

Date Difference Calculation

CCTS Biostatistics Core

January 2023

Calculate the time interval between two dates using a `datediff` function. This type of function can be used to:

- Validate a date field. For example, ensure that participants enter a date that is after the study began but on or before the current date.
- Calculate a participant's age from their date of birth.
- Set branching logic to show or hide fields based on a date calculation.

Syntax

Calculations take the form:

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

where:

- `[date1]` and `[date2]` are the two dates being compared. These may be form fields (e.g., `[dob]`) or text (e.g., "05/01/2020" or "today").
- `"units"` is the unit of time that will be returned. Units may be years ("y"), months ("M"), days ("d"), hours ("h"), minutes ("m"), or seconds ("s").
- `"format"` is the date format, which can be set for a date field using validation rules. Both `[date1]` and `[date2]` must have the same format to use a `datediff` calculation. Options include "ymd", "mdy", and "dmy". If omitted, this value defaults to "ymd".
- `Return Signed Value` indicates whether the returned value will be signed (i.e., can be negative) or unsigned (i.e., absolute value). This value must be either `true` (signed) or `false` (unsigned). If omitted, this value defaults to `false`.

Examples

Real-Time Date Validation

Display a warning on the form if a `datediff` condition is met. For example, warn a user if their enrollment date comes before the date of their first visit.

1. Create a Descriptive Text field. Add a warning that the entered date is not in the valid range.
2. Use branching logic to show the warning if the following rule is violated:

```
datediff([date_enrolled],[date_1stvisit],"d","mdy",true)<0
```

Validation Using a Data Quality Rule

Find records that violate a rule using the Data Quality module. For example, identify records with an enrollment date that comes before the date of their first visit.

1. Open the Data Quality module. Create a new rule to check for invalid data using the following function:
`datediff([date_enrolled], [date_1stvisit], "d", "mdy", true)<0`
2. Execute the rule to see a list of records with invalid date values. Records can be viewed, modified, or excluded.
3. Check `Execute in real time on data entry forms` to run the rule at the time of data entry. Note that this option only works on data entry forms, not on surveys.

Calculated Date on Data Entry Forms

Currently, REDCap doesn't have the function to show the calculated date in a data entry form without using an external module or plugin. Instead, you may utilize Scheduling/Calendar for longitudinal databases.

Tips

- You may use the variables "today" and "now" in `datediff` functions. However, this practice is not recommended for calculated fields or branching logic. If you use these variables in a calculation, the results will change every time you access and save the data entry form, based on the value of the current date. For example, a function might calculate a patient's age at the time of enrollment. If you access the record a year later to review or make updates, the elapsed time (i.e., age) as of "today" will also be updated (+1 year). Instead, you are advised to base time calculations on fixed dates contained in form fields (e.g., screening date or enrollment date).
- If the date field is in a repeated longitudinal instrument, the unique event name, assigned automatically by REDCap, must be added in front of the variable name. For example:
`datediff([event_1_arm_1][date_enrolled], [event_1_arm_1][date_1stvisit], "d", "mdy", true)<0`