CS410P

ASSIGNMENT 7P

Purpose

This assignment is to work with a few different type of plots as well as learn how to customize them. There are two separate programs needed for submission.

The first program should be submitted in a file called **waves.py**, and the second one should be submitted as **rgb.py**

To be able to see the figure window separately, to pop it out of Spyder go to Tools/Preferences/Ipython Console/Graphics tab – Backend dropdown should be set to Automatic. You will then be able to see what you need to adjust in terms of font sizes so that text content doesn't overlap and the plot is not skewed.

Program 1

Let us draw some basic line plots to depict sine, cosine and tangent waves.

A **sine wave** is a geometric waveform that oscillates (moves up, down or side-to-side) periodically, and is defined by the function y = sin x. In other words, it is an s-shaped, smooth **wave** that oscillates above and below zero.

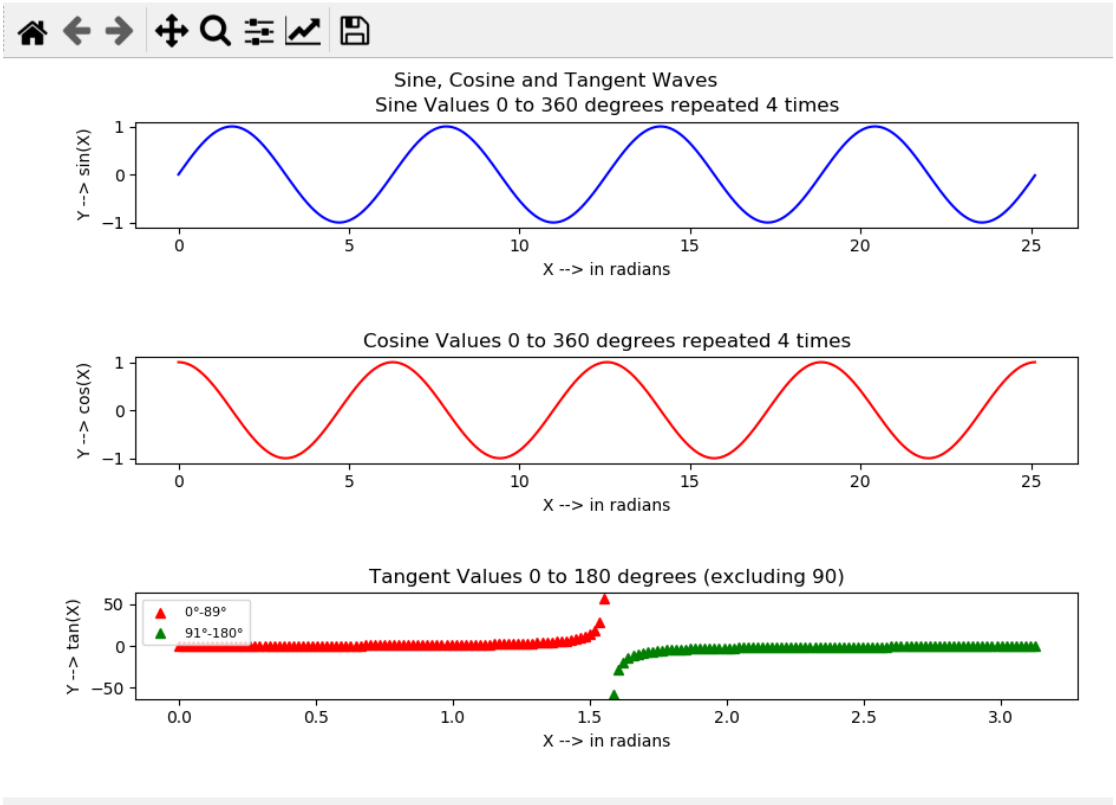
A **cosine wave** is a signal waveform with a shape identical to that of a sine **wave** , except each point on the **cosine wave** occurs exactly 1/4 cycle earlier than the corresponding point on the sine **wave**.

The **tangent** will be zero wherever its numerator (the sine) is zero. The tangent will be *undefined* wherever its denominator (the cosine) is zero. A zero in the denominator means you'll have a vertical asymptote. The tangent will have vertical asymptotes wherever the cosine is zero.

You must use numpy arrays to generate the x and y values for each plot/wave. You cannot use any loops to generate the data points. To make the plots more interesting you will repeat values on the x-axis such that the sine and cosine plots will go from 0 to 360 degrees 4 times. The tangent plot can go from 0 to 180 degrees on the x-axis with the exception of plotting at 90 deg since it is undefined/infinity. The tangent plot must actually be two plots one from 0 to 89 and another from 91 to 180 in two different colors.

Additionally you must divide the figure window into 3 subplots (3 rows) with each plot appearing in a sub plot. You must customize it with axes labels, subplot and figure titles, and a legend for the tangent plot since we want to see two plots for it on the same subplot area.

