REDCAP
Logic Syntax (202)

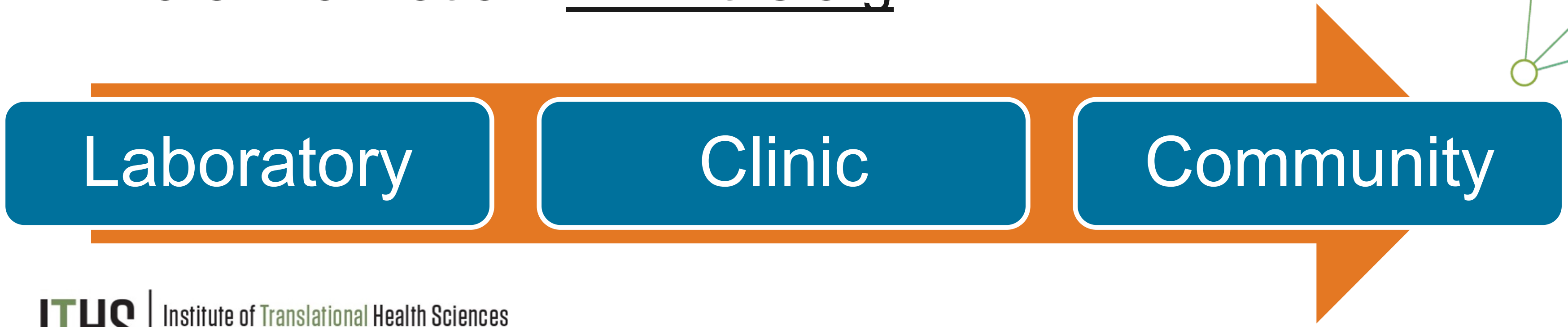
Institute of Translational Health Sciences

ACCELERATING RESEARCH. IMPROVING HEALTH.

Research Electronic Data Capture

ITHS Focus

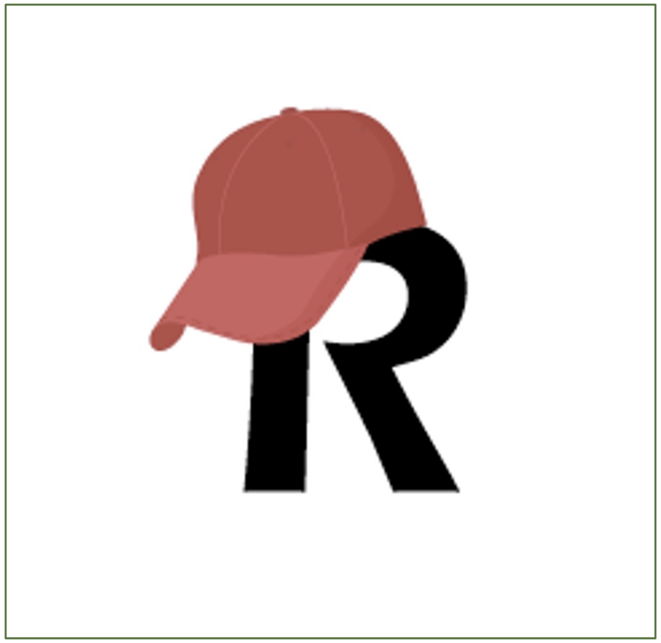
- Speeding science to clinical practice for the benefit of patients and communities.
- Promotes translation of scientific discovery by:
  - ❑ Fostering innovative research
  - ❑ Cultivating multi-disciplinary partnerships
  - ❑ Training the next generation of researchers
- More information: www.iths.org



ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

LEARNING OBJECTIVES

- Branching logic basics
- Simple logic statements
- Complex logic statements
- Special functions
- Longitudinal branching logic
- Interplay with action tags
- Creative Uses and Tips



