REDCap Training 301

Advanced REDCap Features

Sui Tsang
REDCap@Yale Team
5/16/2024
REDCap 301 Learning Objectives

• Understand *Form Display Logic*

• Identify *Special Functions* that are commonly used in *calculations* and *branching logic*

• Know how *Action Tags* can be used to make your data collection more dynamic

• Learn how to incorporate *Smart Variables* to customize questions, forms, and communications.

• Be able to build and display a *Project Dashboard*

• Learn how to manage *Randomization* in REDCap
Form Display Logic

New Record – only Patient Screening Form is available for data entry

Caregiver consent=Yes – all caregiver forms available for data entry
Online Designer -> Form Display Logic

The Online Designer will allow you to make project modifications to fields and data collection instruments very easily using a web browser. NOTE: While in development status, all field changes will take effect immediately in real time.
Form Display Logic is an advanced feature that provides a way to use conditional logic to disable specific data entry forms that are displayed on the Record Status Dashboard, Record Home Page, or the form list on the left-hand menu. You might think of it as ‘form-level branching logic’. Form Display Logic can be very useful if you wish to prevent users from entering data on a specific form or event until certain conditions have been met. The forms will still be displayed on the page, but they will be disabled in order to prevent users from accessing them. Below you may define as many conditions as you want. A form may be selected in multiple conditions, but if so, please note that the form will be enabled if at least one of the conditions is met. The Form Display Logic does not impact data imports but only operates in the data entry user interface to enable/disable forms. Additionally, Form Display Logic is not utilized by the Survey Queue at all but can affect the behavior of the Survey Auto-Continue feature if the checkbox for it is enabled below. LIMITATION: Please note that the conditional logic used below will be evaluated at the record level and not within the context of an event or a repeating instance, which means that it is not possible to use relative instance or relative event Smart Variables - i.e., those with the name ‘current’, ‘next’, or ‘previous’, such as [next-instance] or [previous-event-name].

Optional Settings:
- Keep forms enabled if they contain data
  Only disable empty forms (those with a gray form status icon).
- Hide forms that are disabled
  All forms that are disabled will also be hidden (not visible) on the Data Collection menu and on the Record Home Page.

Condition 1:

Keep the following forms enabled...

- [All Events]
  Patient Screening Form [All Events]
  Caregiver Screening Form [All Events]
  Patient Contact [All Events]
  Caregiver Contact [All Events]
  Patient Interview Start [All Events]
  Patient Baseline Characteristics [All Events]

...when the logic below is TRUE.

[baseline_arm_1][pt_cg_approach]=1

e.g., [enrollment_arm_1][age] > 30

How to use this

View list of selected forms

Save  Cancel
QUESTIONS?
List of functions that can be used in Branching Logic, Calculations, Report filtering, Data Quality Module, Automated Survey Invitations, etc.

REDCap logic can be used in a variety of places, such as Branching Logic, Calculations, Report filtering, Data Quality Module, Automated Survey Invitations, and more. Special functions can be used in the logic, if desired. A complete list of ALL available functions is listed below. Listed below are some examples of common use cases where these functions might be used.

NOTICE: Please be advised that it is not possible to pipe the choice label of a multiple choice field into any of the special functions listed below. In other words, you cannot use the ‘label’ option, such as [my_field:label], to output the text label into a function. The special functions only utilize the value of a field, never its label.

Practical examples for common use cases

1. Calculate the number of days separating today's date and a date/datetime field's value in the past or future.
   
   datediff([date1], 'today', 'd')

2. Calculate a person's age based on date of birth.
   
   rounddown(datediff([date_of_birth], 'today', 'y'))

3. Calculate a person's BMI (in metric units of 'cm' and 'kg') and rounding to the first decimal place.
   
   round(([weight]*10000)/([height]^2), 1)

4. Calculate a person's BMI (in English units of 'lb' and 'in') and rounding to the first decimal place.
   
   round(([weight]/([height]^2)*703), 1)

