

```

4 # Equation from slides 178-181. You can use the built-in function
5 import pandas as pd
6 import numpy as np
7
8 df = pd.read_csv('auto-mpg.csv')
9 df=df.replace('?',np.nan)
10 df=df.dropna()
11 df=df.drop("origin",axis=1)
12 df=df.drop("car name",axis=1)
13 df=df.astype(float)
14 #print(df)
15 #print(df.dtypes)
16 y=np.array(df['mpg'])
17 x=np.array(df.loc[:,'cylinders':'model year'])
18 #print(y)
19 n=len(x)
20 x1=np.ones((n),dtype=int).reshape(n,1)
21 xArr=np.hstack((x1,x))
22 yArr=np.array(y).reshape(n,1)
23 xArrT=xArr.transpose()
24 #print(xArrT)
25 yArrT=yArr.transpose()
26 xMul=np.matmul(xArrT,xArr)
27 #print(xMul)
28 xyMul=np.matmul(xArrT,yArr)
29 xMulInv=np.linalg.inv(xMul)
30 A=np.matmul(xMulInv,xyMul)
31 #print(A)
32 intercept=A[0]
33 print(f'coefficients= {A[1:7]}, intercept= {intercept}')
34

```

- raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/bin/python3 -c "from sklearn.linear\_model import LinearRegression; from sklearn.preprocessing import StandardScaler; import pandas as pd; df = pd.read\_csv('auto-mpg.csv'); df = df.replace('?', np.nan); df = df.dropna(); df = df.drop('origin', axis=1); df = df.drop('car name', axis=1); df = df.astype(float); print(df); y = np.array(df['mpg']); x = np.array(df.loc[:, 'cylinders': 'model year']); xScaled = StandardScaler().fit\_transform(x); #print(xScaled); li = []; for i in range(6): slope, intercept, r, p, std\_error = stats.linregress(xScaled[:, i], y); li.append(r); li = np.array(li); li = abs(li); li = list(li); print(li); sortedLi = sorted(li, reverse=True); #print(type(sortedLi)); topThree = sortedLi[0:3]; #print(topThree); index1 = li.index(topThree[0]); index2 = li.index(topThree[1]); index3 = li.index(topThree[2]); print(index1, index2, index3); print(f'most important features are: {df.columns[index1+1]}, {df.columns[index2+1]}, {df.columns[index3+1]}')"
- raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %

```

4 import pandas as pd
5 import numpy as np
6 from sklearn.linear_model import LinearRegression
7 from sklearn.preprocessing import StandardScaler
8 from scipy import stats
9
10 df = pd.read_csv('auto-mpg.csv')
11 df=df.replace('?',np.nan)
12 df=df.dropna()
13 df=df.drop("origin",axis=1)
14 df=df.drop("car name",axis=1)
15 df=df.astype(float)
16 print(df)
17 y=np.array(df['mpg'])
18 x=np.array(df.loc[:, 'cylinders': 'model year'])
19 xScaled = StandardScaler().fit_transform(x)
20 #print(xScaled)
21 li=[]
22 for i in range(6):
23     slope, intercept, r, p, std_error = stats.linregress(xScaled[:,i], y)
24     li.append(r)
25 li=np.array(li)
26 li=abs(li)
27 li=list(li)
28 print(li)
29 sortedLi=sorted(li,reverse=True)
30 #print(type(sortedLi))
31 topThree=sortedLi[0:3]
32 #print(topThree)
33 index1=li.index(topThree[0])
34 index2=li.index(topThree[1])
35 index3=li.index(topThree[2])
36 print(index1,index2,index3)
37 print(f'most important features are: {df.columns[index1+1]}, {df.columns[index2+1]}, {df.columns[index3+1]}')

```

```

/usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/Lab6_2ML.py
● raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython
    mpg    cylinders  displacement  horsepower  weight  acceleration  model  year
0     18.0         8.0         307.0         130.0   3504.0          12.0    70.0
1     15.0         8.0         350.0         165.0   3693.0          11.5    70.0
2     18.0         8.0         318.0         150.0   3436.0          11.0    70.0
3     16.0         8.0         304.0         150.0   3433.0          12.0    70.0
4     17.0         8.0         302.0         140.0   3449.0          10.5    70.0
..     ...         ...         ...         ...         ...         ...     ...
393    27.0         4.0         140.0          86.0   2790.0          15.6    82.0
394    44.0         4.0          97.0          52.0   2130.0          24.6    82.0
395    32.0         4.0         135.0          84.0   2295.0          11.6    82.0
396    28.0         4.0         120.0          79.0   2625.0          18.6    82.0
397    31.0         4.0         119.0          82.0   2720.0          19.4    82.0

[392 rows x 7 columns]
[0.7776175081260216, 0.805126946710458, 0.7784267838977761, 0.8322442148315753, 0.42332853690278727, 0.5805409660907854]
3 1 2
most important features are: weight, displacement, horsepower

```

```

3 goodness of fit of the model (you will have to use R2)
4 import pandas as pd
5 import numpy as np
6 from scipy import stats
7
8 df = pd.read_csv('auto-mpg.csv')
9 df=df.replace('?',np.nan)
10 df=df.dropna()
11 df=df.drop("origin",axis=1)
12 df=df.drop("car name",axis=1)
13 df=df.astype(float)
14 print(df)
15 y=np.array(df['mpg'])
16 x=np.array(df.loc[:, 'cylinders':'model year'])
17
18 y = np.array(df['mpg'])
19 x1 = np.array(df['weight'])
20 x2 = np.array(df['displacement'])
21 x3 = np.array(df['horsepower'])
22 slope, intercept, r, p, std_error = stats.linregress(x1, y)
23 print('Slope: ', slope, 'y-intercept: ', intercept, 'r: ', r)
24 slope, intercept, r, p, std_error = stats.linregress(x2, y)
25 print('Slope: ', slope, 'y-intercept: ', intercept, 'r: ', r)
26 slope, intercept, r, p, std_error = stats.linregress(x3, y)
27 print('Slope: ', slope, 'y-intercept: ', intercept, 'r: ', r)

```

```

● raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/WorkSpaceVSPython/m1.py
mpg    cylinders  displacement  horsepower  weight  acceleration  model  year
0      18.0       8.0         307.0       130.0    3504.0         12.0   70.0
1      15.0       8.0         350.0       165.0    3693.0         11.5   70.0
2      18.0       8.0         318.0       150.0    3436.0         11.0   70.0
3      16.0       8.0         304.0       150.0    3433.0         12.0   70.0
4      17.0       8.0         302.0       140.0    3449.0         10.5   70.0
..      ...       ...         ...       ...      ...         ...   ...
393    27.0       4.0         140.0       86.0    2790.0         15.6   82.0
394    44.0       4.0          97.0       52.0    2130.0         24.6   82.0
395    32.0       4.0         135.0       84.0    2295.0         11.6   82.0
396    28.0       4.0         120.0       79.0    2625.0         18.6   82.0
397    31.0       4.0         119.0       82.0    2720.0         19.4   82.0

[392 rows x 7 columns]
Slope: -0.007647342535779578 y-intercept: 46.21652454901758 r: -0.8322442148315754
Slope: -0.06005142781220625 y-intercept: 35.12063593840391 r: -0.8051269467104579
Slope: -0.15784473335365348 y-intercept: 39.93586102117045 r: -0.7784267838977758
○ raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %

```

```

3 import pandas as pd
4 import numpy as np
5 from sklearn.linear_model import LinearRegression
6
7 df = pd.read_csv('auto-mpg.csv')
8 df=df.replace('?',np.nan)
9 df=df.dropna()
10 df=df.drop("origin",axis=1)
11 df=df.drop("car name",axis=1)
12 df=df.astype(float)
13 print(df)
14 #print(df.dtypes)
15 y=np.array(df['mpg'])
16 x=np.array(df.loc[:,'cylinders':'model year'])
17 p=np.array([8, 306, 129, 3508, 11, 70])
18
19 reg=LinearRegression()
20 reg.fit(x,y)
21 c=np.array(reg.coef_)
22 print(f'coefficients: {c}')
23 yIntercept=reg.intercept_
24 print(f'Intercept: {yIntercept}')
25
26 predict=0
27 for i in range(len(p)):
28     predict+=c[i]*p[i]
29
30 predict+=yIntercept
31 print(f'prediction with given data are\n{predict}')

```

```

● raulrodriguez@Rauls-MacBook-Air WorkspaceVSPython % /usr/local/bin/python3 /Users/ra
    mpg  cylinders  displacement  horsepower  weight  acceleration  model  year
0    18.0      8.0      307.0      130.0  3504.0      12.0    70.0
1    15.0      8.0      350.0      165.0  3693.0      11.5    70.0
2    18.0      8.0      318.0      150.0  3436.0      11.0    70.0
3    16.0      8.0      304.0      150.0  3433.0      12.0    70.0
4    17.0      8.0      302.0      140.0  3449.0      10.5    70.0
..     ...      ...      ...      ...      ...      ...      ...
393  27.0      4.0      140.0      86.0  2790.0      15.6    82.0
394  44.0      4.0      97.0      52.0  2130.0      24.6    82.0
395  32.0      4.0      135.0      84.0  2295.0      11.6    82.0
396  28.0      4.0      120.0      79.0  2625.0      18.6    82.0
397  31.0      4.0      119.0      82.0  2720.0      19.4    82.0

[392 rows x 7 columns]
coefficients: [-3.29859089e-01  7.67843024e-03 -3.91355574e-04 -6.79461791e-03
 8.52732469e-02  7.53367180e-01]
Intercept: -14.535250480506118
prediction with given data are
14.963180251333352
○ raulrodriguez@Rauls-MacBook-Air WorkspaceVSPython %

```