The Stata Journal (2017)
17, Number 1, pp. 130–138

Capturing a Stata dataset and releasing it into REDCap

Seth T. Lirette
Department of Data Science
University of Mississippi Medical Center
Jackson, MS
and
Department of Biostatistics
University of Alabama at Birmingham
Birmingham, AL
slirette2@umc.edu

Samantha R. Seals
Department of Mathematics and Statistics
University of West Florida
Pensacola, FL

Chad Blackshear
Department of Data Science
University of Mississippi Medical Center
Jackson, MS

Warren May
School of Health Related Professions
University of Mississippi Medical Center
Jackson, MS

Abstract. With technology advances, researchers can now capture data using web-based applications. One such application, Research Electronic Data Capture (REDCap), allows for data entry from any computer with an Internet connection. As the use of REDCap has increased in popularity, we have observed the need to easily create data dictionaries and data collection instruments for REDCap. The command presented in this article, `redcapture`, demonstrates one method to create a REDCap-ready data dictionary using a loaded Stata dataset, illustrated by examples of starting from an existing dataset or completely starting from scratch.

Keywords: dm0091, redcapture, reproducibility, REDCap, web based, data entry

1 Introduction

Research Electronic Data Capture (REDCap) is a secure, web-based application designed to support data capture for research studies. It provides 1) an intuitive interface for validated data entry; 2) audit trails for tracking data manipulation and export procedures; 3) automated export procedures for seamless data downloads to common statistical packages; and 4) procedures for importing data from external sources (Harris et al. 2009). While developed at Vanderbilt, REDCap is hosted at the individual institution level; that is, all data collected using REDCap are stored on a local server.

REDCap has many desirable properties for researchers interested in electronic data collection. Because the data are stored on a centrally managed server at the local host institution, researchers do not have to worry about losing the data file (for example, an Excel file maintained on a USB flash drive), and the security of the data can be more easily ensured. Clinicians have begun to recognize the need for securing patient data (Anand and Spalding 2015; Niehaus, Boimbo, and Akuthota 2015; and Morinville et al. 2014). Further, REDCap has double data-entry functionality, including a data comparison tool that allows users to select the correct data-entry mode for the type of study using article forms or direct entry. Other useful features of REDCap are its ease of use and intuitive interface. With minimal training, staff can enter data into the system consistently. For relatively simple data-entry forms, REDCap offers a form development platform that is straightforward and easy to use without any knowledge of database programming. Certain functions, such as audit trails, are automatically activated when the project's database is put into production.

In addition to data collected within an institution, REDCap allows data capture among multiple sites using a common web-based interface. As an example, consider a multicenter clinical trial where all sites enter data into the same database through REDCap. When the statistician needs access to the data to create the data safety monitoring board report, the data can be exported for immediate use in several softwares: Excel, SPSS, SAS, R, and Stata. These exported data are formatted specifically for the statistical software of choice and save the statistician from extensive data management. Other useful features include a data import tool, a randomization tool, and a longitudinal study planner with a calendar function to inform the data collector of the progress of each individual subject.

Because REDCap is becoming more popular, we discovered a need to easily create a REDCap instrument from an existing dataset. Researchers either want to import their existing data into REDCap or want to collect the same data as a previous project. Outside of basic single-study clinical research, longitudinal cohort studies may be interested in collecting future data in REDCap. In that case, studies often reuse questionnaires across visits, so it may be helpful for statisticians to be able to use existing data to create a data dictionary to upload to REDCap. For example, most large epidemiologic studies use standard instruments at each clinic exam to take blood pressure and medical health history. If another clinic visit is funded, these, among other, instruments will be repeated and REDCap is an attractive choice in terms of ease of use, security, accessibility, and cost for the continuation.

Some practical published examples are warranted at this point. Shared libraries are very efficient, both for multisite and multivisit studies and for sharing across studies. REDCap shines here (Obeid et al. 2013). REDCap offers low-cost functionality in developing countries (Tuti et al. 2016). Finally, studies have demonstrated the ease-of-use capabilities of REDCap (Pang et al. 2014; Franklin, Guidry, and Brinkley 2011).

