Superstore Sales Prediction using Time Series Analysis

In [28]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
```

In [2]:

```
### Load the Dataset
data = pd.read_csv("D:/HelloTech Softwares - Data Science Intern Projects/Su
```

In [4]:

```
### Display first few rows
data.head()
```

Out[4]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States

In [5]:

```
### Display last 5 rows
data.tail(5)
```

Out[5]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Countr
9795	9796	CA-2017-125920	21/05/2017	28/05/2017	Standard Class	SH-19975	Sally Hughsby	Corporate	Unite State
9796	9797	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	Unite State
9797	9798	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	Unite State
9798	9799	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	Unite State
9799	9800	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	Unite State

In [9]:

```
### Identifying the column names

print(data.columns)
```

```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'Stat
e',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales'],
      dtype='object')
```

In [10]:

```
### Total no.of rows and columns

data.shape
```

Out[10]:

```
(9800, 18)
```

Data Preprocessing

In [11]:

```
### Checking null values

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

Out[11]:

```
Row ID            0
Order ID          0
Order Date        0
Ship Date         0
Ship Mode         0
Customer ID       0
Customer Name     0
Segment           0
Country           0
City              0
State             0
Postal Code      11
Region            0
Product ID        0
Category          0
Sub-Category      0
Product Name      0
Sales             0
dtype: int64
```

In [12]:

```
### Removing empty rows

data.dropna(inplace=True)
```

In [13]:

```
data.shape   # Size of rows and columns after removing empty rows
```

Out[13]:

```
(9789, 18)
```

Data Preparation

In [14]:

```
# Convert Order Date to datetime format

data['Order Date'] = pd.to_datetime(data['Order Date'],format = '%d/%m/%Y')
```

In [17]:

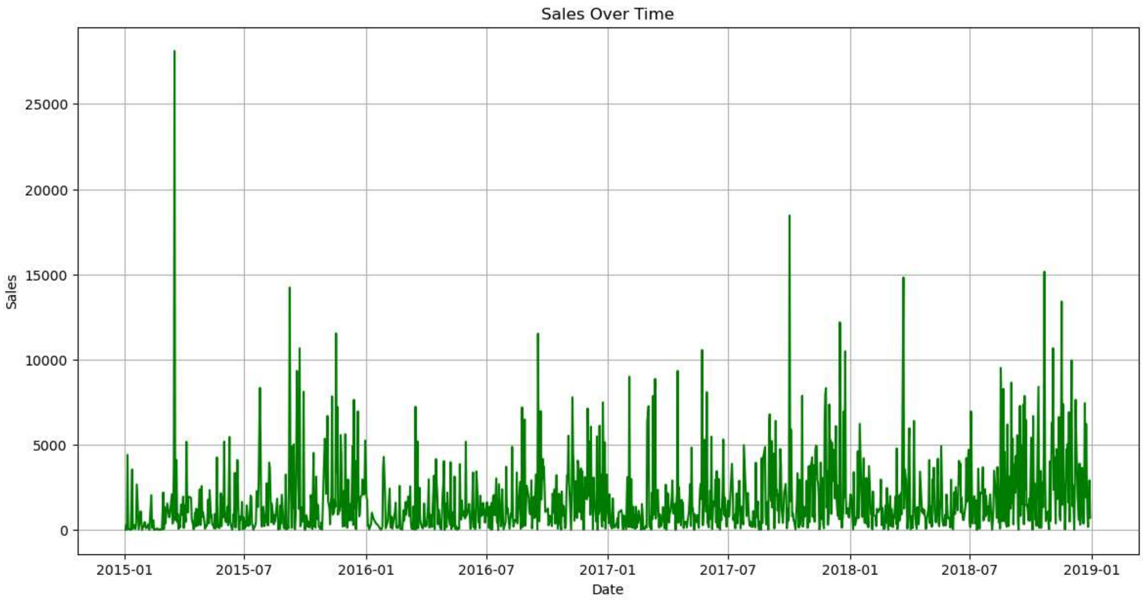
```
# Aggregate sales by order date

sales_data = data.groupby('Order Date')['Sales'].sum().reset_index()
```

