

Calculated Fields – REDCap How-To Guide

What are calculated fields?

REDCap has the ability to make real-time calculations on data entry forms and surveys, however it is recommended that calculation field types are not *excessively* utilized on REDCap data collection instruments as they can slow down the webpages.

How do I format calculated fields?

In order for the calculated field to function, it will need to be formatted in a particular way. This is somewhat similar to constructing equations in Excel or with certain scientific calculators.

The variable names/field names can be used as variables in the equation, but you must place [] brackets around each variable. Please be sure that you follow the mathematical order of operations when constructing the equation or else your calculated results might end up being incorrect.

Calculations in REDCap are formatted the same way as Excel, but instead of using the cell names ([A2]+[A3]), the variable names are used ([bpi_q1]+[bpi_q2]).

How do I create a calculated field?

On the Online Designer page, you would select *Calculated Field* as the field type. Type the calculation into the *Calculation Equation* box. If you have test data in the project, you can test the calculation to see if it's valid by selecting a study ID from the *Test calculation with a record* dropdown list.

The screenshot shows the 'Edit Field' dialog box in REDCap. It contains the following fields and options:

- Field Type:** A dropdown menu set to 'Calculated Field', with a red arrow pointing to it.
- Question Number (optional):** A text input field.
- Field Label:** A text input field containing 'Age', with a red arrow pointing to it. A link 'How to use Piping' is next to it.
- Calculation Equation:** A text input field containing the formula `round(datediff([demo_visit_date],[demo_dob],"y"),1)`, with a red arrow pointing to it. A link 'How do I format the equation?' is next to it.
- Variable Name (utilized during data export):** A text input field containing 'demo_age'. A checkbox 'Enable auto naming of variable based upon its Field Label?' is next to it.
- Required?*** Radio buttons for 'No' (selected) and 'Yes'. A note '* Prompt if field is blank' is below.
- Identifier?** Radio buttons for 'No' (selected) and 'Yes'. A note 'Does the field contain Identifying Information (e.g., name, SSN, address)?' is below.
- Custom Alignment:** A dropdown menu set to 'Right / Vertical (RV)'.
- Field Note (optional):** A text input field.
- Test calculation with a record:** A dropdown menu set to '-- select record --', with a red arrow pointing to it. A link 'Clear calculation' is next to it.
- Action Tags / Field Annotation (optional):** A text input field. A link 'Learn about Action Tags or using Field Annotation' is below.
- Buttons:** 'Save' and 'Cancel' buttons at the bottom right.

Can I create a calculation that returns text or a date as a result (Ex: "True" or "False," "[visit_date] + 30 days")?

No, calculations can only result in **numbers**.

What mathematical operations are available for calculated fields?

+	Add
-	Subtract
*	Multiply
/	Divide

Null or blank values can be referenced using "" or "NaN"

Performing calculations on multiple choice questions (Radio Buttons or Dropdown Lists):

REDCap uses the numerical value assigned to the answer as the answer's value/score.

- 0, Never
- 1, Occasionally
- 2, Often
- 3, Always

In this case, REDCap will score *Never* as '0,' *Occasionally* as '1,' *Often* as '2,' and *Always* as '3.'

For scoring questions that are radio buttons/dropdown lists, the *Calculation Equation* would contain the field/variable names of the questions you want to sum.

Edit Field

You may add a new project field to this data collection instrument by completing the fields below and clicking the Save button at the bottom. When you add a new field, it will be added to the form on this page. For an overview of the different field types available, you may view the [Field Types video \(4 min\)](#).

Field Type: Calculated Field

Field Label: Total Score

Calculation Equation: `sum([gcsi_nausea],[gcsi_retch],[gcsi_vomit],[gcsi_fullness],[gcsi_cant_finish_meal],[gcsi_excess_full],[gcsi_appetite_loss],[gcsi_bloating],[gcsi_belly_larger])`

Variable Name: avg_1

Required?* No Yes

Identifier? No Yes

Custom Alignment: Right / Vertical (RV)

Field Note (optional):

Action Tags / Field Annotation (optional):

Save **Cancel**

Note: If two or more answers need to have the same scoring, i.e., both *Never* and *Always* have a score of '4', conditional logic will have to be used instead since REDCap will not allow two answer choices to have the same values/scores.