Behind any good study is a well-designed data structure. REDCap uses the data dictionary to allow professional databases to be built with minimal programming but accounts for different variable types and limiters, variable and value labels, skip patterns,

and on-screen appearance of the data-entry screens. If starting a new project, one can build the dictionary within REDCap, but REDCap also offers one the ability to build the data dictionary externally and upload it to create the necessary forms. Further, REDCap has built-in functionality for downloading existing data dictionaries once they are created. However, to upload data from another data source, one must first create the REDCap data dictionary.

By using existing data to create a data dictionary for REDCap, statisticians can create the framework for the REDCap instrument by first uploading the data dictionary. This can then be used to either upload existing data or, in our case, easily generate forms for our new study visit based on existing data currently stored in Stata data files. These goals are easily accomplished using `redcapture`.

2 The redcapture command

The essential starting block for a new project in REDCap is a data dictionary. The `redcapture` command seamlessly automates this process by using the loaded Stata dataset instead of the point-and-click interface of the web-based REDCap. In essence, this data dictionary is merely a CSV file that contains a certain set of column headers and formatting rules. The user can then simply load the CSV file into REDCap.

The heart of `redcapture` is built around the `text()`, `dropdown()`, and `radio()` options. At least one of these must be specified for `redcapture` to run, and anything listed in the main *varlist* must be listed in one of these three options' *varlist*s. We do note that check boxes are also available in the web version of REDCap; however, we are attune to the headaches and inconsistencies involved with check-box answers and decided to omit them as an option in `redcapture`. Also note that for the variables declared to be drop downs or radios (that is, categorical in nature), `redcapture` searches the dataset and compiles the form based solely upon each unique label of the categorical variables. Therefore, first, drop-down or radio variables need to be numerical with value labels attached. Second, all values the user wishes to include in the data dictionary need to be defined in the value label, although that actual value does not need to be present in the loaded data.

REDCap also offers the ability to require validation for variables entered as text fields, and `redcapture` provides this option as well. The user simply declares validation variables in the `validate()` option. This option requires the declared variables to also be declared as text variables in the `text()` option. In addition to validating the variables, REDCap and `redcapture` offer the option of providing required minimums and maximums to the validated variables, done in `redcapture` through the `validmin()` and `validmax()` options. If the user wishes to omit validation minimums and maximums for any or all the validation variables, `none` should be entered into the corresponding location in the `validmin()` and `validmax()` options. REDCap version 6.10.1 has 24 different validation types. Each is listed in table 1 with an example of a minimum and maximum. Blanks indicate minimums and maximums are not legitimate for that particular validation type. Examples will follow.

As one last option, questions with the same responses can be grouped together in REDCap using the matrix functionality. An example would be 25 survey questions each with a Likert-type response. `redcapture` allows for up to 10 different matrices.

Table 1. Validation types and examples of minimums and maximums

validtypes	Example min	Example max
`date_dmy`	01/01/1900	12/31/3000
`date_mdy`	01/01/1900	12/31/3000
`date_ymd`	01/01/1900	12/31/3000
`datetime_dmy`	01/01/1900 00:01	12/31/3000 23:59
`datetime_mdy`	01/01/1900 00:01	12/31/3000 23:59
`datetime_ymd`	01/01/1900 00:01	12/31/3000 23:59
`datetime_seconds_dmy`	01/01/1900 00:00:01	12/31/3000 23:59:59
`datetime_seconds_mdy`	01/01/1900 00:00:01	12/31/3000 23:59:59
`datetime_seconds_ymd`	01/01/1900 00:00:01	12/31/3000 23:59:59
`email`		
`integer`	1	100
`alpha_only`		
`mrn_10d`		
`number`	1	100
`number_1dp`	0	100.1
`number_2dp`	0	10
`number_3dp`	0	2.012
`number_4dp`	1.3132	5.3216
`phone`		
`ssn`		
`time`	01:23	14:55
`time_mm_ss`	01:23:01	59:59:00
`vmrn`		
`zipcode`		

