$$\theta = (\mathbf{X}^\intercal \cdot X)^{-1} \cdot (\mathbf{X}^\intercal \cdot Y)$$

In [2]:
```
def prepare_country_stats(oecd_bli, gdp_per_capita):
    oecd_bli = oecd_bli[oecd_bli["INEQUALITY"] == "TOT"]
    oecd_bli = oecd_bli.pivot(index="Country", columns="Indicator", values="Value")
    gdp_per_capita.rename(columns={"2015":"GDP per capita"}, inplace=True)
    gdp_per_capita.set_index("Country", inplace=True)
    full_country_stats = pd.merge(left=oecd_bli, right=gdp_per_capita,
                                  left_index = True, right_index = True)
    full_country_stats.sort_values(by="GDP per capita", inplace= True)
    remove_indices = [0,1,6,8,33,34,35]
    keep_indices = list(set(range(36)) - set(remove_indices))
    return full_country_stats[["GDP per capita","Life satisfaction"]].iloc[keep_indices]
```

In [3]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [4]:
```
# Load the data
oecd_bli = pd.read_csv("handson-ml\datasets\lifesat\oecd_bli_2015.csv", thousands=',')
gdp_per_capita = pd.read_csv("handson-ml\datasets\lifesat\gdp_per_capita.csv", thousands=",", delimiter='\t',
                             encoding='latin1', na_values="n/a")
```

In [5]:
```
# Brief description of the dataset
gdp_per_capita.head(5)
```

Out[5]:

	Country	Subject Descriptor	Units	Scale	Country/Series-specific Notes	2015	Estimates Start After
0	Afghanistan	Gross domestic product per capita, current prices	U.S. dollars	Units	See notes for: Gross domestic product, curren...	599.994	2013.0
1	Albania	Gross domestic product per capita, current prices	U.S. dollars	Units	See notes for: Gross domestic product, curren...	3995.383	2010.0
2	Algeria	Gross domestic product per capita, current prices	U.S. dollars	Units	See notes for: Gross domestic product, curren...	4318.135	2014.0
3	Angola	Gross domestic product per capita, current prices	U.S. dollars	Units	See notes for: Gross domestic product, curren...	4100.315	2014.0
4	Antigua and Barbuda	Gross domestic product per capita, current prices	U.S. dollars	Units	See notes for: Gross domestic product, curren...	14414.302	2011.0

In [6]:
```
 # Prepare the data
country_stats = prepare_country_stats(oecd_bli, gdp_per_capita)
```

In [7]:
```
# Build the pices for Normal equation
thetas = np.zeros((country_stats.size,1))
ones = np.ones((country_stats[country_stats.columns[0]].count(),1))
X = country_stats["GDP per capita"].to_frame()
Y = country_stats["Life satisfaction"].to_frame()
```

In [8]:
```
# Append an extra column of ones to the fearute vector (X)
X.insert(loc=0, column='X0', value=ones)
```

In [9]:
```
# Apply the formula
thetas = np.linalg.inv(np.transpose(X).dot(X)).dot((np.transpose(X).dot(Y)))
```

In [10]:
```
print(thetas)
X.shape
```

```
[[4.85305280e+00]
 [4.91154459e-05]]
```

Out[10]: (29, 2)

Create a function to predict new values taking account the hypotesis function $h_\theta(x) = \theta_0 + \theta_1 X_1$

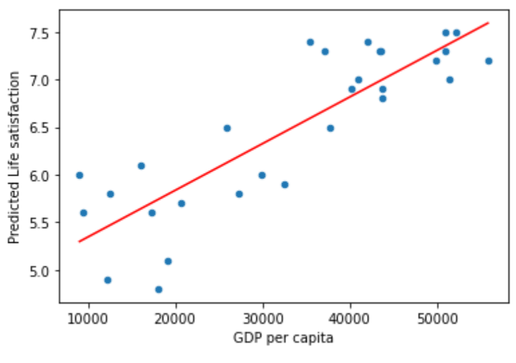
In [11]:
```
def predit_value(x_new, thetas):
    predicted_value = thetas[0] + thetas[1]*x_new
    return predicted_value
```

In [12]:
```
# Predict the output (Life satisfaction) for the X input = 22587 GDP
print(predit_value(22587, thetas))
print(predit_value(40000, thetas))
```

```
[5.96242338]
[6.81767064]
```

In [13]:
```
# For plot purposes let's calculate all the "calculated" y for given x
y_calculated = X.dot(thetas)
X_for_plot = np.array(X['GDP per capita'].values.reshape(1,29)).ravel()
Y_for_plot = np.array(y_calculated.values.reshape(1,29)).ravel() # ravel() to remove extra bracket in numpy array
```

In [14]:
```
# Visualize the initial data set
country_stats.plot(kind="scatter", x="GDP per capita", y="Life satisfaction")
plt.plot(X_for_plot, Y_for_plot, 'r')
plt.ylabel('Predicted Life satisfaction')
plt.xlabel('GDP per capita')
plt.show()
```



In [ ]: