Extracting Metadata from Stata Datasets

Suzanna Vidmar and Luke Stevens

Clinical Epidemiology and Biostatistics Unit
Murdoch Children’s Research Institute
Data sharing and storage

• To enable data sharing, the data should be stored in a format that does not required a particular version of a particular statistical package

• At the conclusion of a study, data should be stored in a retrievable format, and not one that may become obsolete

• The safest retrievable format is to have the data stored in CSV or text files

• Stata’s `export delimited` command writes data from a Stata dataset to a text file
But what do the data mean?

Without a description of the data, the data file is of limited use

<table>
<thead>
<tr>
<th>partid</th>
<th>plpt_nassda</th>
<th>ccpp_actical</th>
<th>p1sh_valudec</th>
</tr>
</thead>
<tbody>
<tr>
<td>121</td>
<td>6</td>
<td>25461</td>
<td>.32</td>
</tr>
<tr>
<td>207</td>
<td>3</td>
<td>19483</td>
<td>.44</td>
</tr>
<tr>
<td>153</td>
<td>5</td>
<td>25618</td>
<td>.036</td>
</tr>
<tr>
<td>114</td>
<td>4</td>
<td>20159</td>
<td>.87</td>
</tr>
<tr>
<td>215</td>
<td>7</td>
<td>23876</td>
<td>.02</td>
</tr>
</tbody>
</table>
• **Metadata** is data that describes other data

- My focus is on variable-level metadata, also known as a data dictionary
- Examples of variable-level metadata are data types, variable labels and value labels

**Metadata is a love note to the future**
Extracting the data dictionary from Stata

filename .CSV
But wait, there’s more!

Data and metadata can be imported into data capture software such as REDCap

<table>
<thead>
<tr>
<th>Variable / Field N Form Name</th>
<th>Section Header</th>
<th>Field Type</th>
<th>Field Label</th>
<th>Choices, Calculations, OR Slider Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>enrolid</td>
<td>demographics</td>
<td>text</td>
<td>Enrolment ID</td>
<td></td>
</tr>
<tr>
<td>screenid</td>
<td>demographics</td>
<td>text</td>
<td>Screening ID</td>
<td></td>
</tr>
<tr>
<td>dob</td>
<td>demographics</td>
<td>text</td>
<td>DOB</td>
<td></td>
</tr>
<tr>
<td>gender</td>
<td>demographics</td>
<td>dropdown</td>
<td>Gender</td>
<td>1, Male</td>
</tr>
<tr>
<td>dtenrol</td>
<td>demographics</td>
<td>text</td>
<td>Enrolment Date</td>
<td></td>
</tr>
<tr>
<td>ageenrol_error</td>
<td>demographics</td>
<td>descriptive</td>
<td>Age is out of range</td>
<td></td>
</tr>
<tr>
<td>constipation_yrs</td>
<td>demographics</td>
<td>text</td>
<td>Length of time constipated (years)</td>
<td></td>
</tr>
<tr>
<td>constipation_mths</td>
<td>demographics</td>
<td>text</td>
<td>Length of time constipated (months)</td>
<td></td>
</tr>
<tr>
<td>hpac_yh</td>
<td>demographics</td>
<td>dropdown</td>
<td>Has the child’s constipation been managed by a medical or allied health practitioner?</td>
<td>0, No</td>
</tr>
<tr>
<td>hpac_type</td>
<td>demographics</td>
<td>checkbox</td>
<td>What type of health practitioner has been treating your child’s constipation?</td>
<td>1, GP</td>
</tr>
<tr>
<td>hpac_spec</td>
<td>demographics</td>
<td>text</td>
<td>Specify who managed your child’s constipation.</td>
<td></td>
</tr>
<tr>
<td>treated_yrs</td>
<td>demographics</td>
<td>text</td>
<td>Years treated for constipation</td>
<td></td>
</tr>
<tr>
<td>treated_mths</td>
<td>demographics</td>
<td>text</td>
<td>Months treated for constipation</td>
<td></td>
</tr>
<tr>
<td>meconium_24</td>
<td>demographics</td>
<td>dropdown</td>
<td>Meconium passed &gt;24hrs</td>
<td>0, Absent</td>
</tr>
<tr>
<td>meconium_48</td>
<td>demographics</td>
<td>dropdown</td>
<td>Meconium passed &gt;48hrs</td>
<td>0, Absent</td>
</tr>
</tbody>
</table>
Features of REDCap

