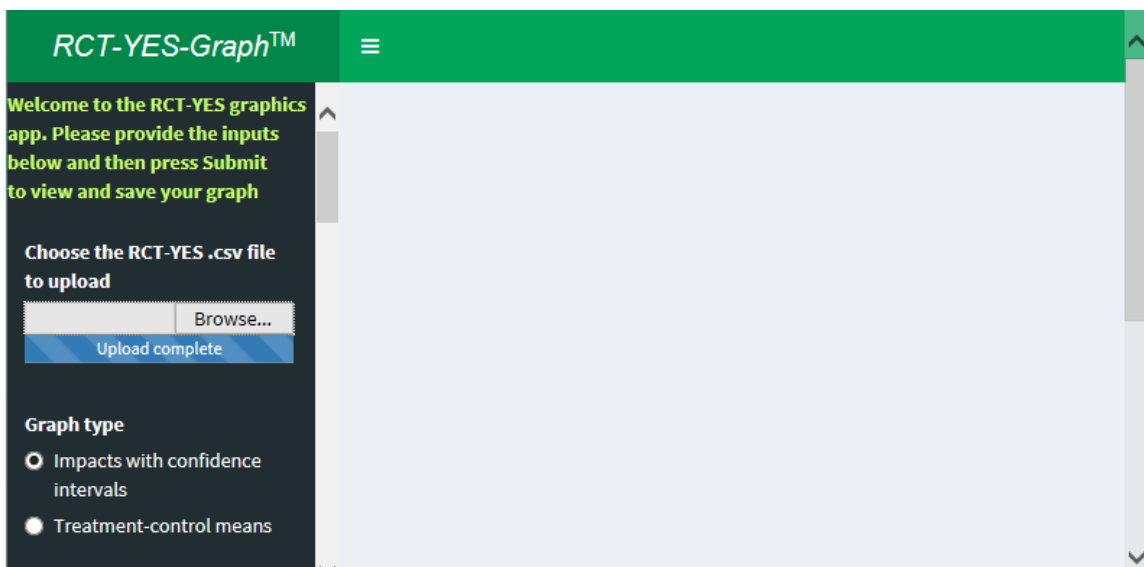
Once R, shiny, shinydashboard, and ggplot2 have been installed, the impact findings can be graphed using the following steps, where we use our after-school RCT example for illustration:

- Open the R interactive workspace by double-clicking the R icon on the desktop or in the **Start** menu
- Click the **File** menu in the toolbar and select **Source R code**
- A file selection menu will appear. Locate the .R graph program produced by the R or Stata computer program in the directory where it was saved: it will be located in the same directory as the .html file and will have the same base name except with a “\_graph” suffix. Click the **Open** button.

## 7. Graphing the impact results

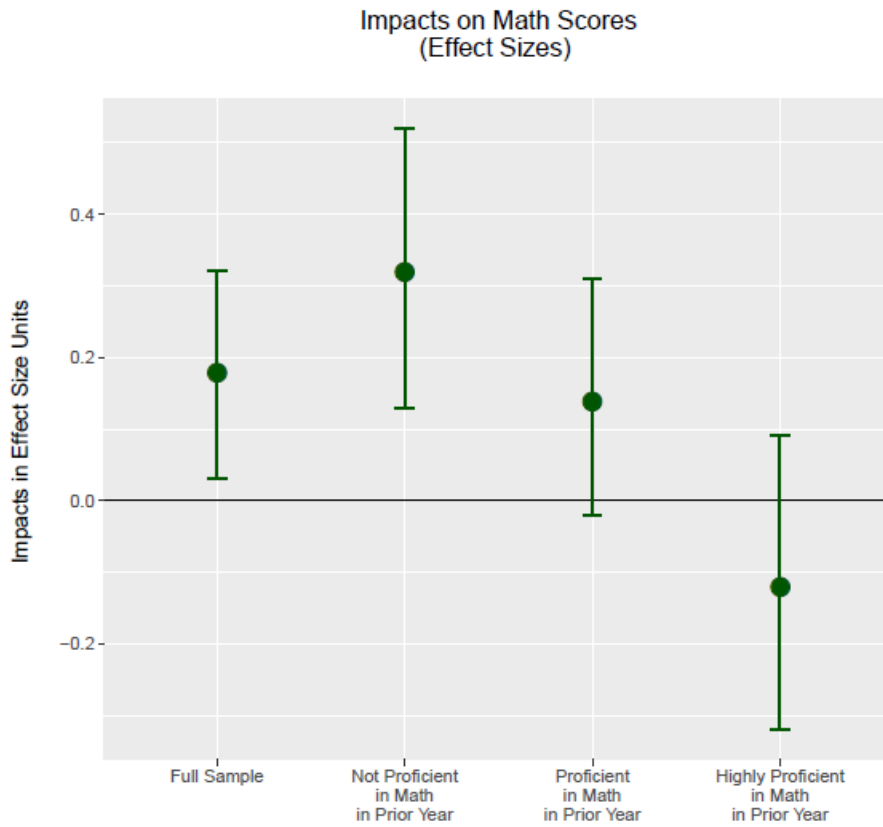
- After a few seconds, the graphics dashboard will appear and request input information for the graph. Make sure to maximize the screen to view the full dashboard. The scroll bars in the dashboard can be used for navigation. The black area at the left of the dashboard is for the inputs and the grey area at the right is where the graph will be displayed after all inputs have been entered and the **Submit** button has been pressed.

