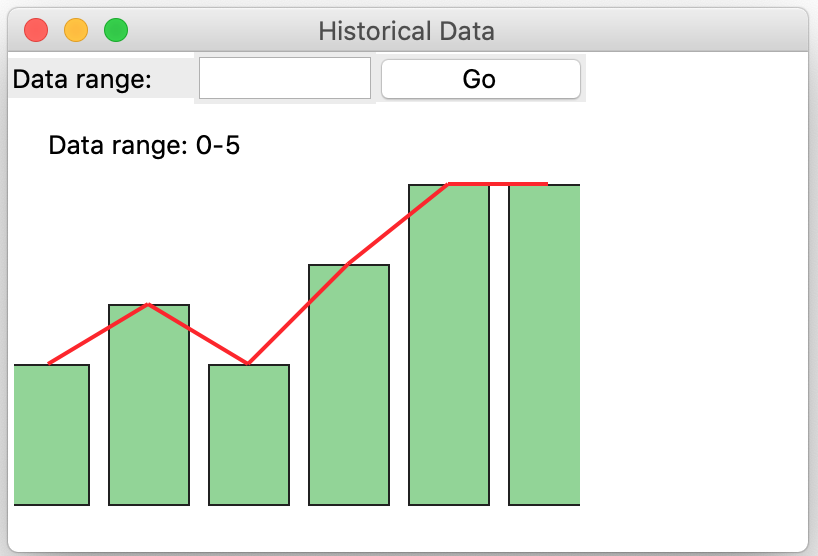
Networking for Software Developers

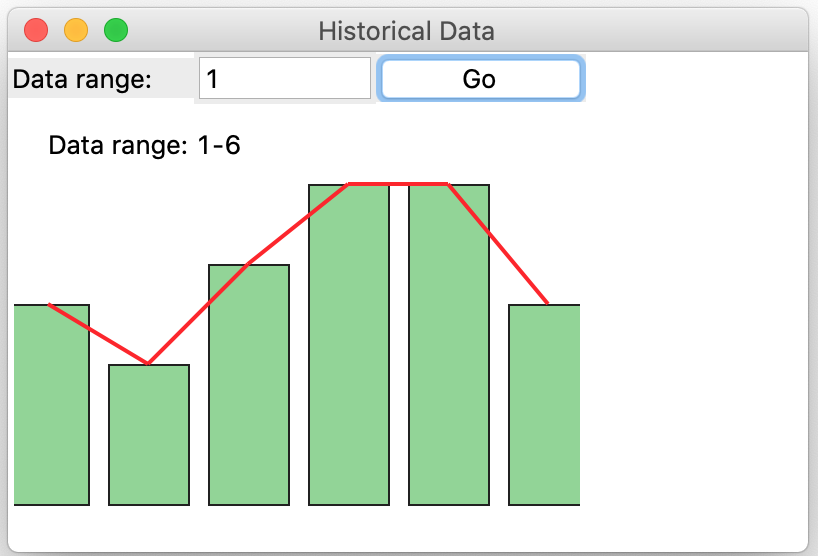
This is a group lab that will contribute towards your final project. One person will submit this work to the group dropbox.

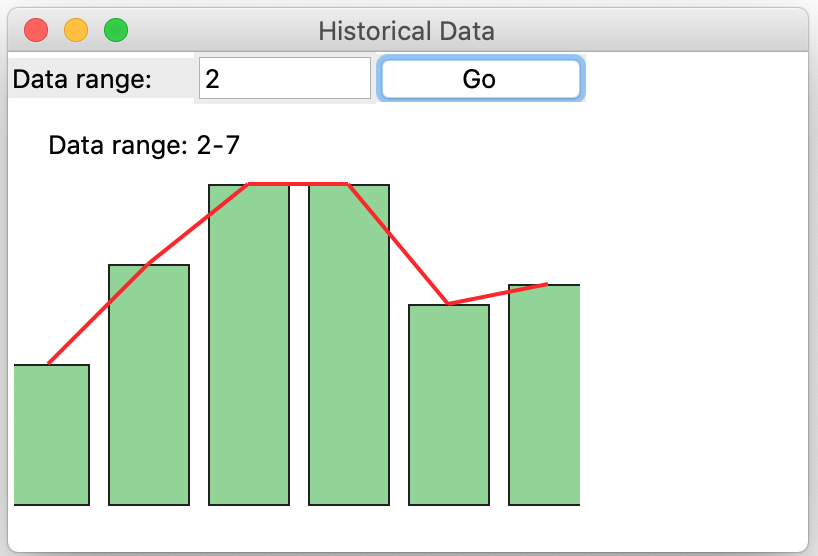
# Lab 11 – Display chart.

We will build a GUI to display a set of values. You will display the values using both a line chart and a bar chart on the same app. Most of the coding will come from Week\_09\_lab\_09\_Display.docx. You will make the following modification to your solution of lab09 or lab10:

1. In the constructor create a list of 20 values from your generator class in Lab 8. This will not change for the life of the application. (You must get these values explicitly form your previous code)
2. In the initUI() method do the following:
   * Add the code that will create the three widgets at the top of the window as shown in the screenshot below.
   * Call the method below that will draw the rectangles and lines as shown in the screenshot.
   * Wire-up the button to read the input from the textbox and call the method in step 3 with the appropriate arguments.
3. Define a method that takes the start values (the list of values is available as a class attribute) and will draw the six rectangles and the lines.







## Due:

Before the start of week 10

## Requirements:

1. You will use the same quantity that you selected in the previous lab (from temperature, humidity, barometric pressure, customers at a mall, or just with an alternate descriptor).
2. Design and build a GUI application class that will model a display for your sensor reasonably well.
3. You must provide an Entry (Textbox) and a button to read the value and call the method to draw the rectangles and lines.
4. There are marks for aesthetics.

See the appendix of the previous week lab for some code sample and possible directions to explore.

#### Submission

1. Your code file will be named group\_«your\_group\_number»\_display\_chart.py e.g. group\_1\_display\_chart.py.
2. You must also submit the data generator class that you built in your previous assessment. (You should not copy the statements into your current work)
3. Must be uploaded to course dropbox.
4. See course shell for due date.

#### Rubrics

|  |
| --- |
| [Class] 4/4 |
| [unitUI Method] 3/3 |
| [Button Handler] 2/2 |
| [Draw rectangle] 6/6 |
| [Gui] 2/2 |
| [Aesthetics] 3/3 |
| Total 20/20 |

#### Appendix

Because you will be drawing multiple rectangles and lines on your canvas, you will have to “clear the output”. There are two suggested techniques:

1. Remove the canvas from your tkinter app. So, to draw you will have to create a new canvas and re-draw your rectangles etc. This is not a bad option, but if you must redraw axis and other embellishments, it might be too expensive.   
   Code to remove a canvas widget:

self.canvas = Canvas(«host\_container») # reference to the canvas

«host\_container».delete(self.canvas)

1. A slightly more efficient approach is to remove all the items in the canvas.  
   Code to remove the items:

self.canvas = Canvas(«host\_container») # reference to the canvas

self.canvas.delete('all') # clears the canvas

1. Another option is to remove just the temporary items. This is more efficient but slightly little more complicated.  
   Code to remove the items:

self.items = [] # this will store all temp items

self.canvas = Canvas(«host\_container») # reference to the canvas

self.items.append(self.canvas.create\_text( # add a temp item

«left», «top»), text='some text'))

self.items.append(self.canvas.create\_line( # add another temp item

«left», «top», «right», «bottom»))

for item in self.items:

self.canvas.delete(item) # remove each item

self.items = [] # re-initialize the temp storage