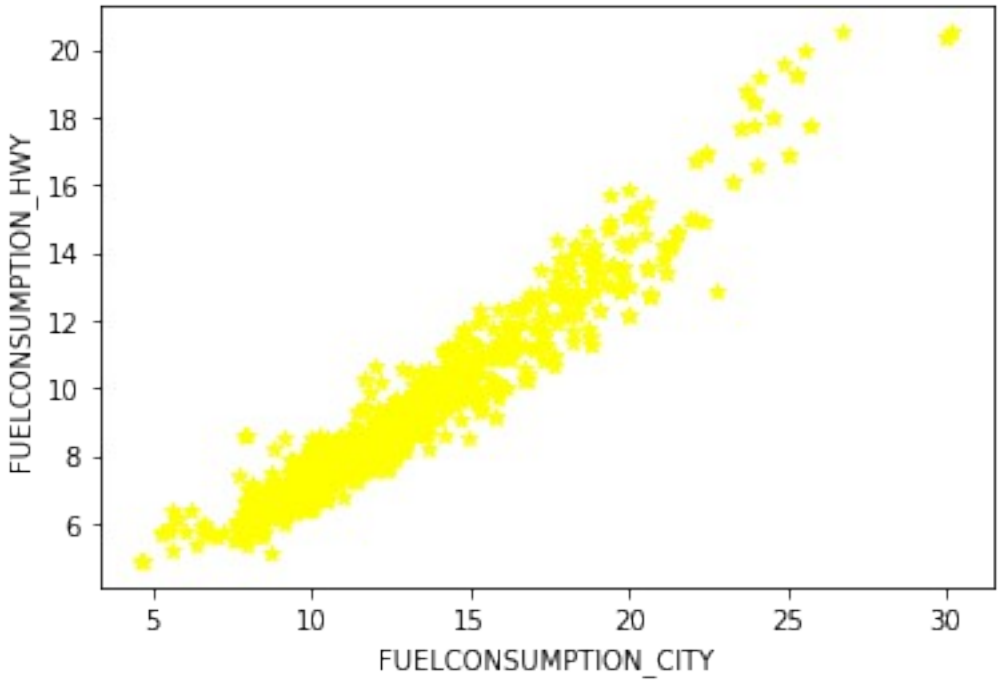
```
import pandas as pd
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt

#Read the Database file
df = pd.read_csv('FuelConsumption.csv')
df
```

```
      MODELYEAR   MAKE       MODEL    VEHICLECLASS  ENGINESIZE
CYLINDERS  \
0          2014  ACURA         ILX          COMPACT         2.0
4
1          2014  ACURA         ILX          COMPACT         2.4
4
2          2014  ACURA  ILX HYBRID          COMPACT         1.5
4
3          2014  ACURA     MDX 4WD     SUV - SMALL         3.5
6
4          2014  ACURA     RDX AWD     SUV - SMALL         3.5
6
...         ...    ...         ...              ...         ...
...
1062       2014  VOLVO    XC60 AWD     SUV - SMALL         3.0
6
1063       2014  VOLVO    XC60 AWD     SUV - SMALL         3.2
6
1064       2014  VOLVO    XC70 AWD     SUV - SMALL         3.0
6
1065       2014  VOLVO    XC70 AWD     SUV - SMALL         3.2
6
1066       2014  VOLVO    XC90 AWD  SUV - STANDARD         3.2
6

     TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY
\
0             AS5        Z                   9.9                  6.7

1              M6        Z                  11.2                  7.7

2             AV7        Z                   6.0                  5.8

3             AS6        Z                  12.7                  9.1

4             AS6        Z                  12.1                  8.7

...           ...      ...                   ...                  ...

1062          AS6        X                  13.4                  9.8
```

```
1063          AS6        X                  13.2                   9.5

1064          AS6        X                  13.4                   9.8

1065          AS6        X                  12.9                   9.3

1066          AS6        X                  14.9                  10.2


      FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                      8.5                        33           196
1                      9.6                        29           221
2                      5.9                        48           136
3                     11.1                        25           255
4                     10.6                        27           244
...                    ...                       ...           ...
