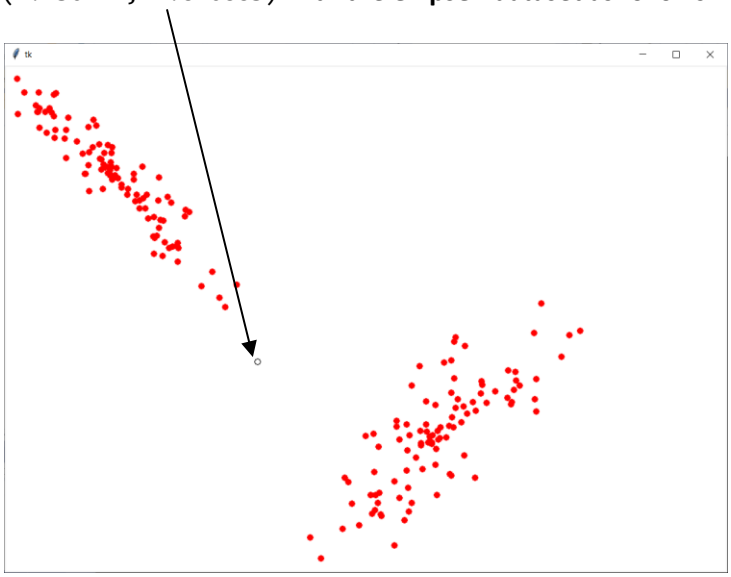
Answers to Week 4 Tutorial Questions

Question 1: Modify your program (**Week4Tutorial.py**) to display the following data sample `unknown_sample = (2.236779, 2.896883)` with the **ellipse1** dataset as follows



Answer:

```
#Question 1
unknown_sample = (2.236779, 2.896883)

#Question 1
#Display unknown sample
(x1, y1) = unknown_sample
x1 = x1*s + 150
y1 = y1*s + 150
C.create_oval(x1-r, y1-r, x1+r, y1+r)
```

Question 2: Write a function named **find_nearest_neighbour** that takes **unknown_sample** and **data_list** as its input parameters and returns the *nearest* data sample of **unknown_sample.** Place this function in **io_data_module.**

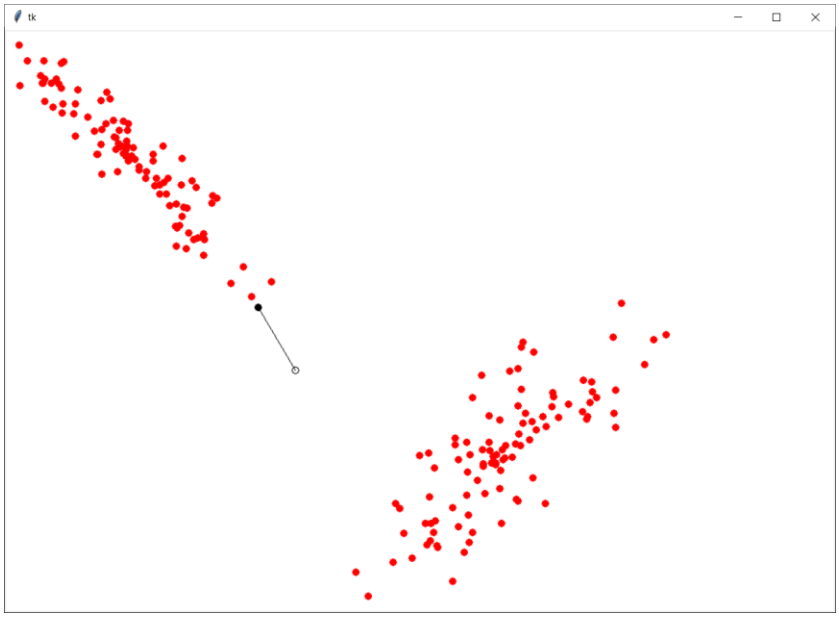
```
#define function
def find_nearest_neighbour(unknown_sample, data_list):

    #write your code here

    return nearest_sample
#end function
```

Answer:

```
#Example 5 in Week 3
#distance d = square root of (x2-x1)*(x2-x1) + (y2-y1)*(y2-y1)
#Two points are two tuples p1 = (x1, y1) and p2 = (x2, y2)
#define function
def calculate_distance(p1, p2):
    d = 0
    for i in range(len(p1)):
        d += (p2[i] - p1[i]) * (p2[i] - p1[i])
    d = d**0.5
    return d
#end function
```

```
#define function
def find_nearest_neighbour(unknown_sample, data_list):
    shortest_distance = 100000000
    nearest_sample = () #tuple
    for sample in data_list:
        dist = calculate_distance(unknown_sample, sample)
        if shortest_distance > dist:
            shortest_distance = dist
            nearest_sample = sample
    return nearest_sample
#end function
```

Question 3: Modify your program (**Week4Tutorial.py**) to call the **find_nearest_neighbour** function to get the nearest sample of the **unknown_sample**. Change colour of this nearest sample to black and draw a black line between this nearest sample and the unknown sample as follows



Answer:

```
import io_data_module as iodata
import tkinter as tk

#Open file and read data
data_list = iodata.read_data_file('ellipse1.txt')
#print(data_list)

#Question 1
unknown_sample = (2.236779, 2.896883)

#Question 2
nearest_sample = iodata.find_nearest_neighbour(unknown_sample, data_list)


#Create canvas
top = tk.Tk()
C = tk.Canvas(top, bg="white", height=700, width=1000)

#Display data
s = 90 #scale factor
```

```
r = 4 #radius
for (x,y) in data_list:
    x = x*s + 150
    y = y*s + 150
    C.create_oval(x-r, y-r, x+r, y+r, outline = "red", fill="red")

#Question 1
#Display unknown sample
(x1, y1) = unknown_sample
x1 = x1*s + 150
y1 = y1*s + 150
C.create_oval(x1-r, y1-r, x1+r, y1+r)

#Question 2
#Display nearest sample
(x2, y2) = nearest_sample
x2 = x2*s + 150
y2 = y2*s + 150
C.create_oval(x2-r, y2-r, x2+r, y2+r, outline = "black", fill="black")
C.create_line(x1, y1, x2, y2, fill = "black")

C.pack()
top.mainloop()
```