5. Remove a prefix and dash from the beginning of a record name (e.g., convert '4890-2318' to '2318').
   
   mid([record-name], find('-', [record-name]) + 1, length([record-name]) - find('-', [record-name]) + 1)

6. Convert a person's first and last name into a username-looking format (e.g., convert 'John' and 'Doe' to 'john_doe'). We may want to trim the values just in case there were spaces accidentally entered.
   
   lower( concat( trim([first_name]), '_', trim([last_name]) ) )

7. Add leading zeros to an integer, in which the number near the end of the equation represents the maximum length of the result after
Example 1 – Date Functions

- `year([date_field])`
- `month([date_field])`
- `day([date_field])`

<table>
<thead>
<tr>
<th>Interview Date</th>
<th>[pt_int_start_date]</th>
<th>05-10-2024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interview Year</td>
<td><code>year([pt_int_start_date])</code></td>
<td>2024</td>
</tr>
<tr>
<td>Interview Month</td>
<td><code>month([pt_int_start_date])</code></td>
<td>05</td>
</tr>
<tr>
<td>Interview Day</td>
<td><code>day([pt_int_start_date])</code></td>
<td>10</td>
</tr>
</tbody>
</table>
Special Functions
Example 2 – Text Functions

1. Joins the text from multiple text strings with a separator – concat_ws

When is the best time of day to call?

- Morning
- Afternoon
- Evening

Patient Best Time to Call

- Morning | Afternoon

Action Tags / Field Annotation (optional)

@CALCTEXT(concat_ws(" | ", (if([pt_besttime(1])=1, "Morning ","")), (if([pt_besttime(2)])=1, " Afternoon ","")), (if([pt_besttime(3)])=1, " Evening",""))))

Syntax:

concat_ws function: concat_ws (separator, text, text, ...)
2. Extract a text string from a text field – mid, find

**Participant ID**

<table>
<thead>
<tr>
<th>Extract last three numbers of record ID</th>
<th>034</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convert the last three numbers of record ID to number</td>
<td>34</td>
</tr>
</tbody>
</table>

**Action Tags / Field Annotation** (optional)

@CALCTEXT(mid([partid], (find('-', [partid])+1), 3))

**Calculation Equation**

[extract_number]*1

**Syntax:**
- mid function: `mid (text, start position, number of characters)`
- find function: `find (needle, haystack)`
QUESTIONS?
Action Tags

Action tags allow you to modify fields in very specific ways. Once applied, a corresponding action is performed.

In REDCap, action tags begin with the '@' sign and are placed inside a field’s “Action Tags/Field Annotation” box.

Field Label
Test default value

Choices (one choice per line)
1, Yes
2, No

Action Tags / Field Annotation (optional)
@DEFAULT='2'

Adding new Participant ID code 1111
Event Name: Flu Clinic (Arm 1: U19)
Participant ID code
1111
test secondary unique field
Test default value

Learn about @ Action Tags or Using Field Annotation
Action Tags
Example 1 - @IF

- Allows action tags to be set based on conditional logic provided inside an @IF() function.

- Syntax:

  @IF(CONDITION, ACTION TAGS if condition is TRUE, ACTION TAGS if condition is FALSE)

  Example: @IF([yes_no] = '1', @HIDDEN, @HIDECHOICE='3' @READ-ONLY)

- If you wish not to output action tags for a certain condition, set it with a pair of apostrophes/quotes as a placeholder

  Example: @IF([my_radio]='1', @READONLY, ‘’)

*Note: The conditional logic is evaluated only when the survey page or data entry form initially loads; thus, action tag will not be evaluated in real-time as data is entered.*
@CALCTEXT

Evaluates logic that is provided inside a @CALCTEXT() function for Text Box fields *only* and outputs the result as text, typically performed with an if(x,y,z) function - e.g.,

@CALCTEXT(if([gender]='1', 'male', 'female'))

@CALCDATE

Performs a date calculation by adding or subtracting a specified amount of time from a specified date or datetime field and then provides the result as a date or datetime value - e.g.,

@CALCDATE([visit_date], 7, 'd')
On the Interview End Form, a prompt is shown on the form for the interviewer to tell the patient when they will be contacted again for the next interview.