Plotting

In [23]:
```
# Plot the time series data

plt.figure(figsize=(14,7))
plt.plot(sales_data['Order Date'],sales_data['Sales'],color = 'green')
plt.title('Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.grid(True)
plt.show()
```



In [24]:
```
# Set the date as index

sales_data.set_index('Order Date',inplace=True)
```

Modelling

In [25]:
```
# Split data into train and test sets

train_data,test_data = sales_data[:-30],sales_data[-30:]
```

Fit an ARIMA Model

In [29]:
```
model = ARIMA(train_data,order=(5,1,0)) # You may need to adjust the order (
model_fit=model.fit()
```

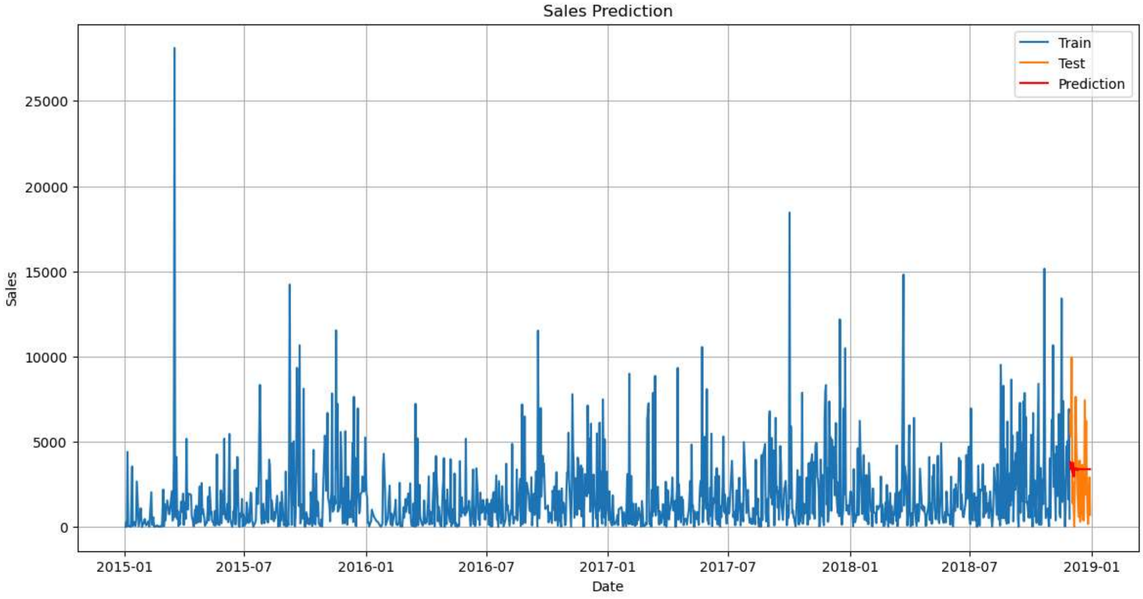
```
E:\Anaconda Software\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:4
71: ValueWarning: A date index has been provided, but it has no associated
frequency information and so will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
E:\Anaconda Software\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:4
71: ValueWarning: A date index has been provided, but it has no associated
frequency information and so will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
E:\Anaconda Software\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:4
71: ValueWarning: A date index has been provided, but it has no associated
frequency information and so will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

Prediction and Visualization

In [30]:
```
# Make predictions
pred = model_fit.forecast(steps=len(test_data))

# Plot the predictions vs actual sales
plt.figure(figsize=(14,7))
plt.plot(train_data.index, train_data, label = 'Train')
plt.plot(test_data.index, test_data, label = 'Test')
plt.plot(test_data.index, pred, label = 'Prediction', color='red')
plt.title('Sales Prediction')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.legend()
plt.grid(True)
plt.show()
```

```
E:\Anaconda Software\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:8
34: ValueWarning: No supported index is available. Prediction results will
be given with an integer index beginning at `start`.
  return get_prediction_index(
```



Mean Squared Error (MSE)

In [31]:
```
# Evaluate the model
mse = mean_squared_error(test_data,pred)
print(f' Mean Squared Error: {mse}')
```

```
 Mean Squared Error: 6261646.15971704
```