Raul Rodriguez

```python
'''Exercise 1
Given the dataset: (-5, -10), (-1, -3), (3, 5), (7, 8), (5, 7), (9, 9), find the regression line using
Matrix Algebra Least Squares (slides 323-325). Find the parameters of the line (slope and y-
intercept) and plot the data points along with the regression line
Note 1: You can use the x.transpose() method to transpose a matrix, the np.matmul(x, y)
method to multiply matrices x and y, and the np.linalg.inv(x) to find the inverse of a matrix
Note 2: Your algorithm should be able to work with any dataset not just a dataset of 5 data
points'''
import numpy as np
from matplotlib import pyplot as plt
x = [-5, -1, 3, 7, 5, 9]
y = [-10, -3, 5, 8, 7, 9]
n=len(x)
x1=np.ones((n),dtype=int).reshape(n,1)
x2=np.array(x).reshape(n,1)
xArr=np.hstack((x1,x2))
yArr=np.array(y).reshape(n,1)
xArrT=xArr.transpose()
yArrT=yArr.transpose()
xMul=np.matmul(xArrT,xArr)
xyMul=np.matmul(xArrT,yArr)
xMulInv=np.linalg.inv(xMul)
A=np.matmul(xMulInv,xyMul)
print(A)
b=A[0]
m=A[1]
Ymodel=b+(m*x2)
plt.scatter(x2,yArr)
plt.plot(x2,Ymodel,color='r')
plt.title("Least Squares Regression Line")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
```

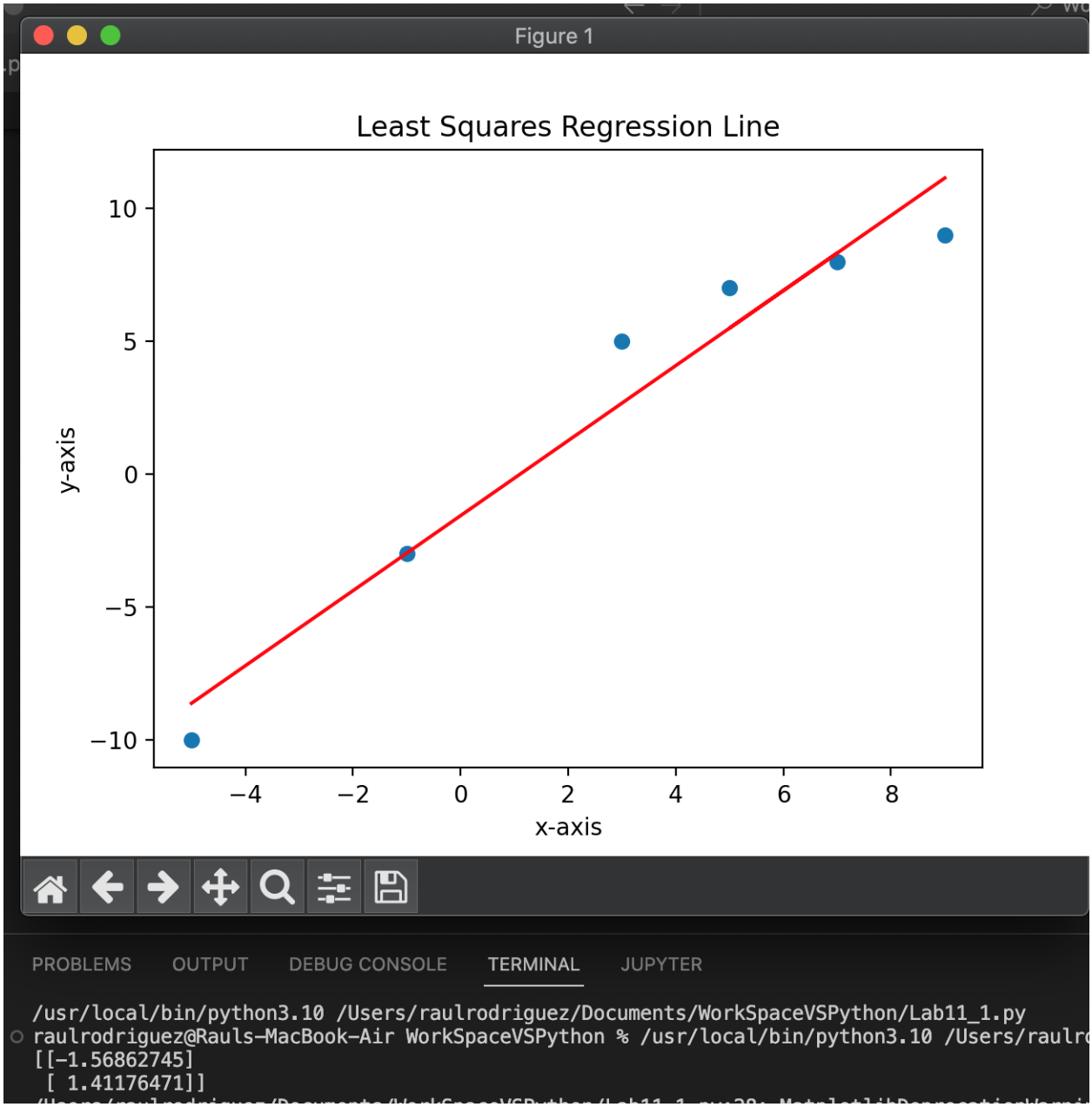
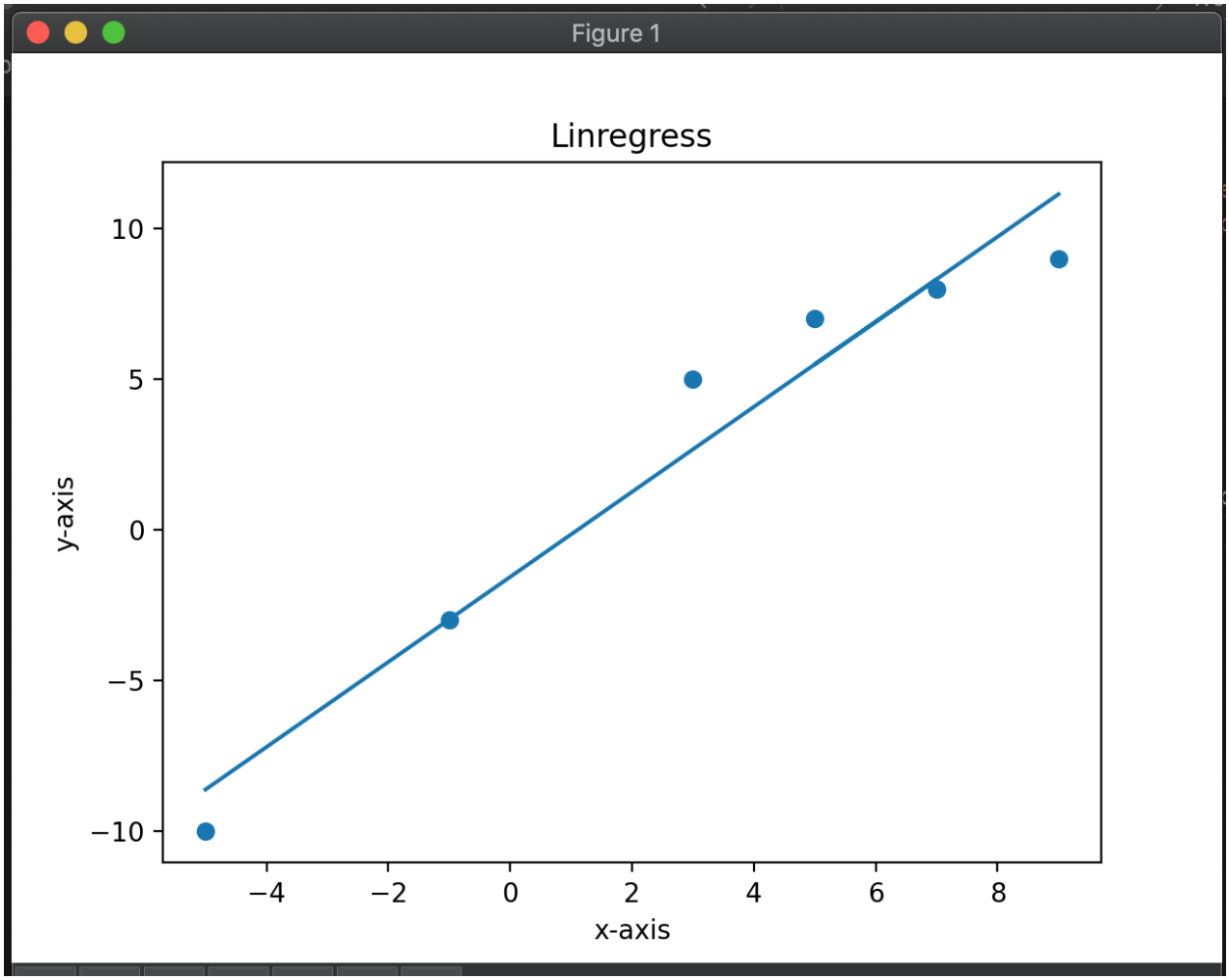


Figure 1
Least Squares Regression Line
y-axis
x-axis
10
5
0
−5
−10
−4
−2
0
2
4
6
8
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
JUPYTER
/usr/local/bin/python3.10 /Users/raulrodriguez/Documents/WorkSpaceVSPython/Lab11_1.py
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3.10 /Users/raulr
[[-1.56862745]
[1.41176471]]

