

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

from sklearn.metrics import mean_squared_error , mean_absolute_error


dataset=pd.read_csv("/content/all_energy_statistics1 (3) (1).csv")

print(dataset.head())
print(dataset.describe())

dataset=dataset.drop(columns=["commodity_transaction","unit","category"])

print(dataset.columns)

dataset["country_or_area_as_num"]=dataset["country_or_area"].astype('category').cat.codes

X=dataset[["country_or_area_as_num", "year"]].values

country_code=dataset["country_or_area_as_num"].unique()
country_name=dataset["country_or_area"].unique()

for c in range(country_name.shape[0]):

    print("country code: ",country_code[c]," country name: ",country_name[c])

#country code: 32 country name: Greece
#country code: 31 country name: Germany

Y=dataset["quantity"].values

df_2014 = dataset.loc[dataset["year"]==2014]
```

```
dataset=dataset.drop(dataset[dataset.year==2014].index)
```

```
model = LinearRegression().fit(X, Y)
```

```
r_sq = model.score(X, Y)
```

```
y_pred = model.predict(X)
```

```
print("coefficient of determination:", r_sq)
```

```
print("slope:", model.coef_)
```

```
print("Intercept:",model.intercept_)
```

```
y_pred = model.predict(X) # Enter year you are going to predict the SEP production
```

```
print("predicted response:", y_pred)
```

```
#y_pred1 = model.intercept_ + model.coef_ * X
```

```
#print('#predicted response1:', y_pred1)
```

```
dataset.plot(x="year", y="quantity", style="")
```

```
plt.title("QTY KWT Per Hr by Million")
```

```
plt.xlabel("Year")
```

```
plt.ylabel("Quantity (KW mil. per Hr)")
```

```
ypoints = np.array(y_pred)
```

```
plt.plot(X,ypoints)
```

```
plt.show()
```

```
#day one, the age and speed of 13 cars:
```

```
x1 = np.array([2012,2013,2014,2015,2016,2017,2018,2019,2020,2021,2022,2023,2024,2025])
```

```
for count in range(country_name.shape[0]):
```

```
    temp = np.full((14, 2), count, dtype=np.int64)
```

```
    temp[:, 0] = x1
```

```
    for itr in range(14):
```

```
        predcit = model.predict(temp[itr,:].reshape(1,-1))
```

```
        print("country: "+country_name[count]+" year: "+str(temp[itr,0])+" prediction: " +
```

```
print('++++++')
```

```
germany=df_2014.loc[(df_2014["country_or_area_as_num"]==31)]
X=germany[["country_or_area_as_num", "year"]].values
test_pred = model.predict(X)

result = mean_squared_error(germany["quantity"] , test_pred)

print("germany Mean squared error result: ",result)
result = mean_absolute_error(germany["quantity"] , test_pred)
print("germany Mean absolute error result: ",result)
```

```
greese=df_2014.loc[(df_2014["country_or_area_as_num"]==32)]
X=greese[["country_or_area_as_num", "year"]].values
test_pred = model.predict(X)

result = mean_squared_error(greese["quantity"] , test_pred)

print("Greese Mean squared error result: ",result)

result = mean_absolute_error(greese["quantity"] , test_pred)
print("Greese Mean absolute error result: ",result)
```

```

country_or_area ... category
0 American Samoa ... solar_electricity
1 American Samoa ... solar_electricity
2 American Samoa ... solar_electricity
3 Argentina ... solar_electricity
4 Argentina ... solar_electricity

```

```
[5 rows x 6 columns]
```

```

count    year    quantity
mean    2006.773347    612.694175
std      6.410260    2870.113133
min     1990.000000     0.000000
25%     2003.000000     1.000000
50%     2009.000000     7.000000
75%     2012.000000    50.000000
max     2014.000000   36056.000000

```

```
Index(['country_or_area', 'year', 'quantity'], dtype='object')
```

```

country code: 0 country name: American Samoa
country code: 1 country name: Argentina
country code: 2 country name: Australia
country code: 3 country name: Austria
country code: 4 country name: Azerbaijan
country code: 5 country name: Bangladesh
country code: 6 country name: Belarus
country code: 7 country name: Belgium
country code: 8 country name: Bolivia (Plur. State of)
country code: 9 country name: Botswana
country code: 10 country name: Brazil
country code: 11 country name: Brunei Darussalam
country code: 12 country name: Bulgaria
country code: 13 country name: Cabo Verde
country code: 14 country name: Cambodia
country code: 15 country name: Canada
country code: 16 country name: Chile
country code: 17 country name: China
country code: 18 country name: Costa Rica
country code: 19 country name: Croatia
country code: 20 country name: Cyprus
country code: 21 country name: Czechia
country code: 22 country name: Denmark
country code: 23 country name: Ecuador
country code: 24 country name: Egypt
country code: 25 country name: Eritrea
country code: 26 country name: Finland
country code: 27 country name: France
country code: 28 country name: French Guiana
country code: 29 country name: French Polynesia
country code: 30 country name: Gabon
country code: 31 country name: Germany
country code: 32 country name: Greece
country code: 33 country name: Guadeloupe
country code: 34 country name: Guatemala
country code: 35 country name: Hungary
country code: 36 country name: India
country code: 37 country name: Indonesia
country code: 38 country name: Iran (Islamic Rep. of)
country code: 39 country name: Ireland
country code: 40 country name: Israel
country code: 41 country name: Italy

```