### Screenshot 7.1: The *RCT-YES-Graph* dashboard



- The first input item will request the name and location of the .csv file produced by the R or Stata computer program that contains the impact findings. Use the **Browse** button to locate this file and click it to fill the input box. The application will now read the .csv file and list the domains, outcomes, and subgroups to select for the graphs.
- Select the graph type that either shows (i) impact estimates and associated lower and upper confidence limits or (ii) treatment and control group means
- Select whether the graph should be a bar chart or line graph. Screenshots 7.2 and 7.3 display examples of bar graphs for the two different graph types using our after-school RCT example. Line graphs are appropriate for longitudinal evaluations where the same outcome can be plotted over time (for example, monthly school attendance rates or third to fifth grade test scores). For both bar charts and line graphs, x-axis labels can be specified as inputs. For line graphs, the x-axis can instead display a time trend corresponding to when the outcomes were measured.

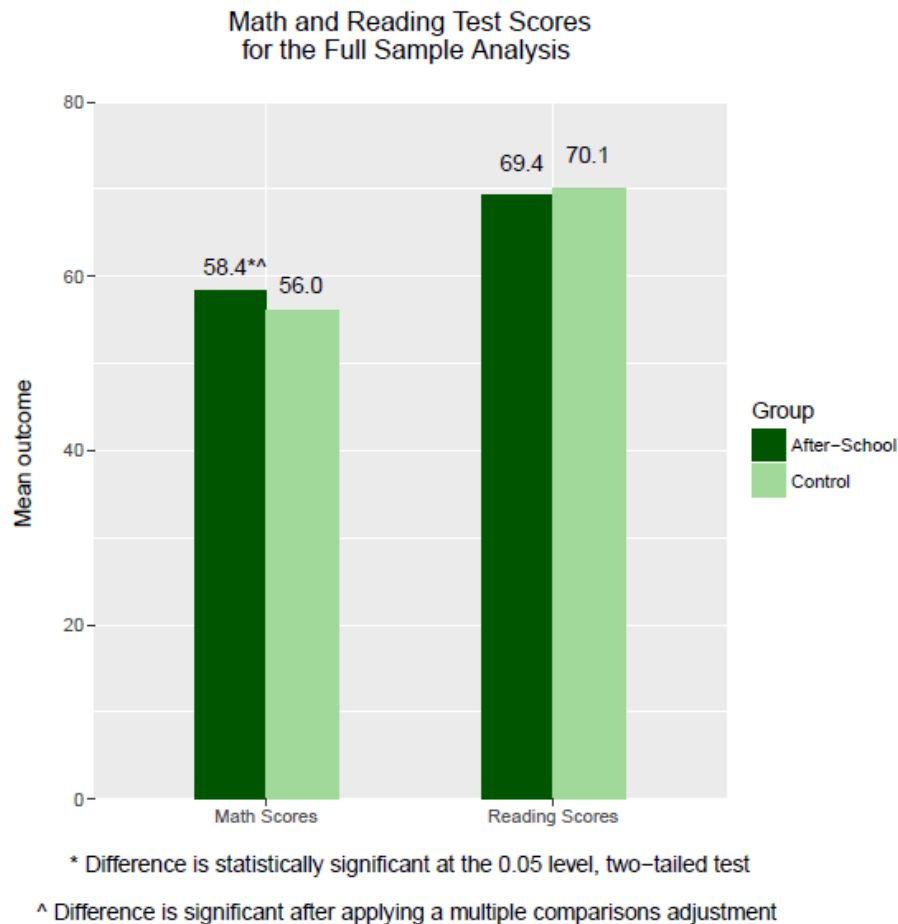
Screenshot 7.2: Example of a bar graph with impacts and confidence intervals



