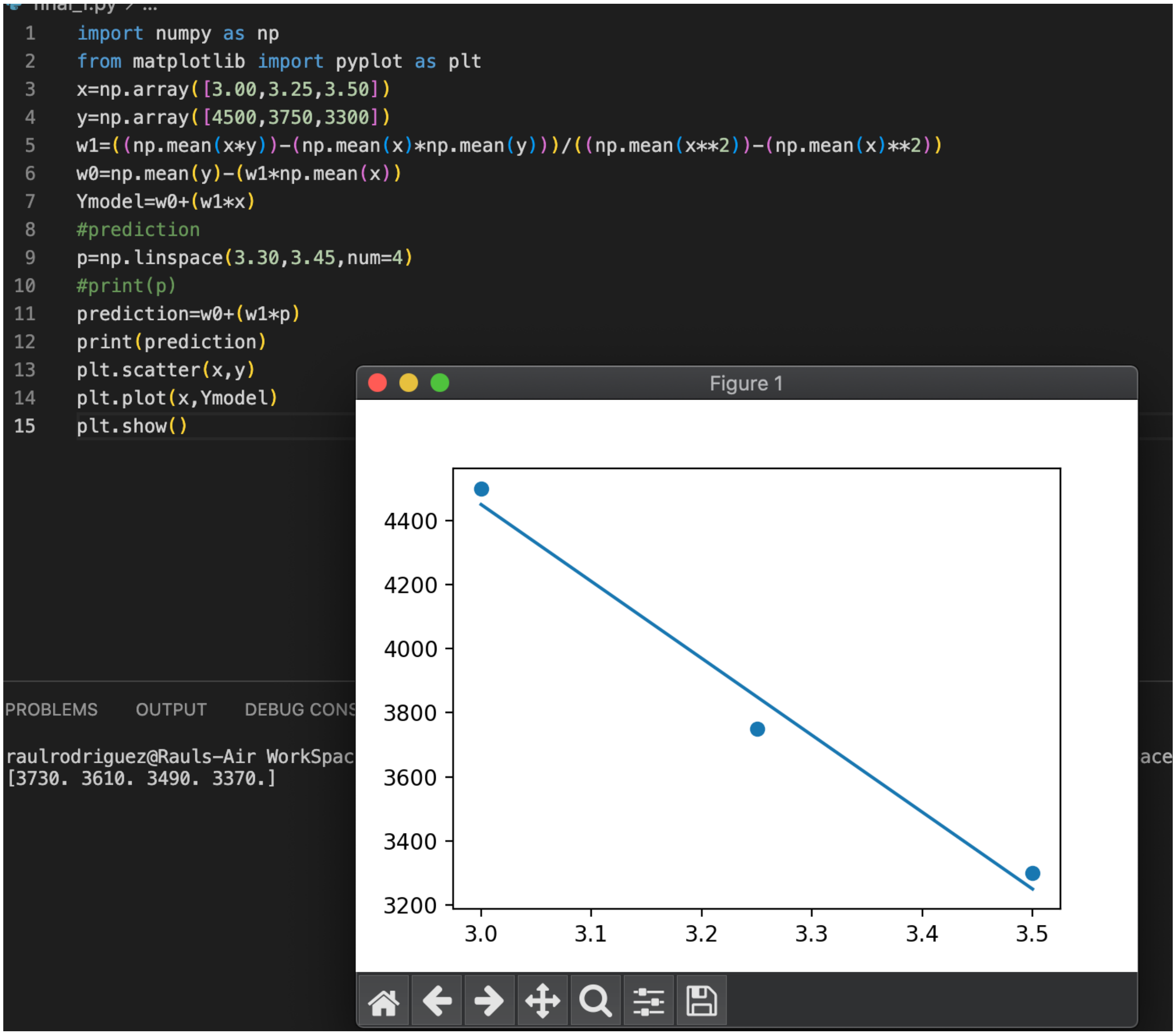
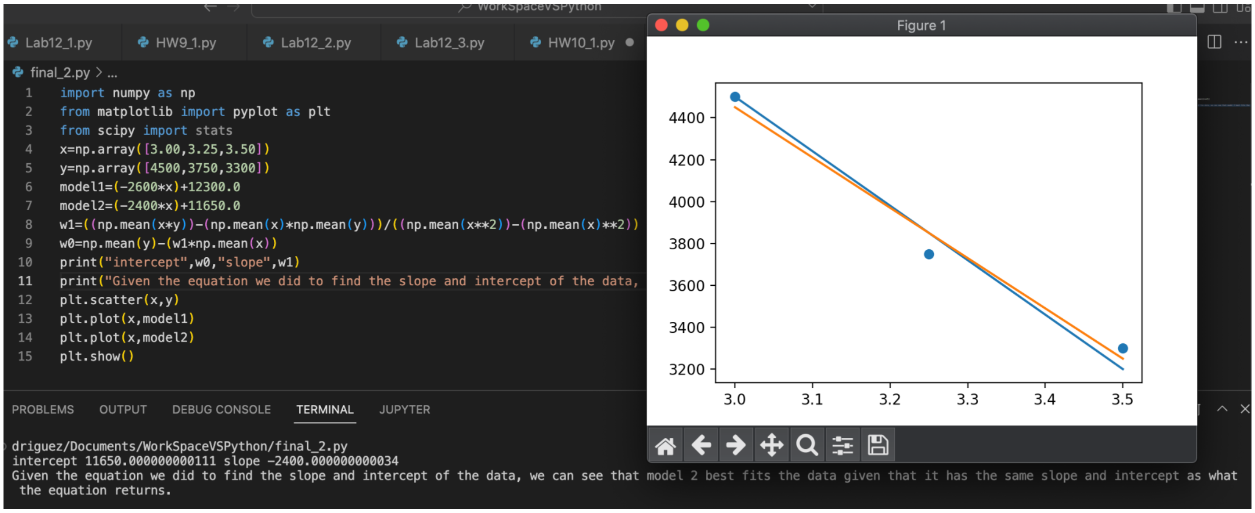
Raul Rodriguez



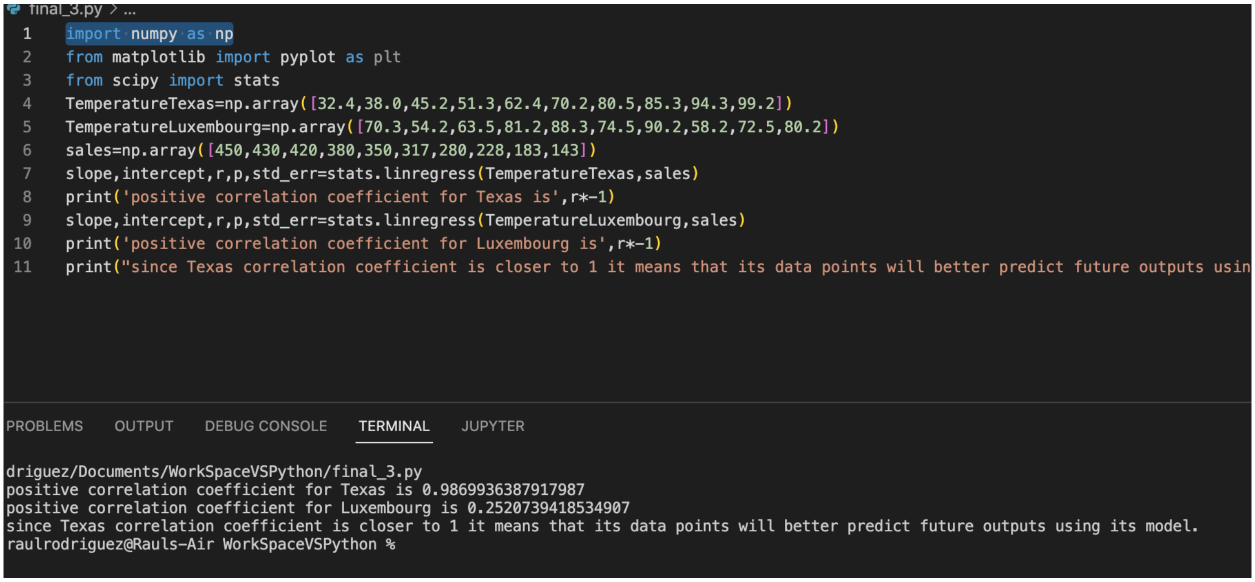
```python
import numpy as np
from matplotlib import pyplot as plt
x=np.array([3.00,3.25,3.50])
y=np.array([4500,3750,3300])
w1=((np.mean(x*y))-(np.mean(x)*np.mean(y)))/((np.mean(x**2))-(np.mean(x)**2))
w0=np.mean(y)-(w1*np.mean(x))
Ymodel=w0+(w1*x)
#prediction
p=np.linspace(3.30,3.45,num=4)
#print(p)
prediction=w0+(w1*p)
print(prediction)
plt.scatter(x,y)
plt.plot(x,Ymodel)
plt.show()
```

```
raulrodriguez@Rauls-Air WorkSpac                                    ace
[3730. 3610. 3490. 3370.]
```



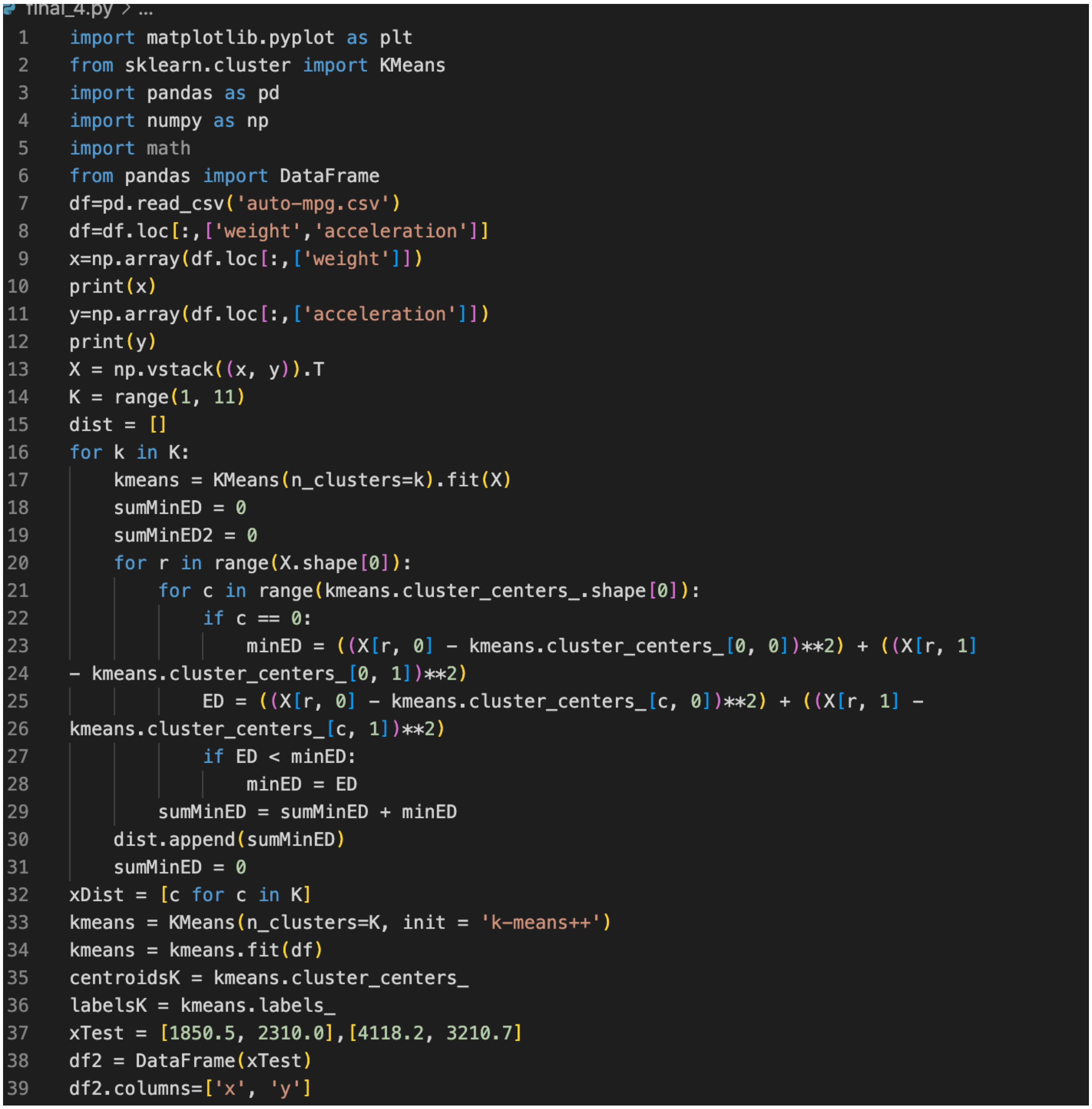
```python
import numpy as np
from matplotlib import pyplot as plt
from scipy import stats
x=np.array([3.00,3.25,3.50])
y=np.array([4500,3750,3300])
model1=(-2600*x)+12300.0
model2=(-2400*x)+11650.0
w1=((np.mean(x*y))-(np.mean(x)*np.mean(y)))/((np.mean(x**2))-(np.mean(x)**2))
w0=np.mean(y)-(w1*np.mean(x))
print("intercept",w0,"slope",w1)
print("Given the equation we did to find the slope and intercept of the data,
plt.scatter(x,y)
plt.plot(x,model1)
plt.plot(x,model2)
plt.show()
```

```
driguez/Documents/WorkSpaceVSPython/final_2.py
intercept 11650.000000000111 slope -2400.000000000034
Given the equation we did to find the slope and intercept of the data, we can see that model 2 best fits the data given that it has the same slope and intercept as what
 the equation returns.