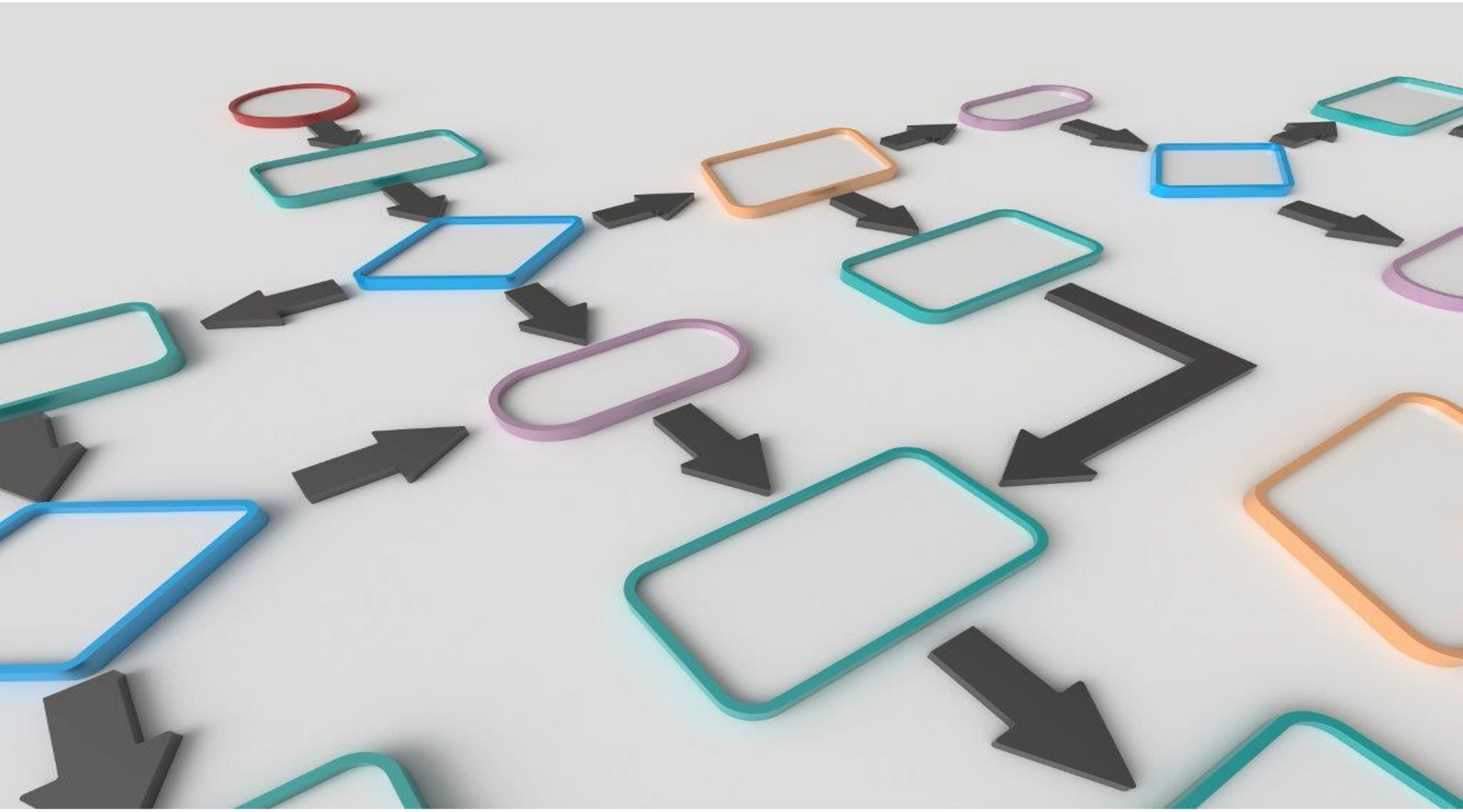
What is syntax in REDCap?

Allows you to customize your REDCap project

- Syntax refers to:
  - The set of rules that determine the arrangement of sentences in a language.
- REDCap utilizes specialized logic syntax for features such as:
  - Branching logic
  - Calculations
  - Report Building
  - Data Quality Rules
  - Custom Dashboards