**Prompt at 45-day visit:**

<table>
<thead>
<tr>
<th>Event: FU 45d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participant ID</td>
</tr>
<tr>
<td><strong>PROMPT:</strong> Those are all the questions I have for you today. Thank you very much for taking the time to complete this interview! We will be contacting you again in four and a half months, around 10-30-2023 - 11-15-2023.</td>
</tr>
</tbody>
</table>

**Prompt at 6-month visit:**

<table>
<thead>
<tr>
<th>Event: FU 6m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participant ID</td>
</tr>
<tr>
<td><strong>PROMPT:</strong> Those are all the questions I have for you today. Thank you very much for taking the time to complete this interview! We will be contacting you again in six months, around 05-02-2024 - 05-18-2024.</td>
</tr>
</tbody>
</table>
### Action Tags

#### Example 2 - \(@CALCDATE\) and \(@CALCTEXT\)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Text</th>
<th>Field Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[calc_fu_interval]</td>
<td>Calctext fu interval for piping</td>
<td>text</td>
<td>Field Annotation: @HIDDEN @CALCTEXT(if(event-name)=&quot;fu_45d_arm_1&quot;, 'four and a half months', If(event-name)=&quot;fu_6m_arm_1&quot;, 'six months', ''))</td>
</tr>
<tr>
<td>[calc_6m_start]</td>
<td>6 month interview window start date</td>
<td>text (date_mdy)</td>
<td>Field Annotation: @HIDDEN @CALCDATE([baseline_arm_1] [random_date], 179, 'd')</td>
</tr>
<tr>
<td>[calc_6m_end]</td>
<td>6 month interview window end date</td>
<td>text (date_mdy)</td>
<td>Field Annotation: @HIDDEN @CALCDATE([baseline_arm_1] [random_date], 195, 'd')</td>
</tr>
<tr>
<td>[calc_12m_start]</td>
<td>12 month interview window start date</td>
<td>text (date_mdy)</td>
<td>Field Annotation: @HIDDEN @CALCDATE([baseline_arm_1] [random_date], 364, 'd')</td>
</tr>
<tr>
<td>[calc_12m_end]</td>
<td>12 month interview window end date</td>
<td>text (date_mdy)</td>
<td>Field Annotation: @HIDDEN @CALCDATE([baseline_arm_1] [random_date], 380, 'd')</td>
</tr>
<tr>
<td>[calc_int_window_start]</td>
<td>Window start date for piping</td>
<td>text (date_mdy)</td>
<td>Field Annotation: @HIDDEN @CALCTEXT(if(event-name)=&quot;fu_45d_arm_1&quot;, [calc_6m_start], if(event-name)=&quot;fu_6m_arm_1&quot;, [calc_12m_start], ''))</td>
</tr>
<tr>
<td>[calc_int_window_end]</td>
<td>Window end date for piping</td>
<td>text (date_mdy)</td>
<td>Field Annotation: @HIDDEN @CALCTEXT(if(event-name)=&quot;fu_45d_arm_1&quot;, [calc_6m_end], if(event-name)=&quot;fu_6m_arm_1&quot;, [calc_12m_end], ''))</td>
</tr>
</tbody>
</table>

**PROMPT:** Those are all the questions I have for you today. Thank you very much for taking the time to complete this interview! We will be contacting you again in [calc_fu_interval], around [calc_int_window_start] - [calc_int_window_end].

