Exercise

Predict Canada's per capita income in year 2020. There is an exercise folder here on github download that and you will find `canada_per_capita_income.csv` file. Using this build a regression model and predict the per capita income fo Canadian citizens in year 2020.

> Answer `41288.69409442`

In [1]:
```
#Required imports

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

In [2]:
```
# Reading csv file to dataframe

df = pd.read_csv('./data/canada_per_capita_income.csv')
df.head()
```

Out[2]:

	year	per capita income (US$)
0	1970	3399.299037
1	1971	3768.297935
2	1972	4251.175484
3	1973	4804.463248
4	1974	5576.514583

In [3]:
```
# Last 5 rows

df.tail()
```

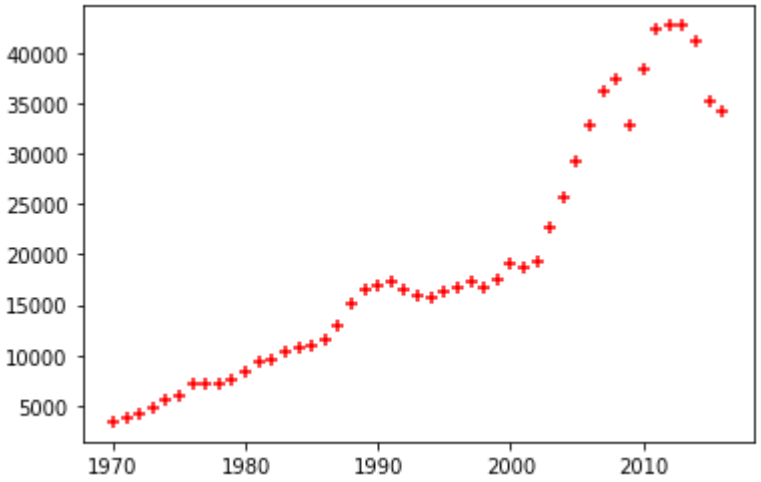
Out[3]:

	year	per capita income (US$)
42	2012	42665.25597
43	2013	42676.46837
44	2014	41039.89360
45	2015	35175.18898
46	2016	34229.19363

In [4]:
```
# Scatter plot for the dataset

%matplotlib inline
plt.xlabel = "year"
plt.ylabel = "per capita income  (US$)"
plt.scatter(df.year, df['per capita income (US$)'], color='red', marker='+')
```

Out[4]: <matplotlib.collections.PathCollection at 0x178e865f790>



Preparing data for linear regression,

In [5]:
```
x_df = df.drop('per capita income (US$)', axis='columns')
x_df.head()
```

Out[5]:

	year
0	1970
1	1971
2	1972
3	1973
4	1974

In [6]:
```
y_df = df['per capita income (US$)']
y_df.head()
```

Out[6]:
```
0    3399.299037
1    3768.297935
2    4251.175484
3    4804.463248
4    5576.514583
Name: per capita income (US$), dtype: float64
```

Applying Linear Regression

In [7]:
```
model = linear_model.LinearRegression()
model.fit(x_df, y_df)
```

Out[7]: LinearRegression()

Predicting 'per capita income for Canadian citizens in year 2020'

In [8]:
```
ans = model.predict([[2020]])
print('The per capita income for Canadian citizens in year 2020 will be, \nUS${}'.fo
```

The per capita income for Canadian citizens in year 2020 will be, US$[41288.69409442]

Proof

In [9]:
```
m = model.coef_
c = model.intercept_
print('Coefficient, m = ', m)
print('Intercept, c = ', c)
```

Coefficient, m = [828.46507522]
Intercept, c = -1632210.7578554575

Calculating y for y = m*x +c ,

In [10]:
```
y = m*2020 + c
print('y = m*x +c =', y)
```

y = m*x +c = [41288.69409442]

Here, we can see that y = ans = 41288.69409442 .