Inertial Balance in python

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY
كلية العلوم والدراسات الإنسانية - قسم الفيزياء
College of Scienece and Humanities - Physics Department

Noura Albaker-2190004926
Farah Alsardi 2190002998
Fatima Albenali- 2200006291

Introducton

In today's world, especially in the sciences, it is crucial to have the fundamental abilities of modeling abstraction, analysis, simulation, and validation. Some pupils in numerous underdeveloped nations acquire these skills hypothetically. People will be able to model any scientific experiment and adjust any parameters using only a laptop with modeling using Python.

objective

1-chose one of some physical experimental lap and use python programming like Inertial Balance

2- use the programming skills we learned in a scientific computing and modeling course

Methodology

1-Create the data table1 by using pandas library (func : DataFrame).

```
#Table 1
import pandas as pd
data={'m_in_gram':[500,200,300,600,150,400],
      't10':[6.07,4.54,4.88,7.06,3.72,5.25]}
df=pd.DataFrame(data)
print(df)
```

2.Calculate the requirement data.

```
df['t_10']= df.apply(lambda row: row.t10/10, axis= 1)
print(df)
```

```
df['t2'] = df.apply(lambda row: row.t_10**2, axis =1)
print(df)
```

3.Styling the data table by using seaborn library
(func : style.background_gradient)

```
[14] import seaborn as sns

     cm=sns.light_palette("blue",as_cmap=True)

     b=df.style.background_gradient(cmap=cm)
     b
```

4-we repeat step 1,2and 3 to great table 2 and 3

Methodology

5-Plotting the data table1 by using matplotlib library

```
xpoints = np.array([100, 600])
ypoints = np.array([0, 0.50])
plt.xlabel("Mass(g)")
plt.ylabel("Time(s^2)")
plt.title("Mass vs time^2", fontsize=20)
plt.plot(xpoints, ypoints, color='hotpink')
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([150,200,300,400,500,600])
ypoints = np.array([0.14,0.2,0.23,0.27,0.36,0.49])
plt.xlabel("Mass(g)")
plt.ylabel("Time(s^2)")
plt.title("Mass vs time^2", fontsize=20)
plt.plot(xpoints, ypoints,'o', ms=10, mec='hotpink', mfc='hotpink')
plt.show()
```

6- calculate the ratio mi and mg

```
#find the ratio mg1/mg2
mi1=102.040
mi2=225.102
ratio2= mi1/mi2
r2=round(ratio2,3)
print(r2,"g")
```

```
#find the ratio mi1/mi2
mi1=150
mi2=400
ratio= mi1/mi2
r=round(ratio,3)
print(r,"g")
```

Result

1. Data table

	m_in_gram	t10	t_10	t2
0	500	6.070000	0.607000	0.368449
1	200	4.540000	0.454000	0.206116
2	300	4.880000	0.488000	0.238144
3	600	7.060000	0.706000	0.498436
4	150	3.720000	0.372000	0.138384
5	400	5.250000	0.525000	0.275625

table 1

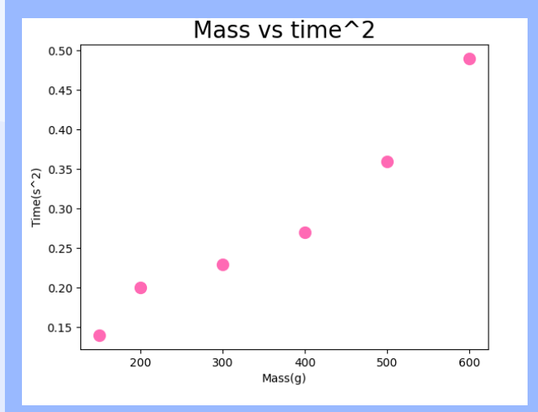
	body	T10	T10_10	T2	mi
0	light Body (m1)	3.870000	0.387000	0.149769	230
1	Heavy Body(m**2)	4.690000	0.469000	0.219961	300

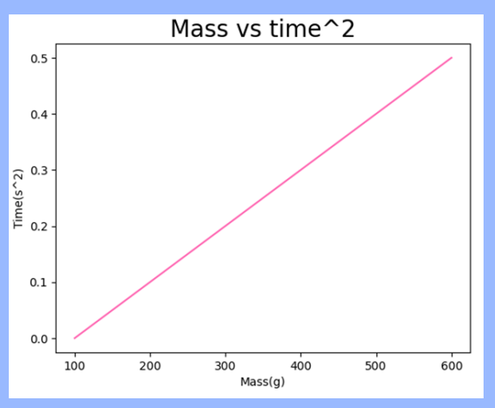
table 2

	W (dyne)	mg (g)
0	100000	102.040000
1	250000	225.102000

table 3

2. plotting





3. Results

ratio mi = 0.453 g Ratio mg= 0.375 g

Discussion

1. We faced an issue while plotting, the line wasn't a straight line connecting the dots. We searched online for a way to make it streamline which worked (self-learning).
2. We had a problem with the second table, but after a brief research, it was solved.