**descriptive**
QUESTIONS?
Smart Variables

Allow reference information other than data fields (e.g., event, repeat instance, DAG or users)

<table>
<thead>
<tr>
<th>Name of Smart Variable</th>
<th>Description</th>
<th>Example of Usage</th>
<th>Example Input</th>
<th>Example output</th>
</tr>
</thead>
<tbody>
<tr>
<td>[user-name]</td>
<td>The current user’s REDCap username.</td>
<td>[user-name]</td>
<td>jane_doe</td>
<td></td>
</tr>
<tr>
<td>[user-fullname]</td>
<td>The current user’s first and last name (as listed on their My Profile page).</td>
<td>[user-fullname]</td>
<td>Doe Jane</td>
<td></td>
</tr>
<tr>
<td>[user-email]</td>
<td>The current user’s primary email address (as listed on their My Profile page).</td>
<td>[user-email]</td>
<td><a href="mailto:janedoe@example.edu">janedoe@example.edu</a></td>
<td></td>
</tr>
<tr>
<td>[user-dag-name]</td>
<td>The Data Access Group (the unique group name) to which the current user belongs (blank if not in a DAG).</td>
<td>[user-dag-name]</td>
<td>vanderbilt_group</td>
<td></td>
</tr>
<tr>
<td>[user-dag-id]</td>
<td>The group ID number of the Data Access Group to which the current user belongs (blank if not in a DAG).</td>
<td>[user-dag-id]</td>
<td>324</td>
<td></td>
</tr>
<tr>
<td>[user-dag-label]</td>
<td>The name/label of the Data Access Group to which the current user belongs (blank if not in a DAG).</td>
<td>[user-dag-label]</td>
<td>vanderbilt_group</td>
<td></td>
</tr>
<tr>
<td>[record-name]</td>
<td>The record name of the current record.</td>
<td>[record-name]</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>[record-dag-name]</td>
<td>The Data Access Group (the unique group name) to which the current record belongs (blank if not in a DAG).</td>
<td>[record-dag-name]</td>
<td>harvard_site</td>
<td></td>
</tr>
<tr>
<td>[record-dag-id]</td>
<td>The group ID number of the Data Access Group to which the current record belongs (blank if not in a DAG).</td>
<td>[record-dag-id]</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>[record-dag-label]</td>
<td>The name/label of the Data Access Group to which the current record belongs (blank if not in a DAG).</td>
<td>[record-dag-label]</td>
<td>harvard_site</td>
<td></td>
</tr>
<tr>
<td>[is-form]</td>
<td>Detects if the current instrument is being viewed as a data entry form (returns 1 for True, 0 for False), as opposed to a survey.</td>
<td>[is-form]</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Smart Variables

**Example 1:**
Set the option selected at the previous event as the default choice in the current event.

<table>
<thead>
<tr>
<th>Choices (one choice per line)</th>
<th>Copy existing choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Amlodipine (Norvasc)</td>
<td></td>
</tr>
<tr>
<td>2. Diltiazem (Cardizem, Tiazac, etc.)</td>
<td></td>
</tr>
<tr>
<td>3. Nicardipine (Cardene SR)</td>
<td></td>
</tr>
<tr>
<td>4. Nifedipine (Procardia, Adalat)</td>
<td></td>
</tr>
<tr>
<td>5. Verapamil (Calan SR, Verelan)</td>
<td></td>
</tr>
<tr>
<td>8. Other {mr_ccb_oth}</td>
<td></td>
</tr>
</tbody>
</table>

**Action Tags / Field Annotation (optional)**

@DEFAULT='[previous-event-name][mr_ccb:value]'

- **Append ':value' to the variable name inside brackets to pipe the value (not the label) of a multiple-choice field.**
- **To pipe the value of a multiple-choice field as the default value inside the @DEFAULT Action Tag, ':value' must be used.**
Example 2:
Branching logic based on data access groups -
[record-dag-name]='bwh' or [record-dag-name]='inova'

Example 3:
Branching logic based on user roles -
[user-role-label]='medical monitor' or [user-role-label]='admin' or [user-role-label]='unblinded dcc staff'
Smart Variables

Example 4:
Customize message for alert and notification

An unanticipated problem has been reported by [record-dag-label].

Participant ID: [partid]

Event: [event-label]  Event Label

SAE form instance number: [current-instance]  Current repeat instance number

[form-link:safety_report_sae_upirso:Click here to open the form]  Form link

This event is serious, unexpected and related to study participation. This event must be reviewed by the Medical Monitor immediately.