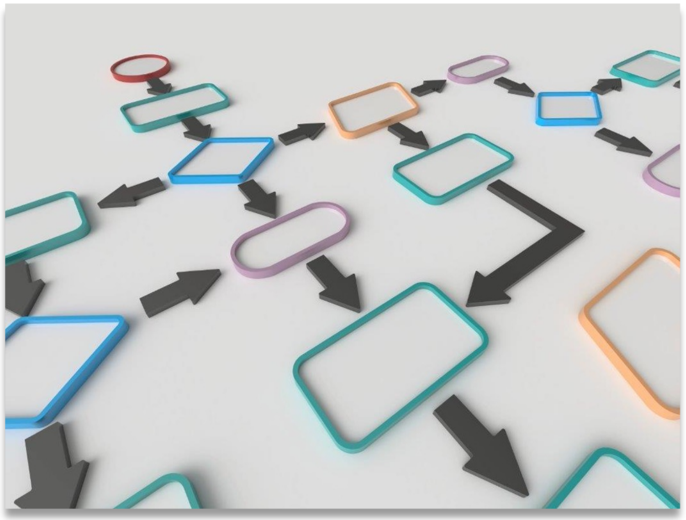
ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

BRANCHING LOGIC



ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

What is Branching Logic?



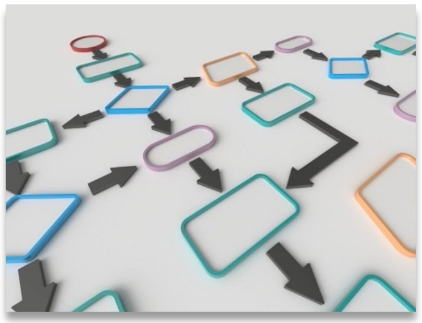
Adds flexibility to your instruments

- The art of hiding or showing fields
- Based on previously entered values
- Limited to a single project
- Drag and Drop Method:
  - The “easy” way
  - Reduced flexibility
- Advanced Syntax Method:
  - The “hard” way
  - Programming experience helps
  - Allows you to get creative
  - Can be used both in the online interface and in the data dictionary

Reverse your thought process

- Logic located in the "source" question
- Directs you to "skip" to a question down the line
- Very linear
- Hard to account for complex logic
- Mostly found on paper forms

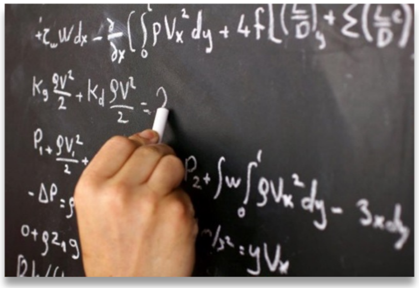
Classic skip logic



- Logic located in the "destination" question
- Hides or shows the question
- Allows for multiple logic pathways (e.g., Inclusion criteria)
- Can get very complex
- Extensively used in REDCap

Branching logic

The language of logic in REDCap



- AND Statement
  - **and**
- OR statement
  - **or**
- Equals statement
  - **=**
- NOT statement
  - **<>**

Logic

- Standard math
  - **+, -, /, ***
- Comparisons
  - **>, <, >=, <=**
- Order of operations
  - **Parentheses ()**

Math

ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Building a basic logic statement

```
}
if (!isIdentityAssertion) {
String passwordWant = null;
try {
    passwordWant = database.getUserPassword(userName);
} catch (NotFoundException shouldNotHappen) {}
    String passwordHave = getPasswordHave(userName, callbacks);
    if (passwordWant == null || !passwordWant.equals(passwordHave)) {
        throwFailedLoginException(
            "Authentication Failed: User " + userName + " bad password. " +
            "Have " + passwordHave + ". Want " + passwordWant + ".";
        );
    }
} else {
    // anonymous login - let it through?
    System.out.println("\tempty userName");
}
loginSucceeded = true;
principalsForSubject.add(new WLSUserImpl(userName));
addGroupsForSubject(userName);
return loginSucceeded;
```

- End result always needs to be a “true” or a “false”
- Define the **Variable**
    - Use brackets (e.g., **[variable1]**)
    - Use the variable name instead of the field label
- Put in an **Operator**
    - e.g., **= , <>, >=, <=, >, <**
- Declare your **Comparison Value**
    - Can be a ”hard” value like a number or a date
    - Can be another variable
    - Using single quotes vs double quotes
    - When not to use quotes
        - **>=, <=, >, <**

What is the logic?

ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Building a basic logic statement

```
}
if (!isIdentityAssertion) {
 String passwordWant = null;
 try {
   passwordWant = database.getUserPassword(userName);
 } catch (NotFoundException shouldNotHappen) {}
   String passwordHave = getPasswordHave(userName, callbacks);
   if (passwordWant == null || !passwordWant.equals(passwordHave)) {
     throwFailedLoginException(
       "Authentication Failed: User " + userName + " bad password. " +
       "Have " + passwordHave + ". Want " + passwordWant + "."
     );
   }
 }
} else {
 // anonymous login - let it through?
 System.out.println("\tempty userName");
}
loginSucceeded = true;
principalsForSubject.add(new WLSUserImpl(userName));
addGroupsForSubject(userName);
return loginSucceeded;
```

- End result always needs to be a “true” or a “false”
- Define the **Variable**
  - Use brackets (e.g., **[variable1]**)
  - Use the variable name instead of the field label
- Put in an **Operator**
  - e.g., **= , <>, >=, <=, >, <**
- Declare your **Comparison Value**
  - Can be a ”hard” value like a number or a date
  - Can be another variable
  - Using single quotes vs double quotes
  - When not to use quotes
    - **>=, <=, >, <**

[age_of_child] < 18

ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Branching logic example 1

Simple statement (single/radio)

Basic statements

Simple (Single/Radio)

And

Or

Not

Empty

Complex statements

Logic context

- You want to ask the question:
  Is the participant on Medicare?
  But this is only relevant for people 65 and over.

Needed elements

- Variable: [age]
- Operator: >=
- Comparison value: 65

Branching logic statement

- **[age] >= 65**

Branching logic example 2

Simple statement (single/radio)

Basic statements

Simple (Single/Radio)

And

Or

Not

Empty

Complex statements

Logic context

- You want to show a warning when the age of a child is greater than or equal to the age of the parent

Needed elements

- Variable: [agechild]
- Operator: >=
- Comparison value: [ageparent]

Branching logic statement

- **[agechild] >= [ageparent]**

ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Branching logic example 3

Simple statement (checkbox)

Logic context

- You want to ask the question:
  Did the participant get vaccinated for malaria?
  But this is only relevant for people who recently went to a country where malaria is prevalent.

Needed elements

- Variable: [country(3)]
- Operator: =
- Comparison value: ‘1’

Branching logic statement

- **[country(3)] = ‘1’**

Branching logic example 4

And statement

Basic statements

Simple (Single/Checkbox)

And

Or

Not

Empty

Complex statements

Logic context

- You want to ask the question:
  How did the medication affect your allergy symptoms?
  But this is only relevant for people who have allergy symptoms and take the medication.

Needed elements

- Variables: [allergy] and [symptoms]
- Operator: =
- Comparison value: ‘1’

Branching logic statement

- **[allergy] = ‘1’ and [symptoms] = ‘1’**

Branching logic example 5

Or statement

Basic statements

Simple (Single/Checkbox)

And

Or

Not

Empty

Complex statements

Logic context

- You display a warning to warn for ineligibility when a participant is either a smoker or a drug user.

Needed elements

- Variables: [smoker] and [drugs]
- Operator: =
- Comparison value: ‘1’

Branching logic statement

- **[smoker] = ‘1’ or [drugs] = ‘1’**

Branching logic example 6
Or statement

Basic statements

- Simple (Single/Checkbox)
- And
- **Or**
- Not
- Empty

Complex statements

Logic context

- You ask further questions when the participant indicates they are currently a smoker or previously a smoker.

Needed elements

- Variables: [smoker]
- Operator: =
- Comparison value: ‘1’, ‘2’

Branching logic statement

- **[smoker] = ‘1’ or [smoker] = ‘2’**

ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Branching logic example 7
Not statement

Basic statements

- Simple (Single/Checkbox)
- And
- Or
- **Not**
- Empty

Complex statements

Logic context

- You want to ask the question:
  How many packs of cigarettes do you smoke on average currently or when you were a smoker?
  But you only want to ask this when people have indicated they are or were a smoker.

Needed elements

- Variable: [smoker]
- Operator: <>
- Comparison value: '0' *(Never been a smoker is coded as 0)*

Branching logic statement

- **[smoker] <> '0'**

ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Branching logic example 8
Empty statement

Basic statements

Simple (Single/Checkbox)

And

Or

Not

Empty

Complex statements

ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Logic context

- You want to display a warning when the date of birth field has not been filled out. But the warning needs to disappear if the date of birth field has a value in it.

