

```

1 import csv
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 # declare variables
6 nextVal = None
7 x = 0
8 y = 0
9 sumX = 0
10 sumY = 0
11 sumProdXY = 0
12 sumSqX = 0
13 sumSqY = 0
14 numPoints = 0
15 slope = 0
16 intercept = 0
17 maxX = 0
18 data = []
19
20 # open and read downloads file
21 path = r"C:\Users\jeffp\OneDrive\Documents\GitHub\Machine-Learning-Project-1\downloads.txt"
22 file = open(path)
23 reader = csv.reader(file)
24
25 # populate two dimensional list (data) with entries
26 for row in reader:
27     hour = int(row[0])
28     if row[1] != 'nan':
29         downloads = int(row[1])
30     else:
31         downloads = None
32     data.append([hour, downloads])
33
34 file.close()
35
36 # replace missing values with mean of two adjacent valid values
37 for i in range(len(data)):
38     if data[i][1] is None and i != 0 and i != (len(data) - 1) and data[i - 1][1] is not None:
39         j = 1
40         while data[i + j][1] is None:
41             j += 1
42             if (i + j) > (len(data) - 1):
43                 break
44             else:
45                 continue
46             break
47         if (i + j) <= (len(data) - 1):
48             nextVal = data[i + j][1]
49             data[i][1] = (nextVal + data[i-1][1]) // 2
50
51 # linear regression using simple least squares method
52 for i in range(len(data)):
53     if data[i][1] is not None:
54         curX = data[i][0]
55         curY = data[i][1]
56         sumX += curX
57         sumY += curY
58         sumProdXY += curX * curY
59         sumSqX += curX ** 2
60         sumSqY += curY ** 2
61         numPoints += 1
62         if data[i][0] > maxX:

```

```

63         maxX = data[i][0]
64
65 slope = ((numPoints * sumProdXY) - (sumX * sumY)) / ((numPoints * sumSqX) - (sumX ** 2))
66 intercept = (sumY - (slope * sumX)) / numPoints
67
68 x = np.array(range(0, maxX))
69 y = intercept + (slope * x)
70
71 # graph data and trend line
72 plt.plot(x, y, 'red')
73 plt.scatter(*zip(*data))
74 plt.title('Previous Month Downloads')
75 plt.xlabel('Hour')
76 plt.ylabel('Downloads')
77 plt.show()
78
79 # estimate downloads on noon of fifth day of following month
80 x = maxX + (24 * 4) + 12
81 y = intercept + (slope * x)
82 print(round(y))

```