# 2   The Stata user interface

## The windows

This chapter introduces the core of Stata's interface: its main windows, its toolbar, its menus, and its dialogs.

Past commands appear here    Current dataset name appears here    Results are displayed here    Variable list appears here    Data properties appear here



Current working directory appears here

Commands are typed here

Current log status appears here

Command log status appears here

The five main windows are the History, Results, Command, Variables, and Properties windows. Except for the Results window, each window has its name in its title bar. The main interface features a title bar that shows the name of the current dataset and, if applicable, the current (working) frame when multiple frames are present. These five windows are typically in use the whole time Stata is open. There are other, more specialized windows such as the Viewer, Data Editor, Variables Manager, Do-file Editor, Graph, and Graph Editor windows—these are discussed later in this manual.

To open any window or to reveal a window hidden by other windows, select the window from the **Window** menu, or select the proper item from the toolbar. You can also use *Ctrl+Tab* to cycle through all open windows inside the main Stata window or *Alt+Tab* to cycle through all open windows (Stata and other) if you want to change windows from the keyboard. Many of Stata's windows have functionality that can be accessed by clicking on the right mouse button (right-clicking) within the window. Right-clicking displays a contextual menu that, depending on the window, allows you to copy text, set the preferences for the window, or print the contents of the window. When you copy text or print, we recommend that you always right-click on the window rather than use the menu bar or toolbar so that you can be sure of where and what you are copying or printing.

1

# The toolbar

This is the toolbar:



The toolbar contains buttons that provide quick access to Stata's more commonly used features. If you forget what a button does, hold the mouse pointer over the button for a moment, and a tooltip will appear with a description of that button.

Buttons that include both an icon and an arrow display a menu if you click on the arrow. Here is an overview of the toolbar buttons and their functions:

**Open** opens a Stata dataset. Click on the button to open a dataset with the *Open* dialog.

**Save** saves the Stata dataset currently in memory to disk.

**Print** displays a list of windows. Select a window name to print its contents.

**Log** begins a new log or closes, suspends, or resumes the current log. See [GSW] **16 Saving and printing results by using logs** for an explanation of log files.

**Viewer** opens the Viewer or brings a Viewer to the front of all other windows. Click on the button to open a new Viewer. Click on the arrow to select a Viewer to bring to the front. See [GSW] **3 Using the Viewer** for more information.

**Graph** brings a Graph window to the front of all other windows. Click on the button to bring the Graph window to the front. Click on the arrow to select a Graph window to bring to the front. See *The Graph button* in [GSW] **14 Graphing data** for more information.

**Do-file Editor** opens the Do-file Editor or brings a Do-file Editor to the front of all other windows. Click on the button to open a new Do-file Editor. Click on the arrow to select a Do-file Editor to bring to the front. See [GSW] **13 Using the Do-file Editor—automating Stata** for more information.

**Data Editor (Edit)** opens the Data Editor or brings the Data Editor to the front of the other Stata windows. See [GSW] **6 Using the Data Editor** for more information.

**Data Editor (Browse)** opens the Data Editor in browse mode. See *Browse mode* in [GSW] **6 Using the Data Editor** for more information.

**Variables Manager** opens the Variables Manager. See [GSW] **7 Using the Variables Manager** for more information.

**Show more results** tells Stata to continue when it has paused in the middle of long output. Click on the arrow to choose whether to run the command to completion. See [GSW] **B.8 More** for more information.

**Break** stops the current task in Stata. See [GSW] **10 Listing data and basic command syntax** for more information.

# The Command window

Commands are submitted to Stata from the Command window. The Command window supports basic text editing, copying and pasting, a command history, function-key mapping, filename completion, and variable-name completion.

From the Command window, pressing

| | |
|---|---|
| *Page Up* | steps backward through the command history. |
| *Page Down* | steps forward through the command history. |
| *Tab* | autocompletes a partially typed variable name when possible or presents a list of similar names if there could be more than one completion. Further typing will narrow the list. As soon as the name is complete, the full name will be inserted. If the name starts with a double quote, *Tab* attempts to autocomplete a filename in the same manner. |

See [U] **10 Keyboard use** for more information about keyboard shortcuts for the Command window.

The command history allows you to recall a previously submitted command, edit it if you wish, and then resubmit it. Commands submitted by Stata's dialogs are also included in the command history, so you can recall and submit a command without having to open the dialog again.

# The Results window

The Results window contains all the commands and their textual results you have entered during the Stata session.

You can scroll through the Results window to look at work you have done, it is much simpler to search within the Results window by using the find bar. By default, the find bar is hidden. You can expose it by selecting **Edit > Find...**.

