In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:

```
data = pd.read_csv("World_Per_Capita_Electricity_Consumption.csv")
```

In [4]:

```
data.head()
```

Out[4]:

	Entity	Year	Per capita electricity (kWh)
0	Afghanistan	2000	22.474
1	Afghanistan	2001	27.399
2	Afghanistan	2002	30.397
3	Afghanistan	2003	39.652
4	Afghanistan	2004	36.155

In [5]:

```
data.tail()
```

Out[5]:

	Entity	Year	Per capita electricity (kWh)
5615	Zimbabwe	2015	679.553
5616	Zimbabwe	2016	486.509
5617	Zimbabwe	2017	513.434
5618	Zimbabwe	2018	632.533
5619	Zimbabwe	2019	617.672

In [6]:

```
data.shape
```

Out[6]:

```
(5620, 3)
```

Shivam Modi

Founder & CEO of learneverythingAI

Hello World, Learn Everything AI is my attempt to teach Data Science and coding basics to people in a short time, which took me ages to learn. Artificial Intelligence, Data Analytics, Mathematics, Machine Learning, I simplify these topics to help you understand how they work. Using this knowledge, you can build wealth and live a happier, more meaningful life.

Teaching is my passion. I strongly believe "Anyone Can Code"! I aim to fulfill this vision by teaching Data Science in a most simplistic and practical way!

In [7]:

```
data.columns
```

Out[7]:

```
Index(['Entity', 'Year', 'Per capita electricity (kWh)'], dtype='object')
```

In [8]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5620 entries, 0 to 5619
Data columns (total 3 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Entity                        5620 non-null   object 
 1   Year                          5620 non-null   int64  
 2   Per capita electricity (kWh)  5620 non-null   float64
dtypes: float64(1), int64(1), object(1)
memory usage: 131.8+ KB
```

In [9]:

```
data.describe()
```

Out[9]:

	Year	Per capita electricity (kWh)
count	5620.000000	5620.000000
mean	2005.826157	4095.887211
std	9.238193	5403.908499
min	1985.000000	0.000000
25%	2000.000000	646.116250
50%	2007.000000	2555.821500
75%	2013.000000	5561.172750
max	2021.000000	58863.361000

In [10]:

```
data.isnull().sum()
```

Out[10]:

```
Entity                          0
Year                            0
Per capita electricity (kWh)    0
dtype: int64
```

In [11]:

```
data['Entity'].unique()
```

Out[11]:

```
array(['Afghanistan', 'Africa', 'Albania', 'Algeria', 'American Samoa',
       'Angola', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Aruba',
       'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
       'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
       'Bermuda', 'Bhutan', 'Bolivia', 'Bosnia and Herzegovina',
       'Botswana', 'Brazil', 'British Virgin Islands', 'Brunei',
       'Bulgaria', 'Burkina Faso', 'Burundi', 'Cambodia', 'Cameroon',
       'Canada', 'Cape Verde', 'Cayman Islands',
       'Central African Republic', 'Chad', 'Chile', 'China', 'Colombia',
       'Comoros', 'Congo', 'Cook Islands', 'Costa Rica', 'Croatia',
       'Cuba', 'Cyprus', 'Czechia', 'Democratic Republic of Congo',
       'Denmark', 'Djibouti', 'Dominica', 'Dominican Republic', 'Ecuador',
       'Egypt', 'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia',
       'Ethiopia', 'Europe', 'European Union (27)', 'Falkland Islands',
       'Fiji', 'Finland', 'France', 'French Guiana', 'French Polynesia',
       'Gabon', 'Gambia', 'Georgia', 'Germany', 'Ghana', 'Gibraltar',
       'Greece', 'Greenland', 'Grenada', 'Guadeloupe', 'Guam',
       'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti',
       'Honduras', 'Hong Kong', 'Hungary', 'Iceland', 'India',
       'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
       'Jamaica', 'Japan', 'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati',
       'Kuwait', 'Kyrgyzstan', 'Laos', 'Latvia', 'Lebanon', 'Lesotho',
       'Liberia', 'Libya', 'Lithuania', 'Luxembourg', 'Madagascar',
       'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta', 'Martinique',
       'Mauritania', 'Mauritius', 'Mexico', 'Mongolia', 'Montenegro',
       'Montserrat', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia',
       'Nauru', 'Nepal', 'Netherlands', 'Netherlands Antilles',
       'New Caledonia', 'New Zealand', 'Nicaragua', 'Niger', 'Nigeria',
       'Niue', 'North Korea', 'North Macedonia', 'Norway', 'Oman',
       'Pakistan', 'Palestine', 'Panama', 'Papua New Guinea', 'Paraguay',
       'Peru', 'Philippines', 'Poland', 'Portugal', 'Puerto Rico',
       'Qatar', 'Reunion', 'Romania', 'Russia', 'Rwanda', 'Saint Helena',
       'Saint Kitts and Nevis', 'Saint Lucia',
       'Saint Pierre and Miquelon', 'Saint Vincent and the Grenadines',
       'Samoa', 'Sao Tome and Principe', 'Saudi Arabia', 'Senegal',
       'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore', 'Slovakia',
       'Slovenia', 'Solomon Islands', 'Somalia', 'South Africa',
       'South Korea', 'South Sudan', 'Spain', 'Sri Lanka', 'Sudan',
       'Suriname', 'Sweden', 'Switzerland', 'Syria', 'Taiwan',
       'Tajikistan', 'Tanzania', 'Thailand', 'Togo', 'Tonga',
       'Trinidad and Tobago', 'Tunisia', 'Turkey', 'Turkmenistan',
       'Turks and Caicos Islands', 'Uganda', 'Ukraine',
       'United Arab Emirates', 'United Kingdom', 'United States',
       'United States Virgin Islands', 'Uruguay', 'Uzbekistan', 'Vanuatu',
       'Venezuela', 'Vietnam', 'Western Sahara', 'World', 'Yemen',
       'Zambia', 'Zimbabwe'], dtype=object)
```

In [13]:

```
len(data['Entity'].unique())
```

Out[13]:

```
213
```

In [14]:

```
data['Entity'].value_counts()
```

Out[14]:

```
Turkey                  37
Portugal                37
Austria                 37
Cyprus                  37
Netherlands             37
                        ..
Mauritius               20
Ethiopia                20
Guinea                  20
Cambodia                20
Netherlands Antilles    18
Name: Entity, Length: 213, dtype: int64
```

In [15]:

```
india_electricity_consumption = data[data['Entity']=='India']
```

In [18]:

```
india_electricity_consumption.head()
```

Out[18]:

	Entity	Year	Per capita electricity (kWh)
2307	India	1985	237.628
2308	India	1986	252.723
2309	India	1987	272.873
2310	India	1988	288.136
2311	India	1989	318.526

In [19]:

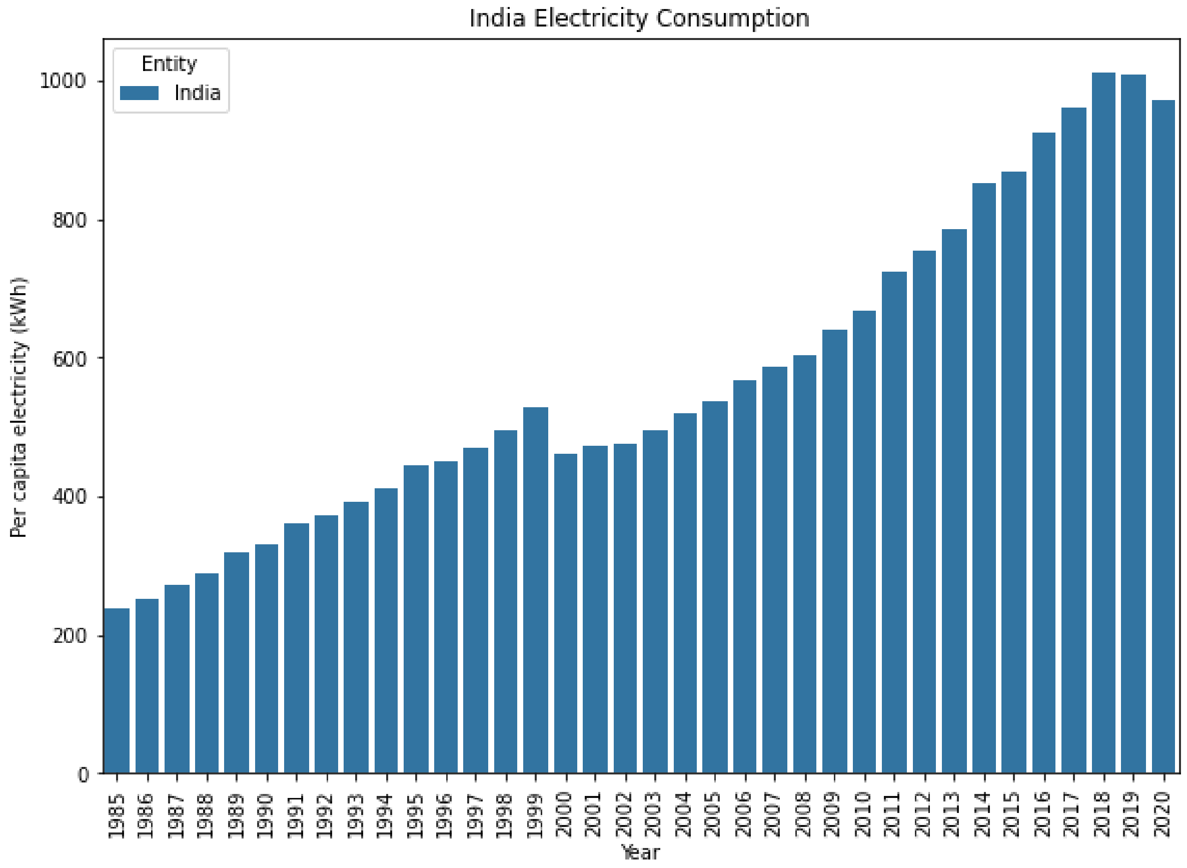
```
india_electricity_consumption.tail()
```

Out[19]:

	Entity	Year	Per capita electricity (kWh)
2338	India	2016	924.206
2339	India	2017	960.455
2340	India	2018	1011.229
2341	India	2019	1008.604
2342	India	2020	972.437

In [22]:

```
plt.figure(figsize = (10,7))
sns.barplot(x = 'Year',
            y = 'Per capita electricity (kWh)',
            hue = 'Entity',
            data = india_electricity_consumption)
plt.xticks(rotation = 90)
plt.title("India Electricity Consumption")
plt.show()
```



In [24]:

```
china_electricity_consumption = data[data['Entity']=='China']
```

In [25]:

```
china_electricity_consumption.head()
```

Out[25]:

	Entity	Year	Per capita electricity (kWh)
1022	China	1985	381.828
1023	China	1986	410.524
1024	China	1987	445.542
1025	China	1988	479.211
1026	China	1989	504.861

In [26]:

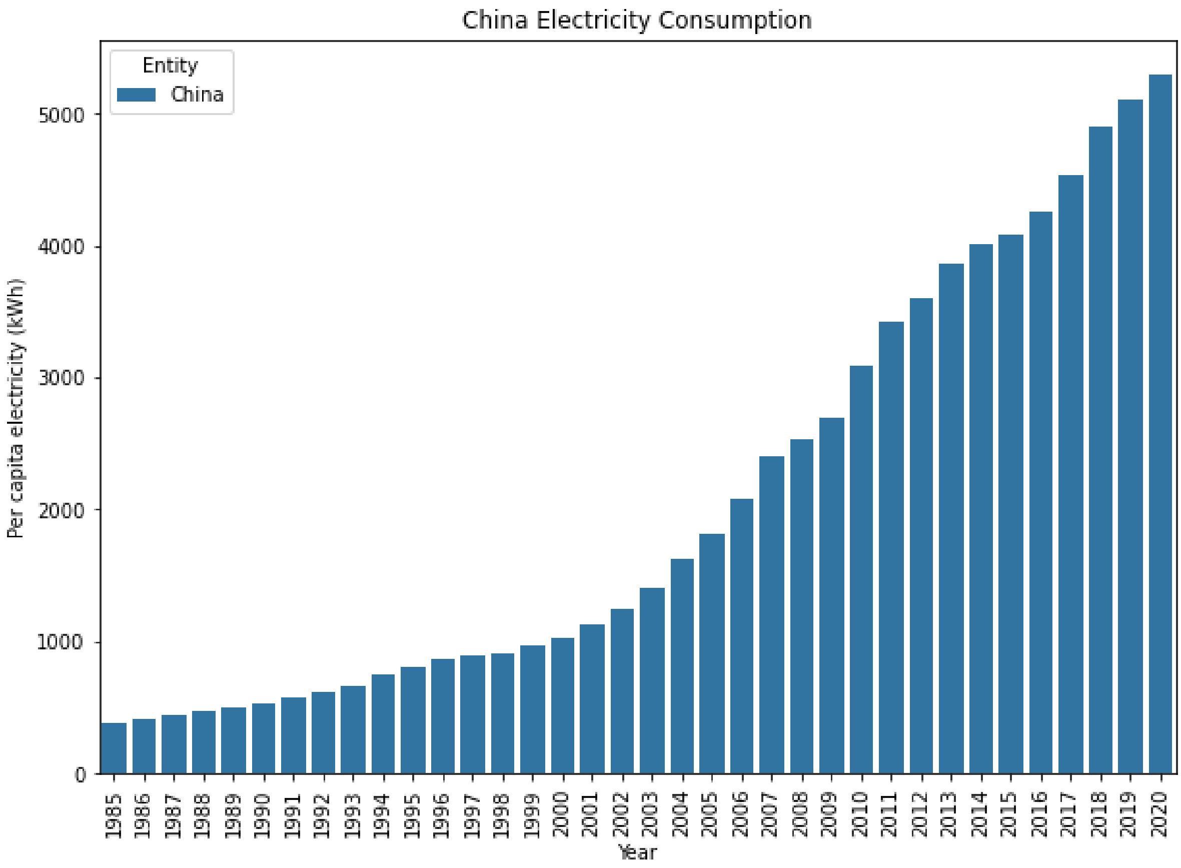
```
china_electricity_consumption.tail()
```

Out[26]:

	Entity	Year	Per capita electricity (kWh)
1053	China	2016	4259.257
1054	China	2017	4538.073
1055	China	2018	4900.508
1056	China	2019	5110.046
1057	China	2020	5296.721

In [27]:

```
plt.figure(figsize = (10,7))
sns.barplot(x = 'Year',
            y = 'Per capita electricity (kWh)',
            hue = 'Entity',
            data = china_electricity_consumption)
plt.xticks(rotation = 90)
plt.title("China Electricity Consumption")
plt.show()
```



In [28]:

```
US_electricity_consumption = data[data['Entity']=='United States']
```

In [29]:

```
US_electricity_consumption.head()
```

Out[29]:

	Entity	Year	Per capita electricity (kWh)
5301	United States	1985	11048.449
5302	United States	1986	11023.556
5303	United States	1987	11312.686
5304	United States	1988	11781.619
5305	United States	1989	12635.657

In [30]:

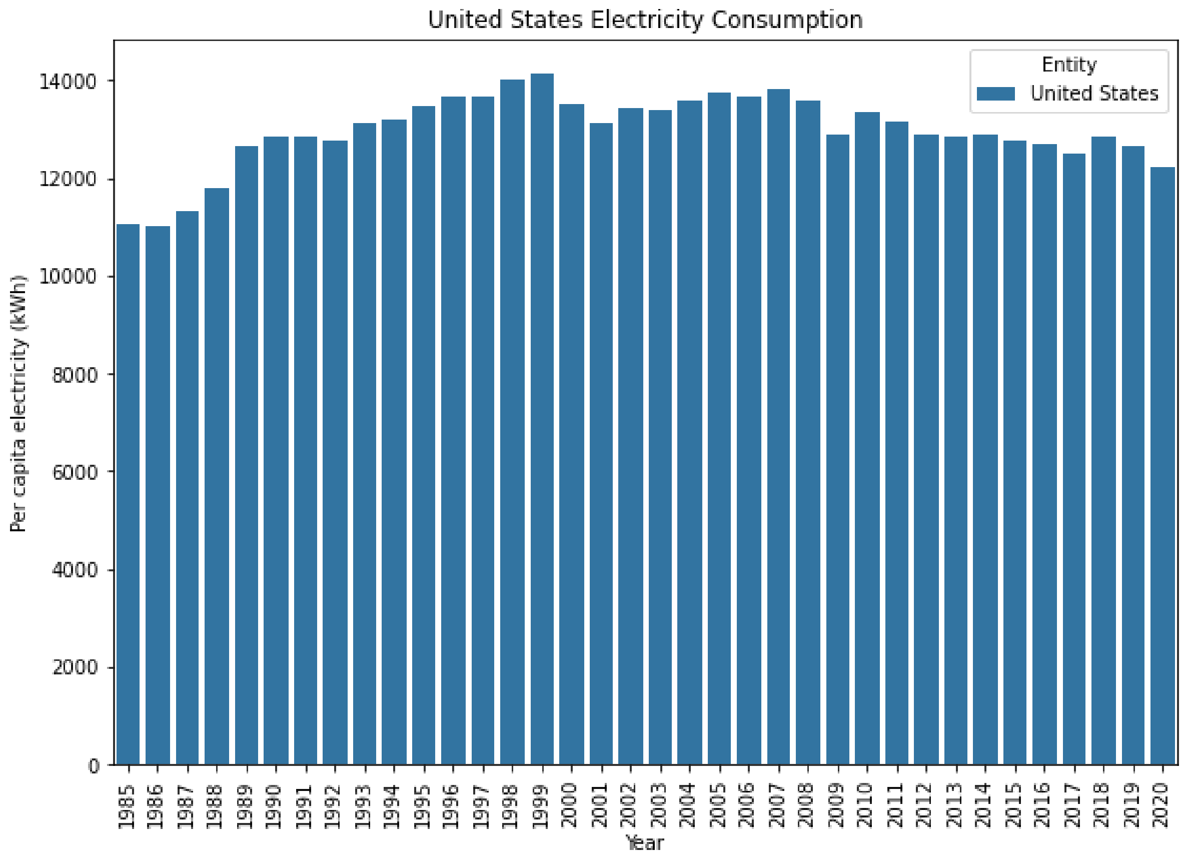
```
US_electricity_consumption.tail()
```

Out[30]:

	Entity	Year	Per capita electricity (kWh)
5332	United States	2016	12678.900
5333	United States	2017	12483.701
5334	United States	2018	12852.293
5335	United States	2019	12647.457
5336	United States	2020	12235.088

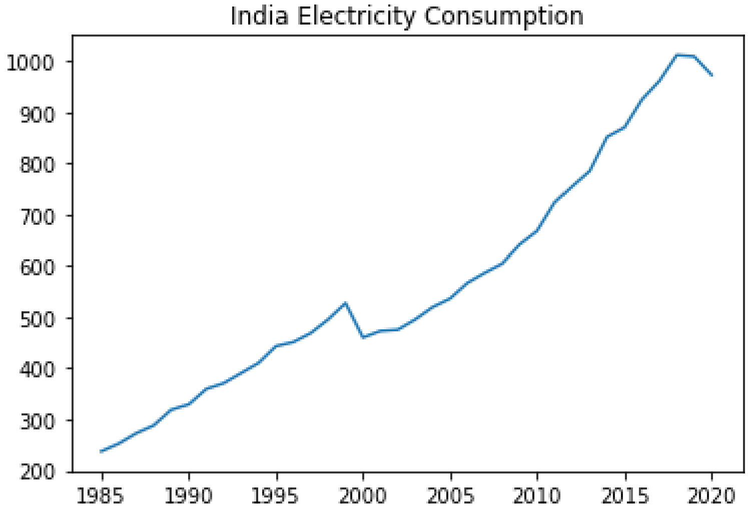
In [31]:

```
plt.figure(figsize = (10,7))
sns.barplot(x = 'Year',
            y = 'Per capita electricity (kWh)',
            hue = 'Entity',
            data = US_electricity_consumption)
plt.xticks(rotation = 90)
plt.title("United States Electricity Consumption")
plt.show()
```



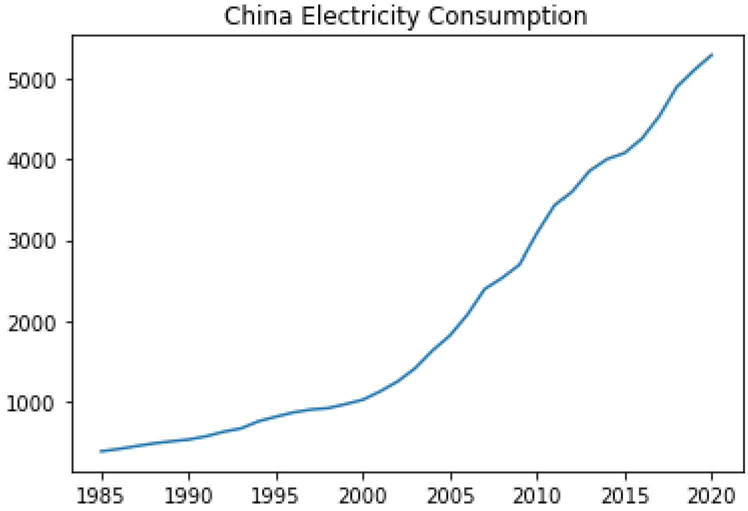
In [43]:

```
plt.plot(data['Year'][data['Entity']=='India'],
         data['Per capita electricity (kWh)'][data['Entity']=="India"])
plt.title("India Electricity Consumption")
plt.show()
```



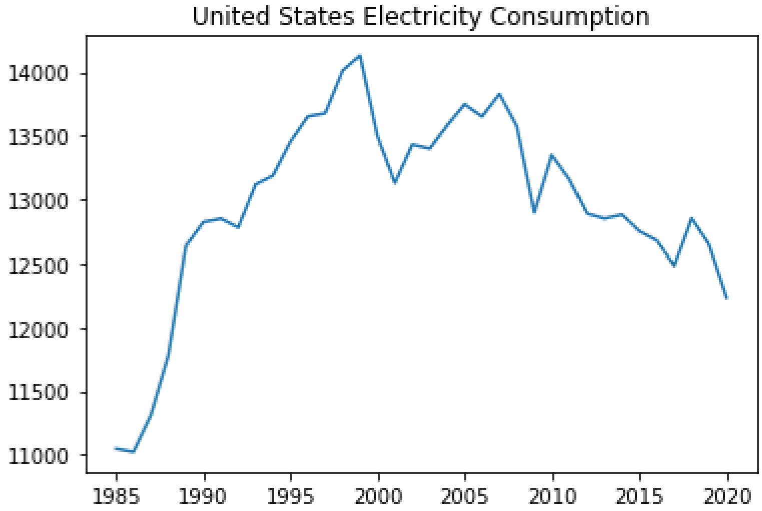
In [44]:

```
plt.plot(data['Year'][data['Entity']=='China'],
         data['Per capita electricity (kWh)'][data['Entity']=="China"])
plt.title("China Electricity Consumption")
plt.show()
```



In [45]:

```
plt.plot(data['Year'][data['Entity']=='United States'],
         data['Per capita electricity (kWh)'][data['Entity']=="United States"])
plt.title("United States Electricity Consumption")
plt.show()
```



In [52]:

```
x = india_electricity_consumption.drop(['Per capita electricity (kWh)', 'Entity'], axis
y = india_electricity_consumption['Per capita electricity (kWh)']
```

In [53]:

```
x.shape
```

Out[53]:

```
(36, 1)
```

In [54]:

```
y.shape
```

Out[54]:

```
(36,)
```

In [55]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [56]:

```
model_india=LinearRegression()
model_india.fit(x,y)
```

Out[56]:

```
LinearRegression()
```

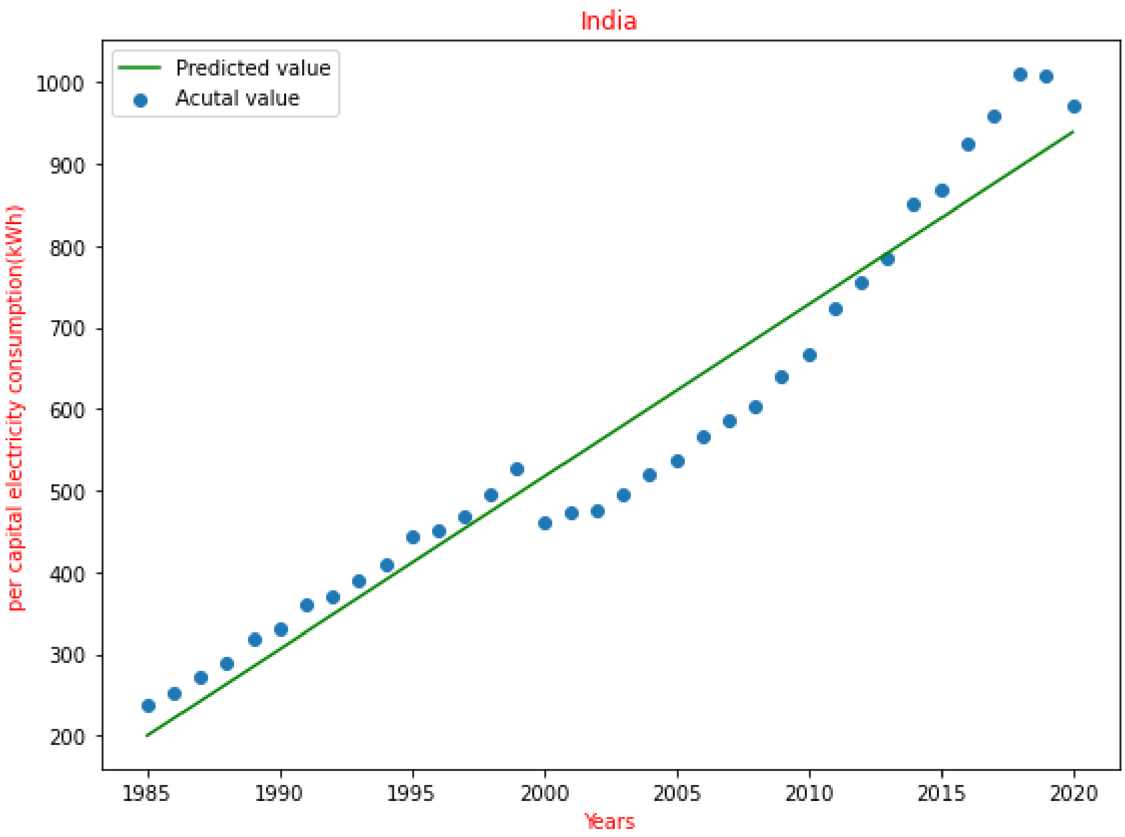
In [57]:

```
print("Coefficient: ",model_india.coef_)
print("intercept: ",model_india.intercept_)
pre = model_india.predict(x)
```

```
Coefficient:  [21.11990232]
intercept:  -41723.12313899615
```

In [60]:

```
plt.figure(figsize=(8,6))
plt.scatter(x,y,label='Acutal value')
plt.plot(x,pre,color='g',label='Predicted value')
plt.legend()
plt.title("India",color='r')
plt.xlabel("Years",color='r')
plt.ylabel("per capital electricity consumption(kWh)",color='r')
plt.tight_layout()
plt.show()
```



In [62]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [63]:

```
years=[2021,2022,2023,2024,2025]
for i in years:
    print(model_india.predict([[i]]))
```