Note: The error bars are 95 percent confidence intervals

## 7. Graphing the impact results

**Screenshot 7.3: Example of a bar graph with treatment and control group means**



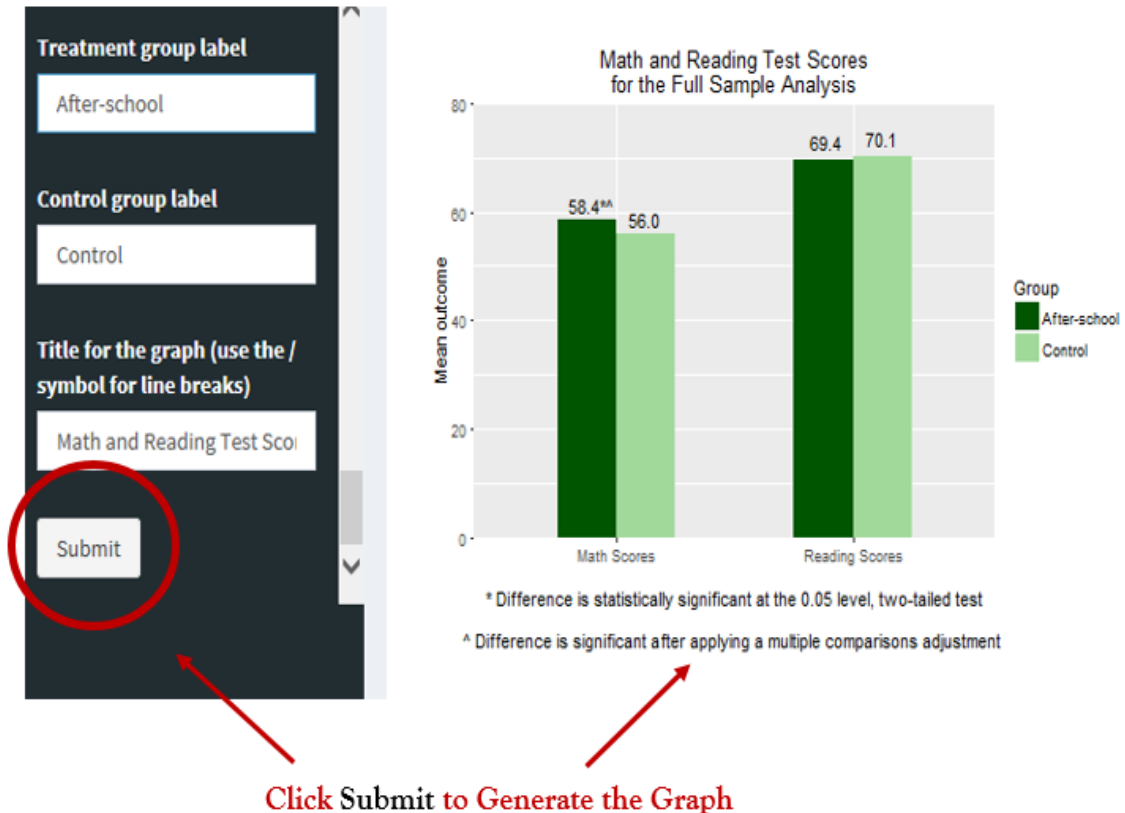
- Select the outcome domain; only one can be selected at a time.
- Fill in the following inputs: the table number in the .html file for the graph (Table 9 if no CACE analyses are specified, and Table 9a, 9b, or 9c otherwise); the outcome and subgroup variables for the graph; the graph title; and x-axis labels (to override the default variable names) or time trends (for line graphs). The list of outcomes and subgroups will automatically update if the selected domain is changed. For line graphs, only a single subgroup can be selected. *Importantly, the graph will plot all combinations of specified outcome and subgroup variables and the plot will be ordered by outcome variable first and then subgroup.* There are no options for changing the ordering of outcomes and subgroups in the input checklists.
- Provide inputs for several graph features (specific to the graph type) to override default values, including (i) whether impacts should be displayed in nominal or effect size units; (ii) options for displaying numbers and significance levels above the data points; (iii) labels for the treatment and control groups; (iv) a label for the y-axis; and (v) minimum and maximum values for the y-axis, where *both* values must be entered to override the zero default values.



## 7. Graphing the impact results

- Press **Submit** after all required information for the graph has been entered. The graph will appear in the grey area on the right-hand side of the dashboard (Screenshot 7.4). Importantly, the **Submit** button will only need to be pressed once per session: subsequent changes to the graph inputs will automatically update the graph shown on the screen.

