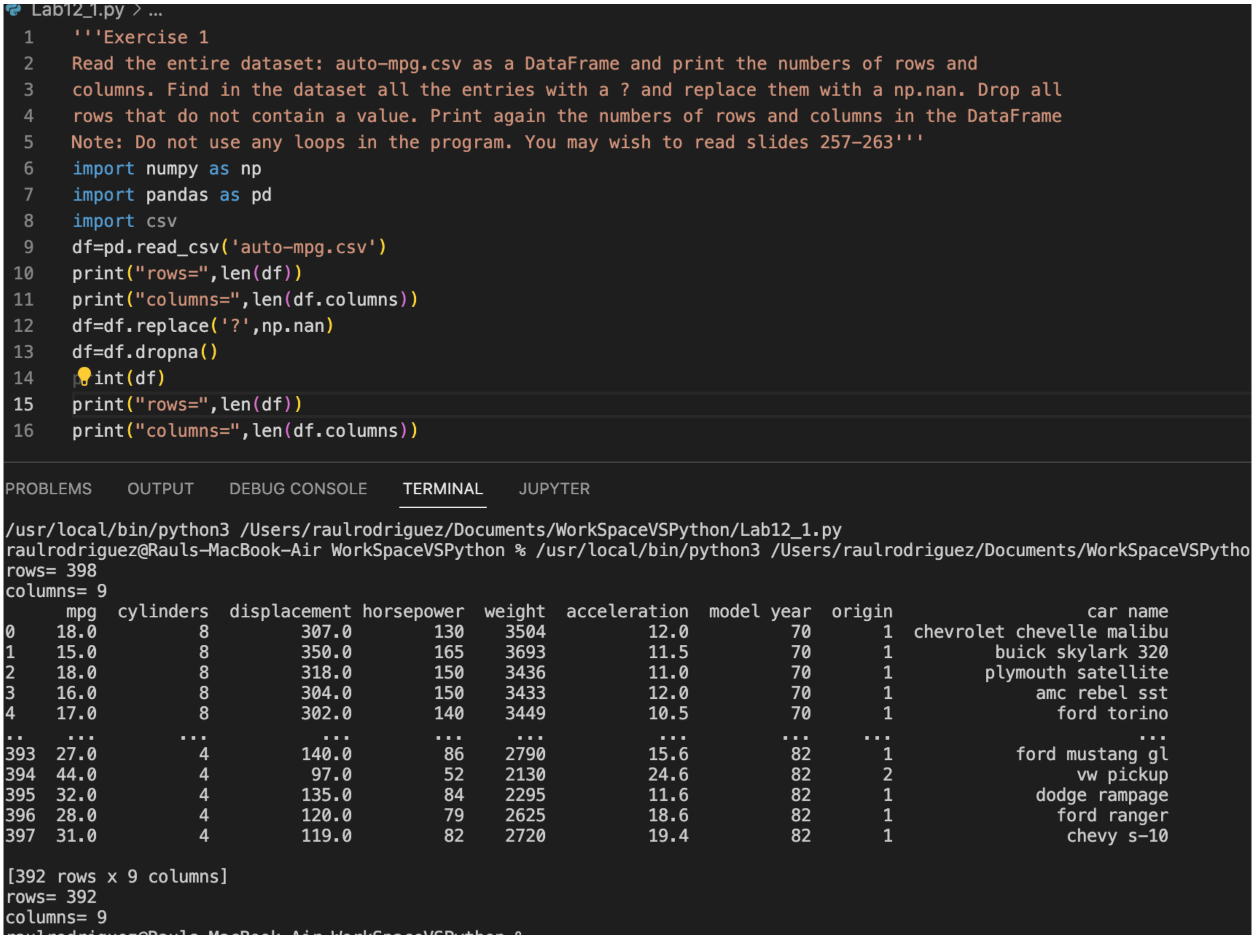
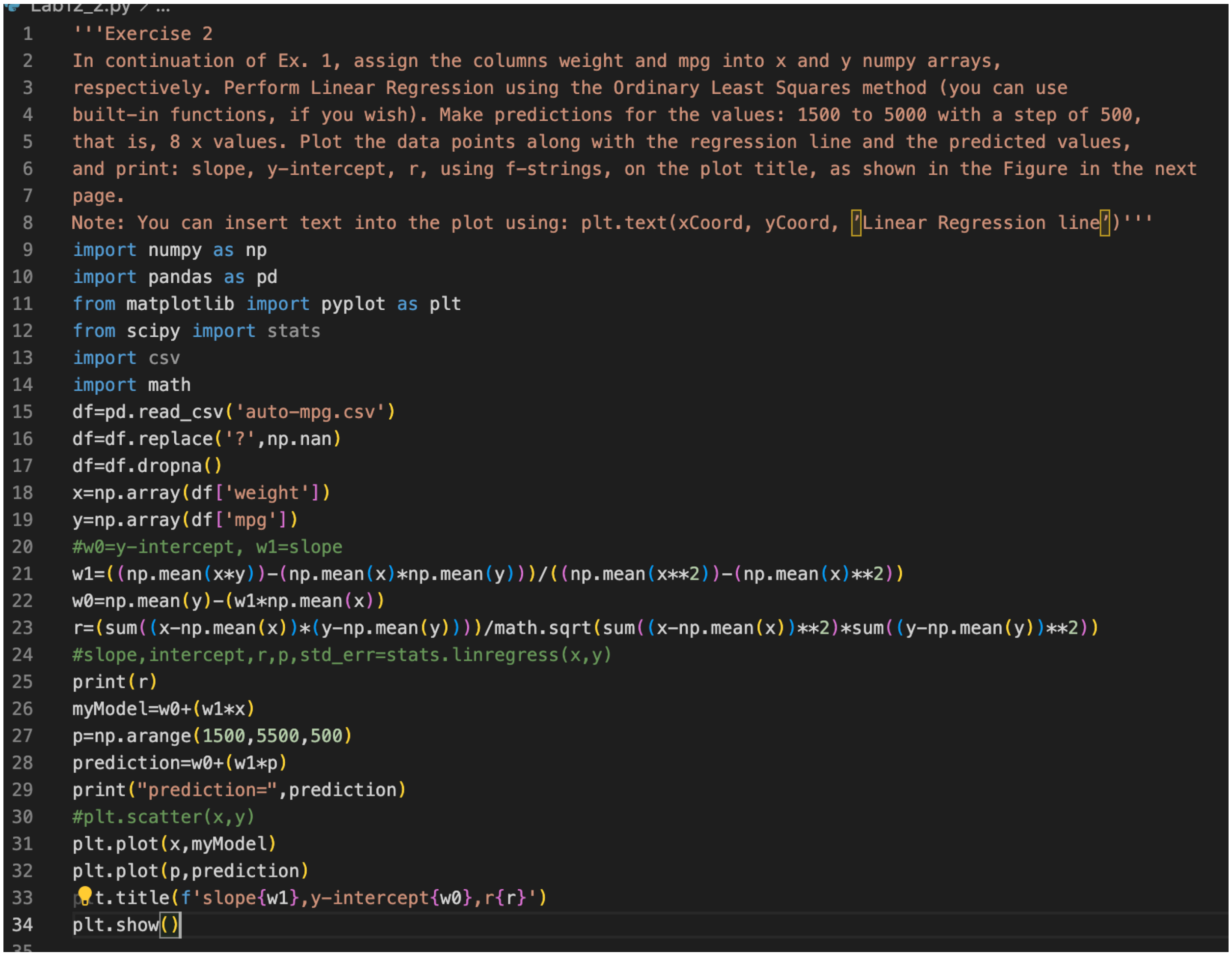
Raul Rodriguez



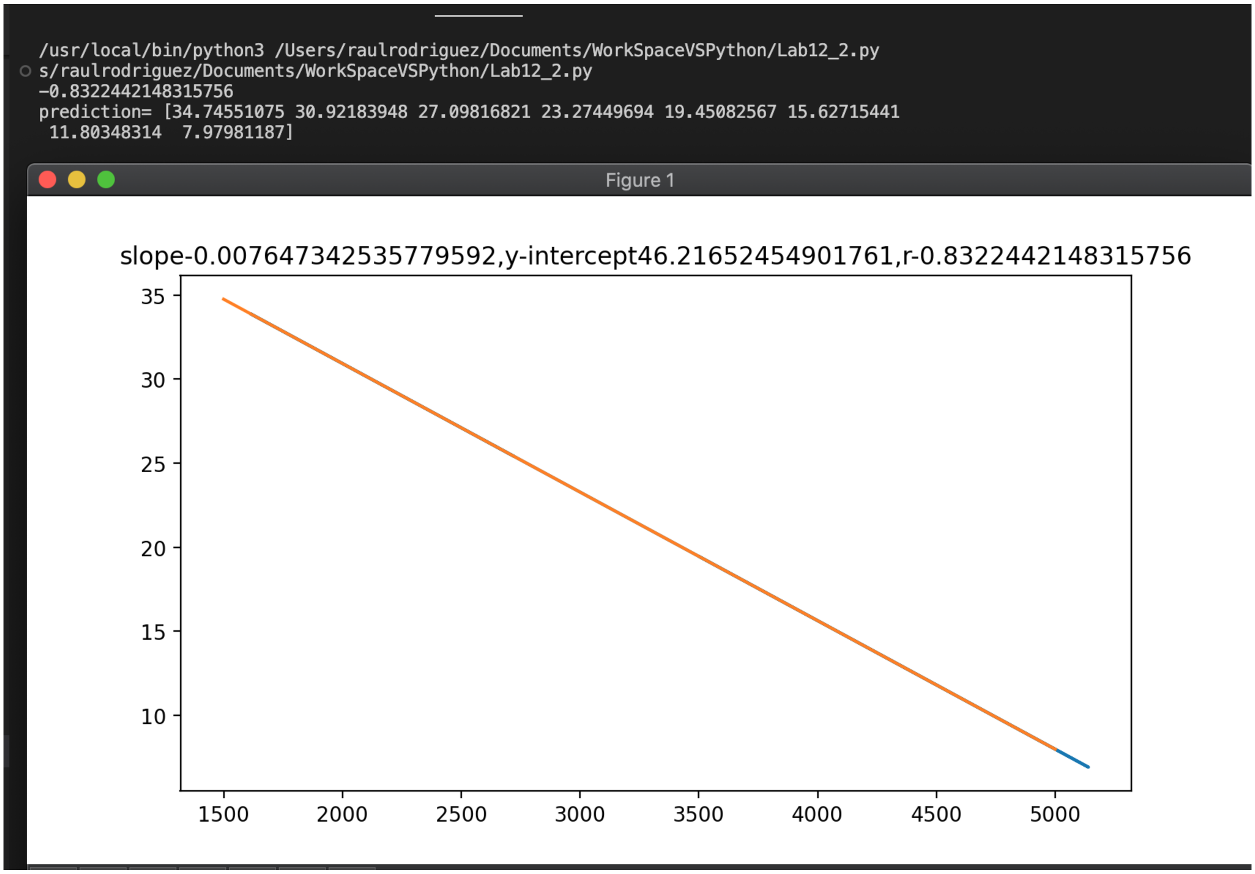
```python
'''Exercise 1
Read the entire dataset: auto-mpg.csv as a DataFrame and print the numbers of rows and
columns. Find in the dataset all the entries with a ? and replace them with a np.nan. Drop all
rows that do not contain a value. Print again the numbers of rows and columns in the DataFrame
Note: Do not use any loops in the program. You may wish to read slides 257-263'''
import numpy as np
import pandas as pd
import csv
df=pd.read_csv('auto-mpg.csv')
print("rows=",len(df))
print("columns=",len(df.columns))
df=df.replace('?',np.nan)
df=df.dropna()
print(df)
print("rows=",len(df))
print("columns=",len(df.columns))
```

```
/usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/Lab12_1.py
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPytho
rows= 398
columns= 9
      mpg  cylinders  displacement horsepower  weight  acceleration  model year  origin                   car name
0    18.0          8         307.0        130    3504          12.0          70       1  chevrolet chevelle malibu
1    15.0          8         350.0        165    3693          11.5          70       1          buick skylark 320
2    18.0          8         318.0        150    3436          11.0          70       1         plymouth satellite
3    16.0          8         304.0        150    3433          12.0          70       1              amc rebel sst
4    17.0          8         302.0        140    3449          10.5          70       1                ford torino
..    ...        ...           ...        ...     ...           ...         ...     ...                        ...
