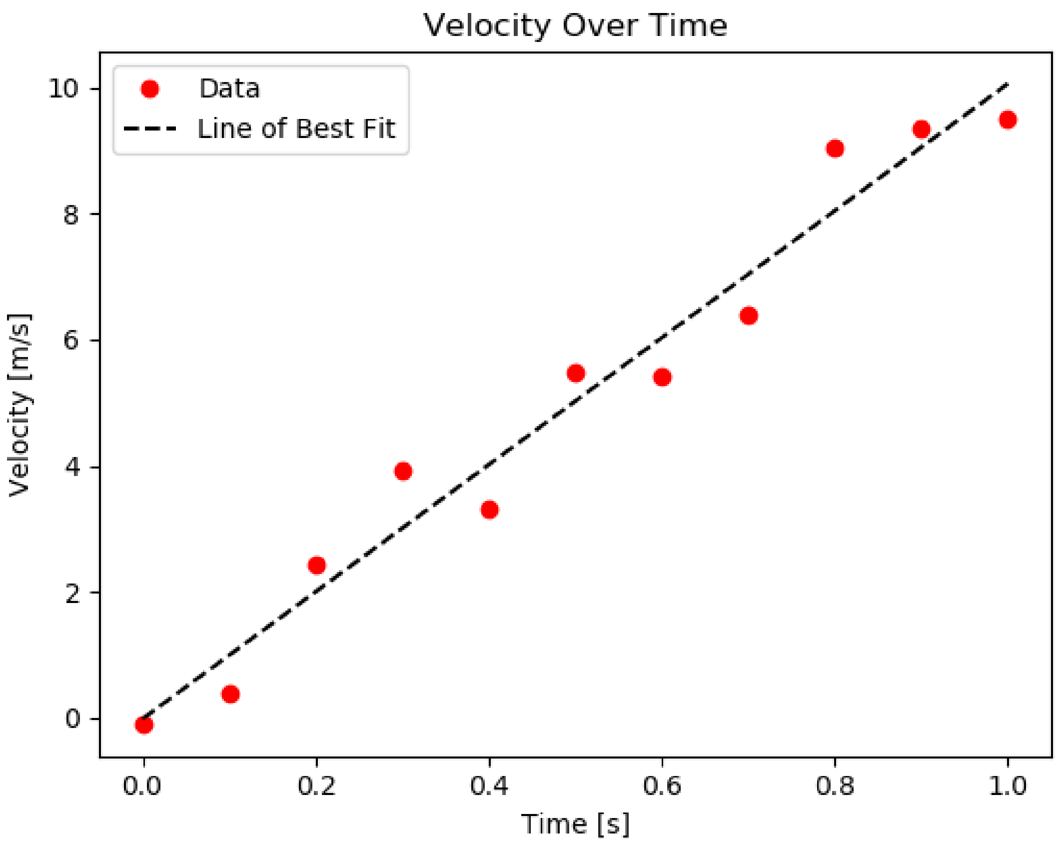
Homework 8

Ryan Branagan

March 4, 2019

Problem 1

```
#Ryan Branagan
#Collaborators: N/A
#Branagan_hw8_p1.py
#2/28/19

import numpy as np
import pylab as p
#%%
#Import Data
Dat = np.loadtxt("C:\\Users\\ryan-\\Documents\\251\\Homework 8\\freefall.data")
t = Dat[:,0]
v = Dat[:,1]

#Fitting Parameters
Sumt = np.sum(t)
Sumt2 = np.sum(t**2)
Sumv = np.sum(v)
Sumtv = np.sum(t*v)
N = len(t)

#Using LinAlg
A = np.matrix([[Sumt2,Sumt],[Sumt,N]])
r = np.matrix([[Sumtv],[Sumv]])
Soln = np.linalg.solve(A,r)
Fit = Soln[0,0]*t+Soln[1,0]

#Plotting
fig,ax = p.subplots(1,1)
ax.plot(t,v,'ro',label="Data")
ax.plot(t,Fit,'k--',label="Line of Best Fit")
```