```
[960.19944286]
[981.31934517]
[1002.43924749]
[1023.55914981]
[1044.67905212]
```

In [65]:

```
from sklearn.metrics import mean_squared_error
```

In [68]:

```
print("By function: ",mean_squared_error(y,model_india.predict(x)))
```

```
By function:  3107.018301824428
```

In [69]:

```
x = china_electricity_consumption.drop(['Per capita electricity (kWh)', 'Entity'], axis
y = china_electricity_consumption['Per capita electricity (kWh)']
```

In [70]:

```
x.shape
```

Out[70]:

```
(36, 1)
```

In [71]:

```
y.shape
```

Out[71]:

```
(36,)
```

In [72]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [73]:

```
model_china=LinearRegression()
model_china.fit(x,y)
```

Out[73]:

```
LinearRegression()
```

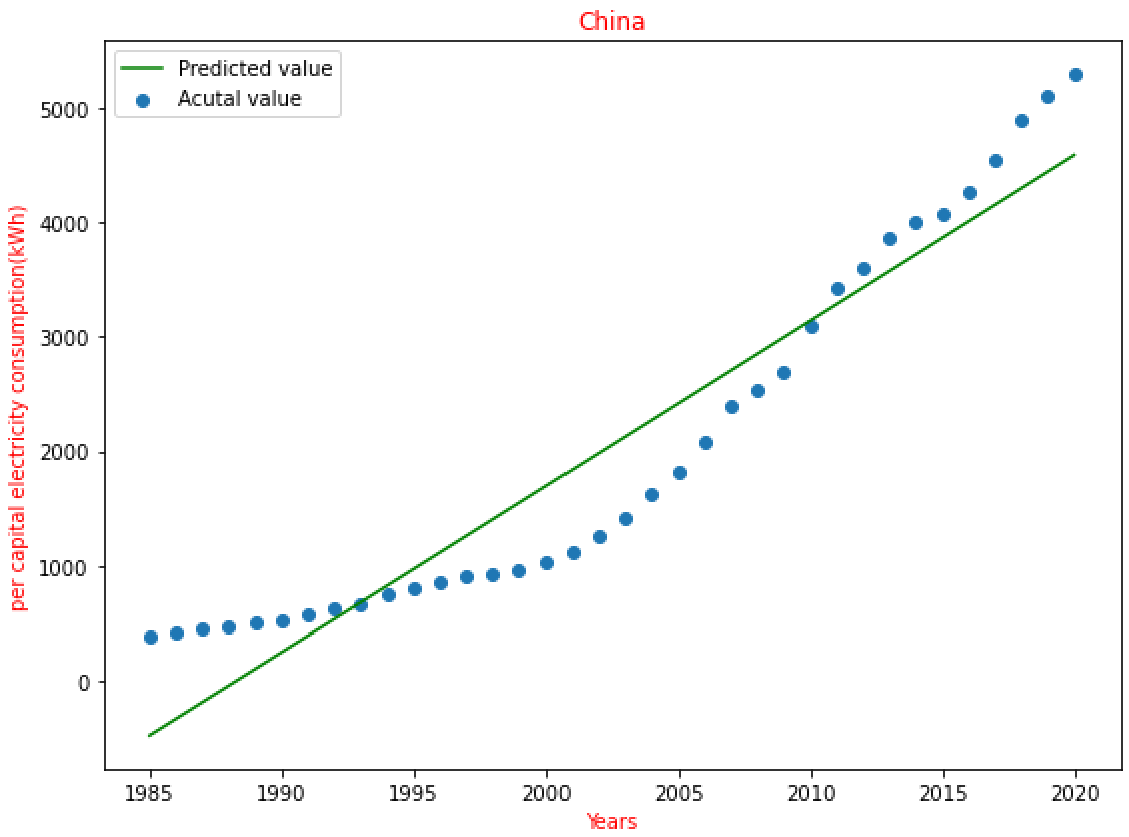
In [74]:

```
print("Coefficient: ",model_china.coef_)
print("intercept: ",model_china.intercept_)
pre = model_china.predict(x)
```

```
Coefficient:  [144.66853256]
intercept:  -287644.1720647791
```

In [75]:

```
plt.figure(figsize=(8,6))
plt.scatter(x,y,label='Acutal value')
plt.plot(x,pre,color='g',label='Predicted value')
plt.legend()
plt.title("China",color='r')
plt.xlabel("Years",color='r')
plt.ylabel("per capital electricity consumption(kWh)",color='r')
plt.tight_layout()
plt.show()
```



In [76]:

```
years=[2021,2022,2023,2024,2025]
for i in years:
    print(model_china.predict([[i]]))
```

```
[4730.93224127]
[4875.60077383]
[5020.26930639]
[5164.93783895]
[5309.60637151]
```

In [78]:

```
print("By function: ",mean_squared_error(y,model_china.predict(x)))
```

```
By function:  225359.5087121302
```

In [79]:

```
x = US_electricity_consumption.drop(['Per capita electricity (kWh)', 'Entity'], axis =
y = US_electricity_consumption['Per capita electricity (kWh)']
```

In [80]:

```
x.shape
```

Out[80]:

```
(36, 1)
```

In [81]:

```
y.shape
```

Out[81]:

```
(36,)
```

In [82]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [83]:

```
model_US=LinearRegression()
model_US.fit(x,y)
```

Out[83]:

```
LinearRegression()
```

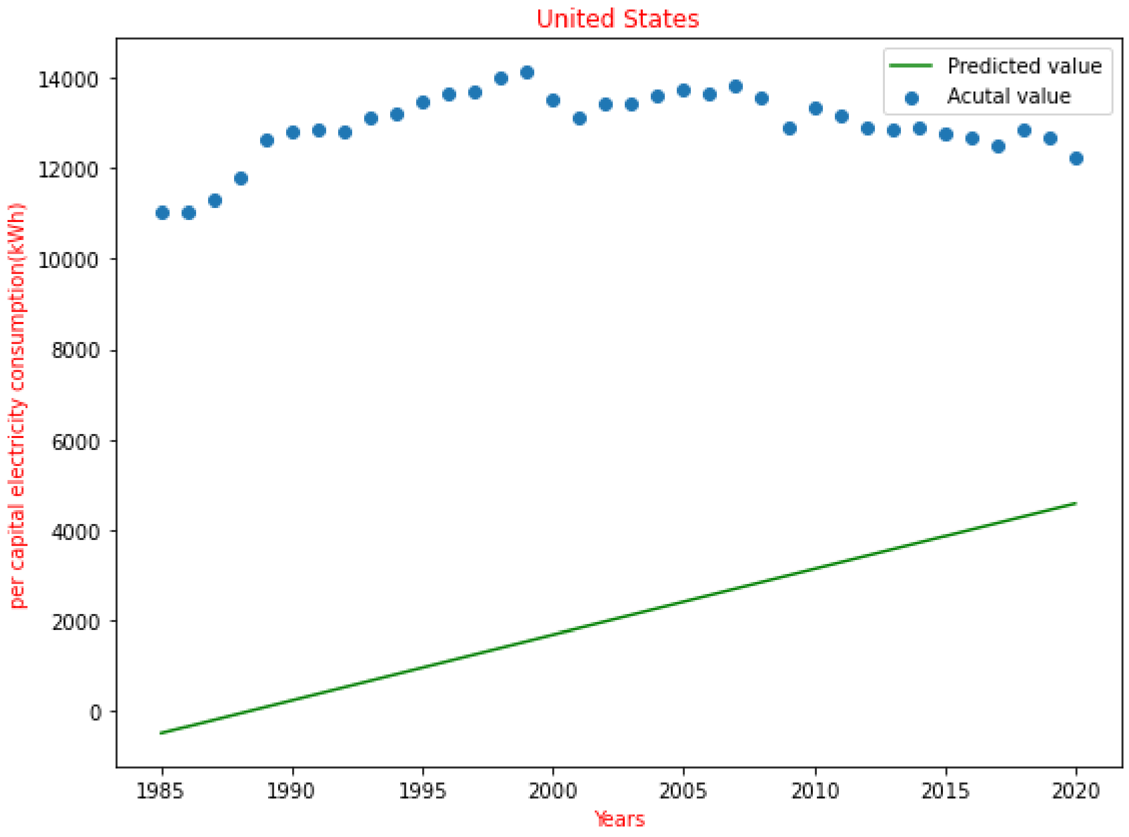
In [84]:

```
print("Coefficient: ",model_US.coef_)
print("intercept: ",model_US.intercept_)
pre = model_china.predict(x)
```

```
Coefficient:  [18.66015328]
intercept:  -24394.778085800142
```

In [85]:

```
plt.figure(figsize=(8,6))
plt.scatter(x,y,label='Acutal value')
plt.plot(x,pre,color='g',label='Predicted value')
plt.legend()
plt.title("United States",color='r')
plt.xlabel("Years",color='r')
plt.ylabel("per capital electricity consumption(kWh)",color='r')
plt.tight_layout()
plt.show()
```



In [86]:

```
years=[2021,2022,2023,2024,2025]
for i in years:
    print(model_US.predict([[i]]))
```

```
[13317.39169683]
[13336.05185011]
[13354.71200339]
[13373.37215667]
[13392.03230995]
```

In [87]:

```
print("By function: ",mean_squared_error(y,model_US.predict(x)))
```

```
By function:  518207.58128774806
```