MA1008 Introduction to Computation Thinking
Take Home Mini Project – Weeks 10 - 13

Project Objectives

This project requires you to use the graphics module in Python called Turtle to plot a bar chart. You have to compose the chart from basic graphic elements like points, lines and texts using functions in Turtle.

You can jump to the section titled "Step 4 Plot the bar chart of a dataset" for the project requirements.

Preamble

This project description is somewhat long. That's because we are giving you a step-by-step guide on learning how to execute it – four major steps. You need to go through the computational thinking behind the problem, break it down into its basic components, and compose the steps for the solution. It requires you to use some tools (the relevant turtle functions) that are new to you. The four steps more or less map nicely to the four weeks you have for the project.

Bar chart is something familiar to us all, and so it is very well understood and should not present you with much of a conceptual problem, if at all. There are different kinds of charts. To keep it straightforward, we will confine ourselves to vertical bar charts of the type shown in the examples in Figure 1. You should be able to come up with other examples.

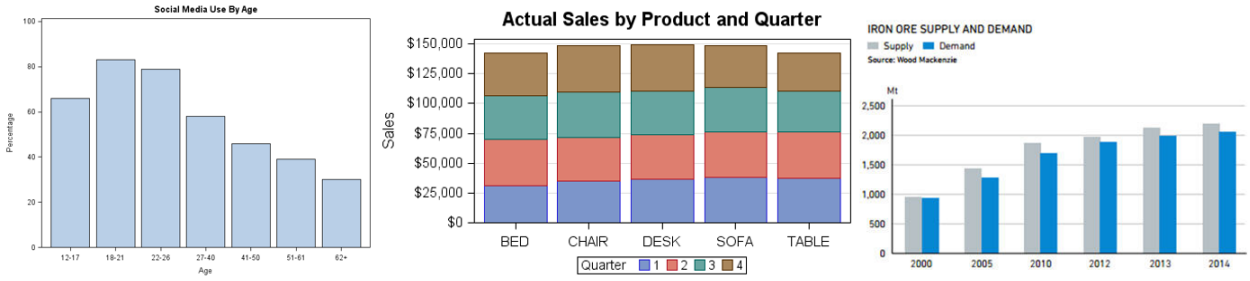

Figure 1: Different line graphs

Even within this limited type of chart, there are many variations – bar size, colour, text size and position, etc. – that allow you to plot the chart of your choice, as you can see from the three examples. Such variations can be easily controlled in software, and it is your task in this project to produce a program to plot such charts, and control their appearance.

You are not restricted on how your chart should look, except that it should display the information you need to present clearly and neatly. This will allow you to exploit the flexibility of software to create interesting results.

Turtle

A graph is essentially a drawing. So you will need to be able to write a program to do drawing. Drawing capabilities in software are provided by a graphics library. Turtle is a graphics library provided within the Python environment. It contains many functions you need in creating a drawing. But there is no need for you to learn everything in the library; only those functions you need for plotting graphs.

The Turtle reference manual can be found in the Python documentation, which you can find by pressing the F1 key in the Python Shell, and then search for "turtle (module)". That should bring you to this

section: “24.1 turtle – Turtle graphics”. There is no need for you to read it from start to finish. Read the introduction to learn what it is about, and then scan the list of functions provided. The names of the functions should give you the indication of what each does. Then you decide which functions you need for your purpose, and then go and read the details of those functions.

There are some demonstration programs available for Turtle. Select “Help” in the menu list at the top of the Python shell, and you will see at the bottom of the pop-up menu “Turtle demo”. Run some of these demos to see the interesting things that it can do. The source programs of the demos are visible as well, so you can get an appreciation of what the programs look like. Don’t worry if you cannot comprehend them. That will take some work.

The project

What you need to produce and submit at the end of this project is a program that plots a bar chart for a given dataset. You are free to design the program in the way you want, so long as it is neat and clear. We will describe the problem later. Here, we first give you a progressive guide on how to do drawings.

You have four weeks for the project. The guide below breaks your progress down to four steps, and you can take one step per week, more or less. If you follow them well, you should be able to produce the desired results at the end. However, we expect that some of you will be able to do it ahead of time.

Step 1 – learn to use Turtle functions

Learn how to use Turtle functions to do some simple drawings. Try these:

- Draw a line, using the direction functions like forward(), backward(), right(), left() etc., with appropriate parameters, of course.
- Draw a line, by specifying coordinates to draw to, using the goto() function, with appropriate control of penup() and pendown().
- Vary the widths of the lines you draw by using pensize().
- Draw multiple lines in different directions, connected and not connected. Try drawing lines to write the word “turtle” or any word you choose.
- Draw a few closed shapes: triangles, rectangles and shapes of your own design, of different sizes, and fill them with colours, especially rectangles, of course, since the bars are rectangular.
- Write some text in the picture.
- Try different colours, for the lines, the texts and the filling of the shapes.

