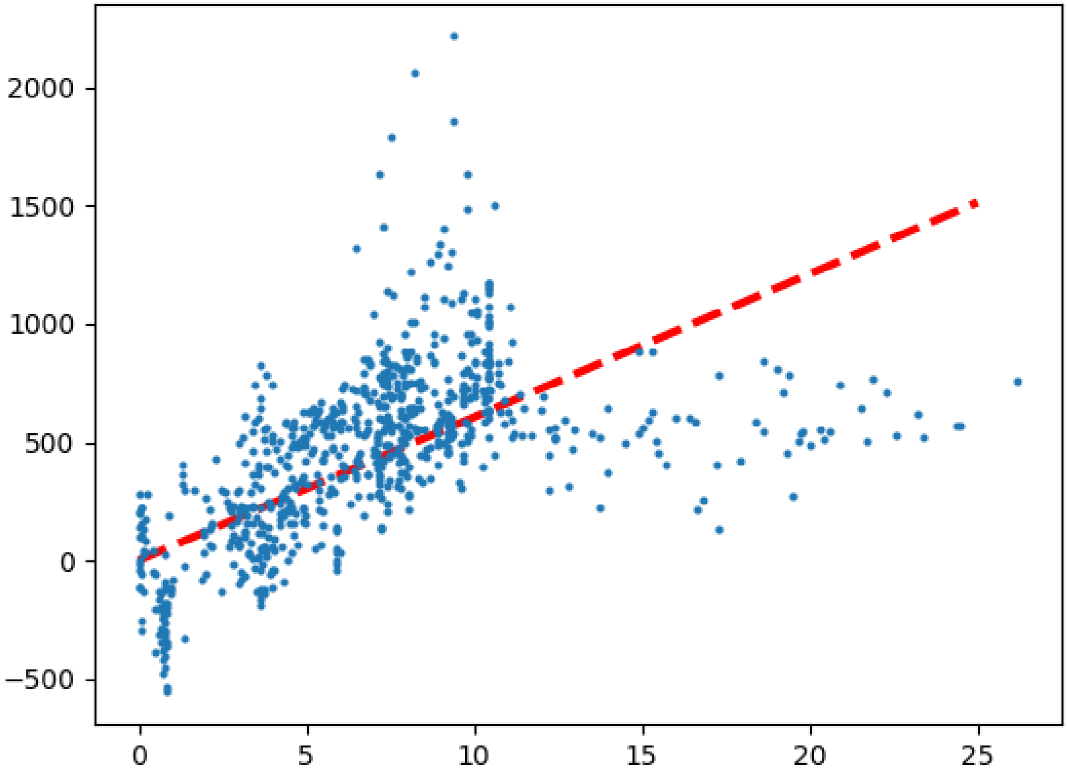
Trombini_Quentin_Week10

S10

a)

```
import numpy as np
import scipy as sp
import matplotlib.figure as figure
import matplotlib.backends.backend_agg as back

with open("./Astro/S10/EA.txt","r") as f:
    lines = f.readlines()

lines = lines[111:]
dist = []
vel = []

for line in lines:
    splitted = line.split()
    try:
        curr_dist = float(splitted[-2])
        curr_vel = float(splitted[-3])
    except:
        continue

    dist = np.append(dist,curr_dist)
    vel = np.append(vel,curr_vel)

def func (x , H0 ):
    y = H0 * x
    return (y)

H0 = 100

popt , pcov = sp.optimize.curve_fit(func, dist, vel, p0 = [H0])
H0_bestfit = popt[0]
fitted_x = np.linspace(0.0, 25.0, 50)
fitted_y = func(fitted_x, H0_bestfit)

fig = figure.Figure()
canvas = back.FigureCanvasAgg(fig)
ax = fig.add_subplot()

ax.plot(dist,vel, linestyle = "None", markersize = 2, marker= "o")
ax.plot(fitted_x, fitted_y, linestyle = "--", linewidth = 3, color = "red", zorder = 0.1)

fig.savefig("./Astro/S10/EA.png")
```

b)



c)

Once we get the value in the file we use $lines = lines[111 :]$ to remove all the documentation line.

Then we use:

```
def func (x , H0 ):
    y = H0 * x
    return (y)

H0 = 100

popt , pcov = sp.optimize.curve_fit(func, dist, vel, p0 = [H0])
H0_bestfit = popt[0]
fitted_x = np.linspace(0.0, 25.0, 50)
fitted_y = func(fitted_x, H0_bestfit)
```

We create a function so that we can use the optimize function from scipy. this will return us the best value in each case and allow us to trace this redline