393  27.0          4         140.0         86    2790          15.6          82       1            ford mustang gl
394  44.0          4          97.0         52    2130          24.6          82       2                  vw pickup
395  32.0          4         135.0         84    2295          11.6          82       1              dodge rampage
396  28.0          4         120.0         79    2625          18.6          82       1                ford ranger
397  31.0          4         119.0         82    2720          19.4          82       1                 chevy s-10

[392 rows x 9 columns]
rows= 392
columns= 9
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```

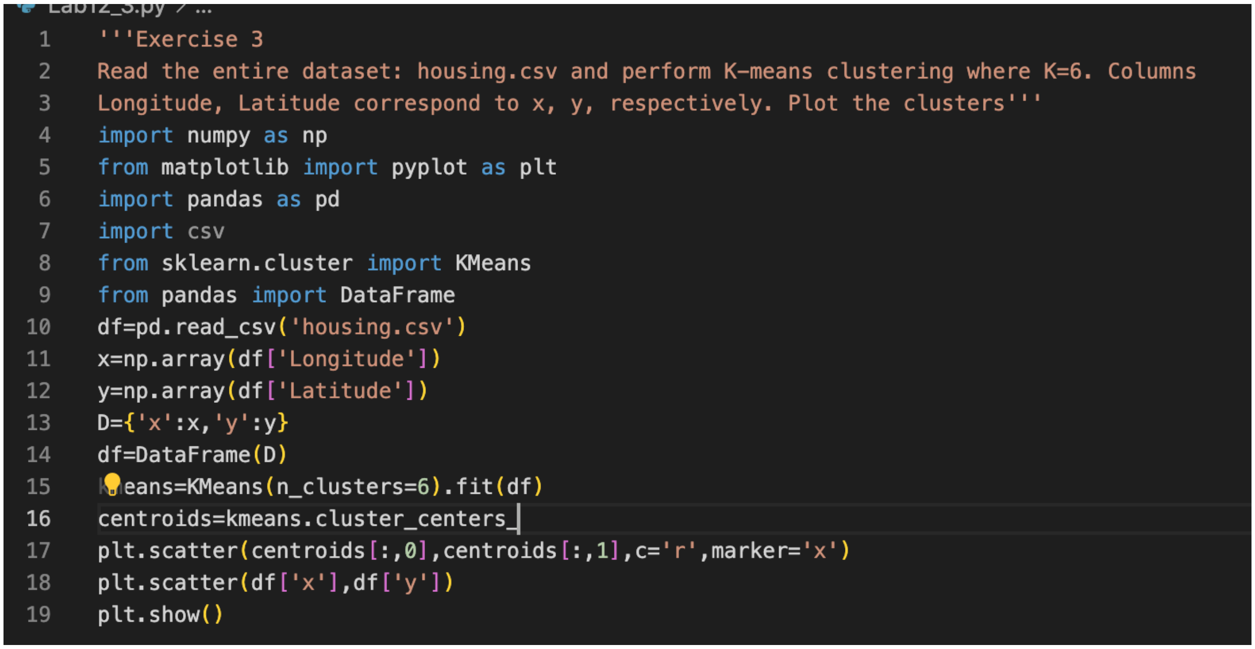


```
'''Exercise 2
In continuation of Ex. 1, assign the columns weight and mpg into x and y numpy arrays,
respectively. Perform Linear Regression using the Ordinary Least Squares method (you can use
built-in functions, if you wish). Make predictions for the values: 1500 to 5000 with a step of 500,
that is, 8 x values. Plot the data points along with the regression line and the predicted values,
and print: slope, y-intercept, r, using f-strings, on the plot title, as shown in the Figure in the next
page.
Note: You can insert text into the plot using: plt.text(xCoord, yCoord, 'Linear Regression line')'''
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from scipy import stats
import csv
import math
df=pd.read_csv('auto-mpg.csv')
df=df.replace('?',np.nan)
df=df.dropna()
x=np.array(df['weight'])
y=np.array(df['mpg'])
#w0=y-intercept, w1=slope