- Secure, web-based application for research databases and surveys
- Very easy to use
- Audit trail
- User permission controls
- Data quality measures
- Data export to statistical software
- Generate summary report and letters

https://projectredcap.org/
Building a REDCap database

• As with all data capture software, data entry forms can be developed within REDCap
• A REDCap database can also be built by uploading an external data dictionary
Example using metadatacsv.ado

declaration
decl_name analysis
dict_example.csv

declaration
decl_name analysis
dict_example.csv

<table>
<thead>
<tr>
<th>idno</th>
<th>hosp</th>
<th>cob</th>
<th>gender</th>
<th>qu1</th>
<th>qu2</th>
<th>qu3</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RC</td>
<td>Australia</td>
<td>Male</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>7.442102</td>
</tr>
<tr>
<td>2</td>
<td>RC</td>
<td>Australia</td>
<td>Female</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>6.8682</td>
</tr>
<tr>
<td>3</td>
<td>WCH</td>
<td>Australia</td>
<td>Missing</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>5.956524</td>
</tr>
<tr>
<td>4</td>
<td>RC</td>
<td>Missing</td>
<td>Missing</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>5.199898</td>
</tr>
<tr>
<td>5</td>
<td>PCH</td>
<td>Australia</td>
<td>Missing</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>11.08253</td>
</tr>
<tr>
<td>6</td>
<td>WCH</td>
<td>Australia</td>
<td>Missing</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>7.455984</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>name</th>
<th>varlab</th>
<th>type</th>
<th>isnumeric</th>
<th>format</th>
<th>vallab</th>
<th>choices</th>
</tr>
</thead>
<tbody>
<tr>
<td>idno</td>
<td>ID number</td>
<td>float</td>
<td>1</td>
<td>%9.0g</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hosp</td>
<td>Hospital</td>
<td>str3</td>
<td>0</td>
<td>%9s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cob</td>
<td>Country of birth</td>
<td>float</td>
<td>1</td>
<td>%14.0g</td>
<td>coblab</td>
<td>-1, Missing</td>
</tr>
<tr>
<td>gender</td>
<td>Gender</td>
<td>float</td>
<td>1</td>
<td>%9.0g</td>
<td>genderlab</td>
<td>-1, Missing</td>
</tr>
<tr>
<td>qu1</td>
<td>Question 1</td>
<td>float</td>
<td>1</td>
<td>%9.0g</td>
<td>noyes</td>
<td>0, No</td>
</tr>
<tr>
<td>qu2</td>
<td>Question 2</td>
<td>float</td>
<td>1</td>
<td>%9.0g</td>
<td>noyes</td>
<td>0, No</td>
</tr>
<tr>
<td>qu3</td>
<td>Question 3</td>
<td>float</td>
<td>1</td>
<td>%9.0g</td>
<td>noyes</td>
<td>0, No</td>
</tr>
<tr>
<td>age</td>
<td>Age (yrs)</td>
<td>float</td>
<td>1</td>
<td>%9.0g</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Directory and file name

describe, replace
local fullpath: char _dta[d_filename]

mata: st_local("fullname", pathbasename("`fullpath'"))

local length=strpos("`fullname'",".")-1

local filestub=substr("`fullname'",1,`length')
Directory and filename

describe, replace
local fullname: char _dta[d_filename]
• di "fullpath"
• C:\Users\suzanna.vidmar\Documents\Suzanna\Metadata\example.dta