Needed elements

- Variable: [dob]
- Operator: =
- Comparison value: ‘’ *(two single quotes)*

Branching logic statement

- **[dob] = ‘’**

Branching logic example 9

Complex statement

```
}
if (!isIdentityAssertion) {
 String passwordWant = null;
 try {
   passwordWant = database.getUserPassword(userName);
 } catch (NotFoundException shouldNotHappen) {}
   String passwordHave = getPasswordHave(userName, callbacks);
   if (passwordWant == null || !passwordWant.equals(passwordHave)) {
     throwFailedLoginException(
       "Authentication Failed: User " + userName + " bad password, " +
       "Have " + passwordHave + ". Want " + passwordWant + ".");
     );
   }
 }
} else {
 // anonymous login - let it through?
 System.out.println("\tempty userName");
}
loginSucceeded = true;
principalsForSubject.add(new WLSUserImpl(userName));
addGroupsForSubject(userName);
return loginSucceeded;
```

Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Logic context

- You want to display a question about mid life crisis, but only when the participant is outside of the standard mid life crisis age range and the date of birth field has been filled out.

Needed elements

- Variable: [age], [dob]
- Operator: >=, <=, <>
- Comparison value: '0','39','63','120','' *(two single quotes)*

Branching logic statement

- **([age] >= 0 and [age] <= 39 and [dob] <> '') or ([age] >= 63 and [age] <= 120 and [dob] <> '')**
- **(([age] >= 0 and [age] <= 39) or ([age] >= 63 and [age] <= 120)) and [dob] <> ''**

Branching logic example 10

Date differential

Logic context

- You want to display a question about how somebody's heart attack affected their work life, but only if they had the heart attack when they were younger than 65.

Needed elements

- Variable: [dob], [date_of_attack]
- Function: datediff([date1],[date2],"units","format")
- Operator: <=
- Comparison value: '65'

Branching logic statement

- **(datediff([dob],[date_of_attack],'y'))<=65**

ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Branching logic example 11

Date differential

Logic context

- You want to display a question about a person's retirement but only if they are over the age of 65 at the time of the survey.

Needed elements

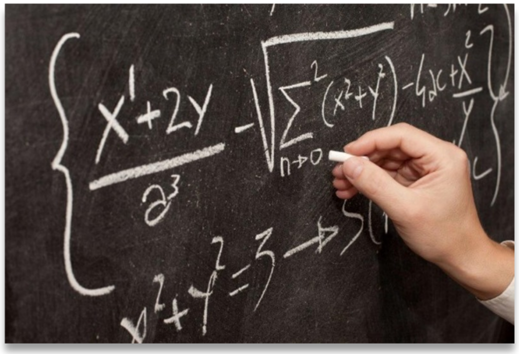
- Variable: [dob], 'today'
- Function: datediff([date1],[date2],"units","format")
- Operator: <=
- Comparison value: '65'

Branching logic statement

- **(datediff([dob],'today','y'))>65**

ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Branching logic example 12
Sum statement



Logic context

- You want to ask a question about depression when the total score of a depression scoring tool reaches above a certain value.

Needed elements

- Variable: [depr1], [depr2], [depr3]
- Function: sum()
- Operator: >=
- Comparison value: '4'

Branching logic statement

- **(sum([depr1],[depr2],[depr3]))>=4**

Branching logic example 13
Contains statement

Logic context

- You want to ask for a survey respondent's private email if they provide a university email address in their initial response.

Needed elements

- Variable: [email]
- Function: contains()
- Comparison value: '.edu'

Branching logic statement

- **contains([email],'.edu')**

ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Branching logic example 14
If statement

Logic context

- You want to ask a question about depression when the total score of a depression scoring tool reaches above a certain value. However, you've built in a "prefer not to answer" response for the first question that you coded as '99'. You need to filter out this response from your logic.

Needed elements

- Variable: [depr1], [depr2], [depr3]
- Function: if(), sum()
- Operator: >=,=
- Comparison value: '4','99','0'

Branching logic statement

- **(sum((*if([depr1]='99','0', [depr1])*) ,[depr2],[depr3]))>=4**

Branching logic example 15

Nested if statement

ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Logic context

- You want to ask a question about depression when the total score of a depression scoring tool reaches above a certain value. However, you've built in a ”prefer not to answer” response for the first question that you coded as ’99’. You've also added an option for “unknown” (98). You need to filter out these responses from your logic.