Sample output is shown below:



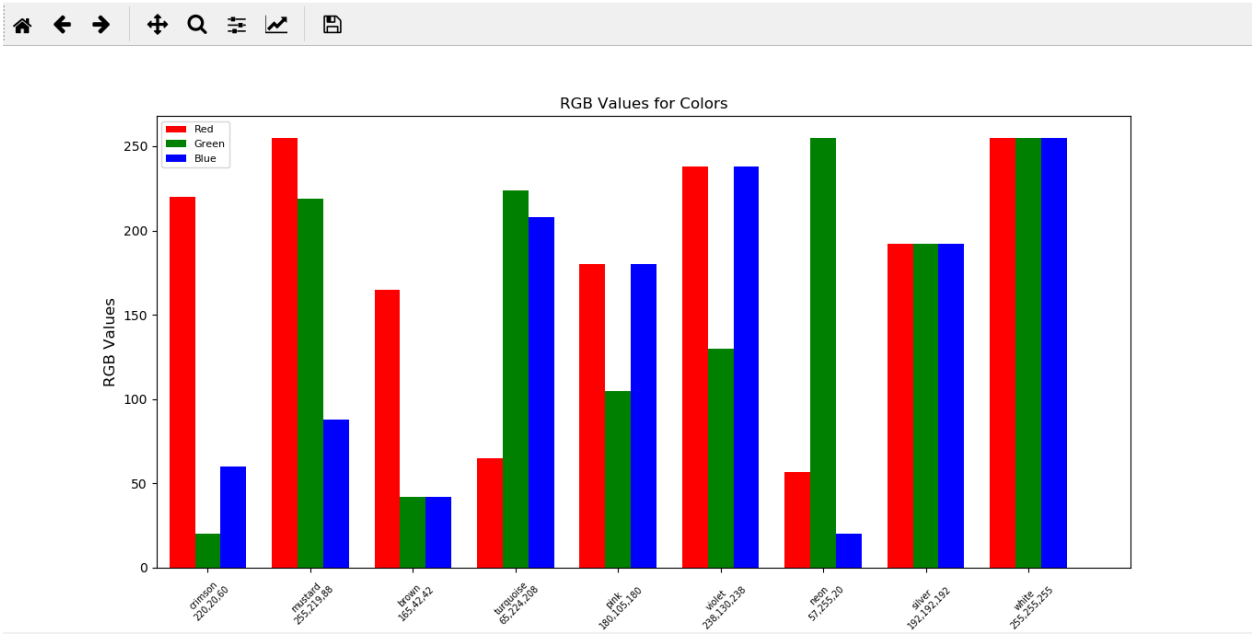
Program 2

A bar plot (or bar chart) is one of the most common type of graphic. It shows the relationship between a set of data items. Let's take the example of working with color schemes - A color's **RGB value** indicates its red, green, and blue intensity. Each intensity value is on a scale of 0 to 255. So each color has 3 intensity components to it.
Examples: Red 255, 0, 0 (Red intensity is 255 other two are 0), Crimson is 220, 20, 60 (Red is 220, Green is 20, Blue is 60).

Some colors have some of the intensity values as 0. We want to pick colors that have a non-zero value for all 3 intensities and plot them using a bar plot to show what their intensity values are. You are free to use the same colors that I am using (shown in the output sample below) or pick other colors but we want to see colors with all 3 intensity values that are non-zero. Choose at least 9 different colors.

Customize your plot as much as you can, the examples in the lecture notes are sufficient to do what I've shown you for the output. (Intensity values under each color name in the horizontal axis is not mandatory, but at least the color name is!). Keep in mind that each "red" bar is an item in an array (or list) that has all the red intensity colors (and similarly you will have array/list objects for green and blue intensities). So each corresponding item in the 3 arrays/lists will come together graphically as one color value of RGB intensities.

You can hard-code your arrays/lists with the color intensities, no need to seek any input from the user. Sample output is shown below.



Grade Key for waves.py

A	Figure window divided correctly 3 subplots, one graph per subplot area (5 each)	**15**
B	Sine wave accurate, repeats at least 4 times (variation is ok)	**5**
C	Cosine wave accurate, repeats at least 4 times (variation is ok)	**5**
D	Tan plot/wave accurate, shows at least 0 to 89, 91 to 180 each drawn once (3 points each)	**6**
E	Each subplot has axes labels and individual titles (2.5 each), plot title (2)	**9**
F	Tan plot has legend	**5**
G	No overlapping text, plots/text looks clear when enlarged	**5**
H	Late penalty	

Grade Key for rgb.py

A	9 colors chosen with each of them having non-zero intensity values for Red, Green and Blue	**5**
B	3 bars correctly plot per color, all 9 or as many colors that the array/list has appear in the plot	**20**
C	Vertical axes labeled (2), Plot title shown (2)	**4**
D	Legend shown correctly	**6**
E	Horizontal axis labeled with each color under the set of bars	**10**
F	No overlapping text, chart looks clear when enlarged	**5**
G	Late penalty	