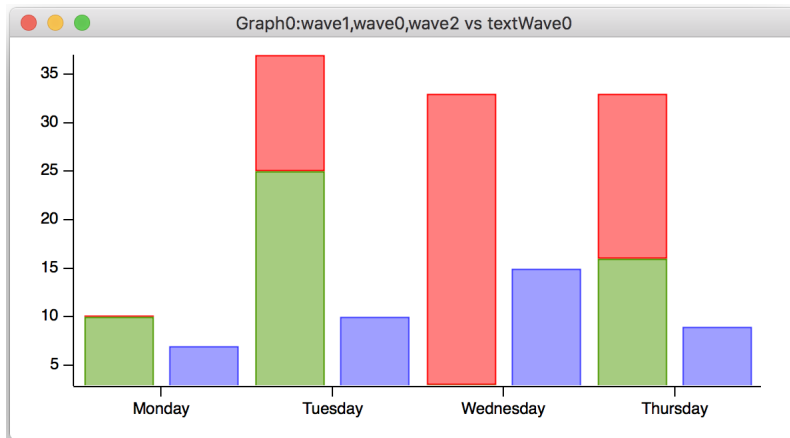


15. Click the Tick In Center checkbox and then click Do It.

Notice the new positions of the tick marks.



16. Double-click the bottom axis again.
17. Click the Axis tab.
18. Change the value of Bar Gap to zero and then click Do It.  
Notice that the bars within a group are now touching.
19. Use the Modify Axis dialog to set the Category Gap to 50%.  
The widths of the bars shrink to 50% of the category width.
20. Choose Graph→Modify Graph.
21. Click the “Swap X & Y Axes” checkbox and then click Do It.  
This is how you create a horizontal bar plot.
22. Choose File→Save Experiment.

## The Command Window

Parts of this tour make use of Igor’s command line to enter mathematical formulae. Let’s get some practice now.

1. Choose File→New Experiment.

This clears any windows and data left over from previous experimentation.

Your command window should look something like this:



If you don’t see the command window, choose Windows→Command Window.

The command line is the space below the red separator whereas the space above the separator is called the history area.

2. Click in the command line, type the following line and press Return or Enter.

```
Print 2+2
```

The Print command as well as the result are placed in the history area.

3. Press the Up Arrow key.

The line containing the print command is selected, skipping over the result printout line.

**4. Press Return or Enter.**

The selected line in the history is copied to the command line.

**5. Edit the command line so it matches the following and press Return or Enter.**

```
Print "The result is ", 2+2
```

The Print command takes a list of numeric or string expressions, evaluates them and prints the results into the history.

**6. Choose the Help→Igor Help Browser menu item.**

The Igor Help Browser appears.

You can also display the help browser by clicking the question-mark icon near the right edge of the command window and, on Windows, by pressing F1.

**7. Click the Command Help tab in the Igor Help Browser.**

**8. Click Advanced Filtering if necessary to reveal the advanced options.**

**9. If there is a Show All link above the left-hand list, click it.**

**10. Uncheck the Functions and Programming checkboxes and check the Operations checkbox.**

A list of operations appears.

**11. In the pop-up menu next to the Operations checkbox, choose About Waves.**

**12. Select PlaySound in the list.**

**Tip:** Click in the list to activate it and then type “p” to jump to PlaySound.

The reference help for the PlaySound operation appears in the help area on the right.

**13. Click the help area on the right, scroll down to the Examples section, and select the first four lines of example text (starting with “Make”, ending with “PlaySound sineSound”).**

**14. Choose the Edit→Copy menu to copy the selection.**

**15. Close the Igor Help Browser.**

**16. Choose Edit→Paste to paste the command into the command line.**

All four lines are pasted into the command line area.

**17. Make the command window taller and then drag the red divider line up so you can see the commands in the command line.**

**18. Press Return or Enter to execute the commands.**

The four lines are executed and a short tone plays.

**19. Click once on the last line in the history area, on “PlaySound sineSound”.**

The entire command is selected just as if you pressed the arrow key.

**20. Press Return or Enter once to transfer the command to the command line and a second time to execute it.**

The tone plays again as the line executes.

We are finished with the “sineSound” wave that was created in this exercise so let’s kill the wave to prevent it from cluttering up our wave lists.

**21. Choose Data→Kill Waves.**

The Kill Waves dialog appears.

**22. Select “sineSound” and click Do It.**

The sineSound wave is removed from memory.

**23. Again click once on the history line “PlaySound sineSound”.**

**24. Press Return or Enter twice to re-execute the command.**

An error dialog is presented because the sineSound wave no longer exists.

**25. Click OK to close the error dialog.**