Design of Lab7– Jacob Bollinger

Problem:

Design graphical software to illustrate Benford's law.

Plan:

I will start by additionally importing turtle. Then I will create the variables FONT and set it to Arial at normal 12 point and HORIZONTAL\_PIXELS set to 400.

I will create a new function called drawGrid(). It will take the inputs turtle, xMin, xMax, yMin, yMax, and places. First make sure the pen is up by using turtle.penup() and set the color to light grey with turtle.color(‘lightgrey’). Set a variable xGab equal to xMax minus xMin both divided by 10.

Next I will create a for loop in the function that will go for I in range(1, 11). In the for loop tell turtle to go to the coordinates (xMin plus i times xGap, yMin) and put the pen down with turtle.pendown(). Then tell turtle to go to (xMin + i \* xGap, yMax) and lift the pen up.

Outside of the for loop set yGap equal to yMax – yMin both divided by 10 and create another for loop that will go for I in range(1, 11) Inside tell turtle to go to (xMin, yMin plus i times yGap) and put the pen down with turtle.pendown(). Then tell turtle to go to (xMax, yMin plus i times yGap) and lift the pen up.

Outside of the for loop set the color to black with turtle.color(‘black’) and creat another for loop that will run for i in range(1, 11). Inside tell turtle to go to (-0.1, yMin + i \* yGap). Create an if statement to test if places > 0, and tell turtle to write the string of round(yMin + i \* yGap, places). Follow it by an else statement that will tell turtle to write the string of round(yMin + i \* yGap). Create another if statement to test if yMin is greater than zero and set vPos to yMin + 0.1. Create an elif statement to test if yMax is less than zero and set vPos to yMax – 0.1. Create an else statement to set vPos to -0.1. tell turtle to go to (xMin + i \* xGap, vPos). Create another if statement to test if yMax – yMin is less than 2 and set places to 2. Finish the for loop and function off by telling turtle to write the string of round(xMin + i \* xGap, places) and align it to the center.

Create another function named drawXAxis() with the parameters turtle, a, b, and step. Start by setting the color to light grey by using turtle.color(‘lightgrey’). Tell turtle to pick the pen up. Tell turtle to go to (a, 0) and put the pen down. Tell turtle to go to (b, 0) and pick the pen up. Tell turtle to then go to (b – step, -step) and write x in the font FONT.

Create an additional function named drawYAxis(). It will take the parameters turtle, a, b, and step. Start by setting the color to light grey by using turtle.color(‘lightgrey’). Tell turtle to pick the pen up. Tell turtle to go to (0, a) and put the pen down. Tell turtle to go to (0, b) and pick the pen up. Tell turtle to then go to (-2 \* step, 0.9 \* b) and write y in the font FONT.

Next create a function named drawGraph(). It will take the inputs turtle, valueL, a, b, yMin, yMax, and step. Start by telling turtle to pick up the pen and set x equal to a. Tell turtle to goto (x, valueL[0]) and put the pen down. Increase x by steps. Set i equal to 1 and tell turtle to put the pen down.

Set screen equal to turtle.Screen(). Set turtle’s speed to zero. Call drawXAxis() with the variables turtle, XMIN, XMAX, and XSTEPS. Call drawYAxis() with the variables turtle, 0, YMAX, and YSTEPS. Call drawGrid() with the variables turtle, XMIN, XMAX, YMIN, YMAX, and 2.