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))
r=(sum((x-np.mean(x))*(y-np.mean(y))))/math.sqrt(sum((x-np.mean(x))**2)*sum((y-np.mean(y))**2))
#slope,intercept,r,p,std_err=stats.linregress(x,y)
print(r)
myModel=w0+(w1*x)
p=np.arange(1500,5500,500)
prediction=w0+(w1*p)
print("prediction=",prediction)
#plt.scatter(x,y)
plt.plot(x,myModel)
plt.plot(p,prediction)
plt.title(f'slope{w1},y-intercept{w0},r{r}')
plt.show()
```



```
/usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/Lab12_2.py
s/raulrodriguez/Documents/WorkSpaceVSPython/Lab12_2.py
-0.8322442148315756
prediction= [34.74551075 30.92183948 27.09816821 23.27449694 19.45082567 15.62715441
 11.80348314  7.97981187]
```

Figure 1
slope-0.007647342535779592,y-intercept46.21652454901761,r-0.8322442148315756
35
30
25
20
15
10
1500
2000
2500
3000
3500
4000
4500
5000



```
'''Exercise 3
Read the entire dataset: housing.csv and perform K-means clustering where K=6. Columns
Longitude, Latitude correspond to x, y, respectively. Plot the clusters'''
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
import csv
from sklearn.cluster import KMeans
from pandas import DataFrame
df=pd.read_csv('housing.csv')
x=np.array(df['Longitude'])
y=np.array(df['Latitude'])
D={'x':x,'y':y}
df=DataFrame(D)
kmeans=KMeans(n_clusters=6).fit(df)
centroids=kmeans.cluster_centers_
plt.scatter(centroids[:,0],centroids[:,1],c='r',marker='x')
plt.scatter(df['x'],df['y'])
plt.show()
```

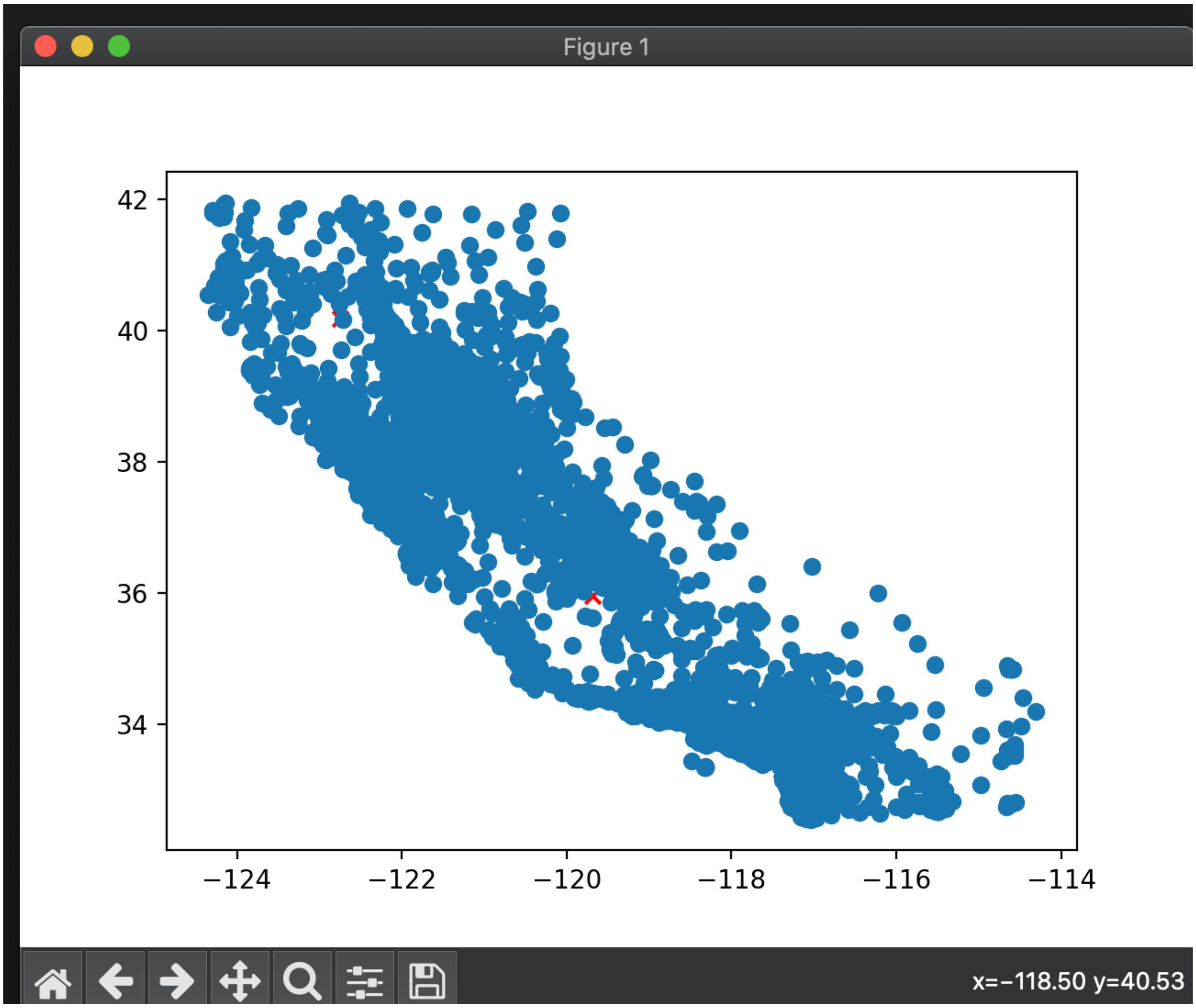


Figure 1
42
40
38
36
34
−124
−122
−120
−118
−116
−114
x=−118.50 y=40.53