```python
'''Exercise 2
Based on the dataset from Ex. 1, find the regression line using Ordinary Least Squares
(slides 326-328). You can use the mean() built-in function'''
import numpy as np
from matplotlib import pyplot as plt
x = np.array([-5, -1, 3, 7, 5, 9])
y = np.array([-10, -3, 5, 8, 7, 9])
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)
plt.scatter(x,y)
plt.plot(x,Ymodel)
plt.title("Ordinary Least Sauares")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
plt.show()
```

```python
'''Exercise 3
Based on the dataset from Ex. 1, find the regression line using the built-in linregress()
method (slides 331-334). Plot the data points along with the regression line and print: slope,
y-intercept, r (correlation coefficient), p-value, standard error'''
import numpy as np
from scipy import stats
from matplotlib import pyplot as plt
x = np.array([-5, -1, 3, 7, 5, 9])
y = np.array([-10, -3, 5, 8, 7, 9])
slope,intercept,r,p,std_err=stats.linregress(x,y)
print('Slope: ',slope,'y-intercept: ',intercept,'Correlation(r): ',r,'p-value: ',p,'Standard Error: ',std_err)
mymodel=(slope*x)+intercept
plt.scatter(x,y)
plt.plot(x,mymodel)
plt.title("Linregress")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
```



```python
'''Exercise 4
Based on the dataset from Ex. 1, compute the: SST, SSR, SSE (slide 336) and the: r, R2 (slide
337). You can use the sum() and mean() built-in functions'''
import numpy as np
from scipy import stats
from matplotlib import pyplot as plt
x = np.array([-5, -1, 3, 7, 5, 9])
y = np.array([-10, -3, 5, 8, 7, 9])