You can clear out the Results window at any time by right-clicking in the Results window and selecting **Clear results** from the contextual menu. This action cannot be undone.

You can also scroll through the Results window when the keyboard focus is in the Command window. Press the *Shift* key and the up arrow or down arrow key to scroll the Results window a line at a time or the *Shift* key and the *Page Up* or *Page Down* key to scroll the Results window a page at a time.

# The Variables window

The Variables window shows the list of variables in the dataset, along with the properties of the variables. By default, it shows all the variables and their variable labels. You can change what properties get displayed by right-clicking on the header of any column of the Variables window.

Click once on a variable in the Variables window to select it. Multiple variables can be selected in the usual fashion, either by *Ctrl*-clicking on nonadjacent variables or by clicking on a variable and *Shift*-clicking on a second variable to select all intervening variables.

Double-clicking on a variable in the Variables window puts the selected variable at the insertion point in the Command window.

The leftmost column of the Variables window is called the one-click paste column. You can also send variables to the Command window by hovering the mouse over the one-click paste column of the Variables window and clicking on the arrow that appears. The one-click paste column can be shown or hidden in the same fashion as the other columns in the Variables window.

The Variables window supports filtering and changing the display order of the variables. Text entered in the *Filter variables here* field will filter the variables appearing in the Variables window. The filter is applied to all visible columns and shows all variables that match the criteria in at least one column. By default, the filter will ignore case and show any variables for which at least one column contains any of the words in the filter. Clicking on the wrench on the left will allow you to change this behavior as well as add or remove additional columns containing information about the variables.

You can change the display order of the variables in the Variables window by clicking on any column header. The first click sorts in ascending order, the second click sorts in descending order, and the third click puts the variables back in dataset order. Thus clicking on the *Name* column header will make the Variables window display the variables in alphabetical order. Sorting in the Variables window is live, so if you change a property of a variable when the Variables window is sorted by that property, it will automatically move the variable to its proper location. Reordering the display order of the variables in the Variables window does not affect the order of the variables in the dataset itself.

Right-clicking on a variable in the Variables window displays a menu from which you can select

- **Keep only selected variables** to keep just the selected variables in the dataset in memory. You will be asked for confirmation. This affects only the dataset in memory, not the dataset as saved on your disk. See [GSW] **12 Deleting variables and observations** for more information.

- **Drop selected variables** to drop, or eliminate, the selected variables from the dataset in memory. You will be asked for confirmation. Just as above, this affects only the dataset in memory, not the dataset as saved on your disk. See [GSW] **12 Deleting variables and observations** for more information.

- **Copy varlist** to copy the selected variable names to the clipboard.

- **Select all** to select all variables in the dataset that satisfy the filter conditions. If no filter has been specified, all variables will be selected.

- **Send varlist to Command window** to send all selected variables to the Command window.

- **Font...** to bring up a *Font* dialog, allowing you to change the font used to display the Variables window contents.

Items from the contextual menu issue standard Stata commands, so working by right-clicking is just like working directly in the Command window.

If you would like to hide the Variables window, click on its close button.

To reveal a hidden Variables window, select **Window > Variables**.

## The Properties window

The Properties window displays variable and dataset properties. If a single variable is selected in the Variables window, its properties are displayed. If there are multiple variables selected in the Variables window, the Properties window will display properties that are common across all selected variables.

Clicking the lock icon in the Properties window title bar toggles the ability to alter properties of the selected variables. By default, changes are not allowed. Once the properties are unlocked, you can make any changes to variable or dataset properties you like. Each change you make will create a command that appears in the Results and Command windows, as well as in any command log, so the changes are reproducible. Using the Properties window is one of the simplest ways of managing notes, changing variable and value labels, and changing display formats. See [D] **notes**, [D] **label**, and [D] **format**.

Clicking the arrow buttons next to the lock icon will select the previous or next variable shown in the Variables window, and that selection will be reflected in the Properties window. If you would like to hide the Properties window, click on its close box. If you would like to reveal a hidden Properties window, select **Window > Properties**.

You should also investigate the Variables Manager, explained in [GSW] **7 Using the Variables Manager**, because it extends these capabilities and provides a good interface for managing variables.

## The History window

The History window shows the history of commands that have been entered, with unsuccessful commands and their error codes in red, by default.

The toolbar has two tools for manipulating the contents of the History window. Clicking on the **Filter** button, ⏷ , in the History window title bar toggles the visibility of these tools. Text entered in the *Filter commands here* field will filter the commands appearing in the History window. By default, the filter ignores case and finds any commands containing any of the words in the filter. Clicking on the wrench on the left allows you to change this behavior. You can hide the commands that produced an error by clicking on the **Filter errors** button, ⬤ .

