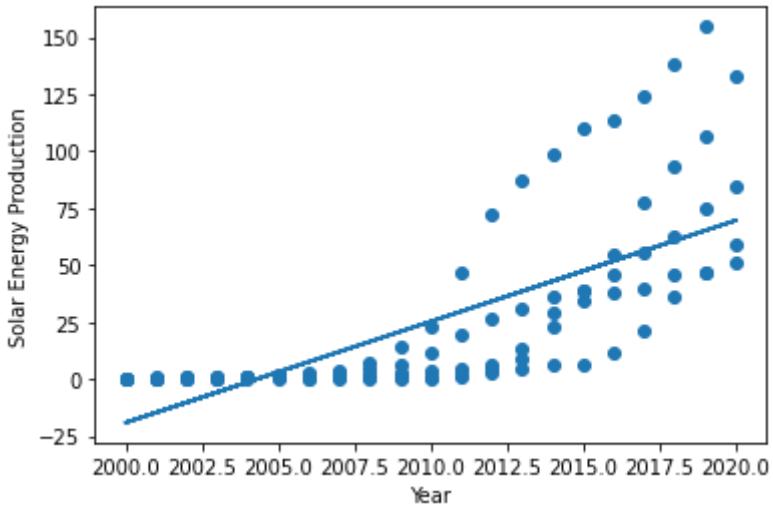
Global Solar Energy Production Estimation

Linear Regression

```
import csv
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats

file = open("Solar_Energy.csv")
csvr = csv.reader(file)
Solar = []
Year = []
for row in csvr:
    if row[0] != "year" and str(row[1])!='World' and str(row[1])!='China':
        Solar.append(float(row[2]))
        Year.append(int(row[0]))

y = np.array(Solar)
x = np.array(Year)
plt.xlabel('Year')
plt.ylabel('Solar Energy Production')

slope, intercept, r, p, std_err = stats.linregress(x, y)
def myfunc(x):
  return slope * x + intercept

mymodel = list(map(myfunc, x))

plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()
print('The coefficient of correlation :',r)
```




```
The coefficient of correlation : 0.7239921530153033
```

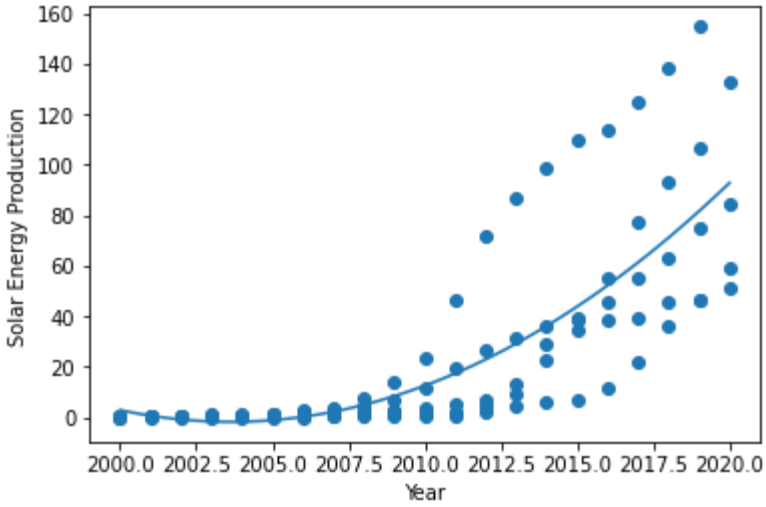
Estimation using Linear regression model

```
year=int(input('Enter year : '))
energy=myfunc(year)
print('Predicted solar energy production in {0} : {1} '.format(year,energy))
```

```
Enter year : 2038
Predicted solar energy production in 2038 : 149.44238872018468
```

