7/1/2019 ppm\_vs\_aperture\_fit.py:

*# import packages to use*

import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits import mplot3d *#new import*

from mpl\_toolkits.mplot3d import Axes3D *#new import*

from mpl\_toolkits.mplot3d.art3d import Poly3DCollection #new import

import matplotlib.ticker as ticker #new import

import pylab #new import

from matplotlib.pyplot import \*

from astropy.utils.data import get\_pkg\_data\_filename

from astropy.table import Table

from astropy.io import fits

import glob

import batman

import lmfit

import corner

# directory='/Users/annaburkholder/exp\_det\_scripts/visit23\_defringed/'

# directory='/home/ian/Desktop/WebbData/visit23\_defringed/'

directory='/visit23\_defringed/' #Change directory to proper location

number\_of\_images=43

number\_of\_rows=64

*#create matrix that stores +value in column0, -value in column1, and ppm value in column2*

*#Matrices are currently manually inputted.*

A = np.array([[6.0, 6.0, 297.69656036629453], [5.0, 5.0, 289.5760836868116], [4.0, 4.0, 636.8248813770422], [3.0, 3.0, 1904.8346594325174], [5.0, 6.0, 293.1929700422862], [4.0, 6.0, 581.8358509634846], [5.0, 4.0, 294.33305061199627], [6.0, 4.0, 274.49171578444094], [7.0, 3.0, 276.24555962003336], [7.0, 4.0, 277.90456690066134]])

*#print each column to check array*

print(A[:,0])

print(A[:,1])

print(A[:,2])

*#create 3D line plot to graph ppm and aperture size*

fig = plt.figure()

ax = plt.axes(projection='3d')

ax = plt.axes(projection='3d')

xl = A[:,0]

yl = A[:,1]

zl = A[:,2]

ax.plot3D(xl, yl, zl)

ax.set\_xlabel('+aperture')

ax.set\_ylabel('-aperture')

ax.set\_zlabel('std. dev. (ppm)')

plt.show()

*#create 3D scatterplot to graph ppm and aperture size*

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

xs = A[:,0]

ys = A[:,1]

zs = A[:,2]

ax.scatter(xs, ys, zs)

ax.set\_xlabel('+aperture')

ax.set\_ylabel('-aperture')

ax.set\_zlabel('std. dev. (ppm)')

plt.show()

#create tri-plot to graph ppm and aperture size

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

x = A[:,0]

y = A[:,1]

z = A[:,2]

verts = [list(zip(x, y, z))]

print(verts)

ax.add\_collection3d(Poly3DCollection(verts), zs='z')

ax.set\_xlim(0,10)

ax.set\_ylim(0,10)

ax.set\_zlim(0,2000)

plt.show()