Needed elements

- Variable: [depr1], [depr2], [depr3]
- Function: if(), sum()
- Operator: >=,=
- Comparison value: ‘4’,’99’,’98’,’0’

Branching logic statement

- **(sum(**
  (if([depr1]=‘99’,’0’,
  (if([depr1]=‘98’,’0’,[depr1]))))
  ,[depr2],[depr3]))>=4

Branching logic

More functions

Location

- For other complex functions see Special Functions

- Examples:
  - Rounding
  - Square root
  - Mean
  - Median
  - Exponents
  - Minimum/Maximum
  - Standard deviation
  - Logarithm
  - Is value a number?

ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Diagram your logic

For complex logic, draw out your logic in a flow chart.

This will help you visualize your logic flow and allow you to troubleshoot any issues.

Longitudinal Branching Logic

- **Classic Branching logic**
  - Define the **variable**
  - Put in an **operator**
  - Declare your **comparison value**

```
[age_of_child] < 18
```

- **Longitudinal Branching Logic**
  - Define the **event**
  - Define the **variable**
  - Put in an **operator**
  - Declare your **comparison value**

```
[baseline_arm_1][age_of_child] < 18
```

ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Interplay with action tags

Action tags usually "win out" over branching logic

- Action tags and branching logic can be used concurrently if desired.
- Most action tags do not affect branching logic
- Exceptions:
  - @HIDDEN
    Hides the field regardless of the logic result
  - @DEFAULT
    Will only prefill a field if the field is shown initially. If it's hidden with logic the @DEFAULT tag will not work!

ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Report Building and Data Quality Rule Tips

Quick way to create advanced logic syntax

- Report Builder Step 3 allows you to create advanced logic syntax utilizing a user friendly, dynamic interface.
- Use the “Switch format: Use advanced logic” link to get the logic syntax after you build the filters.
- Test your logic utilizing the “Data Quality” feature to see if your logic is returning the desired records.

Creative Use: Cascading logic

Use cascading logic to simplify your logic

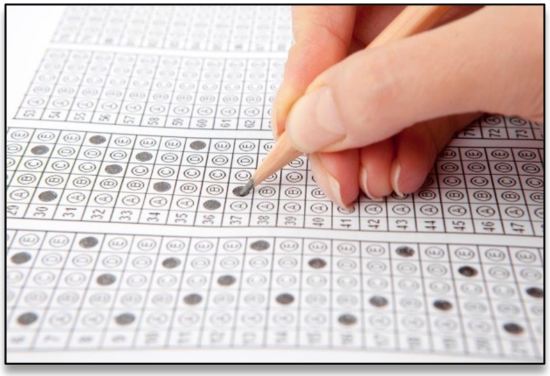
- Cascading logic can greatly simplify all your logic statements.
- Each statement would only need to look at it's preceding variable
- e.g., Cascading medication lists:

Variable	Logic	Non cascading logic
rx1		
rx2	[rx1]<>""	[rx1]<>""
rx3	[rx2]<>""	[rx1]<>"" and [rx2]<>""
rx4	[rx3]<>""	[rx1]<>"" and [rx2]<>"" and [rx3]<>""

rx4 branching logic: [rx3] <> ‘’

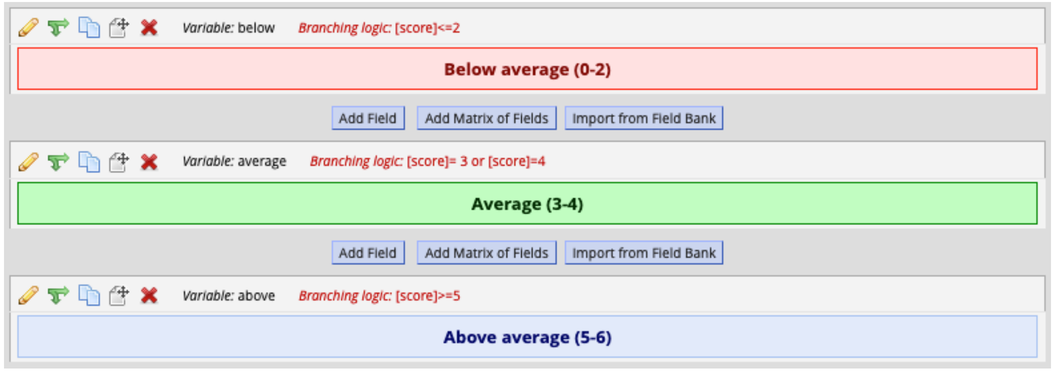
ITHS Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Creative Use: Score evaluation