To enter a command from the History window, you can
- Click once on a past command to copy it to the Command window, replacing the contents of the Command window.
- Double-click on a past command to resubmit it. Executing the command adds the command to the bottom of the History window.

Right-clicking on the History window displays a menu from which you can select various actions:
- **Cut** removes the selected commands from the History window and places them on the Clipboard.
- **Copy** copies the selected commands to the Clipboard.
- **Delete** removes the selected commands from the History window.
- **Select all** selects all the commands in the History window, including those before and after the commands currently displayed.
- **Clear all** clears out all the commands from the History window, including those before and after the commands currently displayed.
- **Do selected** submits all the selected commands and adds them to the bottom of the command history. Stata will attempt to run all the selected commands, even those containing errors, and will not stop even if a command causes an error.
- **Send selected to Do-file Editor** places all the selected commands into a new Do-file Editor window.
- **Save all...** brings up a *Save review contents* dialog, which allows you to save all the commands in the History window, including those before and after the commands currently displayed, in a *do-file*. (See [GSW] **13 Using the Do-file Editor—automating Stata** for more information on do-files.)
- **Save selected...** brings up a *Save review contents* dialog, which allows you to save the selected commands in the History window in a do-file.
- **Font...** brings up a *Font* dialog, allowing you to change the font used to display the History window contents.

## Menus and dialogs

There are two ways by which you can tell Stata what you would like it to do: you can use menus and dialogs, or you can use the Command window. When you worked through the sample session in [GSW] **1 Introducing Stata—sample session**, you saw that both ways have strengths. We will discuss the menus and dialogs here.

Stata's **Data**, **Graphics**, and **Statistics** menus provide point-and-click access to almost every command in Stata. As you will learn, Stata is fully programmable, and Stata users can even create their own dialogs and menus. The **User** menu provides a place for programmers to add their own menu items. Initially, it contains only some empty submenus. As an example, suppose you wish to perform a Poisson regression. You could type Stata's `poisson` command, or you could select **Statistics > Count outcomes > Poisson regression**, which would display this dialog:
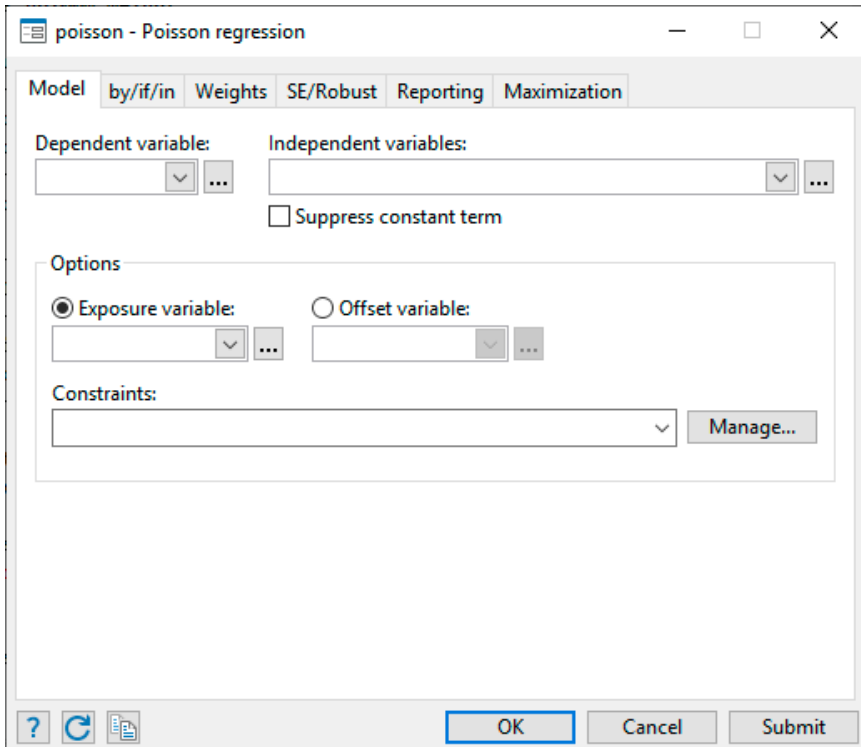


This dialog provides access to all the functionality of Stata's `poisson` command. Because the dependent and independent variables must be numeric, you will find that the combo box will display only numeric variables for choosing. The `poisson` command has many options that can be accessed by clicking on the multiple tabs across the top of the dialog. The first time you use the dialog for a command, it is a good idea to look at the contents of each tab so that you will know all the dialog's capabilities.