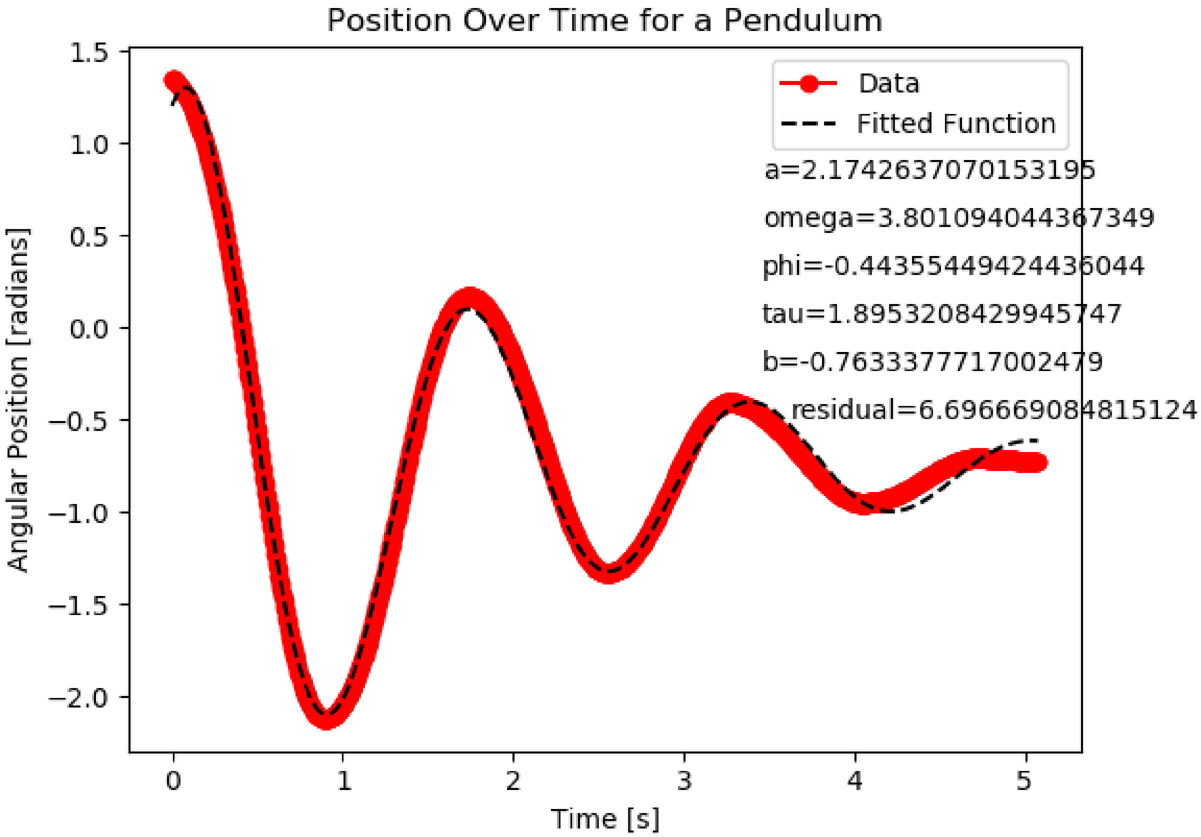

```
ax.set_title('Velocity Over Time')
ax.legend()
ax.set_xlabel('Time [s]')
ax.set_ylabel('Velocity [m/s]')

#Result
print("The experimental value of g for this experiment is approximately 10.089")
```

For this problem I just followed the prompt and used the given data and applied linear algebra methods to get a line of best fit for this data. My result for the measurement of gravity for this data is 10.18 m/s^2

Problem 2

```
#Ryan Branagan
#Collaborators: N/A
```

```
#Branagan_hw8_p2.py
#2/28/19

import numpy as np
import pylab as p
from scipy.optimize import curve_fit as cf
#%%
#Import Data
Dat = np.loadtxt("C:\\Users\\ryan-\\Documents\\251\\Homework 8\\damped_oscillation.txt")
t = Dat[:,0]
theta = Dat[:,1]

#Define our function
def Damped(t,a,w,phi,tau,b):
    return a*np.cos(w*t+phi)*np.exp(-t/tau)+b

#Fitting
par,con = cf(Damped,t,theta)

#Fitted Curve
def Fitted(t):
    a = par[0]
    w = par[1]
    phi = par[2]
    tau = par[3]
    b = par[4]
    return a*np.cos(w*t+phi)*np.exp(-t/tau)+b

#Getting Points
Fit = Fitted(t)

#Residual
R = (Fit - theta)**2
R = np.sum(R)

#Plotting
fig,ax = p.subplots(1,1)
ax.plot(t,theta,'ro-',label="Data")
ax.plot(t,Fit,'k--',label="Fitted Function")
ax.legend()
ax.set_title('Position Over Time for a Pendulum')
ax.set_xlabel('Time [s]')
ax.set_ylabel('Angular Position [radians]')
ax.annotate('a='+str(par[0]),xy=(410, 350),xycoords='figure pixels')
```




```
ax.annotate('omega='+str(par[1]),xy=(410, 325),xycoords='figure pixels')
ax.annotate('phi='+str(par[2]),xy=(410, 300),xycoords='figure pixels')
ax.annotate('tau='+str(par[3]),xy=(410, 275),xycoords='figure pixels')
ax.annotate('b='+str(par[4]),xy=(410, 250),xycoords='figure pixels')
ax.annotate('residual='+str(R),xy=(425, 225),xycoords='figure pixels')
```

For this problem I imported the data then used the scipy fitting to find the fitting parameters. With fitting parameters I defined the fitted function then plotted it. The fitting parameters and residual are annotated on the graph.