You will see that the default setting of Turtle is to display a triangular arrow head, which represents the turtle, pointing to a specific direction that you can change. Also the display is done in slow animation. Learn to change these defaults, as you may want a different behavior when you plot your graph.

Please refer to the “Turtle star” program given at the top of Section 24.1 for a short example of what can be done.

Step 2 – learn to draw proper diagrams

A proper diagram is one that has controlled appearances, such as sizes, shapes and colours. Do these exercises:

- Create a display window of specific size using the setup() function
- In the window, plot the axes of a graph which should be perpendicular to each other. You should be able to control the extent of the axes in both directions.
- Label the axes with appropriate text like “distance”, “time”, etc. depending on the need

- Place tick marks sensibly at regular intervals along the axes
- Label the tick marks with appropriate values
- Learn to place rectangular bars of given sizes where you want them.

Some of the exercises will require you to design simple algorithms to achieve the outcome, such as placing the tick marks at appropriate locations and label them with appropriate values. You decide what "appropriate" is.

Step 3 –Display things in real coordinates

We have not yet mentioned the coordinate system in which you do your drawing. It is likely that you have simply done the drawing so far using the native coordinate system of turtle, which is the pixel coordinates of the screen. That's because you have not specified a coordinate system.

In real life, things come in different shapes and sizes, and they all have to be drawn on a screen that has a fixed maximum size, based on the number of pixels in the x and y directions. That is the "screen coordinates" or "device coordinates". Real things do not conform to the screen coordinates, but have sizes and positions of their own. A thing 10m long can be displayed on a screen, another thing 0.1m long can be displayed on the same screen, but yet looks equally big. Clearly they are displayed at different scales. Hence you need to learn to display things at the appropriate scale so that they appear comfortable to the eyes when displayed.

You should create algorithms/functions to do the following:

- Create a display window of some specific size, as you did last week.
- Decide on the dimension of the object you wish to display, and create a function that would scale it to allow it to be displayed appropriately on the screen
- Generate an object of your choice (such as a rectangle) with its own dimensions, and display it appropriately on the screen

Step 4 – Plot the bar chart of a dataset

By now you should have gathered the tools you need to draw real bar charts. You are required to write a program to display the dataset given in the text file Heights.txt. The file contains one column of numbers, which are the measurements of the heights of people in meters. You are required to plot a bar chart showing the distribution of these heights, by dividing the range of the measurements into 10 segments, from the low to the high, count the number of measurements in each segment and plot the number against the segments as a bar chart.

Computational thinking:

You need to separate the organization of the data from the plotting of the bar chart. The chart is only a display of the data in the way you need to visualize it. So you first need to organize the data, then display it in a bar chart. The description above suggests to you the steps.

1. Find the range of the heights.
2. Divide this range into ten segments, each segment is for 1/10 of the range.
3. Count the number of measurements in each range.
4. Plot the bar chart for the ten segments. You need to decide what the chart should contain.
5. The plotting of the bar chart itself is a major task, and you should have gathered the computational process for it from the earlier exercises.

Submission and Marks

You will need to submit your program by the end of Week 13, i.e. by Friday, 17 November 2018, 5 p.m. You are also required to capture the screen output of your bar chart and submit it either as an image file or a Word or PowerPoint file that contains the bar chart image.

Submit your program through your Assignment link in the course site. Do not submit it through any other link. It is your responsibility to ensure that your program and image are submitted properly.

Please name your file with your full name. The first line of your program should be a comment line bearing your full name and your matric number.

Marks will be awarded as follows:

Design of the algorithm (the logic is correct and efficient) – 25%
Clear and sensible Python code (the code is easy to follow, well organized and modularised) – 25%
Correctness of the solution – 30%
Quality of the output – 20%

Postscript

This is a small independent take-home project, i.e. a project that you need to execute on your own, at your own time. It is expected that you would typically spend 3-5 hours on the project per week over the four weeks. You can use all the resources available, except getting someone else to do it for you. You may refer to documents that discuss algorithms for constructing charts, but you must construct the chart using Turtle functions, and not call a chart plotting routine that is already available elsewhere.

We trust that, upon your honour, you will do this project independently and honestly.