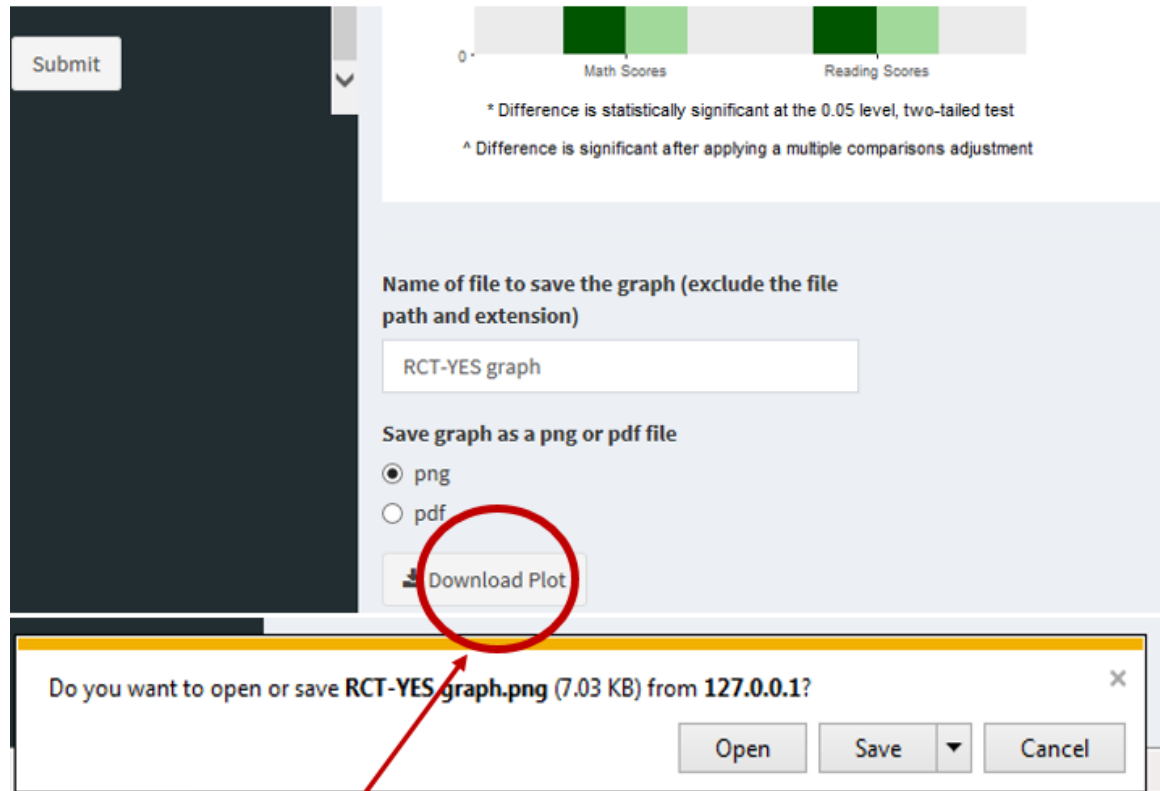
Screenshot 7.4: Pressing Submit to produce the graph



- To save the graph to a file, first specify at the bottom of the graph the base name for the output file (but not the path name) and whether the file type should be a .png or .pdf file. Next, click the **Download Plot** button (Screenshot 7.5). A dialog box will appear requesting whether to open or save the graph to a file with the specified name and file type. To save the file, use the **Save As** command to locate and select the folder for the file, or the **Save** command which will save the file to the default directory. Graphs can also be saved by right-clicking the mouse and using the **Save picture as** command. *It is a good idea to open the file first before saving it because the graph will typically be narrower in the file than in the screen.*

## 7. Graphing the impact results

### Screenshot 7.5: Saving the graph to a file



**Click Download Plot to Save the Graph**

- When the session is over, exit the application (for example, using the **File/Exit** command) to return to the interactive prompt in the R workspace. Importantly, if the application is not exited (but, for example, is instead minimized), it will be necessary to press the esc key or the **stop** button in the toolbar to return to the interactive prompt in R; otherwise, users will be “hung up” in the R workspace.
- Print the saved graph files. Alternatively, insert them first, for example, into a Word document and resize them as needed before printing; this approach will typically provide better printouts of the graphs.

## Appendix A. Downloading and Running R and Stata

In order to use *RCT-YES*, users must know how to (i) create a data file compatible with R or Stata and (ii) run the R or Stata computer program file produced by the interface. This appendix discusses these topics, first for R, and then more briefly for Stata.