The dialogs for many commands have the **by/if/in** and **Weights** tabs. These provide access to Stata's commands and qualifiers for controlling the estimation sample and dealing with weighted data. See [U] **11 Language syntax** for more information on these features of Stata's language.

The dialogs for most estimation commands have the **Maximization** tab for setting the maximization options (see [R] **Maximize**). For example, you can specify the maximum number of iterations for the optimizer.
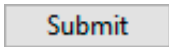
Most dialogs in Stata provide the same six buttons you see at the bottom of the *poisson* dialog above.
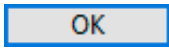
**OK** issues a Stata command based on how you have filled out the fields in the dialog and then closes the dialog.
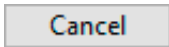
**Cancel** closes the dialog without doing anything—just as clicking on the dialog's close button does.

**Submit** issues a command just like **OK** but leaves the dialog on the screen so that you can make changes and issue another command. This feature is handy when, for example, you are learning a new command or putting together a complicated graph.

**Help** provides access to Stata's help system. Clicking on this button will typically take you to the help file for the Stata command associated with the dialog. Clicking on it here would take you to the poisson help file. The help file will have tabs above groups of options to show which dialog tab contains which options.

**Reset** resets the dialog to its default state. Each time you open a dialog, it will remember how you last filled it out. If you wish to reset its fields to their default values at any time, simply click on this button.

**Copy command to Clipboard** behaves much like the **Submit** button, but rather than issuing a command, it copies the command to the Clipboard. The command can then be pasted elsewhere (such as in the Do-file Editor).

The command issued by a dialog is submitted just as if you had typed it by hand. You can see the command in the Results window and in the History window after it executes. Looking carefully at the full command will help you learn Stata's command syntax.

In addition to being able to access the dialogs for Stata commands through Stata's menus, you can also invoke them by using two other methods. You may know the name of a Stata command for which you want to see a dialog, but you might not remember how to navigate to that command in the menu system. Simply type db *commandname* to launch the dialog for *commandname*:

```
. db poisson
```

You will also find access to the dialog for a command in that command's help file; see [GSW] **4 Getting help** for more details.

As you read this manual, we will present examples of Stata commands. You may type those examples as presented, but you should also experiment with submitting those commands by using their dialogs. Use the db command described above to quickly launch the dialog for any command that you see in this manual.

# The working directory

If you look at the screenshot on page 1, you will notice the status bar at the base of the main Stata window that contains the name of the current working directory you set when you installed Stata. We will use the directory C:\Users\mydir\Documents in our examples; your exact directory will differ. The current working directory is the folder where graphs and datasets will be saved when typing commands such as save *filename*. It does not affect the behavior of menu-driven file actions such as **File > Save** or **File > Open...**. Once you have started Stata, you can change the current working directory with the cd command. See [D] **cd** for full details. Stata always displays the name of the current working directory so that it is easy to tell where your graphs and datasets will be saved.

# Fine control of Stata's windows

When Stata is first launched, the History, Results, Command, Variables, and Properties windows appear within the main Stata window. The History, Variables, Properties, and Command windows are initially linked to one another and attached to the edges of the Stata window. They can be resized by dragging on their edges. They even have two more features: you can drag them by their title bars to reposition them within the main window, and you can hide them as tabs when not in use. To hide them as tabs, click on the pushpin icons in their title bars.

The Results window always occupies whatever space remains inside the main Stata window.

All other windows that open up can be moved freely and independently of the Stata window.

This default behavior is simple and familiar and needs little explanation. Try playing with the windows inside the main Stata window now. If you find that the window behavior is good, you can stop reading this section. If you find that you would like some extra customization or that you would simply like some technical explanation of added window behaviors, keep reading.

# Window types

The Stata for Windows interface has two types of windows: *docking* and *nondocking*. The History, Command, Variables, and Properties windows are docking windows. All other windows are nondocking.

Docking windows have special characteristics that allow them to be used with other docking windows: they can be *linked* (sharing a window with a splitter dividing the windows) or *tabbed* (sharing a window with a tab for each docking window). They can be set to automatically hide when not needed. Nondocking windows have none of these special characteristics.

Here is a brief comparison of the two types of windows:

Docking windows
- can be linked to other docking windows so that they share a window with a splitter dividing them;
- can share one window with other docking windows with a tab for each; and
- can be made to hide automatically when not in use.

Nondocking windows
- are separate from the main Stata window (except for the Results window);
- cannot be linked to other windows;
- cannot be tabbed to another window; and
- cannot be made to hide automatically when not in use.