Per the protocol: If the Medical Safety Monitor believes that a finding or trend suggests a threat to the safety of study participants, the Principal Investigator and chair of the DSMB will be notified and the DSMB convened as necessary. The IRB will be notified of any changes in risk to study participants.

Please note: The report status is [ae_site_rpt_status][current-instance]. If the report is "in process", the project manager will work with the local site to obtain outstanding information while you review the report.
Example 4:

Customize message for alert and notification

An unanticipated problem has been reported by BWH.

Participant ID: 15

Event: FU 45d

SAE form instance number: 7

Click here to open the form

This event is serious, unexpected and related to study participation. This event must be reviewed by the Medical Monitor immediately.

Per the protocol: If the Medical Safety Monitor believes that a finding or trend suggests a threat to the safety of study participants, the Principal Investigator and chair of the DSMB will be notified and the DSMB convened as necessary. The IRB will be notified of any changes in risk to study participants.

Please note: The report status is Ready for review. If the report is "in process", the project manager will work with the local site to obtain outstanding information while you review the report.
QUESTIONS?
• Project Dashboards are pages with dynamic content that can be added to a project

• They can utilize special Smart Variables called Smart Functions, Smart Tables, and Smart Charts that can perform aggregate mathematical functions, display tables of descriptive statistics, and render various types of charts, respectively
Project Dashboard Example 1

SMART FUNCTIONS are aggregate mathematical functions that are applied across ALL records in a project.

Smart Functions include min, max, mean, median, sum, stdev, count, and unique.

This project contains 50 records. The average age of all participants is 47.38 (stdev=30.93). The median weight is 109 (min: 47, max: 197).

SMART TABLES display descriptive statistics for fields with each field as a row in the table.

Smart Tables can be displayed with ALL columns by default:

<table>
<thead>
<tr>
<th>Field</th>
<th>Count</th>
<th>Missing</th>
<th>Unique</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Median</th>
<th>StDev</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height (cm)</td>
<td>50</td>
<td>0</td>
<td>35</td>
<td>133</td>
<td>214</td>
<td>177.22</td>
<td>180</td>
<td>24.64</td>
<td>8,861</td>
</tr>
<tr>
<td>Weight (kilograms)</td>
<td>50</td>
<td>0</td>
<td>44</td>
<td>47</td>
<td>197</td>
<td>114.72</td>
<td>109</td>
<td>42.79</td>
<td>5,736</td>
</tr>
<tr>
<td>Race</td>
<td>50</td>
<td>0</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>50</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Or with only specified columns:

<table>
<thead>
<tr>
<th>Field</th>
<th>Mean</th>
<th>StDev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age (years)</td>
<td>47.38</td>
<td>30.93</td>
</tr>
<tr>
<td>Weight (kilograms)</td>
<td>114.72</td>
<td>42.79</td>
</tr>
<tr>
<td>Height (cm)</td>
<td>177.22</td>
<td>24.64</td>
</tr>
</tbody>
</table>

SMART CHARTS can be used to display many types of charts for one or more fields in the project.

Display a scatter plot of two fields (x vs y):

Use line charts with two fields (x vs y):
Project Dashboard

- Project Dashboards
- Alerts & Notifications
- Calendar
- Data Exports, Reports, and Stats
- Data Import Tool
- Data Comparison Tool
- Logging
- Field Comment Log
- File Repository
- User Rights and DAGs
- Customize & Manage Locking/E-signatures
- Data Quality
- API and API Playground
- REDCap Mobile App
- External Modules
**Dashboard title:** Project Dashboard Example 1

**User access:**
Choose who sees this dashboard on their left-hand project menu
(Note: Users with Setup/Design privileges can still access all dashboards via the Project Dashboards page.)
- All users
- Custom user access
  (Choose specific users, roles, or data access groups who will have access)

**Set as "public":**
Enabling this feature below will auto-generate a public link for viewing the dashboard without needing to log in to REDCap.
- Dashboard is publicly viewable by anyone with the public link

**Dashboard content:**
Add any static or dynamic text to be displayed, including Smart Functions, Smart Tables, and Smart Charts.

