#starting off with all three libraries that were highlighted as well as io and lines for the legends used later on

%matplotlib inline

import matplotlib.pyplot as plt

from matplotlib import pylab

import numpy as np

from numpy import arange,array,ones

import scipy as py

from scipy import stats

from io import StringIO

import matplotlib.lines as mlines

import math

file = "//Users//RenOnly.//Documents//3510.KeplerIII.txt" #reading in data from the file

titles = ["Orbiting", "Orbiter", "Orbiter Type", "Semimajor Axis (AU)", "Orbital Period (Years)"] #titles of columns for lists that may be used later on

np.genfromtxt(StringIO(file)) #generate strings from the file for each column

np.names=titles #shorthand for titles just in case they are used

np.usecols=("Semimajor Axis (AU)", "Orbital Period (Years)") #not sure if this does anything but I listed it just in case

#setting variables to be used for extracted data, this part is for the Sun stuff only, Planets

centralo1a = np.genfromtxt(file, usecols=[0], dtype=str, skip\_header=2, skip\_footer=21)

oplanet1a = np.genfromtxt(file, usecols=[1], dtype=str, skip\_header=2, skip\_footer=21)

otype1a = np.genfromtxt(file, usecols=[2], dtype=str, skip\_header=2, skip\_footer=21)

smaxis1a = np.genfromtxt(file, usecols=[3], skip\_header=2, skip\_footer=21, dtype=float)

per1a = np.genfromtxt(file, usecols=[4], skip\_header=2, skip\_footer=21, dtype=float)

#setting variables to be used for extracted data, this part is for the Sun stuff only, Dwarf Planets

centralo1b = np.genfromtxt(file, usecols=[0], dtype=str, skip\_header=11, skip\_footer=16)

oplanet1b = np.genfromtxt(file, usecols=[1], dtype=str, skip\_header=11, skip\_footer=16)

otype1b = np.genfromtxt(file, usecols=[2], dtype=str, skip\_header=11, skip\_footer=16)

smaxis1b = np.genfromtxt(file, usecols=[3], skip\_header=11, skip\_footer=16, dtype=float)

per1b = np.genfromtxt(file, usecols=[4], skip\_header=11, skip\_footer=16, dtype=float)

#setting variables to be used for extracted data, this part is for the Jupiter stuff only, Moons

centralo2 = np.genfromtxt(file, usecols=[0], dtype=str, skip\_header=19, skip\_footer=10)

oplanet2 = np.genfromtxt(file, usecols=[1], dtype=str, skip\_header=19, skip\_footer=10)

otype2 = np.genfromtxt(file, usecols=[2], dtype=str, skip\_header=19, skip\_footer=10)

smaxis2 = np.genfromtxt(file, usecols=[3], dtype=float, skip\_header=19, skip\_footer=10)

per2 = np.genfromtxt(file, usecols=[4], skip\_header=19, skip\_footer=10, dtype=float)

#setting variables to be used for extracted data, this part is for the Saturn stuff only, Moons

centralo3 = np.genfromtxt(file, usecols=[0], dtype=str, skip\_header=25, skip\_footer=0)

oplanet3 = np.genfromtxt(file, usecols=[1], dtype=str, skip\_header=25, skip\_footer=0)

otype3 = np.genfromtxt(file, usecols=[2], dtype=str, skip\_header=25, skip\_footer=0)

smaxis3 = np.genfromtxt(file, usecols=[3], dtype=float, skip\_header=25, skip\_footer=0)

per3 = np.genfromtxt(file, usecols=[4], skip\_header=25, skip\_footer=0, dtype=float)

#making sure I have the right sets of data by printing what is being read for each variable for the Sun's Planets

print(centralo1a)

print(oplanet1a)

print(otype1a)

print(smaxis1a)

print(per1a)

#making sure I have the right sets of data by printing what is being read for each variable for the Sun's Dwarf Planets

print(centralo1b)

print(oplanet1b)

print(otype1b)

print(smaxis1b)

print(per1b)

#making sure I have the right sets of data by printing what is being read for each variable for Jupiter

print(centralo2)

print(oplanet2)

print(otype2)

print(smaxis2)

print(per2)

#making sure I have the right sets of data by printing what is being read for each variable for Saturn

print(centralo3)

print(oplanet3)

print(otype3)

print(smaxis3)

print(per3)

#changing float type to arrays so that numbers with decimals can be manipulated and scaled

sm1a=np.array(smaxis1a)

print(sm1a)

pd1a=np.array(per1a)

print(pd1a)

sm1b=np.array(smaxis1b)

print(sm1b)

pd1b=np.array(per1b)

print(pd1b)

a=np.array(smaxis2)

print(a)

b=np.array(per2)

print(b)

c=np.array(smaxis3)

print(c)

d=np.array(per3)

print(d)

#scaling data to make sure that the units are the same so they can be accurately represented on the same linear and log scales

sm2=(1/149600000)\*(1000)\*a

pd2=(1/365)\*b

sm3=(1/149600000)\*(1000)\*c

pd3=(1/365)\*d

print(sm2) #making sure that the 10^3 km scale has been changed to AU properly for Jupiter stuff

print(pd2) #making sure that the days scale has been changed to Years properly for Jupiter stuff

print(sm3) #making sure that the 10^3 km scale has been changed to AU properly for Saturn stuff

print(pd3) #making sure that the days scale has been changed to Years properly for Saturn stuff

#testing pie chart of number of P type orbits of the Sun and D type orbits of the Sun

n = 3

Z = np.ones(n)

