1. a.

#-------------------------------------------------------------------------------

# Name: floating point

# Purpose: to enter a floating point number

#

# Author: Clarissa Joyce

#

# Created: 27/09/2014

#-------------------------------------------------------------------------------

#!/usr/bin/env python

i=1.0

i=input("Please enter a floating point: ")

b.

#-------------------------------------------------------------------------------

# Name: output

# Purpose:

#

# Author: Clarissa Joyce

#

# Created: 27/09/2014

#-------------------------------------------------------------------------------

#!/usr/bin/env python

num1=5

if num1>=91:

num2=3

else:

if num1<6:

num2=4

else:

num2=2

x= num2\*num1+1

print "The output is: ",(x,x%7)

c.

Start

Fun1 X:10

120

End

d.

#-------------------------------------------------------------------------------

# Name: Q1d

# Purpose:

#

# Author: Clarissa Joyce

#

# Created: 27/09/2014

#-------------------------------------------------------------------------------

#!/usr/bin/env python

from math import exp, sin

def expsin(x,p,q):

return exp(p\*x)\*sin(q\*x)

def f(x,m,n,r,s):

return expsin(x,r,m)+expsin(x,s,n)

x=2.5

print "The corrected output is: ", f(x,0.1,0.2,1,1)

#Imports must come before they are used.

#Functions must be defined before they are called.

e.

#-------------------------------------------------------------------------------

# Name: Q1e

# Purpose:

#

# Author: Clarissa Joyce

#

# Created: 27/09/2014

#-------------------------------------------------------------------------------

#!/usr/bin/env python

print "For loop:"

for i in range(1,10):

print "i= ",i

print "\nWhile loop:"

j=1

while(j<10):

print"j= ",j

j+=1

1. analysis.py

#-------------------------------------------------------------------------------

# Name: analysis

# Purpose:

#

# Author: Claire Joyce

#

# Created: 27/09/2014

#-------------------------------------------------------------------------------

#!/usr/bin/env python

import numpy as np

from numpy import \*

from math import \*

import matplotlib.pyplot as plt

data = np.loadtxt("extinctdata.txt",skiprows=1)

x=data[:,0]

y=data[:,1]

r=[]

a=np.mean(y)

for i in range(0,len(y)):

r.append(y[i]-a)

s=np.std(y)

print len(y)

print len(r)

print "Average y val: ",a,"\nStandard Dev: ",s

# initialize sums

sum\_x = 0

sum\_y = 0

sum\_xx = 0

sum\_xy = 0

xmin = min(x)

#perform our calculations

for i in range(0,len(x)):

sum\_x = sum\_x+x[i-1]

sum\_y = sum\_y+y[i-1]

xx = x[i-1]\*x[i-1]

sum\_xx = sum\_xx +xx

xy = x[i-1]\*y[i-1]

sum\_xy = sum\_xy+xy

#calculate the coefficients

D = len(x)\*sum\_xx - sum\_x\*sum\_x

A= (sum\_xx\*sum\_y - sum\_x\*sum\_xy)/D

B= (len(x)\*sum\_xy-sum\_x\*sum\_y)/D

#plot data points

plt.subplot(211)

plt.plot(x,y,"bo")

plt.title("Least Squares Plot of Extinct Data")

plt.xlabel("Air Mass")

plt.ylabel("Magnitude")

plt.errorbar(x,y,s,marker='s',mfc='red', mec='green', ms=2, mew=1)

#plot least squares fit line

xc= linspace(xmin,max(x),100)

yc= A+B\*xc

print "y=",B,"x","+",A

plt.plot(xc,yc,"r-")