1062                  11.8                        24           271
1063                  11.5                        25           264
1064                  11.8                        24           271
1065                  11.3                        25           260
1066                  12.8                        22           294

[1067 rows x 13 columns]
```

```
%matplotlib inline
plt.xlabel('FUELCONSUMPTION_CITY')
plt.ylabel('FUELCONSUMPTION_HWY')
plt.scatter(df.FUELCONSUMPTION_CITY,df.FUELCONSUMPTION_HWY,color='yell
ow',marker='*');
```




```
new_df=df
new_df.drop(df.iloc[:,0:5], inplace = True, axis = 1)
new_df
```

```
      FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                           33           196
1                           29           221
2                           48           136
3                           25           255
4                           27           244
...                        ...           ...
1062                        24           271
1063                        25           264
1064                        24           271
1065                        25           260
1066                        22           294

[1067 rows x 2 columns]
```

```
new_df=new_df.drop('CO2EMISSIONS',axis='columns')
new_df
```

```
      FUELCONSUMPTION_COMB_MPG
0                           33
1                           29
2                           48
3                           25
4                           27
...                        ...
```

```
1062                         24
1063                         25
1064                         24
1065                         25
1066                         22

[1067 rows x 1 columns]
```

```
JanTemp = df.CO2EMISSIONS
JanTemp
```

```
0       196
1       221
2       136
3       255
4       244
       ...
1062    271
1063    264
1064    271
1065    260
1066    294
Name: CO2EMISSIONS, Length: 1067, dtype: int64
```

```
print(new_df)
```

```
      FUELCONSUMPTION_COMB_MPG
0                           33
1                           29
2                           48
3                           25
4                           27
...                        ...
1062                        24
1063                        25
1064                        24
1065                        25
1066                        22

[1067 rows x 1 columns]
```

```
#create linear regression object
reg = linear_model.LinearRegression()
reg.fit(new_df,JanTemp)
```

```
LinearRegression()
```

```
#**(2) Predict FOR FC 26
reg.predict([[26]])
```

```
array([259.62358637])
```

```
reg.coef_
```

```
array([-7.6907997])
```

```
reg.intercept_
```

```
459.5843786973706
```

```
#Y = m * X+b (m is coefficient and b is intercept)
-7.6907997*26+459.5843786973706
```

```
259.6235864973706
```

```
#**(2) Predict FOR FC 60
reg.predict([[60]])
```

```
array([-1.86360359])
```

```
X=new_df
Y=JanTemp
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test =
train_test_split(X,Y,test_size=0.25,random_state=2)
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train,Y_train)
Y_pred = lr.predict(X_test)
Y_pred
```

```
array([289.28394791, 281.71206333, 289.28394791, 251.42452502,
       168.13379468, 160.5619101 , 304.42771707, 327.1433708 ,
       311.99960164, 266.56829418, 251.42452502, 205.99321756,
       296.85583249, 266.56829418, 311.99960164, 266.56829418,
       334.71525537, 205.99321756, 213.56510214, 342.28713995,
       221.13698672, 152.99002552, 289.28394791, 168.13379468,
       243.85264045, 327.1433708 , 266.56829418, 266.56829418,
       251.42452502, 236.28075587, 266.56829418, 274.14017876,
       266.56829418, 296.85583249, 357.43090911, 251.42452502,
       190.84944841, 213.56510214, 221.13698672, 327.1433708 ,
       213.56510214, 296.85583249, 198.42133298, 251.42452502,
       266.56829418, 289.28394791, 311.99960164, 289.28394791,
       327.1433708 , 296.85583249, 327.1433708 , 266.56829418,
       289.28394791, 205.99321756, 251.42452502, 281.71206333,
       251.42452502, 183.27756383, 175.70567925, 274.14017876,
       289.28394791, 281.71206333, 281.71206333, 107.55871806,