```

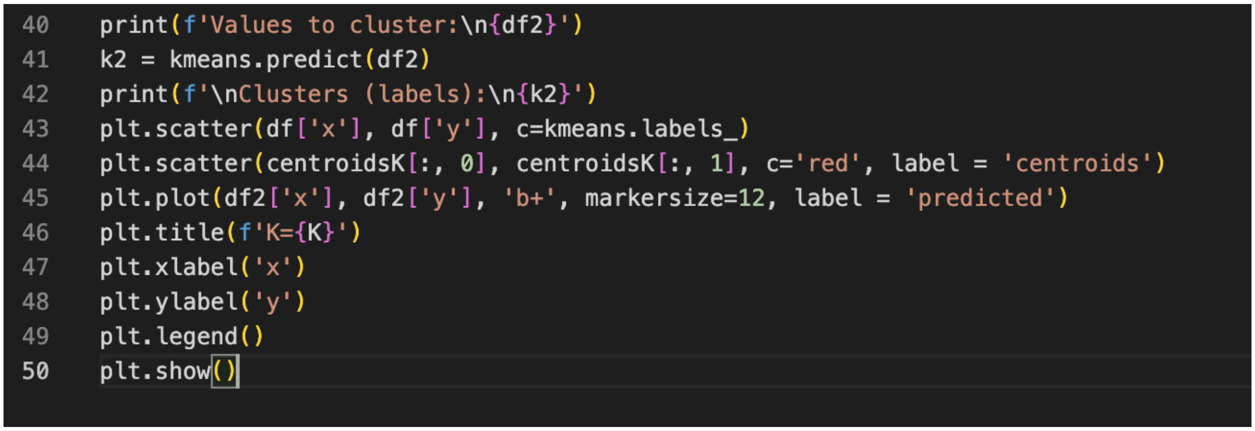


```python
import numpy as np
from matplotlib import pyplot as plt
from scipy import stats
TemperatureTexas=np.array([32.4,38.0,45.2,51.3,62.4,70.2,80.5,85.3,94.3,99.2])
TemperatureLuxembourg=np.array([70.3,54.2,63.5,81.2,88.3,74.5,90.2,58.2,72.5,80.2])
sales=np.array([450,430,420,380,350,317,280,228,183,143])
slope,intercept,r,p,std_err=stats.linregress(TemperatureTexas,sales)
print('positive correlation coefficient for Texas is',r*-1)
slope,intercept,r,p,std_err=stats.linregress(TemperatureLuxembourg,sales)
print('positive correlation coefficient for Luxembourg is',r*-1)
print("since Texas correlation coefficient is closer to 1 it means that its data points will better predict future outputs usin
```

```
driguez/Documents/WorkSpaceVSPython/final_3.py
positive correlation coefficient for Texas is 0.9869936387917987
positive correlation coefficient for Luxembourg is 0.2520739418534907
since Texas correlation coefficient is closer to 1 it means that its data points will better predict future outputs using its model.
raulrodriguez@Rauls-Air WorkSpaceVSPython %
```



```python
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import pandas as pd
import numpy as np
import math
from pandas import DataFrame
df=pd.read_csv('auto-mpg.csv')
df=df.loc[:,['weight','acceleration']]
x=np.array(df.loc[:,['weight']])
print(x)
y=np.array(df.loc[:,['acceleration']])
print(y)
X = np.vstack((x, y)).T
K = range(1, 11)
dist = []
for k in K:
    kmeans = KMeans(n_clusters=k).fit(X)
    sumMinED = 0
    sumMinED2 = 0
    for r in range(X.shape[0]):
        for c in range(kmeans.cluster_centers_.shape[0]):
            if c == 0:
                minED = ((X[r, 0] - kmeans.cluster_centers_[0, 0])**2) + ((X[r, 1]
- kmeans.cluster_centers_[0, 1])**2)
            ED = ((X[r, 0] - kmeans.cluster_centers_[c, 0])**2) + ((X[r, 1] -
kmeans.cluster_centers_[c, 1])**2)
            if ED < minED:
                minED = ED
        sumMinED = sumMinED + minED
    dist.append(sumMinED)
    sumMinED = 0
xDist = [c for c in K]
kmeans = KMeans(n_clusters=K, init = 'k-means++')
kmeans = kmeans.fit(df)
centroidsK = kmeans.cluster_centers_
labelsK = kmeans.labels_
xTest = [1850.5, 2310.0],[4118.2, 3210.7]
df2 = DataFrame(xTest)
df2.columns=['x', 'y']
```



```python
print(f'Values to cluster:\n{df2}')
k2 = kmeans.predict(df2)
print(f'\nClusters (labels):\n{k2}')
plt.scatter(df['x'], df['y'], c=kmeans.labels_)
plt.scatter(centroidsK[:, 0], centroidsK[:, 1], c='red', label = 'centroids')
plt.plot(df2['x'], df2['y'], 'b+', markersize=12, label = 'predicted')
plt.title(f'K={K}')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```

Output not working on exercise 4.