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY
كلية العلوم والدراسات الإنسانية - قسم الفيزياء
College of Scienece and Humanities - Physics Department

Inertial Balance in python

Noura Albaker-2190004926
Farah Alsardi 2190002998
Fatima Albenali- 2200006291

Codes :

```
[ ]  #Table 1
     import pandas as pd
     data={'m_in_gram':[500,200,300,600,150,400],
           't10':[6.07,4.54,4.88,7.06,3.72,5.25]}
     df=pd.DataFrame(data)
     print(df)
```

```
   m_in_gram   t10
0        500  6.07
1        200  4.54
2        300  4.88
3        600  7.06
4        150  3.72
5        400  5.25
```

```
[ ]  df
```

	m_in_gram	t10
0	500	6.07
1	200	4.54
2	300	4.88
3	600	7.06
4	150	3.72
5	400	5.25

```
[ ]  df['t_10']= df.apply(lambda row: row.t10/10, axis= 1)
     print(df)
```

```
   m_in_gram   t10   t_10
0        500  6.07  0.607
1        200  4.54  0.454
2        300  4.88  0.488
3        600  7.06  0.706
4        150  3.72  0.372
5        400  5.25  0.525
```

```
[ ]  df['t2'] = df.apply(lambda row: row.t_10**2, axis =1)
     print(df)
```

```
   m_in_gram   t10   t_10        t2
0        500  6.07  0.607  0.368449
1        200  4.54  0.454  0.206116
2        300  4.88  0.488  0.238144
3        600  7.06  0.706  0.498436
4        150  3.72  0.372  0.138384
5        400  5.25  0.525  0.275625
```

```
[ ]  import seaborn as sns

     cm=sns.light_palette("blue",as_cmap=True)

     b=df.style.background_gradient(cmap=cm)
     b
```

	m_in_gram	t10	t_10	t2
0	500	6.070000	0.607000	0.368449
1	200	4.540000	0.454000	0.206116
2	300	4.880000	0.488000	0.238144
3	600	7.060000	0.706000	0.498436
4	150	3.720000	0.372000	0.138384
5	400	5.250000	0.525000	0.275625

```
[ ]  #table 2
     import matplotlib.pyplot as plt
     import pandas as pd
     data2={"body      ":["light Body (m1)","Heavy Body(m**2)"],
            "T10":[3.87,4.69]}
     df2= pd.DataFrame(data2)
     print(df2)
```

```
           body          T10
0   light Body (m1)   3.87
1  Heavy Body(m**2)   4.69
```

```
[ ]  df2['T10_10']= df2.apply(lambda row: row.T10/10 , axis=1)
     print(df2)
```

```
           body          T10  T10_10
0   light Body (m1)   3.87    0.387
1  Heavy Body(m**2)   4.69    0.469
```

```
[ ]  df2['T2']= df2.apply(lambda row: row.T10_10**2 , axis=1)
     print(df2)
```

```
           body          T10  T10_10        T2
0   light Body (m1)   3.87    0.387  0.149769
1  Heavy Body(m**2)   4.69    0.469  0.219961
```

```
[ ]  df2
```

	body	T10	T10_10	T2
0	light Body (m1)	3.87	0.387	0.149769
1	Heavy Body(m**2)	4.69	0.469	0.219961

```
[ ]  df2
```

	body	T10	T10_10	T2
0	light Body (m1)	3.87	0.387	0.149769
1	Heavy Body(m**2)	4.69	0.469	0.219961

```
[ ]  mass=[230,300]
     df2["mi"]=mass

     df2
```

	body	T10	T10_10	T2	mi
0	light Body (m1)	3.87	0.387	0.149769	230
1	Heavy Body(m**2)	4.69	0.469	0.219961	300

```
[ ]  import seaborn as sns

     cm=sns.light_palette("black",as_cmap=True)

     b=df2.style.background_gradient(cmap=cm)
     b
```

	body	T10	T10_10	T2	mi
0	light Body (m1)	3.870000	0.387000	0.149769	230
1	Heavy Body(m**2)	4.690000	0.469000	0.219961	300

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY
كلية العلوم والدراسات الإنسانية - قسم الفيزياء
College of Scienece and Humanities - Physics Department

Inertial Balance in python

Noura Albaker-2190004926
Farah Alsardi 2190002998
Fatima Albenali- 2200006291

Codes :

```
#table 3
import matplotlib.pyplot as plt
import pandas as pd
data3 = {'W (dyne)' : [100000 , 250000] }
df3 = pd.DataFrame(data3)
df3 [mg] = df3.apply (lambda row : row.W (dyne)/980, axis = 1)
print (df3)
```

```
df3
```

	W (dyne)	mg (g)
0	100000	102.040
1	250000	225.102

```
import seaborn as sns

cm=sns.light_palette("pink",as_cmap=True)

b=df3.style.background_gradient(cmap=cm)
b
```

	W (dyne)	mg (g)
0	100000	102.040000
1	250000	225.102000

```
#find the ratio mg1/mg2
mi1=102.040
mi2=225.102
ratio2= mi1/mi2
r2=round(ratio2,3)
print(r2,"g")
```

```
#find the ratio mi1/mi2
mi1=150
mi2=400
ratio= mi1/mi2
r=round(ratio,3)
print(r,"g")
```

```
df['t2'] = df.apply(lambda row: row.t_10**2, axis =1)
print(df)
```

```
df['t_10']= df.apply(lambda row: row.t10/10, axis= 1)
print(df)
```

```
[14] import seaborn as sns

     cm=sns.light_palette("blue",as_cmap=True)

     b=df.style.background_gradient(cmap=cm)
     b
```

```
xpoints = np.array([100, 600])
ypoints = np.array([0, 0.50])
plt.xlabel("Mass(g)")
plt.ylabel("Time(s^2)")
plt.title("Mass vs time^2", fontsize=20)
plt.plot(xpoints, ypoints, color='hotpink')
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([150,200,300,400,500,600])
ypoints = np.array([0.14,0.2,0.23,0.27,0.36,0.49])
plt.xlabel("Mass(g)")
plt.ylabel("Time(s^2)")
plt.title("Mass vs time^2", fontsize=20)
plt.plot(xpoints, ypoints,'o', ms=10, mec='hotpink', mfc='hotpink')
plt.show()
```