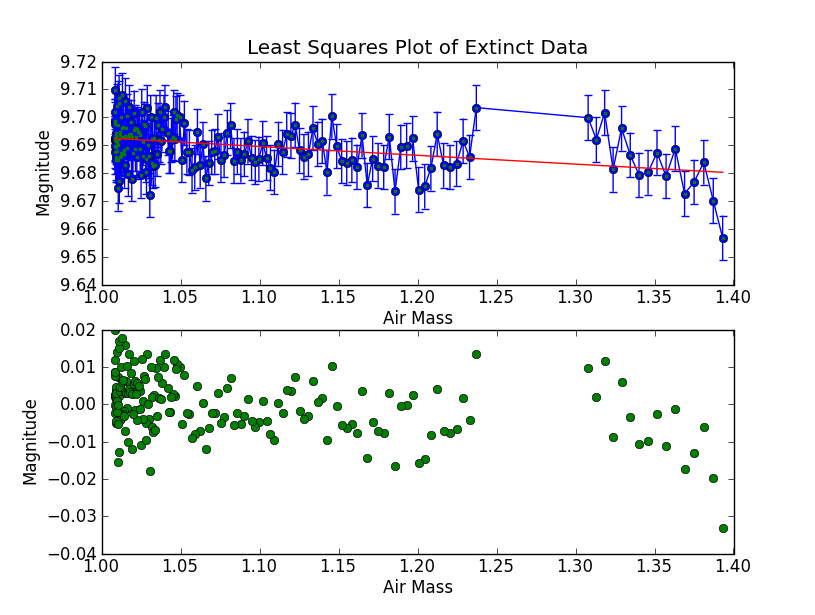
plt.subplot(212)

plt.plot(x,r,"go")

plt.xlabel("Air Mass")

plt.ylabel("Magnitude")

plt.show()



1. pitch.py

#-------------------------------------------------------------------------------

# Name: pitch

# Purpose:

#

# Author: Claire Joyce

#

# Created: 27/09/2014

#-------------------------------------------------------------------------------

#!/usr/bin/env python

from numpy import \*

from math import \*

import matplotlib.pyplot as plt

import numpy as np

from mpl\_toolkits.mplot3d import Axes3D

data = np.loadtxt("pitchfx\_data-FIXED.txt", skiprows = 2,usecols=(25,27,28,29,32,33,34,35,36,37,38,39,40))

x0=data[:,4]

print x0

y0=data[:,5]

z0=data[:,6]

tstart=data[:,0]

ftime=data[:,1]

vx0=data[:,7]

vy0=data[:,8]

vz0=data[:,9]

ax=data[:,10]

ay=data[:,11]

az=data[:,12]

finx=data[:,2]

finz=data[:,3]

N=int(ftime[0]\*100)

x = zeros(N,float)

y = zeros(N,float)

z = zeros(N,float)

x[0],y[0],z[0] = (x0[0],y0[0],z0[0])

print x0[0]

print vx0[0]

print ax[0]

for i in range(0, len(x)):

t = i/100.

x[i] = x0[0] +vx0[0]\*t + 0.5\*ax[0]\*t\*\*2

y[i] = y0[0] +vy0[0]\*t + 0.5\*ay[0]\*t\*\*2

z[i] = z0[0] +vz0[0]\*t + 0.5\*az[0]\*t\*\*2

#plotting in 3D

fig = plt.figure()

ax = fig.gca(projection='3d')

ax.plot(x,y,z)