       183.27756383,  62.12741059, 304.42771707, 304.42771707,
       198.42133298, 251.42452502, 327.1433708 , 274.14017876,
       296.85583249, 342.28713995, 251.42452502, 198.42133298,
       160.5619101 , 327.1433708 , 274.14017876, 243.85264045,
       274.14017876, 160.5619101 , 221.13698672, 175.70567925,
       349.85902453,  99.98683348, 152.99002552, 258.9964096 ,
```

```
       228.70887129, 334.71525537, 160.5619101 , 296.85583249,
       319.57148622, 334.71525537, 205.99321756, 311.99960164,
       228.70887129, 213.56510214, 251.42452502, 228.70887129,
       251.42452502, 190.84944841, 334.71525537, 236.28075587,
       296.85583249, 334.71525537,  69.69929517, 281.71206333,
       205.99321756, 221.13698672, 243.85264045, 266.56829418,
       289.28394791, 289.28394791, 304.42771707, 349.85902453,
       327.1433708 , 281.71206333, 281.71206333, 243.85264045,
       289.28394791, 281.71206333, 251.42452502, 327.1433708 ,
       198.42133298, 365.00279368, 266.56829418, 319.57148622,
       115.13060263, 281.71206333, 289.28394791, 334.71525537,
       213.56510214, 221.13698672, 258.9964096 , 304.42771707,
       221.13698672, 198.42133298, 258.9964096 , 236.28075587,
       319.57148622, 251.42452502, 205.99321756, 221.13698672,
       221.13698672, 266.56829418, 198.42133298, 243.85264045,
       281.71206333, 319.57148622, 243.85264045, 228.70887129,
       243.85264045, 311.99960164, 198.42133298, 281.71206333,
       213.56510214, 251.42452502, 327.1433708 , 243.85264045,
       190.84944841, 205.99321756, 183.27756383, 311.99960164,
       221.13698672, 183.27756383, 311.99960164, 236.28075587,
       258.9964096 , 243.85264045, 274.14017876, 175.70567925,
       289.28394791, 319.57148622, 205.99321756, 228.70887129,
       122.70248721, 221.13698672, 145.41814094, 221.13698672,
       334.71525537, 221.13698672, 281.71206333, 304.42771707,
       319.57148622, 160.5619101 , 190.84944841, 205.99321756,
       258.9964096 , 243.85264045, 160.5619101 , 281.71206333,
       334.71525537, 183.27756383, 289.28394791, 198.42133298,
       183.27756383, 258.9964096 , 296.85583249, 289.28394791,
       258.9964096 , 304.42771707, 221.13698672, 334.71525537,
       243.85264045, 304.42771707, 168.13379468, 228.70887129,
       251.42452502, 243.85264045, 258.9964096 , 266.56829418,
       274.14017876, 357.43090911, 334.71525537, 122.70248721,
       304.42771707, 198.42133298, 342.28713995, 311.99960164,
       296.85583249, 251.42452502, 289.28394791, 266.56829418,
       251.42452502, 183.27756383, 319.57148622, 221.13698672,
       289.28394791, 183.27756383, 304.42771707, 349.85902453,
       266.56829418, 266.56829418, 311.99960164, 160.5619101 ,
       175.70567925, 228.70887129, 236.28075587, 243.85264045,
       258.9964096 , 274.14017876, 327.1433708 , 160.5619101 ,
       258.9964096 , 289.28394791, 183.27756383, 190.84944841,
       221.13698672, 221.13698672, 266.56829418, 213.56510214,
       327.1433708 , 213.56510214, 296.85583249, 319.57148622,
       205.99321756, 198.42133298, 236.28075587, 152.99002552,
       327.1433708 , 175.70567925, 236.28075587, 311.99960164,
       304.42771707, 236.28075587, 258.9964096 ])

from sklearn.metrics import mean_absolute_error
print("MAE",mean_absolute_error(Y_test,Y_pred))

MAE 18.50528579384443
```

```
from sklearn.metrics import mean_squared_error
print("MSE",mean_squared_error(Y_test,Y_pred))
```

```
MSE 761.2896508697498
```

```
print("RMSE",np.sqrt(mean_squared_error(Y_test,Y_pred)))
```

```
RMSE 27.59147786672091
```

```
print("RMSE",np.log(np.sqrt(mean_squared_error(Y_test,Y_pred))))
```

```
RMSE 3.3175069520980287
```

```
print("RMSE",np.log(np.sqrt(mean_squared_error(Y_test,Y_pred))))
```

```
RMSE 3.3175069520980287
```

```
#c. Visualize simple regression model.
%matplotlib inline
plt.xlabel('CO2EMISSIONS')
plt.ylabel('FUELCONSUMPTION_COMB_MPG')
plt.scatter(df.CO2EMISSIONS,df.FUELCONSUMPTION_COMB_MPG,color=
'orange',marker='*')
plt.plot(X_train, lr.predict(X_train), color = "blue")
```

```
[<matplotlib.lines.Line2D at 0x13352709a00>]
```