import numpy as np

from scipy import stats

import matplotlib.pyplot as plt

#Varying the radius

plt.figure("Varying Radius")

r = np.array([0.15, 0.17, 0.18, 0.14, 0.16])

t = np.array([1.597, 1.14, 1.101, 2.36, 1.43])

t\_squared = t\*\*2

goal\_slope = 0.2418/(4\*np.pi\*\*2\*0.1365) #Fc/(4pi^2M)T^2

x,y = t\_squared,r

plt.scatter(x,y, c='r', s=5)

best\_fit = np.poly1d(np.polyfit(x, y, 1))

actual\_slope = best\_fit.c[0]

best\_fit\_s = str(best\_fit)

plt.plot(np.unique(x), np.poly1d(np.polyfit(x, y, 1))(np.unique(x)), 'k--', label=best\_fit\_s)

plt.legend()

plt.title('Varying Radius')

plt.xlabel('Period Squared (s^2)')

plt.ylabel('Radius (m)')

print(f'''

Varying Radius

The ideal slope is {goal\_slope}

The actual slope is {actual\_slope}

The percent difference is {(actual\_slope-goal\_slope)/goal\_slope\*100}%

''')

plt.savefig('VaryingRadius.png')

#Varying the force

plt.figure("Varying Force")

f = np.array([0.5396, 1.0301, 0.7358, 1.2263, 0.4415])

t = np.array([1.335, 1, 1.163, 0.948, 1.386])

inverse\_t\_squared = 1/(t\*\*2)

goal\_slope = 4\*np.pi\*\*2\*0.1365\*0.16 #4pi^2Mr \* 1/T^2

x,y = inverse\_t\_squared,f

plt.scatter(x,y, c='r', s=5)

best\_fit = np.poly1d(np.polyfit(x, y, 1))

actual\_slope = best\_fit.c[0]

best\_fit\_s = str(best\_fit)

plt.plot(np.unique(x), np.poly1d(np.polyfit(x, y, 1))(np.unique(x)), 'k--', label=best\_fit\_s)

plt.legend()

plt.title('Varying Force')

plt.xlabel('Inverse Period Squared (1/s^2)')

plt.ylabel('Force (N)')

print(f'''

Varying Force

The ideal slope is {goal\_slope}

The actual slope is {actual\_slope}

The percent difference is {(actual\_slope-goal\_slope)/goal\_slope\*100}%

''')

plt.savefig('VaryingForce.png')

#Varying the mass

m = np.array([0.1462, 0.1065, 0.2063])

t = np.array([1.211, 1.045, 1.339])

f = 0.7358

for index in range(3):

expected\_force = 4\*np.pi\*\*2\*m[index]\*0.16/t[index]

print(f'''

Varying Mass

The ideal mass is {expected\_force}

The actual slope is {f}

The percent difference is {(f-expected\_force)/expected\_force\*100}%

''')

plt.ion()

plt.show()

plt.pause(10)