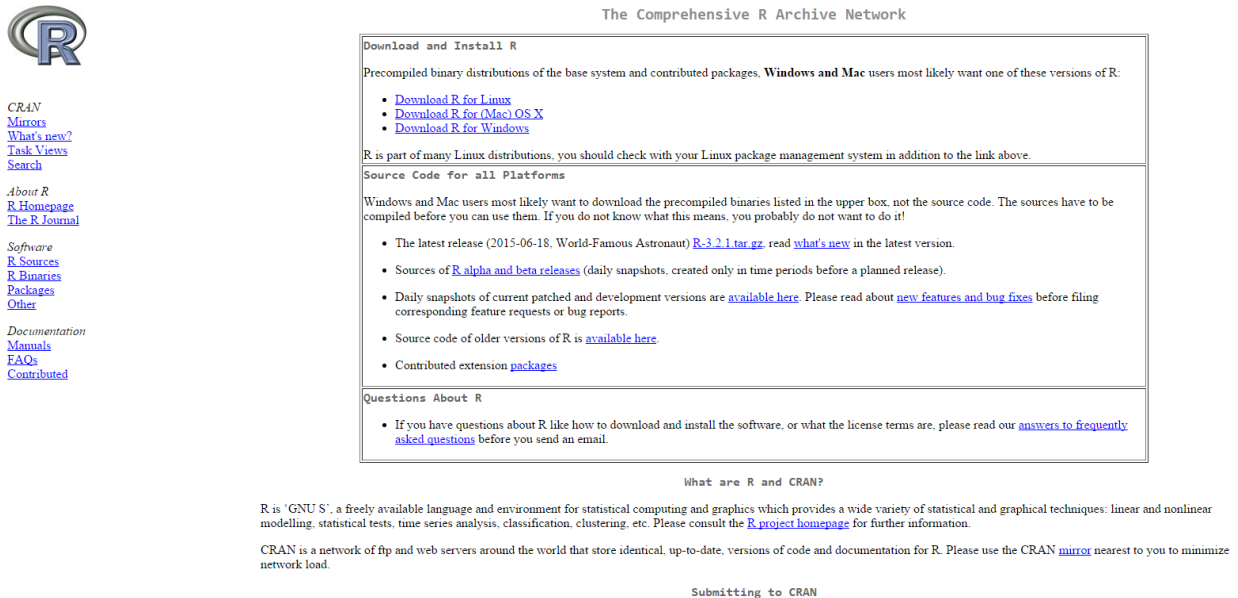
### a. Downloading and Running R

If users plan to work on a computer that is administered by another person or an information technology consultant or department, they should ask for assistance from the appropriate administrator before installing R. There may be conditions that complicate or prevent installation using the instructions below without additional knowledge of the computer system.

#### Downloading and installing R

To download R, users should first go to <https://cran.r-project.org/>. Three links will be located in the top pane of the screen (titled **Download and Install R**). Users should click the link **Download R for Windows**.

#### Screenshot A.1: Downloading R, Step 1



The screenshot shows the CRAN website interface. On the left, there is a navigation menu with links for CRAN, Mirrors, What's new?, Task Views, Search, About R, R Homepage, The R Journal, Software, R Sources, R Binaries, Packages, Other, Documentation, Manuals, FAQs, and Contributed. The main content area is titled 'The Comprehensive R Archive Network' and 'Download and Install R'. It contains the following text and links:

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2015-06-18, World-Famous Astronaut) [R-3.2.1 tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

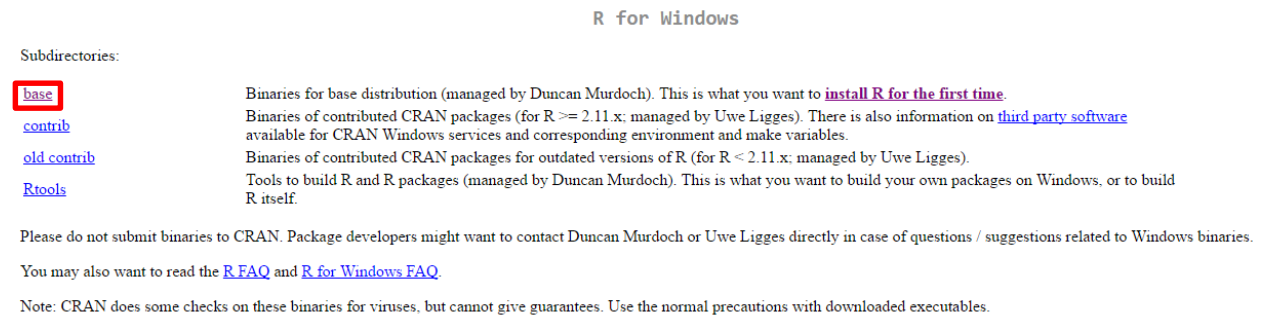
CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN [mirror](#) nearest to you to minimize network load.

Submitting to CRAN

## Appendix A. How to use R and Stata

Users will now be directed to another page with four links under the heading “Subdirectories” (see Screenshot A.2). Users should click the first link [base](#) as shown in the red box in Screenshot A.2.

### Screenshot A.2: Downloading R, Step 2



R for Windows

Subdirectories:

<a href="#">base</a>	Binaries for base distribution (managed by Duncan Murdoch). This is what you want to <a href="#">install R for the first time</a> .
<a href="#">contrib</a>	Binaries of contributed CRAN packages (for R $\geq$ 2.11.x; managed by Uwe Ligges). There is also information on <a href="#">third party software</a> available for CRAN Windows services and corresponding environment and make variables.
<a href="#">old.contrib</a>	Binaries of contributed CRAN packages for outdated versions of R (for R $<$ 2.11.x; managed by Uwe Ligges).
<a href="#">Rtools</a>	Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

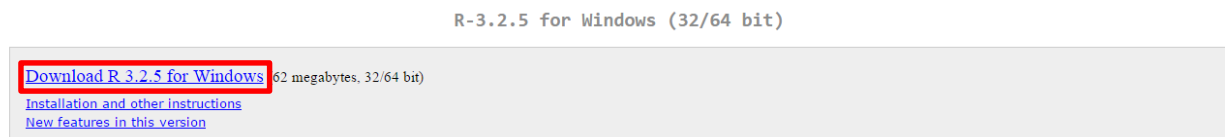
Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

The next page will display a gray box at the top of the page with a link in large font saying, for example, “Download R 3.2.5 for Windows” (see the red box in Screenshot A.3). Note that the version number, in this case 3.2.5, may change in the future as additional versions of R become available. Clicking this link will prompt users to download the installation file, which should be saved in an easy accessible location, such as the default downloads folder for the system.