---

SMART FUNCTIONS are aggregate mathematical functions that are applied across ALL records in a project.

Smart Functions include `min`, `max`, `mean`, `median`, `sum`, `stdev`, `count`, and `unique`.


SMART TABLES display descriptive statistics for fields with each field as a row in the table.

Smart Tables can be displayed with ALL columns by default:

```stats-table:height,weight,race,gender```

Or with only specified columns:

```stats-table:age,weight,height:mean,stdev```

SMART CHARTS can be used to display many types of charts for one or more fields in the project.

Display a scatter plot of two fields (x vs y):

```scatter-plot:height,weight```

Add a third field for grouping (by color):

```scatter-plot:height,weight:gender```

Use line charts with two fields (x vs y):

```line-chart:height,weight```

Add a third field for grouping (by color):

```line-chart:height,weight:gender```
Wizard for creating Smart Functions, Smart Tables, and Smart Charts

This wizard can help you create a new Smart Function, Smart Table, or Smart Chart that you can then use in a Project Dashboard. Simply make selections using the drop-down lists below regarding which Smart Variable you wish to use, which fields to be utilized in it, and other optional features available. Once your selections have been made, you may click the 'Copy to clipboard' button at the bottom, after which you may paste that generated text into the body of your Project Dashboard.

**Step 1) Choose a Smart Variable to create:**

aggregate-count

**Step 2) Choose a field to utilize:**

study_id "Study ID"

[Add another field]

**Step 3) Optional data filtering and other settings:**

By default, Smart Functions, Smart Tables, and Smart Charts will utilize all the data from *all records* in the project. However, you may utilize a subset of the data in the project by limiting them to a specific report's data (using a unique report name), to records belonging to one or more DAGs, and/or to data in specific events (if the project is longitudinal).

Filter the data using a report: -- no filtering by report --

Filter the data using DAGs:

-- no filtering by data access group --

Current user's DAG

Duke

**Step 4) Copy the generated Smart Variable syntax and paste it in your project dashboard:**

[aggregate-count:study_id]  

[Copy to clipboard]
Add a third field for grouping (by color):

Display a bar chart with a single multiple choice field:

Display a pie chart

Or a donut chart
### Smart Variables

<table>
<thead>
<tr>
<th>Name of Smart Variable</th>
<th>Description</th>
<th>Example input</th>
<th>Example output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate Functions, Charts, and Tables</td>
<td>(also known as Smart Functions, Smart Charts, and Smart Tables)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[aggregate-min : fields : parameters]</td>
<td>The minimum value of a field across all records in the project (including all events and/or repeating instances in all records). Multiple fields may be used and must be comma-separated.</td>
<td>[aggregate-min:age]</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[aggregate-min:age,participant_age, other_age]</td>
<td>7</td>
</tr>
<tr>
<td>[aggregate-max : fields : parameters]</td>
<td>The maximum value of a field across all records in the project (including all events and/or repeating instances in all records). Multiple fields may be used and must be comma-separated.</td>
<td>[aggregate-max:age]</td>
<td>95</td>
</tr>
<tr>
<td>[aggregate-mean : fields : parameters]</td>
<td>The mean/average value of a field across all records in the project (including all events and/or repeating instances in all records). Multiple fields may be used and must be comma-separated.</td>
<td>[aggregate-mean:age]</td>
<td>100.1</td>
</tr>
<tr>
<td>[aggregate-median : fields : parameters]</td>
<td>The median value of a field across all records in the project (including all events and/or repeating instances in all records). Multiple fields may be used and must be comma-separated.</td>
<td>[aggregate-median:age]</td>
<td>57</td>
</tr>
<tr>
<td>[aggregate-sum : fields : parameters]</td>
<td>The sum of all values for a field across all records in the project (including all events and/or repeating instances in all records). Multiple fields may be used and must be comma-separated.</td>
<td>[aggregate-sum:age]</td>
<td>9451</td>
</tr>
<tr>
<td>[aggregate-count : fields : parameters]</td>
<td>The count of all values for a field across all records in the project (including all events and/or repeating instances in all records). Multiple fields may be used and must be comma-separated.</td>
<td>[aggregate-count:age]</td>
<td>68</td>
</tr>
</tbody>
</table>
QUESTIONS?
Randomization

Enable randomization:

Project Setup -> Enable optional modules and customizations -> Enable Randomization Module -> Randomization link becomes available on left menu
Randomization

1. Randomization in REDCap works by allowing you to create your custom allocation list, which will serve as a lookup table for deciding how to randomize subjects.