mata: st_local("fullname", pathbasename("``fullpath''"))

local length=strpos("``fullname''","."))-1

local filestub=substr("``fullname''",1,`length')
Directory and filename

describe, replace
local fullname: char _dta[d_filename]
• di "`fullname"
• C:\Users\suzanna.vidmar\Documents\Suzanna\Metadata\example.dta

mata: st_local("fullname", pathbasename("`fullpath'"))) 
• di "`fullname"
• example.dta

local length=strpos("`fullname'",".")-1

local filestub=substr("`fullname'",1,`length')
Directory and file name

describe, replace
local fullpath: char _dta[d_filename]
• di "\fullpath"
• C:\Users\suzanna.vidmar\Documents\Suzanna\Metadata\example.dta

mata: st_local("fullname", pathbasename("\fullpath"))
• di "\fullname"
• example.dta

local length=strpos("\fullname","."|-1
• di "\length"
• 7

local filestub=substr("\fullname",1,\length")
Directory and file name

describe, replace
local fullpath: char _dta[d_filename]
  • di "\fullpath"
  • C:\Users\suzanna.vidmar\Documents\Suzanna\Metadata\example.dta

mata: st_local("fullname", pathbasename("\fullpath"))
  • di "\fullname"
  • example.dta

local length=strpos("\fullname"",".")-1
  • di "\length"
  • 7

local filestub=substr("\fullname",1,\length')
  • di "\filestub"
  • example
Saving data dictionary

export delimited "dict_`filestub'.csv", replace

Saves the data file:

dict_example.csv
describe, replace

• `describe` usually produces a written report

• When the `replace` option is specified, instead of a report the data in memory are replaced with a dataset containing the information that would have been presented in the report. The new dataset has an observation for each variable in the original data.
### describe

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Storage type</th>
<th>Display format</th>
<th>Value label</th>
</tr>
</thead>
<tbody>
<tr>
<td>idno</td>
<td>float</td>
<td>$9.0g</td>
<td>ID number</td>
</tr>
<tr>
<td>hosp</td>
<td>str3</td>
<td>$9s</td>
<td>Hospital</td>
</tr>
<tr>
<td>cob</td>
<td>float</td>
<td>$14.0g</td>
<td>Country of birth</td>
</tr>
<tr>
<td>gender</td>
<td>float</td>
<td>$9.0g</td>
<td>Gender</td>
</tr>
<tr>
<td>qu1</td>
<td>float</td>
<td>$9.0g</td>
<td>noyes</td>
</tr>
<tr>
<td>qu2</td>
<td>float</td>
<td>$9.0g</td>
<td>noyes</td>
</tr>
<tr>
<td>qu3</td>
<td>float</td>
<td>$9.0g</td>
<td>noyes</td>
</tr>
<tr>
<td>age</td>
<td>float</td>
<td>$9.0g</td>
<td>Age (yrs)</td>
</tr>
</tbody>
</table>

### describe, replace

<table>
<thead>
<tr>
<th>Position</th>
<th>Name</th>
<th>Type</th>
<th>Isnumeric</th>
<th>Format</th>
<th>Value label</th>
<th>Variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>idno</td>
<td>float</td>
<td>1</td>
<td>$9.0g</td>
<td></td>
<td>ID number</td>
</tr>
<tr>
<td>2</td>
<td>hosp</td>
<td>str3</td>
<td>0</td>
<td>$9s</td>
<td></td>
<td>Hospital</td>
</tr>
<tr>
<td>3</td>
<td>cob</td>
<td>float</td>
<td>1</td>
<td>$14.0g</td>
<td>coblab</td>
<td>Country of birth</td>
</tr>
<tr>
<td>4</td>
<td>gender</td>
<td>float</td>
<td>1</td>
<td>$9.0g</td>
<td>genderlab</td>
<td>Gender</td>
</tr>
<tr>
<td>5</td>
<td>qu1</td>
<td>float</td>
<td>1</td>
<td>$9.0g</td>
<td>noyes</td>
<td>Question 1</td>
</tr>
<tr>
<td>6</td>
<td>qu2</td>
<td>float</td>
<td>1</td>
<td>$9.0g</td>
<td>noyes</td>
<td>Question 2</td>
</tr>
<tr>
<td>7</td>
<td>qu3</td>
<td>float</td>
<td>1</td>
<td>$9.0g</td>
<td>noyes</td>
<td>Question 3</td>
</tr>
<tr>
<td>8</td>
<td>age</td>
<td>float</td>
<td>1</td>
<td>$9.0g</td>
<td></td>
<td>Age (yrs)</td>
</tr>
</tbody>
</table>
uselabel

Creates a dataset containing value-label information

<table>
<thead>
<tr>
<th>lname</th>
<th>value</th>
<th>label</th>
<th>trunc</th>
</tr>
</thead>
<tbody>
<tr>
<td>coblab</td>
<td>-1</td>
<td>Missing</td>
<td>0</td>
</tr>
<tr>
<td>coblab</td>
<td>1</td>
<td>Australia</td>
<td>0</td>
</tr>
<tr>
<td>coblab</td>
<td>2</td>
<td>United Kingdom</td>
<td>0</td>
</tr>
<tr>
<td>coblab</td>
<td>3</td>
<td>Vietnam</td>
<td>0</td>
</tr>
<tr>
<td>coblab</td>
<td>4</td>
<td>China</td>
<td>0</td>
</tr>
<tr>
<td>coblab</td>
<td>5</td>
<td>Singapore</td>
<td>0</td>
</tr>
<tr>
<td>coblab</td>
<td>6</td>
<td>New Zealand</td>
<td>0</td>
</tr>
<tr>
<td>genderlab</td>
<td>-1</td>
<td>Missing</td>
<td>0</td>
</tr>
<tr>
<td>genderlab</td>
<td>1</td>
<td>Male</td>
<td>0</td>
</tr>
<tr>
<td>genderlab</td>
<td>2</td>
<td>Female</td>
<td>0</td>
</tr>
<tr>
<td>genderlab</td>
<td>3</td>
<td>Intersex</td>
<td>0</td>
</tr>
<tr>
<td>noyes</td>
<td>0</td>
<td>No</td>
<td>0</td>
</tr>
<tr>
<td>noyes</td>
<td>1</td>
<td>Yes</td>
<td>0</td>
</tr>
</tbody>
</table>
Extracting value label names

```stata
gen recnum=_n
```

- `recnum` contains the number of the current observation

```stata
levelsof lname, local(levels) "coblab" "genderlab" "noyes"
```

- These are stored in the local macro `levels'
Creating the contents of each value label

```stata
foreach x of local levels {
    local fullab
    qui su recnum if lname=="`x'"
    local j=r(min)
    local k=r(max)
    forval i=`j'/`k' {
        local val=value[`i']
        local lab=label[`i']
        local fullab `fullab' `val', `lab' |
    }
    local lenlab=strlen("`fullab'")-2
    local fullab=substr("`fullab'",1,`lenlab')
}
```
foreach x of local levels {
    local fullab
    qui su recnum if lname=="`x'"
    local j=r(min)
    local k=r(max)
    forval i=`j'/`k' {
        local val=value[`i']
        local lab=label[`i']
        local fullab `fullab' `val', `lab' |
    }
    local lenlab=strlen("`fullab'")-2
    local fullab=substr("`fullab'",1,`lenlab')
}
Example with coblab

<table>
<thead>
<tr>
<th>lname</th>
<th>value</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>coblab</td>
<td>-1</td>
<td>Missing</td>
</tr>
<tr>
<td>coblab</td>
<td>1</td>
<td>Australia</td>
</tr>
<tr>
<td>coblab</td>
<td>2</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>coblab</td>
<td>3</td>
<td>Vietnam</td>
</tr>
<tr>
<td>coblab</td>
<td>4</td>
<td>China</td>
</tr>
<tr>
<td>coblab</td>
<td>5</td>
<td>Singapore</td>
</tr>
<tr>
<td>coblab</td>
<td>6</td>
<td>New Zealand</td>
</tr>
</tbody>
</table>

```
forval i=`j'/'k' {
    local val=value[`

local lab=label[`

local fullab `fullab' `val', `lab'

}
```

`i'=1

-1, Missing |
Example with coblab

<table>
<thead>
<tr>
<th>lname</th>
<th>value</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>coblab</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>coblab</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>coblab</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>coblab</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>coblab</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>coblab</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>coblab</td>
<td>6</td>
</tr>
</tbody>
</table>

```
forval i=`j'/`k' { 
    local val=value[`i']
    local lab=label[`i']
    local fullab `fullab' `val', `lab'
}
```

`i'=2

-1, Missing | 1, Australia |
Example with coblab

```
forval i=`j'/`k' {  
    local val=value[`i']  
    local lab=label[`i']  
    local fullab `fullab` `val', `lab'
}
```

```
i'=3
```

```
-1, Missing | 1, Australia | 2, United Kingdom |
```
Example with coblab

<table>
<thead>
<tr>
<th>lname</th>
<th>value</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>coblab</td>
<td>-1, Missing</td>
</tr>
<tr>
<td>2</td>
<td>coblab</td>
<td>1, Australia</td>
</tr>
<tr>
<td>3</td>
<td>coblab</td>
<td>2, United Kingdom</td>
</tr>
<tr>
<td>4</td>
<td>coblab</td>
<td>3, Vietnam</td>
</tr>
<tr>
<td>5</td>
<td>coblab</td>
<td>4, China</td>
</tr>
<tr>
<td>6</td>
<td>coblab</td>
<td>5, Singapore</td>
</tr>
<tr>
<td>7</td>
<td>coblab</td>
<td>6, New Zealand</td>
</tr>
</tbody>
</table>

```plaintext
forval i=`j'/'k' {
    local val=value[`i']
    local lab=label[`i']
    local fullab `fullab' `val', `lab'
}
```

`i'=4

-1, Missing | 1, Australia | 2, United Kingdom | 3, Vietnam |
### Example with `coblab`

<table>
<thead>
<tr>
<th><code>iname</code></th>
<th><code>value</code></th>
<th><code>label</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>coblab</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>coblab</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>coblab</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>coblab</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>coblab</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>coblab</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>coblab</td>
<td>6</td>
</tr>
</tbody>
</table>

```Stata
forval i=`j'/`k' { 
    local val=value[`i']
    local lab=label[`i']
    local fullab `fullab' `val', `lab'
}
```

`i'=5

-1, Missing | 1, Australia | 2, United Kingdom | 3, Vietnam | 4, China |
Example with coblab

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>coblab</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>coblab</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>coblab</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>coblab</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>coblab</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>coblab</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>coblab</td>
<td>6</td>
</tr>
</tbody>
</table>

```
forval i=`j'/'k' {
    local val=value[`i']
    local lab=label[`i']
    local fullab `fullab' `val', `lab'
}
```

`i'=6

-1, Missing | 1, Australia | 2, United Kingdom | 3, Vietnam | 4, China | 5, Singapore |
Example with coblab

```
forval i=`j'/`k' {
    local val=value[`i']
    local lab=label[`i']
    local fullab `fullab' `val', `lab'
}
```

```
<table>
<thead>
<tr>
<th>lname</th>
<th>value</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>coblab</td>
<td>-1</td>
<td>Missing</td>
</tr>
<tr>
<td>coblab</td>
<td>1</td>
<td>Australia</td>
</tr>
<tr>
<td>coblab</td>
<td>2</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>coblab</td>
<td>3</td>
<td>Vietnam</td>
</tr>
<tr>
<td>coblab</td>
<td>4</td>
<td>China</td>
</tr>
<tr>
<td>coblab</td>
<td>5</td>
<td>Singapore</td>
</tr>
<tr>
<td>coblab</td>
<td>6</td>
<td>New Zealand</td>
</tr>
</tbody>
</table>
```

`i'=7

-1, Missing | 1, Australia | 2, United Kingdom | 3, Vietnam | 4, China | 5, Singapore | 6, New Zealand |
Example with coblab

foreach x of local levels {
    ...
    forval i=`j'/`k' {
        local val=value[`i']
        local lab=label[`i']
        local fullab `fullab' `val', `lab'
    }
}
local lenlab=strlen("`fullab'")-2
local fullab=substr("`fullab'",1,`lenlab')

-1, Missing | 1, Australia | 2, United Kingdom | 3, Vietnam | 4, China | 5, Singapore | 6, New Zealand | -1, Missing | 1, Australia | 2, United Kingdom | 3, Vietnam | 4, China | 5, Singapore | 6, New Zealand
Allowing for extremely long strings

tempname mem

file write `mem' "`x'" _tab "`fullab'" _newline

- file allows for extremely long string values, up to 2-billion characters
- With postfile the limit is 2045 characters
One week after submitting my abstract for this meeting ...
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  
slrette2@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
Beaten to the punch

Alfred Russel Wallace

Seth Lirette et al
The redcapture command
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)
First, some background on REDCap.
REDCap field types

Field Type: Select a Type of Field

- Text Box (Short Text, Number, Date/Time, ...)
- Notes Box (Paragraph Text)
- Calculated Field
- Multiple Choice - Drop-down List (Single Answer)
- Multiple Choice - Radio Buttons (Single Answer)
- Checkboxes (Multiple Answers)
- Yes - No
- True - False
- Signature (draw signature with mouse or finger)
- File Upload (for users to upload files)
- Slider / Visual Analog Scale
- Descriptive Text (with optional Image/Video/Audio/File Attachment)
- Begin New Section (with optional text)
REDCap validations for text fields

Field Type: Text Box (Short Text, Number, Date/Time, ...)

Field Label

Variable Name (utilized during data export)
ONLY letters, numbers, and underscores

Validation? (optional)
- or -

Enable searching with:
-- choose ontology to use

Required? Yes No
* Prompt if field is blank

Identifier? Yes No
Does the field contain unique identifier

Custom Alignment
Align the position of the field

Field Note (optional)
Small reminder text displayed

Action Tags / Field Annotation (optional)
Learn about Action Tags or using Field Annotation

Date (D-M-Y)
Date (Y-M-D)
Datetime (D-M-Y H:M)
Datetime (Y-M-D H:M)
Datetime w/ seconds (D-M-Y H:M:S)
Datetime w/ seconds (Y-M-D H:M:S)
Email
ID Number: 2-digits, dash, 3-digits e.g. 01-234
Integer
Letters only
Name (Latin characters plus space, apostrophe, dash)
Number
Number (1 decimal place)
Number (2 decimal places)
Capturing categorical data in REDCap
## Example Stata dataset