Performing calculations on multiple choice questions (Checkboxes – select all that apply):

Calculations for Checkbox field types are tricky because they are given a value of '1' if checked and '0' if unchecked. If possible, avoid use calculations for Checkbox field types.

Although the field ('exercise') looks like this:

- 1, Monday
- 2, Tuesday
- 3, Wednesday
- 4, Thursday
- 5, Friday

The data comes out like this:

[exercise__1]	[exercise__2]	[exercise__3]	[exercise__4]	[exercise__5]
1	0	1	0	0

So even though Friday has a value of '5' its values are either '0' or '1.' The '5' only refers to it being the 5th choice on the list. This is because statistical packages would not understand [exercise] = '1' and '3.'

To score this field, each answer choice will need its own calculation, which afterwards can be summed:
[variablename(code)] ('code' refers to the position on the list, aka the value assigned to the answer)

if([exercise(1)] = 1, 1, 0) (if Monday is selected, give me 1, otherwise give me 0)
if([exercise(2)] = 1, 1, 0) (if Tuesday is selected, give me 1, otherwise give me 0)
if([exercise(3)] = 1, 1, 0) (if Wednesday is selected, give me 1, otherwise give me 0)
if([exercise(4)] = 1, 1, 0) (if Thursday is selected, give me 1, otherwise give me 0)
if([exercise(5)] = 1, 1, 0) (if Friday is selected, give me 1, otherwise give me 0)

Then you could sum the five fields above, thereby creating a sixth calculated field.

Can I use conditional logic in a calculated field?

Yes, you may use conditional logic (i.e. an IF/THEN/ELSE statement) by using the function:
if(CONDITION, value if condition is TRUE, value if condition is FALSE)

This construction is similar to "IF" statements in Excel. Provide the condition first (e.g. [weight] = 4), then give the resulting value if it is true, and lastly give the resulting value if the condition is false.

if([weight] > 100, 44, 11)

In this example, if the value of the field 'weight' is greater than 100, then it will give a value of 44, but if 'weight' is less than or equal to 100, it will give 11 as the result.

Consider the question ('xxx') with the following answers:

- 0, Never
- 1, Occasionally
- 2, Often
- 3, Always
- 99, N/A

If someone selects 'N/A' as an answer and you do not want REDCap to score that value as '99,' you want the value to be *blank*, you would use the conditional logic:

if([xxx] < 99, [xxx], "NaN")

In this example, if the value of 'xxx' is less than 99, then it will give the value of the answer that was chosen (0-3), but if 'xxx' is equal to 99 then it will give "NaN" (blank) as the result.

Using the "sum" function vs. the "+" function:

sum([q1],[q2],[q3],[q4]) will give the sum of questions 1-4, even if one of the values is blank.

NOTE: All blank values will be ignored and thus will only return the sum total computed from all numerical, non-blank values.

[q1]+[q2]+[q3]+[q4] will give the sum of questions 1-4 but only if *every* question has a value.

NOTE: Blank values will not be ignored and thus will only return the sum total if all the fields in the equation have values.

Weather to use the "+" or "sum" function depends upon how that particular instrument should be scored.

BMI Calculation:

[weight]*10000/([height]*[height]) = units in kilograms and centimeters

([weight]/([height]*[height]))*703 = units in pounds and inches

Note: 'weight' and 'height' need to be field names in your project. If you called these fields something else, use the names you called those fields instead of the ones listed above.

Calculate the difference between two dates or datetimes (i.e., age at enrollment based upon DOB and date of enrollment, length of hospital stay based on admit and discharge dates):

datediff([date1],[date2], "units", "date format", Return Signed Value)

units

"y"	years	1 year = 365.2425 days
"M"	months	1 month = 30.44 days
"d"	days	
"h"	hours	
"m"	minutes	
"s"	seconds	

date format

"ymd"	Y-M-D (default)
"mdy"	M-D-Y
"dmy"	D-M-Y

- If the date format is not provided, it will default to "ymd".
- Both dates MUST be in the format specified in order to work.

Return Signed Value

false	(default)
true	

- The parameter Return Signed Value denotes the result to be signed or unsigned (absolute value), in which the default value is "false", which returns the absolute value of the difference. For example, if [date1] is larger than [date2], then the result will be negative if Return Signed Value is set to true. If Return Signed Value is not set or is set to false, then the result will ALWAYS be a positive number. If Return Signed Value is set to false or not set, then the order of the dates in the equation does not matter because the resulting value will always be positive (although the + sign is not displayed but implied).