2. Set up your randomization model and all its parameters in step 1

3. Download randomization template in step 2 and send to your statistician

4. Ask your statistician to create randomization table using the variable codes in the template

5. You will need one randomization table for testing and one for production. **The two tables cannot be the same.** Ask your statistician to create a randomization table with fewer number of allocations for testing.
Randomization
Step 1

Step 1: Set up your randomization model and all its parameters

**STEP 1: Define your randomization model**

This step will allow you to define the randomization model you will be implementing and all its parameters, which includes defining strata (if applicable) and optionally randomizing subjects per group/site (if a multi-site study).

**NOTE:** This section is currently locked and uneditable because the randomization setup process has already been completed. If you wish to modify the randomization setup below, you will need to click the Erase Randomization Model button below.

**A) Use stratified randomization?**

It is often necessary to ensure equal treatment among a number of factors. Stratified randomization is the solution to achieve balance within one or more subgroups, such as gender, race, diabetics/non-diabetics, etc. By choosing strata (multiple choice criteria fields), you may then be able to ensure balance within those subgroups. [Tell me more]

**Choose strata** (criteria fields used for stratification; may specify up to 14 multiple choice fields)

| scrn_nitrate (1. Is the patient prescribed a long-acting nitrate?) | for Baseline |

**B) Randomize by group/site?**

If this is a multi-center/multi-site project (or something similar), you may want to stratify the randomization by each group/site. You can select an existing multiple choice field that represents the groups/sites, OR you can use Data Access Groups to stratify by group/site.

- Use Data Access Groups to designate each group/site (5 groups currently exist)
- Use an existing field to designate each group/site

| - select a field - | for |

**C) Choose your randomization field**

This is the field where the allocated randomization (treatment) group will be saved and stored, and is where the Randomize button will appear on your data collection form.

| random_arm (Treatment Arm) | for Baseline |

Save randomization model  Erase randomization model
Randomization
Step 2

Step 2: Download the randomization template and send it to your statistician

**STEP 2: Download template allocation tables (as Excel/CSV files)**

Below are some example files that you may download to get a general idea for how you may structure your own randomization table. You do not have to use any of these. In fact, **we recommend that you NOT use these exact templates** but instead recommend that you merely use them as an example or baseline to start from in order to create your own custom allocation file. After uploading your allocation table in Step 3 below, it will then be used as a lookup table to perform assignments when subjects are being randomized. **NOTE:** Record names (e.g., study ID) should NOT be included as a column in your allocation table, but only the fields listed in the example files below. [More details]

<table>
<thead>
<tr>
<th>Example #1 (basic)</th>
<th>Example #2 (all possible combos)</th>
<th>Example #3 (5x all possible combos)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>random_ascn_nitrate</td>
<td>redcap_data_access_group</td>
<td>NOTES:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>449</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>449</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>449</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0</td>
<td>449</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>450</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>450</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>450</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example #1 (basic)**

**Example #2 (all possible combos)**

**Example #3 (5x all possible combos)**

- Values/labels for "random_ascn_nitrate" (Treatment Arm):
  - 1: Beta Blocker
  - 2: Calcium Channel Blocker

- Values/labels for "ascn_nitrate" (<div class="rich-text-field"></div>):
  - 1: Yes
  - 0: No

- Values/labels for "redcap_data_access_group" (Data Access Group):
  - 449 BWH
  - 450 Duke
  - 451 Inova
  - 452 Kaiser
  - 448 Yale
You will need two randomization tables – one for development and one for production. The two tables must NOT be the same. Ask your statistician to generate a smaller randomization table for testing (development).
Randomization Testing

- Upload the development randomization table for testing
- Test randomization in the development project
- **Very Important** – Do NOT randomize any real participants in your development project

After the randomization table is uploaded, the randomization button will become available in the randomization field. Click the button to randomize test records.
Randomization
Move to Production

• After testing completed and no more changes are needed, upload production randomization table

• Move project to production. All test randomization data will be deleted.