Polynomial Regression

```
import csv
import matplotlib.pyplot as plt
import numpy
from sklearn.metrics import r2_score

file = open("Solar_Energy.csv")
csvr = csv.reader(file)
Solar = []
Year = []
for row in csvr:
    if row[0] != "year" and str(row[1])!='World' and str(row[1])!='China' and str(row[1]!='Europe'):
        Solar.append(float(row[2]))
        Year.append(int(row[0]))

y = np.array(Solar)
x = np.array(Year)
plt.xlabel('Year')
plt.ylabel('Solar Energy Production')


mymodel = numpy.poly1d(numpy.polyfit(x, y, 2))

myline = numpy.linspace(2000, 2020, 100)

plt.scatter(x, y)
plt.plot(myline, mymodel(myline))
plt.show()
print('The r-squared value of relationship :',r2_score(y, mymodel(x)))
```




```
The r-squared value of relationship : 0.6197907965649622
```

Estimation using Polynomial regression

```
year=int(input('Enter year : '))
energy=mymodel(year)
print('Predicted solar energy production in {0} : {1} '.format(year,energy))
```

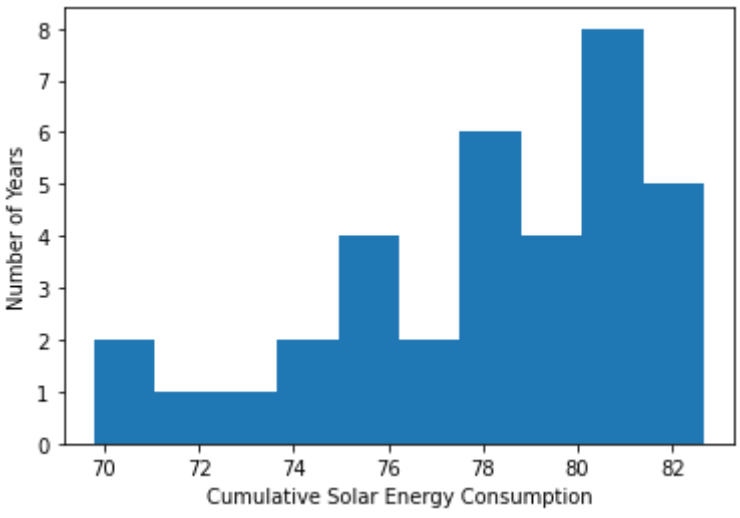
```
Enter year : 2036
Predicted solar energy production in 2036 : 368.12213101168163
```

Estimation of Solar Energy Production

Histogram

```
import csv
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats


file = open("EnergyScenario.csv")
csvr = csv.reader(file)
CumulativeSolar=[]
Solar = []
Year = []
for row in csvr:
    if row[1] != "Year" and row[8]:
        CumulativeSolar.append(float(row[8]))
    if row[1] != "Year" and int(row[1])>2003:
        Solar.append(float(row[7]))
        Year.append(int(row[1]))

x = np.array(Solar)
z = np.array(CumulativeSolar)
y = np.array(Year)

plt.xlabel('Cumulative Solar Energy Consumption')
plt.ylabel('Number of Years')
plt.hist(z)
```

```
(array([2., 1., 1., 2., 4., 2., 6., 4., 8., 5.]),
 array([69.782664  , 71.0717326 , 72.36080119, 73.64986979, 74.93893839,
        76.22800699, 77.51707559, 78.80614419, 80.09521279, 81.38428139,
        82.67334998]),
 <a list of 10 Patch objects>)
```



Standard Deviation

```
sd=np.std(Solar)
print('Standard deviation of Solar Energy production :',sd)
```

```
Standard deviation of Solar Energy production : 34.085743699587134
```

Mean

```
mean=np.mean(Solar)
print('Mean value of Solar Energy produced :',mean)
```

```
Mean value of Solar Energy produced : 20.88522875801963
```

Median

```
median=np.median(Solar)
print('Median value of Solar Energy produced :',median)
```

```
Median value of Solar Energy produced : 3.7748307175709446
```

