Using dialog boxes in Stata to collect user parameters for use in a Stata user written command

Laura C Whiting

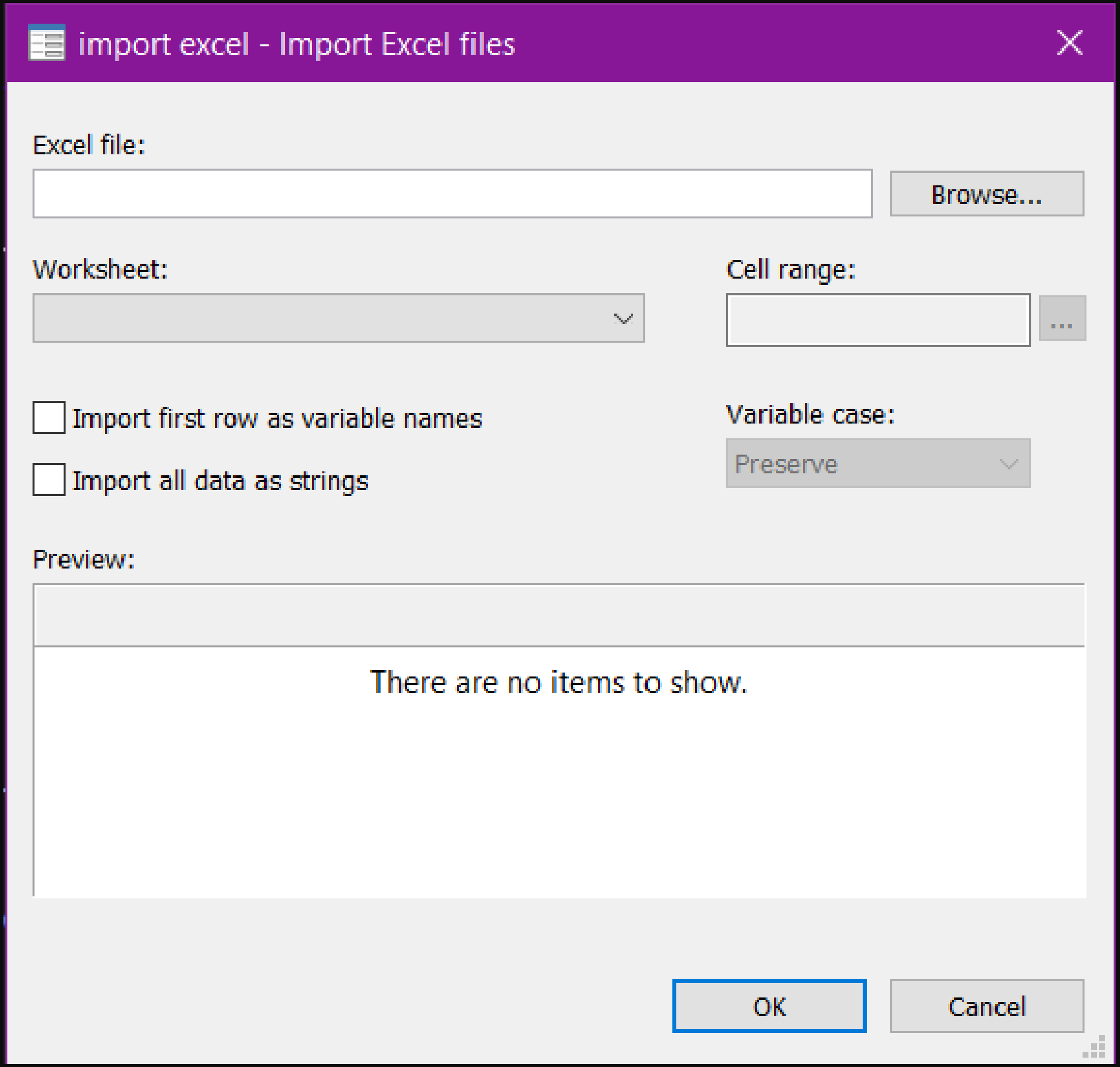
10 February 2022

Oceania Stata Conference

User written dialog boxes in Stata

- Overview of dialog boxes in user written commands
- Creating dialog boxes for your commands
  - Sizing
  - Inputs
  - Buttons and actions
  - Building the command
- Example of a dialog box

Dialog boxes in user written commands



import excel - Import Excel files

Excel file:

Worksheet:

Cell range:

□ Import first row as variable names

□ Import all data as strings

Variable case: Preserve

Preview:

There are no items to show.

Why Make a Dialog Box?

To share a command you built with others who do not understand Stata coding

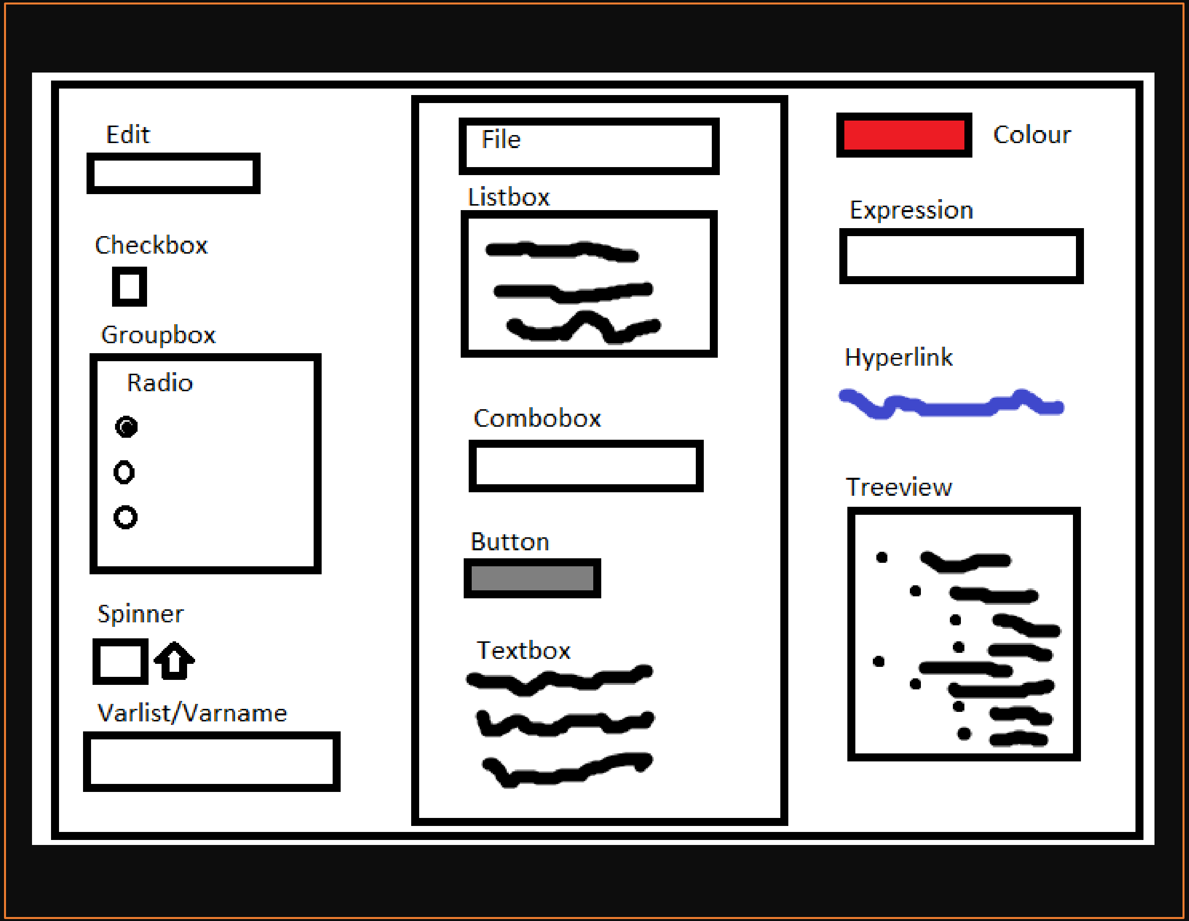
For ease of use for others – they don’t need to understand the command to use it

For faster implementation of the command (make selections and click a button rather than typing everything out)

Creating dialog boxes for your commands

Before You Start

- Make a list of inputs to collect
- Decide on input type
- Do you have related inputs
- Do you need multiple tabs
- Draw a diagram of the box
- Start coding – set the VERSIONVERSION



Setting the Box Size

Sizes given as pixels

Calculate from screen resolution

1920w X 1080h

25% of screen size

- 960w X 540h

50% of screen size

- 960w X 1080h
- 1920w X 540h

POSITION X Y X-size Y-size

A 25% box in the top-left corner:

POSITION 0 0 960 540

A 50% box on the right-side:

POSITION 960 0 960 1080

Define The First Dialog Page

- **DIALOG main, label("Example Dialog Box") title("Main")**
  - **BEGIN**
  - **…**
  - **END**

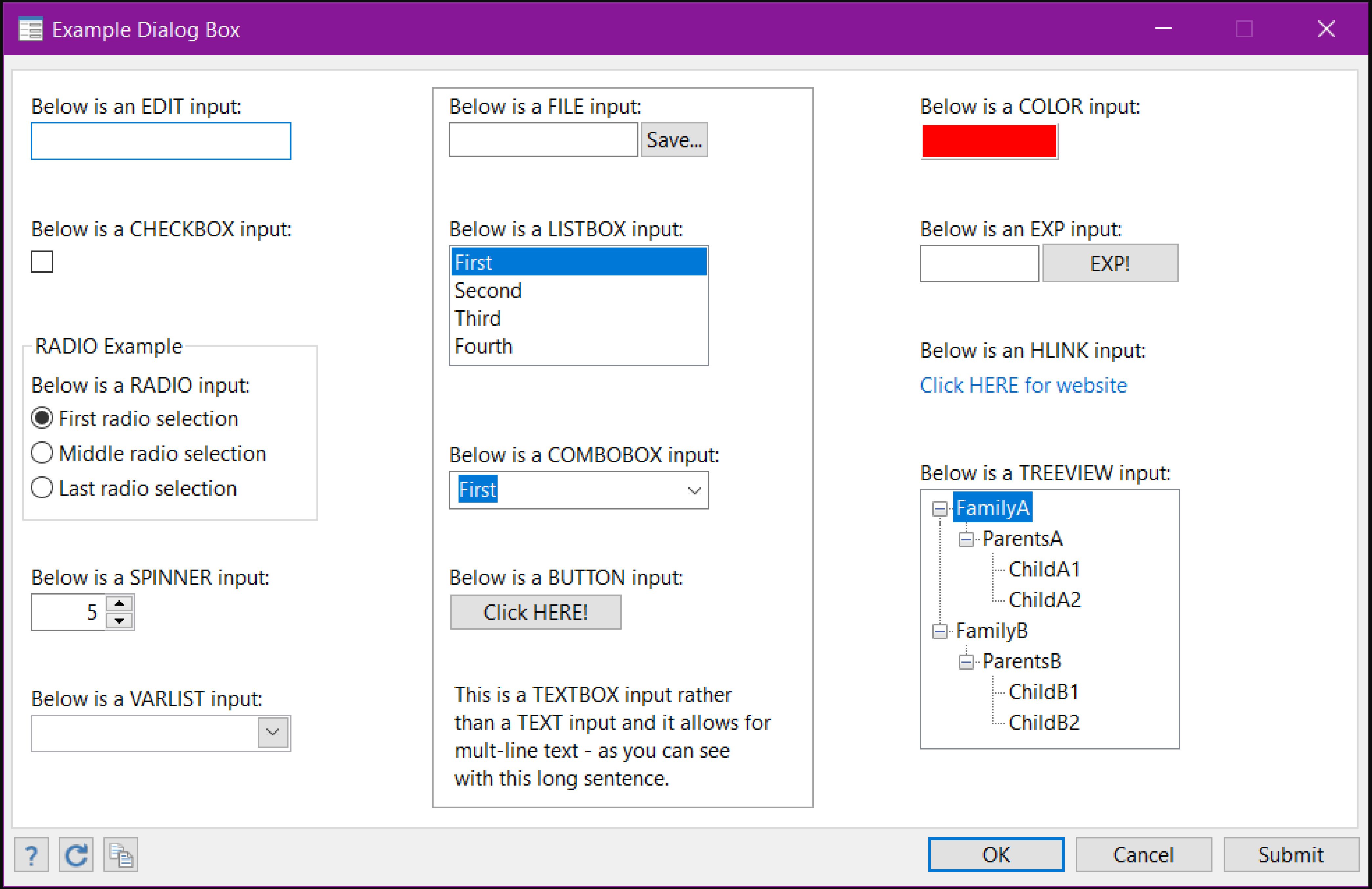
- A Dialog box can have multiple pages (tabs)
- All dialog boxes must have at least one tab
- The first tab is always called "Main"
- The option **label** will appear as the dialog box title
- The option **title** gives text shown on tab
- Once your tab is defined, you open it with **BEGIN** and close with **END**

Define Your Inputs

- An *input control* collects information
- There are several types of input controls:
  - Numeric
  - String
  - Categorical or Binary – mutually exclusive or multiple choices
  - Colour selection
  - Buttons
- There are *static controls* used to inform the user
  - Text is used to explain inputs
  - Borders are used to group inputs

Input Options

- All inputs have their own set of options, some are unique to a particular input
- Most inputs allow you to set a default value/setting
- You can set an error message for each input which will describe the input to the user when an error occurs
- Most inputs can be “remembered” or “forgotten” between instances of the dialog box



Below is an EDIT input:

Below is a CHECKBOX input:
□

RADIO Example

Below is a RADIO input:
● First radio selection
○ Middle radio selection
○ Last radio selection

Below is a SPINNER input:
5

Below is a VARLIST input:

Below is a FILE input:

Below is a LISTBOX input:
First
Second
Third
Fourth

Below is a COMBOBOX input:
First

Below is a BUTTON input:
Click HERE!

This is a TEXTBOX input rather than a TEXT input and it allows for mult-line text - as you can see with this long sentence.

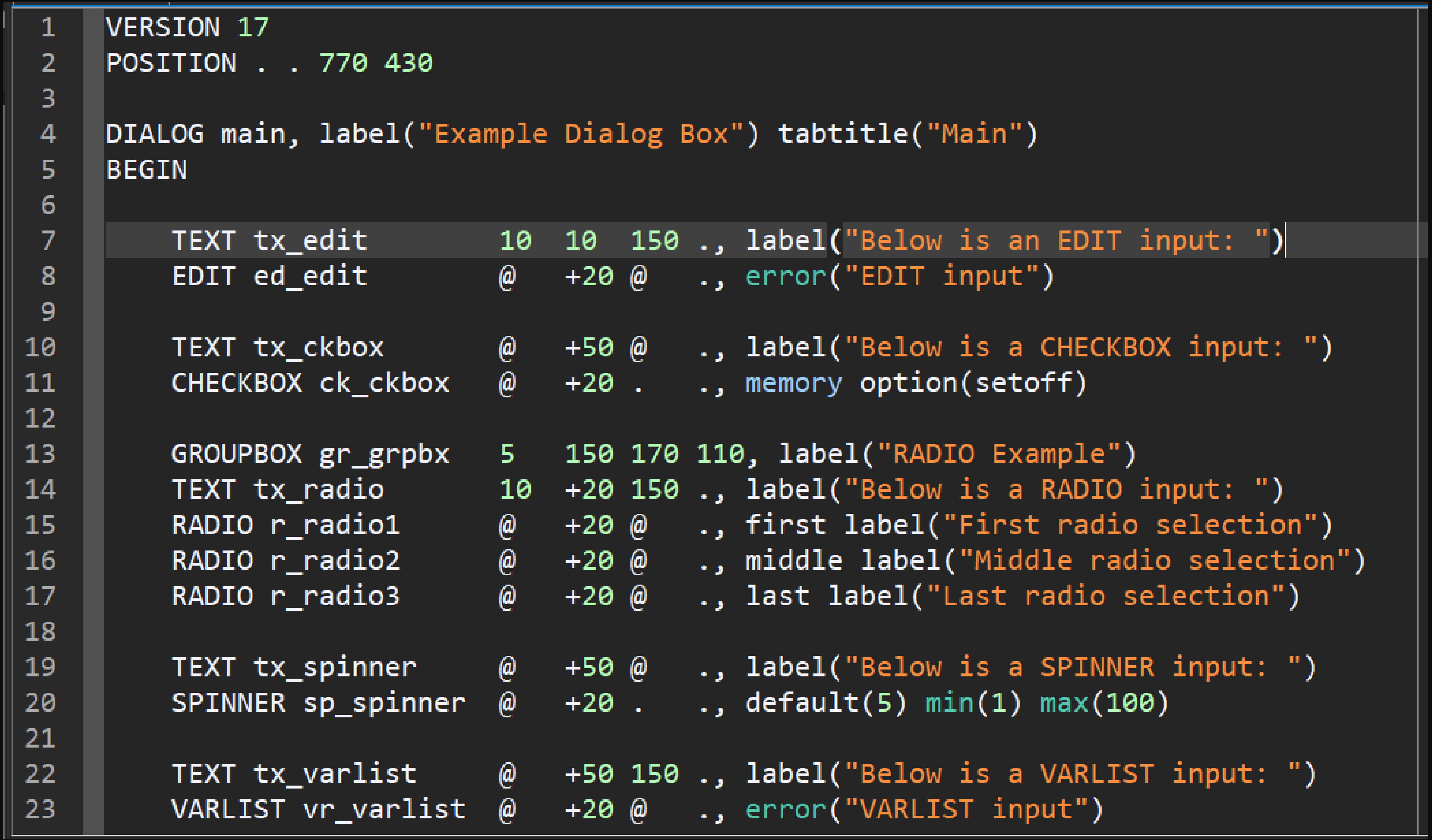
Below is a COLOR input:
🟥

Below is an EXP input:

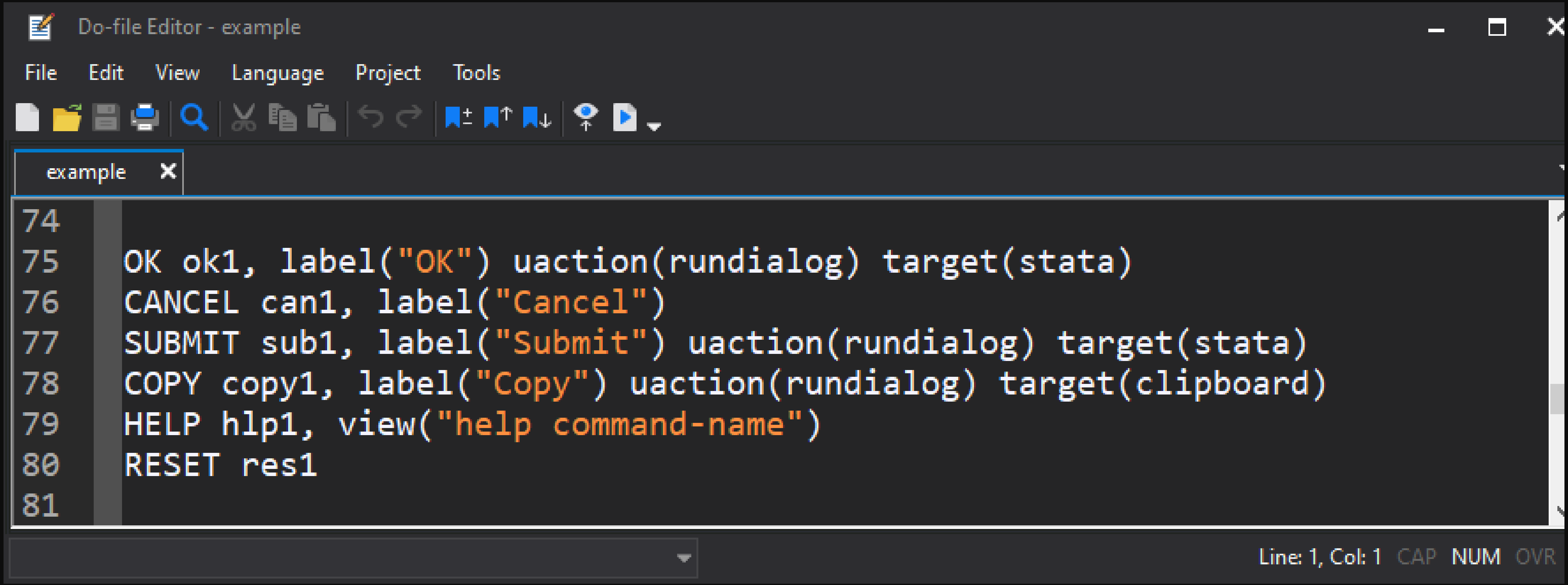
Below is an HLINK input:
Click HERE for website

Below is a TREEVIEW input:

- FamilyA
  - ParentsA
    - ChildA1
    - ChildA2
- FamilyB
  - ParentsB
    - ChildB1
    - ChildB2



```
VERSION 17
POSITION . . 770 430

DIALOG main, label("Example Dialog Box") tabtitle("Main")
BEGIN

    TEXT tx_edit        10  10  150 ., label("Below is an EDIT input: ")
    EDIT ed_edit        @   +20 @   ., error("EDIT input")

    TEXT tx_ckbox       @   +50 @   ., label("Below is a CHECKBOX input: ")
    CHECKBOX ck_ckbox   @   +20 .   ., memory option(setoff)

    GROUPBOX gr_grpbx   5   150 170 110, label("RADIO Example")
    TEXT tx_radio       10  +20 150 ., label("Below is a RADIO input: ")
    RADIO r_radio1      @   +20 @   ., first label("First radio selection")
    RADIO r_radio2      @   +20 @   ., middle label("Middle radio selection")
    RADIO r_radio3      @   +20 @   ., last label("Last radio selection")

    TEXT tx_spinner     @   +50 @   ., label("Below is a SPINNER input: ")
    SPINNER sp_spinner  @   +20 .   ., default(5) min(1) max(100)

    TEXT tx_varlist     @   +50 150 ., label("Below is a VARLIST input: ")
    VARLIST vr_varlist  @   +20 @   ., error("VARLIST input")
```



```

OK ok1, label("OK") uaction(rundialog) target(stata)
CANCEL can1, label("Cancel")
SUBMIT sub1, label("Submit") uaction(rundialog) target(stata)
COPY copy1, label("Copy") uaction(rundialog) target(clipboard)
HELP hlp1, view("help command-name")
RESET res1

```

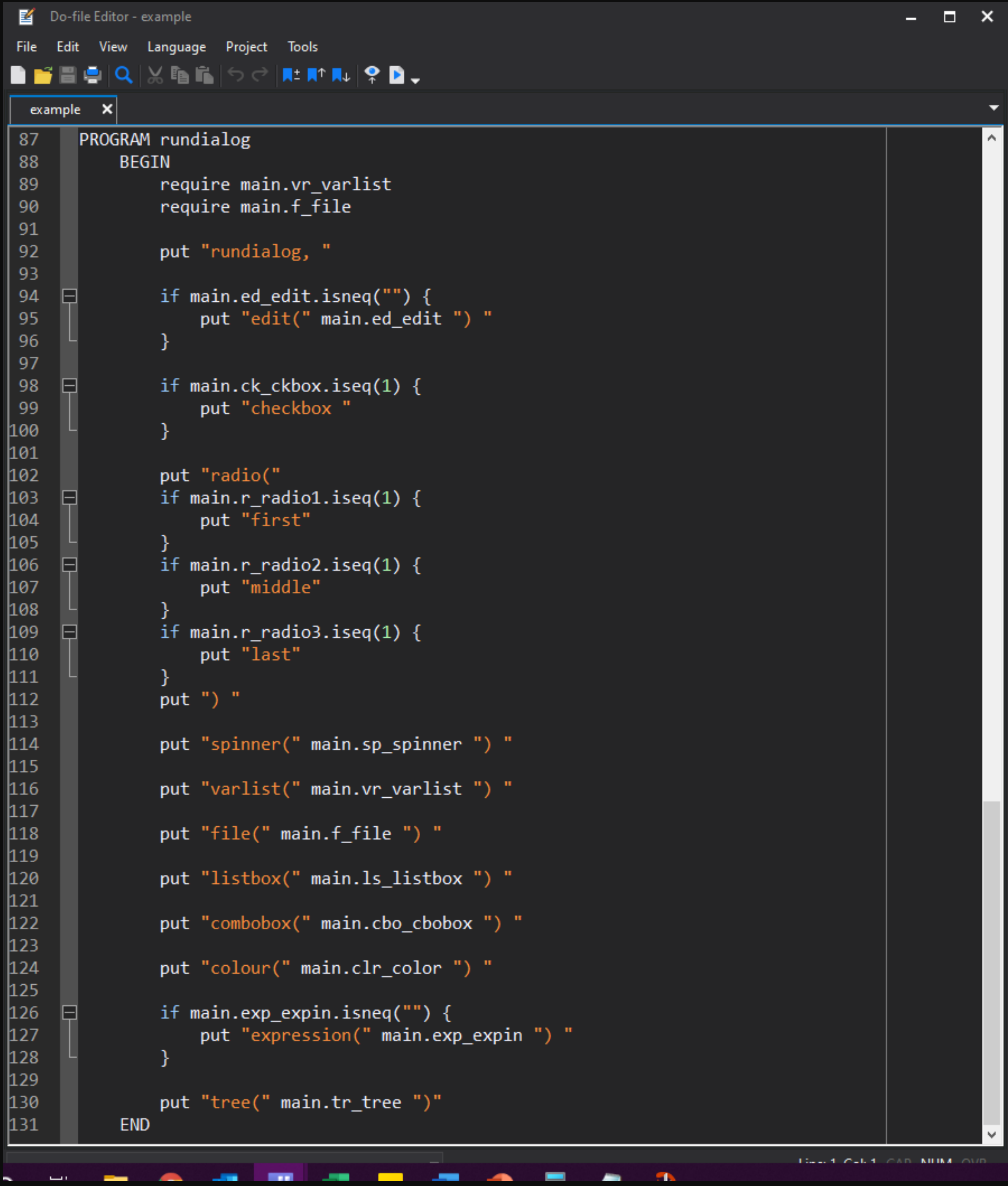
Setting the Standard Dialog Box Buttons

- There are 6 standard dialog buttons that must be enabled:
  - The three buttons on the bottom-left – HELP, RESET, COPY
  - The three buttons on the bottom-right – OK, CANCEL, SUBMIT
- These buttons will not appear if you do not define them
- **BUTTON name, label() uaction() target() view()**

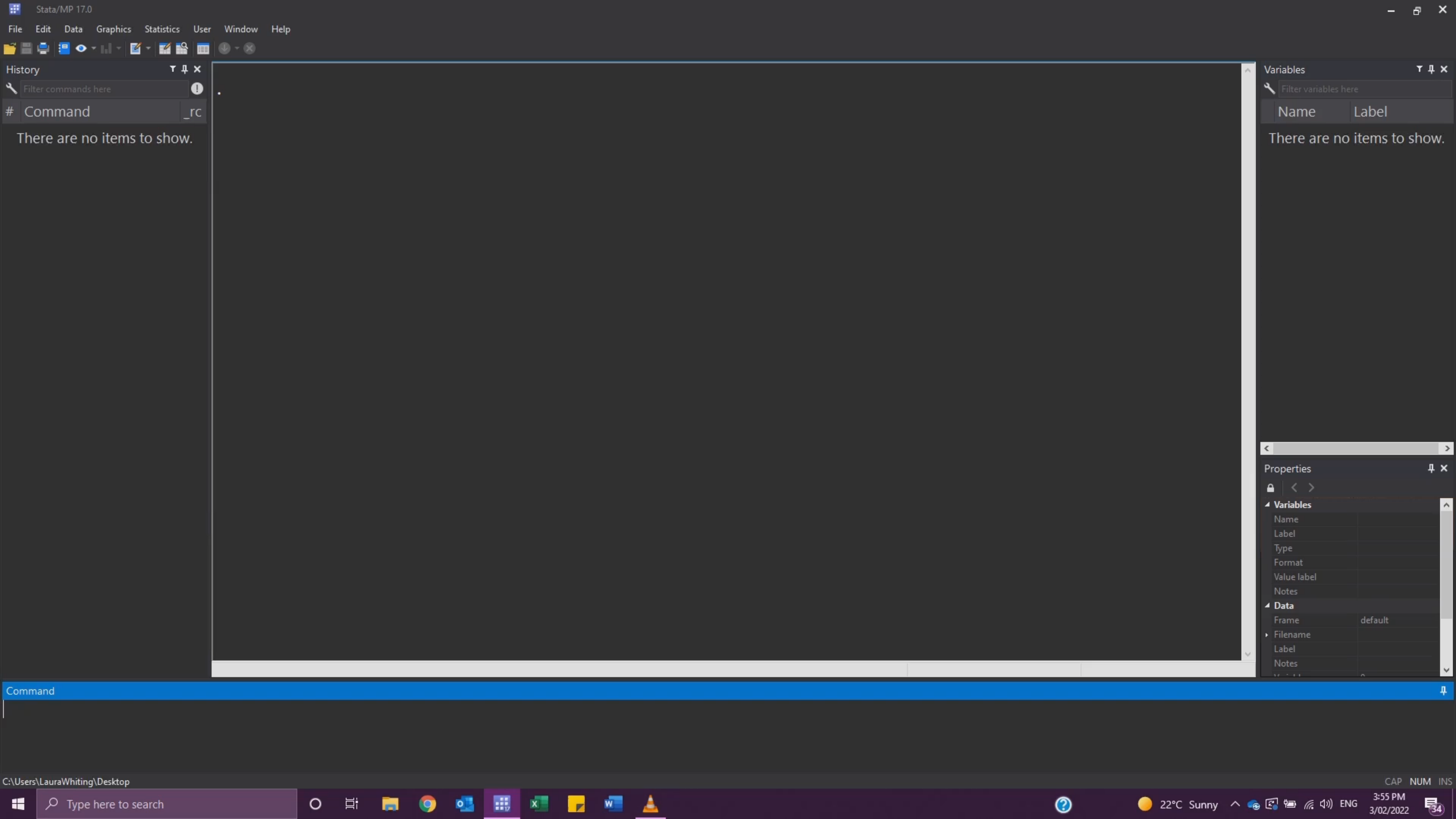
Creating the U-action Program

- Set to run when you click OK or Submit
- **PROGRAM programname**
    - **BEGIN**
        - **put “command text”**
        - **put tabname.inputname**
    - **END**

- Builds the command as text
- *put* adds to the command text



```
PROGRAM rundialog
    BEGIN
        require main.vr_varlist
        require main.f_file

        put "rundialog, "

        if main.ed_edit.isneq("") {
            put "edit(" main.ed_edit ") "
        }

        if main.ck_ckbox.iseq(1) {
            put "checkbox "
        }

        put "radio("
        if main.r_radio1.iseq(1) {
            put "first"
        }
        if main.r_radio2.iseq(1) {
            put "middle"
        }
        if main.r_radio3.iseq(1) {
            put "last"
        }
        put ") "

        put "spinner(" main.sp_spinner ") "

        put "varlist(" main.vr_varlist ") "

        put "file(" main.f_file ") "

        put "listbox(" main.ls_listbox ") "

        put "combobox(" main.cbo_cbobox ") "

        put "colour(" main.clr_color ") "

        if main.exp_expin.isneq("") {
            put "expression(" main.exp_expin ") "
        }

        put "tree(" main.tr_tree ")"
    END
```



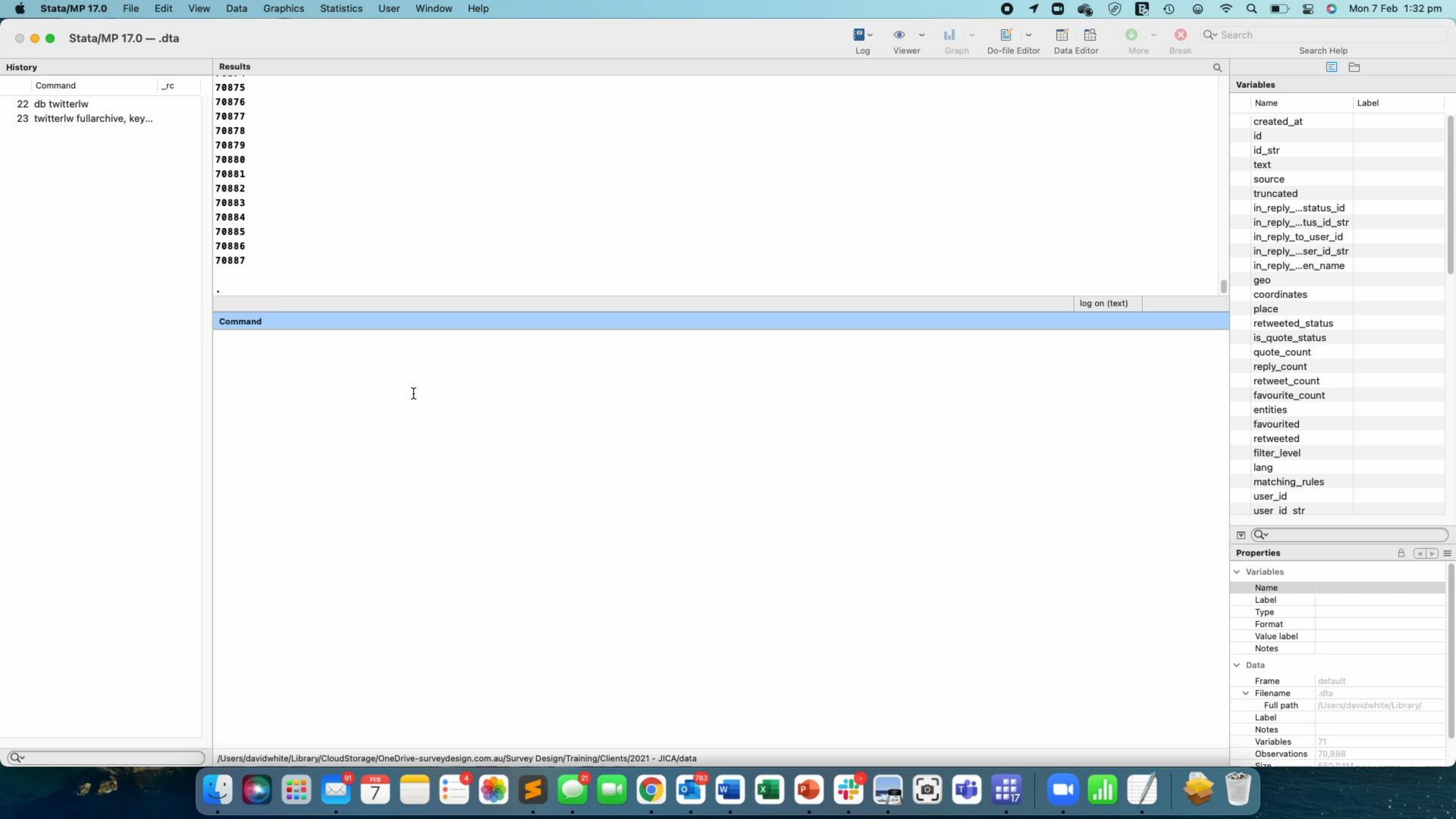
There are no items to show.

There are no items to show.

Frame default

C:\Users\LauraWhiting\Desktop

An Example Using Twitter Data



History

Command	_rc
22 db twitterlw	
23 twitterlw fullarchive, key...	

Results

```
70875
70876
70877
70878
70879
70880
70881
70882
70883
70884
70885
70886
70887

.
```

/Users/davidwhite/Library/CloudStorage/OneDrive-surveydesign.com.au/Survey Design/Training/Clients/2021 - JICA/data

Variables

Name	Label
created_at	
id	
id_str	
text	
source	
truncated	
in_reply_...status_id	
in_reply_...tus_id_str	
in_reply_to_user_id	
in_reply_...ser_id_str	
in_reply_...en_name	
geo	
coordinates	
place	
retweeted_status	
is_quote_status	
quote_count	
reply_count	
retweet_count	
favourite_count	
entities	
favourited	
retweeted	
filter_level	
lang	
matching_rules	
user_id	
user_id_str	

Properties

Variables

Name	
Label	
Type	
Format	
Value label	
Notes	

Data

Frame	default
Filename	.dta
Full path	/Users/davidwhite/Library/
Label	
Notes	
Variables	71
Observations	70,888

Questions