When Stata is first launched, the History, Results, Command, Variables, and Properties windows appear within the main Stata window. The History, Command, Variables, and Properties windows are docking windows and are initially both docked and linked to one another. The Results window is a nondocking window.

## Docking windows

Docking windows may be moved within the main Stata window by dragging on their title bars. When you drag a docking window over the main Stata window or another docking window, docking guides appear (see figure 1). Dragging the title bar over a docking guide gives you a preview of how and where the window will appear if you release the mouse button.

Dropping the box on one of the outer docking guides links the docking window to the window under the guide. For example, figure 1 shows how you would drag the History window over the Variables window, at which point the docking guides appear. When you move the mouse over the bottom docking guide and release the mouse button, the History window becomes linked to the Variables window and is displayed in the lower part of the window.

When docking windows are linked, a divider (splitter) is inserted between them. The splitter allows the linked windows to be resized; increasing the size of one window decreases the size of the other.
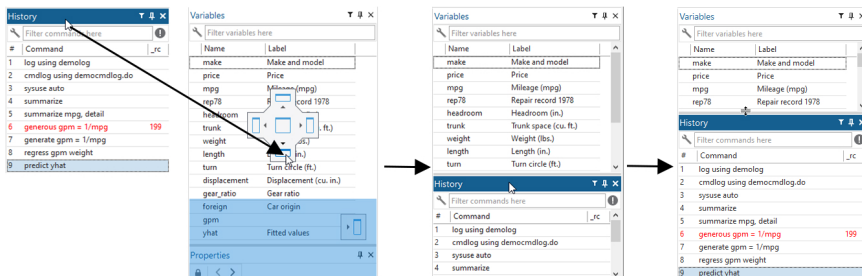


Figure 1: Linking two windows

Dropping the box on the center docking guide (see figure 2) combines the docking window with the window under the guide. One docking window is created with a tab for each docked window at the bottom of the new window. Clicking on a tab displays its window. To undock a tabbed window, click and drag the window's tab out of the docking window to a docking guide. The docking window containing the tabbed windows can also be linked to other docking windows.
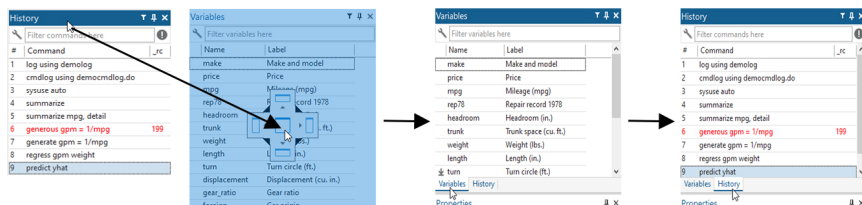


Figure 2: Docking two windows

## Auto Hide and pinning

Docking windows support the Auto Hide feature, which automatically hides the docking windows when they are not in use. Pushpins appear in the title bar of all docked windows in the main Stata window.

To enable Auto Hide for a docking window, click on the pushpin in the title bar (see figure 3) so that the pushpin is horizontal. Click on the pushpin again to disable Auto Hide; the pushpin will be vertical.

After you enable Auto Hide for a docking window, the window is hidden when not in use and is displayed as a tab along an edge of the main Stata window. To show the window, move the mouse pointer over the tab. To hide the window again, move the mouse pointer outside the window. If the window has the focus (its title bar is highlighted), it will be hidden if you click on another window.
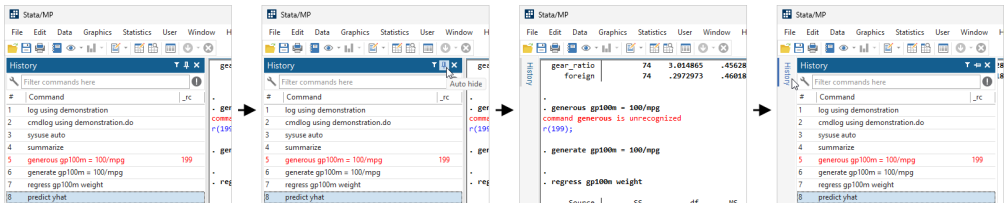


Figure 3: Enabling Auto Hide

If you enable Auto Hide for the History window in its default docked location, it will obscure any text at the beginning of the Command window when it is displayed. To avoid this problem, dock this window at the right edge of the main Stata window before enabling Auto Hide.

## Nondocking windows

Nondocking windows include the Graph window, Data Editor, Do-file Editor, Viewers, and dialogs. These windows are always independent of the main Stata window. The Results window, because of its status as the primary window of Stata, must remain inside the main Stata window.

For suggested citations, see the FAQ on citing Stata documentation.