This is the reason why you should never randomize any real participants in development project.
Reminders:

• Once your project is in production status, the allocation tables will become locked and unmodifiable.

• Be sure to include more assignments in your allocation table than you think you will need (to accommodate possible drop-out and drop-in of subjects).

• Record names (e.g., study ID) should NOT be included as a column in your allocation table, but only the fields listed in the example files from Step 2.

• If you need to upload more allocations to the randomization table, contact us at redcap@yale.edu. We can help you append more rows to the randomization table.
To monitor randomization, go to the dashboard tab of the randomization module.

The table below displays the allocation dashboard for use in DEVELOPMENT status. All assignments are grouped to show in aggregate the count of records that have been randomized for each row (i.e. combinations). Assignments that have been used will get counted in the 'Used' column while those that are still unallocated will get counted in the 'Not Used' column. Once all assignments have been used for a given row/combination, it will display a checkmark icon in its row. The headers in the table may be clicked to sort the table by that column either in ascending or descending order.

<table>
<thead>
<tr>
<th>Used</th>
<th>Not Used</th>
<th>Allocated records</th>
<th>Data Access Group</th>
<th>Treatment Arm (random_arm)</th>
<th>1. Is the patient prescribed a long-act... (scrn_nitrate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>49</td>
<td></td>
<td>Yale (379)</td>
<td>Beta Blocker (1)</td>
<td>No (0)</td>
</tr>
<tr>
<td>7</td>
<td>43</td>
<td>YAI-034, 8, 18, 379, 2, 21, 30, 7</td>
<td>Yale (379)</td>
<td>Beta Blocker (1)</td>
<td>Yes (1)</td>
</tr>
<tr>
<td>2</td>
<td>49</td>
<td>9, 28</td>
<td>Yale (379)</td>
<td>Calcium Channel Blocker (2)</td>
<td>No (0)</td>
</tr>
<tr>
<td>6</td>
<td>44</td>
<td>6, 17, 19, 20, 379, 5, 379-6</td>
<td>Yale (379)</td>
<td>Calcium Channel Blocker (2)</td>
<td>Yes (1)</td>
</tr>
<tr>
<td>2</td>
<td>48</td>
<td>363-018, BWH-018</td>
<td>BWH (380)</td>
<td>Beta Blocker (1)</td>
<td>No (0)</td>
</tr>
<tr>
<td>3</td>
<td>47</td>
<td>15, 380-3, 363-034</td>
<td>BWH (380)</td>
<td>Beta Blocker (1)</td>
<td>Yes (1)</td>
</tr>
<tr>
<td>0</td>
<td>50</td>
<td></td>
<td>BWH (380)</td>
<td>Calcium Channel Blocker (2)</td>
<td>No (0)</td>
</tr>
<tr>
<td>3</td>
<td>47</td>
<td>380-4, 380-5, 380-6</td>
<td>BWH (380)</td>
<td>Calcium Channel Blocker (2)</td>
<td>Yes (1)</td>
</tr>
<tr>
<td>0</td>
<td>51</td>
<td></td>
<td>Duke (381)</td>
<td>Beta Blocker (1)</td>
<td>No (0)</td>
</tr>
</tbody>
</table>

Note: If your study has blinded study staff, make sure that you do not give them access to the randomization module or the form containing the randomization variable.
QUESTIONS?
Wrap Up

• Form Display Logic
• Special Functions
  – Calculations
  – Branching Logic
• Action Tags
• Smart Variables
• Project Dashboard
• Randomization

Visit:

REDCap FAQ
Library

Further Questions:
REDCap@yale.edu

Thank You!