Z[0] = 5

Z[1] = 1.6\*(Z[0])

Z[-1] = 3.2\*(Z[0])

orbiters = ["5 Dwarf Planets",

"8 Major Planets",

"16 Moons"]

data = [float(x.split()[0]) for x in orbiters]

cosmos = [x for x in orbiters]

print(cosmos)

#visual representation parameters for the pie chart

plt.figure(figsize=(10, 6), dpi=80)

plt.axes([0.025, 0.025, 0.95, 0.95])

plt.pie(Z, explode=Z\*.0025, shadow=True, labels=orbiters, textprops=dict(color="black"), autopct='%1.2f%%')

plt.axis(‘equal')

plt.legend(cosmos,

title="Types of Orbiters in our Solar System",

loc="center",

bbox\_to\_anchor=(1, 0, 0.5, 1))

plt.xticks(())

plt.yticks(())

plt.title('Orbiters in Our Solar System')

plt.show()

#making the completely linear plot of data, defining parameters so that the lines are different

plt.figure(figsize=(10, 6), dpi=80)

plt.plot(sm1a,pd1a,'o',linestyle='-',c="green")

plt.plot(sm1b,pd1b,'\*',linestyle='-',c="orange")

plt.plot(sm2,pd2,'s',linestyle='-',c="blue")

plt.plot(sm3,pd3,'<',linestyle='-',c="pink")

line1, = plt.plot(sm1a,pd1a,'o',linestyle='-',c="green")

line2, = plt.plot(sm1b,pd1b,'\*',linestyle='-',c="orange")

line3, = plt.plot(sm2,pd2,'s',linestyle='-',c="blue")

line4, = plt.plot(sm3,pd3,'<',linestyle='-',c="pink")

plt.xlabel("Semimajor Axis (AU)")

plt.ylabel("Orbital Period (Years)”)

plt.xscale('linear')

plt.yscale('linear')

plt.title('Linear Plot of Semimajor Axis vs. Orbital Period')

plt.legend((line1,line2,line3,line4),('Major Planets Orbiting the Sun','Dwarf Planets Orbiting the Sun','Galilean Moons and Himalia Satellite of Jupiter','Moons of Saturn'))

plt.show()

#testing log plot with data on the same graph

plt.figure(figsize=(10, 6), dpi=80)

plt.plot(sm1a,pd1a,'o',linestyle='-',c="green")

plt.plot(sm1b,pd1b,'\*',linestyle='-',c="orange")

plt.plot(sm2,pd2,'s',linestyle='-',c="blue")

plt.plot(sm3,pd3,'<',linestyle='-',c="pink")

line5, = plt.plot(sm1a,pd1a,'o',linestyle='-',c="green")

line6, = plt.plot(sm1b,pd1b,'\*',linestyle='-',c="orange")

line7, = plt.plot(sm2,pd2,'s',linestyle='-',c="blue")

line8, = plt.plot(sm3,pd3,'<',linestyle='-',c="pink")

plt.xlabel("Semimajor Axis (AU)")

plt.ylabel("Orbital Period (Years)")

plt.yscale('log')

plt.xscale('log')

plt.legend((line5,line6,line7,line8),('Major Planets Orbiting the Sun','Dwarf Planets Orbiting the Sun','Galilean Moons and Himalia Satellite of Jupiter','Moons of Saturn’))

plt.title('Logarithmic Plot of Semimajor Axis vs. Orbital Period')

plt.grid(True)

plt.show

#finding appropriate slopes and y-intercepts for Jupiter and Saturn

#since y=mx+b, and m will be defined for both cases, it is possible to solve for y

length1 = sm2

length1.sort()

print(length1)

time1 = pd2

time1.sort()

print(time1)

slope1, intercept1 = np.polyfit(np.log(length1), np.log(time1), 1)

print("The slope of the Jupiter line is:",slope1)

plt.loglog(length1, time1,'s',linestyle='-',c="blue")

#changing points into the correct coordinates for slope

x1 = math.log10(float(sm2[-1]))

y1 = math.log10(float(pd2[-1]))

print(x1)

print(y1)

x2 = 1

print(x2)

m = slope1

yi = (m\*(x2-x1))+y1

k = yi-(m\*x2)

print("The y-intercept of the Jupiter line where semimajor axis is equal to 1 AU is: 10^",k)

M1=((10\*\*k)\*\*2)\*\*-1

print("The mass of Jupiter in Solar-Mass units is",M1)

M2=M1/(3\*(10\*\*-6))

print("This corresponds to a mass of",M2,"Earth-Mass units.")

length2 = sm3

length2.sort()

print(length2)

time2 = pd3

time2.sort()

print(time2)

slope2, intercept2 = np.polyfit(np.log(length2), np.log(time2), 1)

print("The slope of the Saturn line is:”,slope2)

plt.loglog(length2, time2,'<',linestyle='-',c="pink")

x3 = math.log10((float(sm3[-1])))

y3 = math.log10(float(pd3[-1]))

print(x3)

print(y3)

x4 = 1

print(x4)

n = slope2

yj = (n\*(x4-x3))+y3

l = yj-(m\*x4)

print("The y-intercept of the Saturn line where semimajor axis is equal to 1 AU is: 10^",l)

M3=(((10\*\*l)\*\*2)\*\*-1)

print("The mass of Saturn in Solar-Mass units is",M3)

M4=M3/(3\*(10\*\*-6))

print("This corresponds to a mass of",M4,"Earth-Mass units.")