slope,intercept,r,p,std_err=stats.linregress(x,y)
mymodel=(slope*x)+intercept
SST=sum((y-np.mean(y))**2)
print(SST)
SSR=sum((mymodel-np.mean(y))**2)
print(SSR)
SSE=sum((mymodel-y)**2)
print(SSE)
R2=1-(SSE/SST)
r=R2**(1.0/2)
print(r)
print(R2)
```

```
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/loca
285.33333333333337
271.0588235294118
14.274509803921571
0.9746653335219396
0.9499725123694338
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```

```python
'''Exercise 5
Given the grades.csv file, where the first column is hours studied and the second column is
grades, read its data and place them into two numpy arrays. Find the regression line using the
built-in linregress() method (slides 331-334). Plot the data points along with the regression line
and print: slope, y-intercept, r (correlation coefficient), p-value, standard error. Predict a student's
grade given 10.5 hours of study'''
import numpy as np
from scipy import stats
import pandas as pd
from matplotlib import pyplot as plt
import csv
xLi=[]
yLi=[]
with open('grades.csv', 'r', newline='') as grades:
    reader = csv.reader(grades)
    for record in reader:
        hours,grades=record
        xLi.append(float (hours))
        yLi.append(float (grades))
print(xLi)
print(yLi)
x=np.array(xLi)
y=np.array(yLi)
slope,intercept,r,p,std_err=stats.linregress(x,y)
print('Slope: ',slope,'y-intercept: ',intercept,'Correlation(r): ',r,'p-value: ',p,'Standard Error: ',std_err)
myModel=(slope*x)+intercept
p=(slope*10.5)+intercept
print(f'predicted grade is {p}')
plt.scatter(x,y)
plt.plot(x,myModel)
plt.show()
```