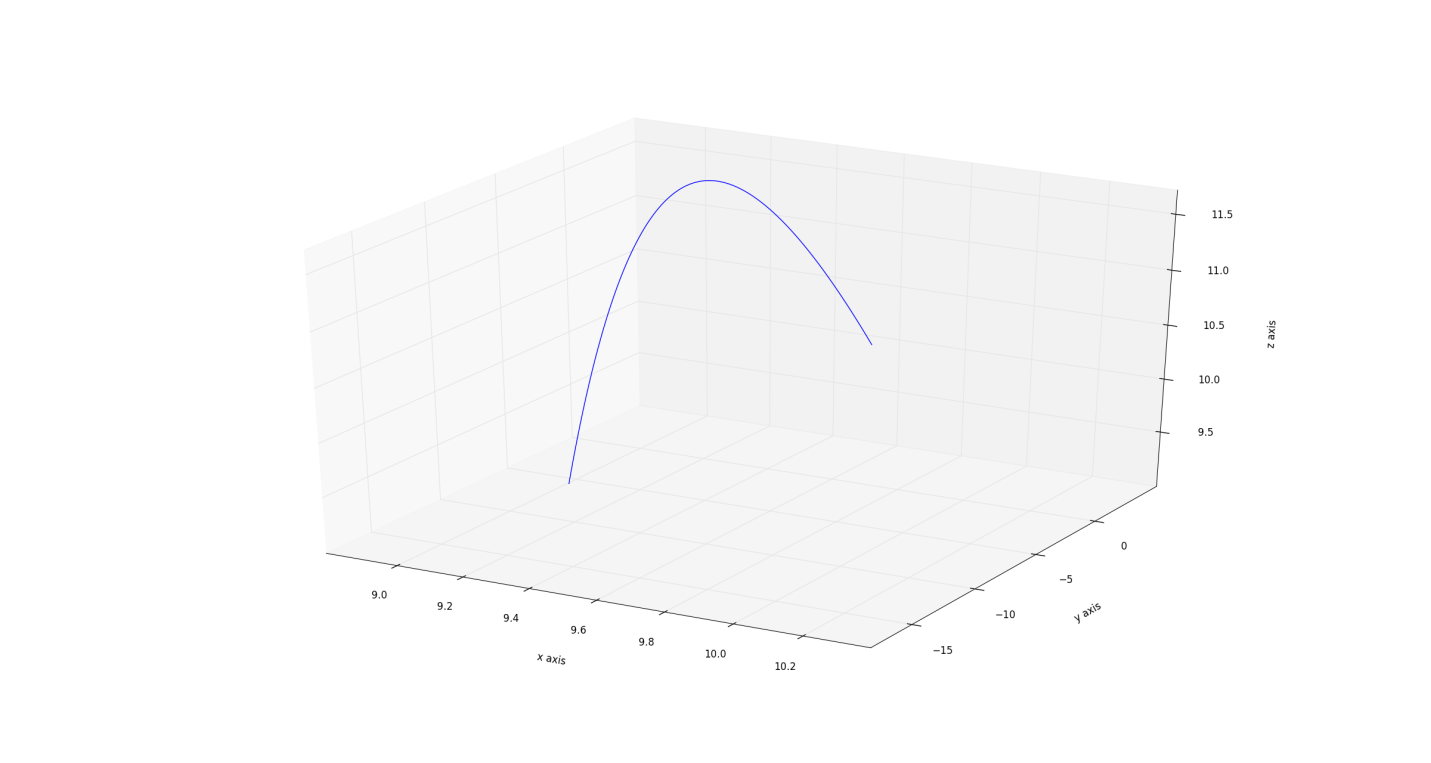
ax.set\_xlabel("x axis")