### Screenshot A.3: Downloading R, Step 3



R-3.2.5 for Windows (32/64 bit)

[Download R 3.2.5 for Windows](#) 62 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

To install R, users should run the file downloaded in the previous step by (i) locating the file in the folder where it was saved and (ii) double-clicking it. Most of the installation process will run automatically, although at several points it will be necessary to make selections. In general, the defaults will be the best choice and make the R installation run smoothly.

### Converting a .csv file into a .rds file

A good way to create a .rds R file is to first create a .csv file, which is a comma-delimited text file. Most spreadsheets (such as Microsoft Excel, Open Office Calc, or Google Spreadsheets) can be saved as .csv files in a single sheet. Screenshot A.4 displays a screenshot of a .csv file for the hypothetical RCT of an after-school program from Chapter 4.

**Screenshot A.4: Example of a .csv file for the hypothetical RCT from Chapter 4**

	A	B	C	D	E	F	G	H	I	J	K
1	SCHOOL	DISTRICT	TREATME	MATH_SC	READ_SCC	PRIOR_MA	PRIOR_RE	GENDER	SG_MATH	SG_READ_PROF	
2	1	1	0	62.41541	60.66273	57.70976	55.42374	0	1	1	
3	1	1	0	42.06018	60.00448	52.16401	50.43956	0	1	1	
4	1	1	0	61.6479	62.46757	54.99372	55.42374	1	1	1	
5	1	1	0	41.51383	59.56616	61.67929	60.40791	1	1	2	
6	1	1	0	49.09638	52.16666	57.62475	45.45539	0	1	1	

Consider a .csv file named “C:/rct\_data.csv”. To convert this file into a .rds file named “C:/rct\_data.rds”, users should proceed as follows:

1. Open the R interactive workspace by double-clicking the R icon on the desktop. Note that **R is case sensitive**, so that all R commands shown below must be entered in the correct upper or lower case (including file names).
2. At the interactive prompt, type the following command to read the .csv file into R (and then hit enter):

```
data <- read.csv("C:/rct_data.csv", header=TRUE, stringsAsFactors=FALSE)
```

3. Type the following command to save the dataset in .rds format (and then hit enter):

```
saveRDS(data, "C:/rct_data.rds")
```

The resulting .rds file can then be used in the *RCT-YES* interface.

Many statistical packages (for example, SAS and SPSS) allow users to output files into .csv format (see Screenshot A.4). Thus, users can create .rds files from these statistical packages by first creating .csv files and then using the steps above to convert .csv files into .rds files. Importantly, users must be careful with how missing values are coded in these packages. For example, SAS formatted missing value codes (.D, .E, etc.) will be written to .csv as letters without the preceding dot, which can cause problems in R when it interprets column types. If possible, it is simplest to convert all numeric missing values to dot (.) before exporting SAS and related files into .csv files.

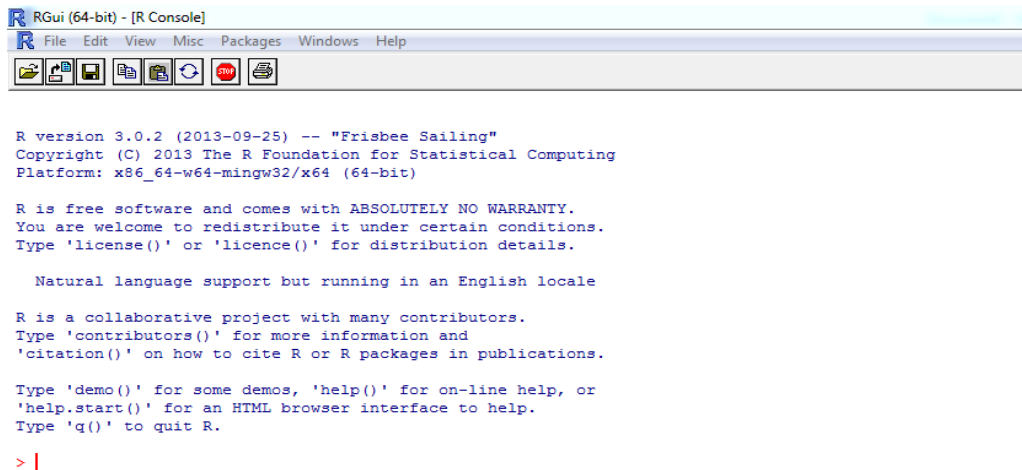
Note in Screenshot A.4 that the first line in the created .csv file should have the variable names and the second line onwards should have the data. In addition, if the top row has a title, users should delete this row before saving, and remove all footnotes.