2.1 Syntax

`redcapture` *varlist*`, file(`*string*`) form(`*string*`)` [`text(`*varlist*`) dropdown(`*varlist*`) radio(`*varlist*`) header(`*string*`) validate(`*varlist*`) validtype(`*validtypes*`) validmin(`*minlist*`) validmax(`*maxlist*`) matrix1(`*varlist*`) matrix2(`*varlist*`) matrix3(`*varlist*`) matrix4(`*varlist*`) matrix5(`*varlist*`) matrix6(`*varlist*`) matrix7(`*varlist*`) matrix8(`*varlist*`) matrix9(`*varlist*`) matrix10(`*varlist*`)]

2.2 Options

`file(`*string*`)` is required and indicates the filename for the CSV file that is output to the active directory. Subdirectories are also supported. This file must be closed before running `redcapture`.

`form(`*string*`)` is required and indicates the form name for later use when loading the instrument into REDCap. This is a REDCap requirement.

Note: At least one of the options `text()`, `dropdown()`, or `radio()` is required.

`text(`*varlist*`)` contains a list of all variables the user wishes to be input as text fields.

`dropdown(`*varlist*`)` contains a list of all variables the user wishes to be input as drop-down lists.

`radio(`*varlist*`)` contains a list of all variables the user wishes to be input as radio buttons.

`header(`*string*`)` gives a header or title to the REDCap instrument.

`validate(`*varlist*`)` contains a list of all variables the user wants to require validation. Only text fields can be validated. Drop downs and radio buttons are inherently validated by only offering valid choices. This option can be left blank, but validating text fields is usually a good idea to ensure quality data.

`validtype(`*validtypes*`)` is required for each variable declared in the `validate()` option and must be listed in the same order as the *varlist* in the `validate()` option. REDCap currently has 24 different validation types. The full list is in table 1 above.

`validmin(`*minlist*`)` contains the minimum values allowed for the validation variables and must be listed in the same order as the *varlist* in the `validate()` option. Only *validtypes* dates, datetimes, integers, numbers, and times use minimum and maximum values. The user is not required to set minimums and maximums and must type `none` where this is desired. `none` must be typed for *validtypes* that are not dates, datetimes, integers, numbers, or times. Dates must be in mm/dd/yyyy format and times in hh:mm:ss format. More details are below.

`validmax(`*maxlist*`)` contains the maximum values allowed for the validation variables and must be listed in the same order as the *varlist* in the `validate()` option. Only *validtypes* dates, datetimes, integers, numbers, and times use minimum and maximum values. The user is not required to set minimums and maximums and must type `none` where this is desired. `none` must be typed for *validtypes* that are not dates, datetimes, integers, numbers, or times. Dates must be in mm/dd/yyyy format and times in hh:mm:ss format. More details are below.

`matrix#(`*varlist*`)` specifies variables the user wishes to cluster. REDCap allows multiple variables to be combined into a single "data matrix". This is most useful for things like combining multiple variables in a Likert scale format into a single cluster in the instrument. `redcapture` allows up to 10 of these matrices.