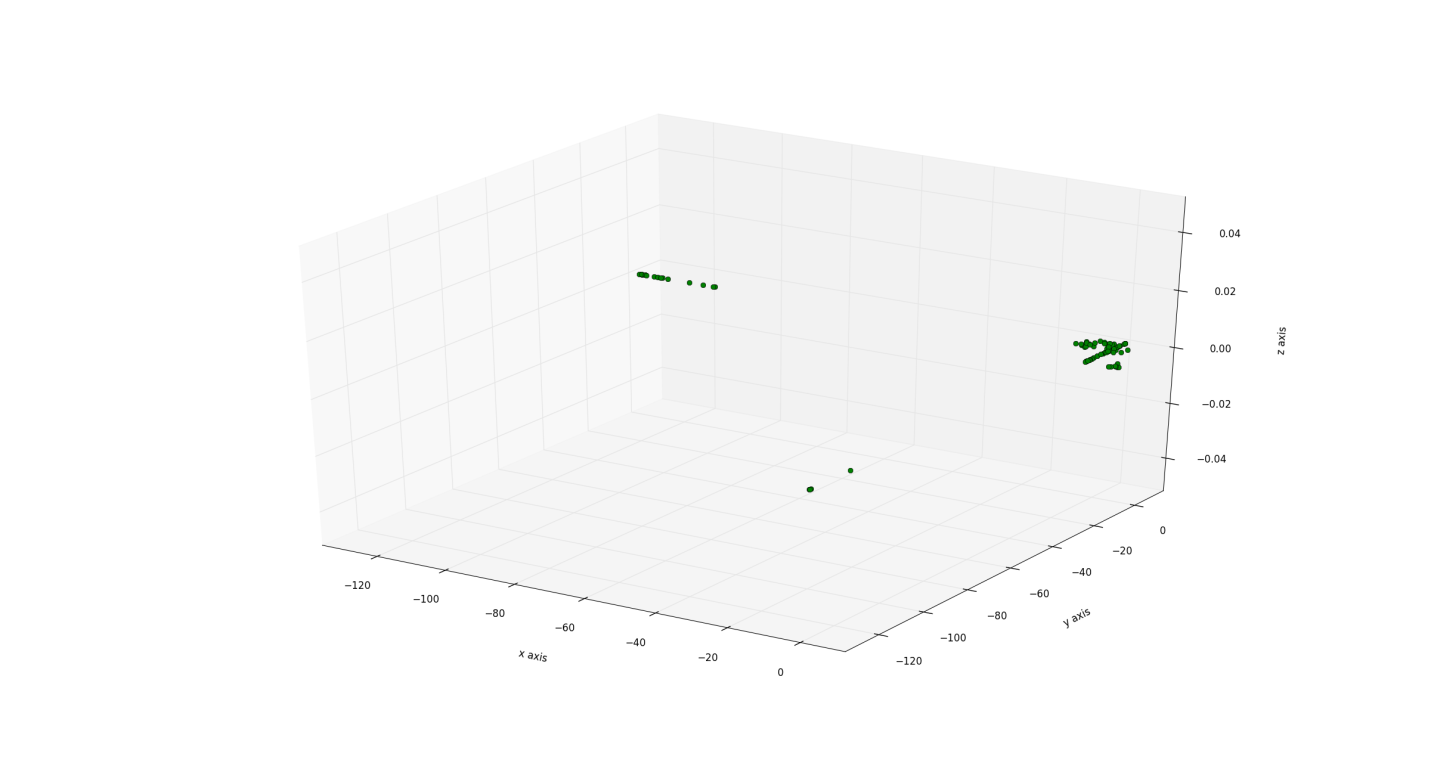
ax.set\_ylabel("y axis")

ax.set\_zlabel("z axis")

plt.plot(finx,finz,"go")

plt.show()





1. decay2.py

#-------------------------------------------------------------------------------

# Name: examdecay

# Purpose:

#

# Author: Claire Joyce

#

# Created: 27/09/2014

#-------------------------------------------------------------------------------

#!/usr/bin/env python

from math import \*

import matplotlib.pyplot as plt

k1=5.76\*10\*\*(-11)

k2=4.85\*10\*\*(-10)

To=0

T=4.5\*10\*\*9

x = 0

K0=10000

Ar0=0

Ca0=0

h=10\*\*7

def dK(K):

return -(k1+k2)\*K

def dAr(K):

return k1\*K

def dCa(K):

return k2\*K

AA=[]

KK=[]

while To<T:

plt.plot(To,K0,"bo")

plt.plot(To,Ar0,"r+")

plt.plot(To,Ca0,"go")

Kend=K0-k1\*dK(K0)\*h

K0+=(dK(K0)+dK(Kend))\*h/2.0

Arend=Ar0 + dAr(K0)\*h

Ar0+=(dAr(K0)+dAr(Arend))\*h/2.0

Caend=Ca0 + dCa(K0)\*h

Ca0+=(dCa(K0)+dCa(Caend))\*h/2.

AA.append(abs(round(dAr(Ar0)/dK(K0),4)))

#KK.append(round(dK(K0),3))

To=To+h

#print AA,"\n",KK

ratio=3.900\*10\*\*-3

'''

for i in range(0,len(AA)):

print AA[i]

'''

year=(AA.index(ratio))\*10\*\*7

print "Age of T-rex: ",year," years"

plt.title("Radioactive decay of K, Ar, and Ca")

plt.xlabel("Years")

plt.ylabel("Number atoms")

plt.show()