Combining logic with descriptive fields to show situational messages

- Adding logic to descriptive fields is a great way of communicating certain results
- e.g., Scoring tool evaluation:
  - Calculated field that generates a score
  - Three descriptive fields:
    - Below average
    - Average
    - Above average
  - Example: https://is.gd/logicdemo



Variable: below *Branching logic:* [score]<=2

Below average (0-2)

Add Field | Add Matrix of Fields | Import from Field Bank

Variable: average *Branching logic:* [score]= 3 or [score]=4

Average (3-4)

Add Field | Add Matrix of Fields | Import from Field Bank

Variable: above *Branching logic:* [score]>=5

Above average (5-6)

ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Complex Branching Logic Example

Longitudinal project – only show the field if multiple screenings and consents have been completed over all arms of the study.

Show the field ONLY if: ([0_arm_1][screener_end_action]="done" OR [1_arm_1][screener_end_action]="done"
OR [2_arm_1][screener_end_action]="done" OR [3_arm_1][screener_end_action]="done" OR
[4_arm_1][screener_end_action]="done" OR [5_arm_1][screener_end_action]="done" OR
[6_arm_1][screener_end_action]="done" OR [7_arm_1][screener_end_action]="done" OR
[8_arm_1][screener_end_action]="done" OR [9_arm_1][screener_end_action]="done") AND
[0_arm_1][study_region]="2" AND (([0_arm_1][manage_consent_start]<>"1" AND
[0_arm_1][consent_form_complete]<>"2" AND [0_arm_1][eligibility_screener_complete]="2") OR
([1_arm_1][manage_consent_start]<>"1" AND [1_arm_1][consent_form_complete]<>"2" AND
[1_arm_1][eligibility_screener_complete]="2") OR ([2_arm_1][manage_consent_start]<>"1" AND
[2_arm_1][consent_form_complete]<>"2" AND [2_arm_1][eligibility_screener_complete]="2") OR
([3_arm_1][manage_consent_start]<>"1" AND [3_arm_1][consent_form_complete]<>"2" AND
[3_arm_1][eligibility_screener_complete]="2") OR ([4_arm_1][manage_consent_start]<>"1" AND
[4_arm_1][consent_form_complete]<>"2" AND [4_arm_1][eligibility_screener_complete]="2") OR
([5_arm_1][manage_consent_start]<>"1" AND [5_arm_1][consent_form_complete]<>"2" AND
[5_arm_1][eligibility_screener_complete]="2") OR ([6_arm_1][manage_consent_start]<>"1" AND
[6_arm_1][consent_form_complete]<>"2" AND [6_arm_1][eligibility_screener_complete]="2") OR
([7_arm_1][manage_consent_start]<>"1" AND [7_arm_1][consent_form_complete]<>"2" AND
[7_arm_1][eligibility_screener_complete]="2") OR ([8_arm_1][manage_consent_start]<>"1" AND
[8_arm_1][consent_form_complete]<>"2" AND [8_arm_1][eligibility_screener_complete]="2") OR
([9_arm_1][manage_consent_start]<>"1" AND [9_arm_1][consent_form_complete]<>"2" AND
[9_arm_1][eligibility_screener_complete]="2"))

Thank You!

Questions?

CONNECT WITH ITHS

www.iths.org

@ITHS_UW

/ithsuw

/InstituteofTranslationalHealthSciences

ITHS | Institute of Translational Health Sciences
ACCELERATING RESEARCH. IMPROVING HEALTH.

Visit ITHS.org to Become an ITHS Member

Join a unique catalyst that accelerates discoveries to practice.

Access

Members gain access the different research services, resources, and tools offered by ITHS, including the ITHS Research Navigator.

Education and Training

Members can access a variety of workforce development and mentoring programs and apply for formal training programs.

Funding

Members can apply for local and national pilot grants and other funding opportunities. ITHS also offers letters of support for grant submissions.

Collaboration

Members can connect with collaborators across the CTSA consortium.