Examples:

datediff([dob],[date_enrolled],"y")	Yields the number of years between the dates for the date_enrolled and dob fields, which must be in Y-M-D format
datediff([dob],"05-31-2007","h","mdy",true)	Yields the number of hours between May 31, 2007, and the date for the dob field, which must be in M-D-Y format. Because returnSignedValue is set to true, the value will be negative if the dob field value is more recent than May 31, 2007.
datediff([dob],[visit_date],"y","mdy")	Yields the number of years between the dates for the dob and visit_date fields, in the M-D-Y format.
datediff("today",[screen_date],"M")	Yields the number of months since the dates for the screen_date and today's date. NOTE: The "today" variable can ONLY be used with date fields and NOT with time, datetime, or datetime_seconds fields. It is strongly recommended, HOWEVER, that you do not use "today" in calculated fields. This is because every time you access and save the form, the calculation will run. For example, if you calculate the age as of today, then a year later you access the form to review or make updates, the elapsed time as of "today" will also be updated (+1 yr). Most users calculate time off of another field (e.g. screening date, enrollment date).

Can fields from different EVENTS be used in calculated fields (longitudinal only)?

Yes, for longitudinal projects (i.e. with multiple events defined), a calculated field's equation may utilize fields from other events (i.e. visits, time-points). The equation format is somewhat different from the normal format because the unique event name must be specified in the equation for the target event. The unique event name must be prepended (in square brackets) to the beginning of the variable name (in square brackets), i.e. **[unique_event_name][variable_name]**. Unique event names can be found listed on the project's *Define My Event's* page on the right-hand side of the events table, in which the unique name is automatically generated from the event name that you have defined.

For example, if the first event in the project is named "Enrollment", in which the unique event name for it is "enrollment_arm_1", then we can set up the equation as follows to perform a calculation utilizing the "weight" field from the Enrollment event: **[enrollment_arm_1][weight]/[visit_weight]**. Thus, presuming that this calculated field exists on a form that is utilized on multiple events, it will always perform the calculation using the value of weight from the Enrollment event while using the value of visit_weight for the current event the user is on.

Can REDCap perform advanced functions in calculated fields?

Yes, it can perform many, which are listed below. NOTE: All function names (e.g. roundup, abs) listed below are case sensitive.

Function	Name/Type of function	Notes / examples
round(number,decimal places)	Round	If the "decimal places" parameter is not provided, it defaults to 0. E.g. To round 14.384 to one decimal place: round(14.384,1) will yield 14.4
roundup(number,decimal places)	Round Up	If the "decimal places" parameter is not provided, it defaults to 0. E.g. To round up 14.384 to one decimal place: roundup(14.384,1) will yield 14.4
rounddown(number,decimal places)	Round Down	If the "decimal places" parameter is not provided, it defaults to 0. E.g. To round down 14.384 to one decimal place: rounddown(14.384,1) will yield 14.3
sqrt(number)	Square Root	E.g. sqrt([height]) or sqrt([(value1]*34)/98.3)
(number)^(exponent)	Exponents	Use caret ^ character and place both the number and its exponent inside parentheses: For example, (4)^(3) or ([weight]+43)^(2)
abs(number)	Absolute Value	Returns the absolute value (i.e. the magnitude of a real number without regard to its sign). E.g. abs(-7.1) will return 7.1 and abs(45) will return 45.
min(number,number,...)	Minimum	Returns the minimum value of a set of values in the format min([num1],[num2],[num3],...)
max(number,number,...)	Maximum	Returns the maximum value of a set of values in the format max([num1],[num2],[num3],...).
mean(number,number,...)	Mean	Returns the mean (i.e. average) value of a set of values in the format mean([num1],[num2],[num3],...).
median(number,number,...)	Median	Returns the median value of a set of values in the format median([num1],[num2],[num3],...).

sum(number,number,...)	Sum	Returns the sum total of a set of values in the format sum([num1],[num2],[num3],...).
stdev(number,number,...)	Standard Deviation	Returns the standard deviation of a set of values in the format stdev([num1],[num2],[num3],...).