<table>
<thead>
<tr>
<th>variable name</th>
<th>type</th>
<th>format</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>str9</td>
<td>%9s</td>
<td>Participant ID</td>
</tr>
<tr>
<td>consented</td>
<td>float</td>
<td>%9.0g</td>
<td>Is a consent document on file?</td>
</tr>
<tr>
<td>age</td>
<td>float</td>
<td>%9.0g</td>
<td>How old were you on your last birthday?</td>
</tr>
<tr>
<td>race</td>
<td>float</td>
<td>%16.0g</td>
<td>What is your race?</td>
</tr>
<tr>
<td>sex</td>
<td>float</td>
<td>%9.0g</td>
<td>What is your sex?</td>
</tr>
<tr>
<td>bdate</td>
<td>float</td>
<td>%td</td>
<td>What is your date of birth?</td>
</tr>
<tr>
<td>sbp</td>
<td>float</td>
<td>%9.0g</td>
<td>What was your last known systolic blood pressure?</td>
</tr>
<tr>
<td>dbp</td>
<td>float</td>
<td>%9.0g</td>
<td>What was your last known diastolic blood pressure?</td>
</tr>
<tr>
<td>happy1</td>
<td>float</td>
<td>%26.0g</td>
<td>The staff greeted me in a professional and courteous manner.</td>
</tr>
<tr>
<td>happy2</td>
<td>float</td>
<td>%26.0g</td>
<td>The waiting time to see a doctor was satisfactory.</td>
</tr>
<tr>
<td>happy3</td>
<td>float</td>
<td>%26.0g</td>
<td>I would return to this hospital.</td>
</tr>
<tr>
<td>comment</td>
<td>str4</td>
<td>%9s</td>
<td>Comments</td>
</tr>
</tbody>
</table>
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)

• Metadata are saved in example.csv. This is the data dictionary that will be uploaded to REDCap.
• The form/instrument name in REDCap is example_form
• Its header is "Example"
Example script

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)

For categorical variables. They must be numeric with value labels attached.
Example script

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)

• These are text fields
• All variables in the validate() option must be declared as text fields
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)

- id is a social security number
- bdate is a date field in YMD format
- dbp is an integer
- comment is a string
Example script

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)

• To omit range checks for any or all of the validation variables, "none" should be entered into the corresponding location
• These are soft checks
Example script

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)

• Radio fields with a common set of response options can be grouped in a matrix
• See next slide
## Matrix of fields

### Variable: happy1

<table>
<thead>
<tr>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neither Agree nor Disagree</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The staff greeted me in a professional and courteous manner.

### Variable: happy2

<table>
<thead>
<tr>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neither Agree nor Disagree</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The waiting time to see a doctor was satisfactory.

### Variable: happy3

<table>
<thead>
<tr>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neither Agree nor Disagree</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I would return to this hospital.
Data dictionary

The `redcapture` command created this data dictionary

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>VariableFieldName</td>
<td>FormName</td>
<td>SectionHeader</td>
<td>FieldType</td>
<td>FieldLabel</td>
<td>Choices</td>
</tr>
<tr>
<td>id</td>
<td>example_form</td>
<td>Example</td>
<td>text</td>
<td>Participant ID</td>
<td></td>
</tr>
<tr>
<td>consented</td>
<td>example_form</td>
<td>dropdown</td>
<td>text</td>
<td>Is a consent document on file?</td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>example_form</td>
<td>text</td>
<td>How old were you on your last birthday?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>race</td>
<td>example_form</td>
<td>dropdown</td>
<td>What is your race?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sex</td>
<td>example_form</td>
<td>text</td>
<td>What is your sex?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bdate</td>
<td>example_form</td>
<td>text</td>
<td>What is your date of birth?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>slbp</td>
<td>example_form</td>
<td>text</td>
<td>What was your last known systolic blood pressure?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbp</td>
<td>example_form</td>
<td>text</td>
<td>What was your last known diastolic blood pressure?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>happy1</td>
<td>example_form</td>
<td>radio</td>
<td>The staff greeted me in a professional and courteous manner.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>happy2</td>
<td>example_form</td>
<td>radio</td>
<td>The waiting time to see a doctor was satisfactory.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>happy3</td>
<td>example_form</td>
<td>radio</td>
<td>I would return to this hospital.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>example_form</td>
<td>text</td>
<td>Comments</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...which can be uploaded into REDCap
In conclusion ...

1. Ensure data will be retrievable 10 or 20 years from now
2. Ensure the next generation of researchers will be able to understand currently archived data

How?

By storing both data and metadata in text files

*Stata's* `export delimited` and `redcapture` commands facilitates this

Data and metadata can be uploaded to data capture software such as REDCap