## Running the R program produced by the interface

RCT-YES will generate a computer program file (.R extension) with the name and location specified in the interface. There are two ways to run this program in R. One is interactive via point-and-click menus, which is probably better for R beginners, and one is from the command line, a useful alternative for those who want to run analyses programmatically.

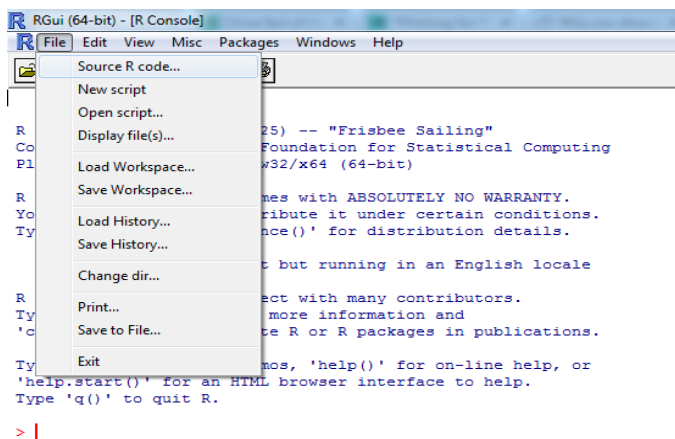
**Running R interactively.** To run R interactively, users should open R by double-clicking the R icon on the desktop. An interactive workspace will open (Screenshot A.5).

### Screenshot A.5: Interactive workspace in R



Next, users should click the **File** menu in the top left corner and select **Source R code** from the list of options (Screenshot A.6).

### Screenshot A.6: Interactive workspace in R



A file selection menu will appear. Users should now locate the R program (with the .R file extension) that was created by the RCT-YES interface, and click the **Open** button. The program will now run

without any additional input. When it completes successfully, the *RCT-YES* output .html file will be written to the folder specified in the interface, and the following message will appear:

```
RCT-YES successfully completed:

You can now view the analysis results by opening the file:
  C:/RCT-YES Files/after_school_rct_r.html

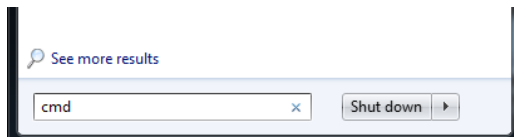
As an option, you can then run the following program in R to plot the impact results using the associated .csv file
(see the User's Manual or Quick Start Guide for directions):
  C:/RCT-YES Files/after_school_rct_r_graph.R
> |
```

Users can now view the analysis results by exiting or minimizing R and locating and opening the .html file. As an option, users can then run the graphics program produced by the R program to plot the impact findings using R.

**Running R from the command line.** To run R from the command line, users should first find the file called “Rscript.exe”. For a 64-bit Windows operating system with R 3.2.5 installed, this file is likely to be located in the folder “C:/Program Files/R/R-3.2.5/bin/x64”. For other versions of R, the text “3.2.5” should be replaced with the appropriate version number. If needed, users can also search their system for “Rscript.exe”.

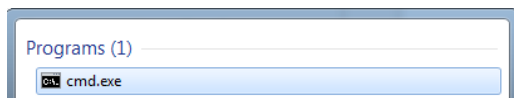
In what follows, we assume that “Rscript.exe” was found in its usual location, and we want to run an R program that was output from the *RCT-YES* interface and saved as “C:/example.R” on a computer with R installed. To run this program, users should first click the **Start** button on their desktop and type “cmd” in the search box (Screenshot A.5).

### Screenshot A.7: Search box in the Windows Start menu



The **cmd.exe** icon will appear, and users should open the file by clicking on it (Screenshot A.7).

### Screenshot A.8: The cmd.exe icon



Users will be directed to a Windows command prompt and should type the path name of “Rscript.exe” followed by the path name of the R program in quotes. For our example, as shown below, users should type “C:/Program Files/R/R-3.2.5/bin/x64/Rscript.exe” “C:/example.R”:

```
C:\>"C:/Program Files/R/R-3.2.5/bin/x64/Rscript.exe" "C:/example.R"
```

After pressing enter, the program will now run and will be complete when a new prompt for an input is requested on the screen.

## b. Downloading and Running Stata

### Downloading and installing Stata

Stata can be downloaded for a fee from [www.stata.com](http://www.stata.com). For help on installing the software, users should reference the Stata installation guide for Windows that can be found at [www.stata.com/install-guide/windows/download/](http://www.stata.com/install-guide/windows/download/).

### Converting a .csv file into a .dta file

A good way to create a Stata .dta file is to first create a .csv file, which is a comma-delimited text file. Most spreadsheets (such as Microsoft Excel, Open Office Calc, or Google Spreadsheets) can be saved as .csv files in a single sheet. Screenshot A.4 above provides an example of a .csv data file.