3 Examples

```
. *Setting up the data
. clear
. input str9 id consented age race sex str10 bdate sbp dbp happy1 happy2 happy3
> str4 comment
           id  consented        age       race        sex       bdate
> sbp        dbp    happy1     happy2     happy3    comment
  1.       "1" 1 54 1 1 "1953-06-21" 110 80 2 2 1 "none"
  2.       "2" 1 64 2 2 "1943-05-19" 140 90 3 4 2 "none"
  3.       "3" 0 85 3 2 "1929-04-03" 160 85 1 5 3 "none"
  4.       "4" 0 80 1 1 "1934-10-23" 120 95 4 3 4 "none"
  5.       "5" 1 60 2 2 "1947-09-18" 115 70 5 1 5 "none"
  6. end
. *Fixing the birth date
. generate bdate2=date(bdate,"YMD")
. format bdate2 %td
. drop bdate
. rename bdate2 bdate
. order bdate, after(sex)
. *Variable labels
. label variable id        "Participant ID"
. label variable consented "Is a consent document on file?"
. label variable age       "How old were you on your last birthday?"
. label variable race      "What is your race?"
. label variable sex       "What is your sex?"
. label variable bdate     "What is your date of birth?"
. label variable sbp       "What was your last known systolic blood pressure?"
. label variable dbp       "What was your last known diastolic blood pressure?"
. label variable happy1    "The staff greeted me in a professional and
> courteous manner."
. label variable happy2    "The waiting time to see a doctor was satisfactory."
. label variable happy3    "I would return to this hospital."
. label variable comment   "Comments"
. *Value Labels
. label define ynlab 0 "No" 1 "Yes", replace
. label values consented ynlab
. label define rlab 1 "Caucasian" 2 "African American" 3 "Other", replace
. label values race rlab
. label define llab 1 "Strongly Agree" 2 "Agree"
>   3 "Neither Agree nor Disagree" 4 "Disagree" 5 "Strongly Disagree", replace
. label values happy1 llab
. label values happy2 llab
. label values happy3 llab
```

This first example is a case where a researcher is starting a study from scratch. He or she desires to include variables for an id variable, whether the participant consented, age, race, sex, birth date, systolic blood pressure, diastolic blood pressure, three Likert-scale variables relating to satisfaction, and an open comment field. The above shows

how the researcher can initiate such a dataset completely within a Stata do-file. Notice that variables the researcher will later declare to be drop downs and radios (`consented`, `race`, `happy1`, `happy2`, and `happy3`) have all values he or she wishes to make available. The rest of the setup regards attaching variable labels and value labels.

```
. redcapture *, file(example) form(example_form) header(Example)
>         text(id age sex bdate sbp dbp comment)
>         dropdown(consented race)
>         radio(happy1 happy2 happy3)
>         validate(id bdate dbp comment)
>         validtype(ssn date_ymd integer alpha_only)
>         validmin(none 1/1/1900 20 none)
>         validmax(none 12/31/2014 200 none)
>         matrix1(happy1 happy2 happy3)
REDcap data dictionary example.csv has been created in the active directory
```

Next comes the `redcapture` call. We wish to include all variables, with `consented` and `race` chosen from drop-down menus, `happy1`, `happy2`, and `happy3` being radio buttons collected into a matrix, and the rest typed in as text fields. We have chosen to validate `id` as a social security number (not a good practice), `bdate` as a `date_ymd` date with a minimum of `1/1/1900` and maximum of `12/31/2014`, `dbp` as an integer with a minimum of `20` and maximum of `200`, and `comment` as a free-text field. We have chosen not to validate `age`, `sex`, and `sbp` (also not a good practice). Once this runs, we get a CSV file whose first few columns are displayed as a screenshot in figure 1. This file can then be uploaded into REDCap.

VariableF	FormNam	SectionHe	FieldType	FieldLabel	ChoicesCalculationsSlider
id	example_	Example	text	Participant ID	
consentec	example_form		dropdowr	Is a consent document on file?	0,No | 1,Yes
age	example_form		text	How old were you on your last birthday?	
race	example_form		dropdowr	What is your race?	1,Caucasian | 2,African Am
sex	example_form		text	What is your sex?	
bdate	example_form		text	What is your date of birth?	
sbp	example_form		text	What was your last known systolic blood pressure?	
dbp	example_form		text	What was your last known diastolic blood pressure?	
happy1	example_form		radio	The staff greeted me in a professional and c	1,Strongly Agree | 2,Agree
happy2	example_form		radio	The waiting time to see a doctor was satisfa	1,Strongly Agree | 2,Agree
happy3	example_form		radio	I would return to this hospital.	1,Strongly Agree | 2,Agree
comment	example_form		text	Comments	