Mode

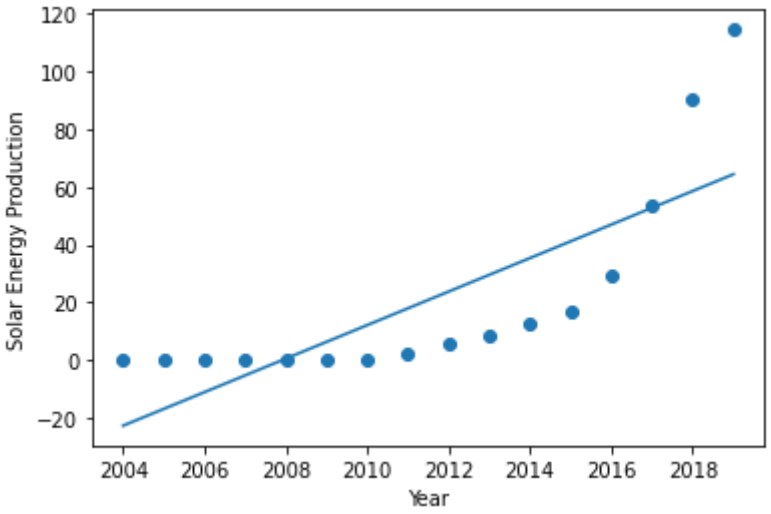
```
mode=stats.mode(Solar)
print('Modal value of Solar Energy produced :',mode)
```

```
Modal value of Solar Energy produced : ModeResult(mode=array([0.0192453]), count=array([1]))
```

Linear Regression

```
y = np.array(Solar)
x = np.array(Year)
plt.xlabel('Year')
plt.ylabel('Solar Energy Production')

slope, intercept, r, p, std_err = stats.linregress(x, y)

def myfunc(x):
  return slope * x + intercept

mymodel = list(map(myfunc, x))

plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()
```

```
print('The coefficient of correlation :',r)
```




```
The coefficient of correlation : 0.7853434398660488
```

Prediction using Linear regression

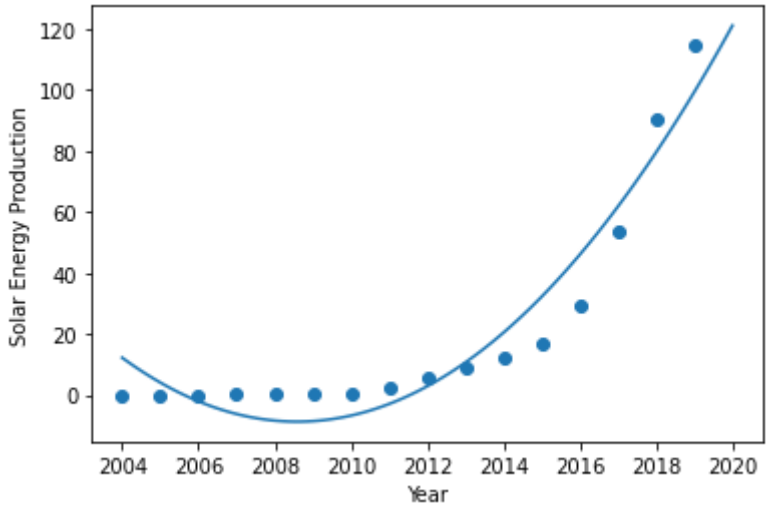
```
year=int(input('Enter year : '))
energy=myfunc(year)
print('Predicted solar energy production in {0} : {1} '.format(year,energy))
```

```
Enter year : 2024
Predicted solar energy production in 2024 : 93.47291281178877
```

Polynomial Regression

```
import csv
import matplotlib.pyplot as plt
import numpy
from sklearn.metrics import r2_score

file = open("EnergyScenario.csv")
csvr = csv.reader(file)
Solar = []
Year = []
for row in csvr:
    if row[1] != "Year" and int(row[1])>2003:
        Solar.append(float(row[7]))
        Year.append(int(row[1]))

y = np.array(Solar)
x = np.array(Year)
plt.xlabel('Year')
plt.ylabel('Solar Energy Production')

mymodel = numpy.poly1d(numpy.polyfit(x, y, 2))

myline = numpy.linspace(2004, 2020, 100)
```

```
plt.scatter(x, y)
plt.plot(myline, mymodel(myline))
plt.show()
print('The r-squared value of relationship :',r2_score(y, mymodel(x)))
```




```
The r-squared value of relationship : 0.9219019443191674
```

Prediction using Polynomial regression model

```
year=int(input('Enter year : '))
energy=mymodel(year)
print('Predicted solar energy production in {0} : {1} '.format(year,energy))
```

```
Enter year : 2023
Predicted solar energy production in 2023 : 130.61410084343515
```