Consider a .csv file named “C:/rct\_data.csv”. If users have the Stat/Transfer software ([www.stattransfer.com](http://www.stattransfer.com)), they can use it to convert the .csv file into a .dta file. Otherwise, users can go into Stata and convert the .csv file into a .dta file named “C:/rct\_data.dta” as follows:

1. Open Stata by double-clicking the Stata icon or using the **Start** menu on the desktop to navigate to the Stata program. Note that **Stata is case sensitive**, so that all Stata commands shown below must be entered in the correct upper or lower case (including file names).
2. In the command window at the bottom of the screen, type the following command to read the .csv file into Stata (and then hit enter):

insheet using “C:/rct\_data.csv”, comma

Alternatively, the .csv file can be read in by clicking **File** in the toolbar, clicking **Import**, and then clicking **Text data (delimited, \*.csv,...)**. Users should then use the **Browse** button to locate the .csv file to import into Stata and click **OK**.

3. In the command window, type the following command to save the dataset in .dta format (and then hit enter):

save “C:/rct\_data.dta”, replace

The resulting .dta file can then be used in the RCT-YES interface.



Many statistical packages (for example, SAS and SPSS) allow users to output files into .csv format. Thus, users can create .dta files from these statistical packages by first creating .csv files and then using the steps above to convert .csv files into .dta files. Importantly, there are some rules to be followed when saving a .csv file for reading into Stata (see Screenshot A.4 above):

- The first line in the spreadsheet should have the variable names and the second line onwards should have the data. If the top row has a title, delete this row before saving. Also, remove all footnotes.
- The variable names cannot start with a number (for example, 1995 is an invalid name)
- Make sure that there are no commas in the data, which will produce errors in Stata
- Missing data should be coded as a single dot (.) or blank field. Stata will read codes such as double dots or hyphens as text, and thus these codes should be avoided for missing data.

### Running the Stata program produced by the interface

RCT-YES will generate a computer program file (.do extension) with the name and location specified in the interface. To run this Stata program, users should proceed as follows:

1. Open Stata by double clicking on the Stata icon or using the **Start** menu on the desktop to navigate to the Stata program
2. Under the **File** menu, select **Do** and navigate to the program file [.do extension] created by the interface
3. Click **Open**. Stata will now run the program and create the .html file specified in the interface ([after\\_school\\_rct.html](#) in the example below). When completed successfully, the following statement will appear in the Stata window:

RCT-YES successfully completed:

You can now view the analysis results by opening the file:

`C:/RCT-YES Files/after_school_rct.html`

As an option, you can then run the following program in R to plot the impact results using the associated .csv file (see the User's Manual or Quick Start Guide for directions):

`C:/RCT-YES Files/after_school_rct_graph.R`

```
.
.
end of do-file
```

Users can now view the analysis results by clicking the link to the .html file or exiting or minimizing Stata and locating and opening the .html file. As an option, users can then run, in R, the graphics program produced by the Stata program to plot the impact findings.



## References

- Benjamini, Y. & Y. Hochberg (1995). Controlling the false discovery rate: A new and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, vol. 57, pp. 1289–1300.
- Imbens, G. & D. Rubin (2015). *Causal inference for statistics, social, and biomedical sciences: an introduction*. Cambridge UK: Cambridge University Press.
- Jones, M. (1996). Indicator and Stratification Methods for Missing Explanatory Variables in Multiple Linear Regression. *Journal of the American Statistical Association*, 91, 222-230.
- May, H., I. Perez-Johnson, J. Haimson, S. Sattar & P. Gleason (2009). Using state tests in education experiments: A discussion of the issues (NCEE 2009-013). Washington, DC: National Center for Education Evaluation and Regional Assistance, Institute of Education Sciences, U.S. Department of Education.
- National Center for Education Statistics (2010a). Basic concepts and definitions for privacy and confidentiality in student education records (NCES 2011-601). SLDS Technical Brief 1. Washington DC: Institute of Education Sciences.
- National Center for Education Statistics (2010b). Statistical methods for protecting personally identifiable information in aggregate reporting (NCES 2011-603). SLDS Technical Brief 3. Washington DC: Institute of Education Sciences.
- Puma, M. J., R. B. Olsen, S. H. Bell, & C. Price (2009). What to do when data are missing in group randomized controlled trials (NCEE 2009-0049). Washington, DC: National Center for Education Evaluation and Regional Assistance, Institute of Education Sciences, U.S. Department of Education.
- Raudenbush, S. W. & A. Bryk (2002). *Hierarchical linear models: Applications and data analysis methods*. Newbury Park, CA: Sage.
- Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. New York: J. Wiley and Sons.
- Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. London: Chapman and Hall.
- 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, Analytic Technical Assistance and Development. Retrieved from <http://ies.ed.gov/ncee/edlabs>.
- What Works Clearinghouse (2014): Procedures and standards handbook, Version 3.0. Washington DC: Institute of Education Sciences.