Figure 1. Screenshot of CSV file for example dataset

Suppose we would like to create a new REDCap data dictionary from an already existing dataset, such as the extract of the U.S. National Longitudinal Survey for employed women in 1988 (a Stata installed dataset). We simply load the dataset, set file and form names, declare radio, drop-down, and text variables, and set all text variables validation types, minimums, and maximums (a good practice). Upload the data dictionary created by `redcapture`, and we are now ready to start data collection for our study.

```
. sysuse nlsw88, clear
(NLSW, 1988 extract)

. redcapture * , file(nlsw_instrument) form(nlsw) header("NLSW Instrument")
> radio(race married collgrad smsa)
> dropdown(industry occupation union)
> text(idcode age never_married grade south c_city wage hours ttl_exp tenure)
> validate(age never_married grade south c_city wage hours ttl_exp tenure)
> validtype(integer integer integer integer integer number integer
>   number_4dp number_2dp)
> validmin(20 0 0 0 0 0 0 0 0)
> validmax(none 1 18 1 1 none 80 30 none)

REDcap data dictionary nlsw_instrument.csv has been created in the active
> directory
```

4 References

Anand, V., and S. J. Spalding. 2015. Leveraging electronic tablets and a readily available data capture platform to assess chronic pain in children: The PROBE system. *Studies in Health Technology and Informatics* 216: 554–558.

Franklin, J. D., A. Guidry, and J. F. Brinkley. 2011. A partnership approach for electronic data capture in small-scale clinical trials. *Journal of Biomedical Informatics* 44: S103–S108.

Harris, P. A., R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde. 2009. Research Electronic Data Capture (REDCap)—A metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of Biomedical Informatics* 42: 377–381.

Morinville, V. D., M. E. Lowe, M. Ahuja, B. Barth, M. D. Bellin, H. Davis, P. R. Durie, B. Finley, D. S. Fishman, S. D. Freedman, C. E. Gariepy, M. J. Giefer, T. Gonska, M. B. Heyman, R. Himes, S. Husain, S. Kumar, C. Y. Ooi, J. F. Pohl, S. J. Schwarzenberg, D. Troendle, S. L. Werlin, M. Wilschanski, E. Yen, and A. Uc. 2014. Design and implementation of INSPPIRE. *Journal of Pediatric Gastroenterology and Nutrition* 59: 360–364.

Niehaus, W., S. Boimbo, and V. Akuthota. 2015. Physical medicine and rehabilitation resident use of iPad mini mobile devices. *PM&R* 7: 512–518.

Obeid, J. S., C. A. McGraw, B. L. Minor, J. G. Conde, R. Pawluk, M. Lin, J. Wang, S. R. Banks, S. A. Hemphill, R. Taylor, and P. A. Harris. 2013. Procurement of

shared data instruments for Research Electronic Data Capture (REDCap). *Journal of Biomedical Informatics* 46: 259–265.

Pang, X., N. Kozlowski, S. Wu, M. Jiang, Y. Huang, P. Mao, X. Liu, W. He, C. Huang, Y. Li, and H. Zhang. 2014. Construction and management of ARDS/sepsis registry with REDCap. *Journal of Thoracic Disease* 6: 1293–1299.

Tuti, T., M. Bitok, C. Paton, B. Makone, L. Malla, N. Muinga, D. Gathara, and M. English. 2016. Innovating to enhance clinical data management using non-commercial and open source solutions across a multi-center network supporting inpatient pediatric care and research in Kenya. *Journal of the American Medical Informatics Association* 23: 184–192.

About the authors

Seth Lirette is a biostatistician at the University of Mississippi Medical Center and a PhD student at the University of Alabama at Birmingham.

Samantha Seals is a visiting assistant professor of statistics at the University of West Florida.

Chad Blackshear is a biostatistician at the University of Mississippi Medical Center.

Warren May is a professor of biostatistics at the University of Mississippi Medical Center.