In [7]:
```
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 [8]:
```
# Load The Dateset
```

In [13]:
```
data = pd.read_csv("E:/Hello Tech DS Project/Superstore Sales Dataset/train.csv"
```

In [14]:
```
# Display first few rows
data.head()
```

Out[14]:

	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	Hende
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Hende
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Ang
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lauder
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Lauder

In [15]:
```
# Display last Last 5 rows
```

In [16]:
```
data.tail(5)
```

Out[16]:

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

In [17]:
```
# Identifying the column names
```

In [18]:
```
print(data.columns)
```

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

In [19]:
```
# total number of rows and columns
```

In [20]:
```
data.shape
```

Out[20]:
```
(9800, 18)
```

DATA PREPROCESSING

In [21]:
```
# Checking null values
```

In [22]:
```
data.isnull().sum()
```

Out[22]:
```
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 [23]:
```
# Removing empty rows
```

In [26]:
```
data.dropna(inplace=True)
```

In [27]:
```
# Size of rows and columns after removing empty rows
```

In [28]:
```
data.shape
```

Out[28]:
```
(9789, 18)
```

DATA PREPARATION

In [29]:
```
# Convert Order Data to datatime formate
```

In [34]:
```
data['Order Date'] = pd.to_datetime(data['Order Date'],format = '%d/%m/%Y')
```

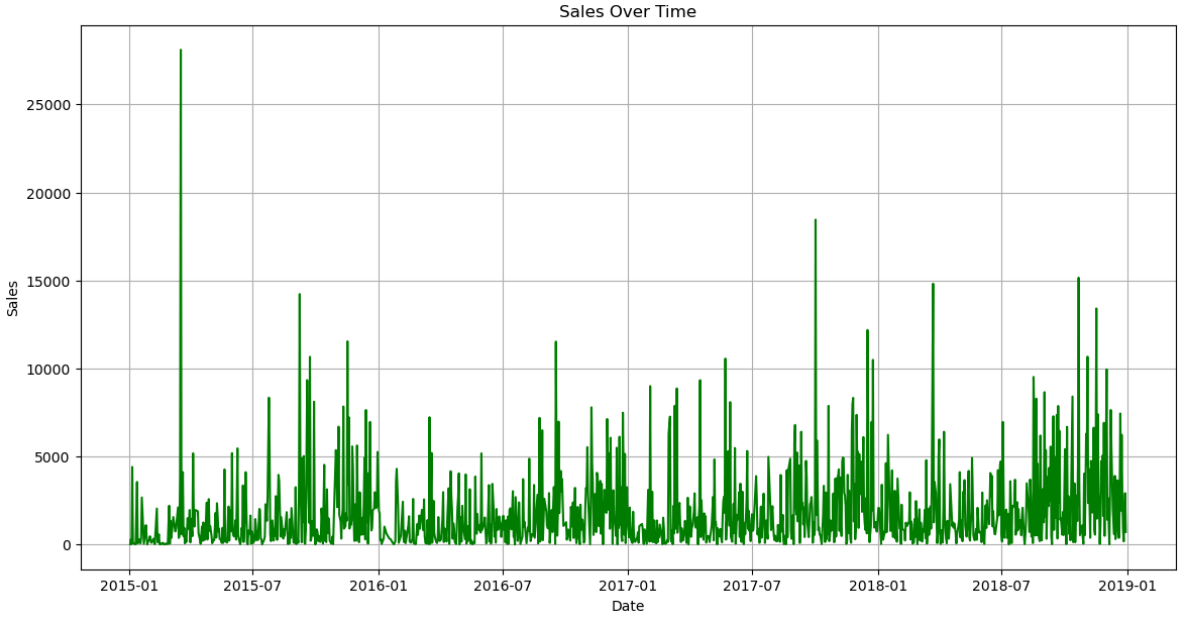
In [35]:
```
# Aggregate sales by order date
```

In [36]:
```
sales_data = data.groupby('Order Date')['Sales'].sum().reset_index()
```

Plotting

In [37]:
```
# Plot the time series data
```

In [40]:
```
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 [41]:
```
# Set the data as index
```

In [46]:
```
sales_data.set_index('Order Date',inplace=True)
```

MODELLING

In [47]:
```
# Split Data into train and test sets
```

In [48]:
```
train_data,test_data = sales_data[:-30],sales_data[-30:]
```

Fit an ARIMA Model

In [51]:
```
# You may need to adjust the order
```

In [53]:
```
model = ARIMA(train_data,order=(5,1,0))
model_fit=model.fit()
```

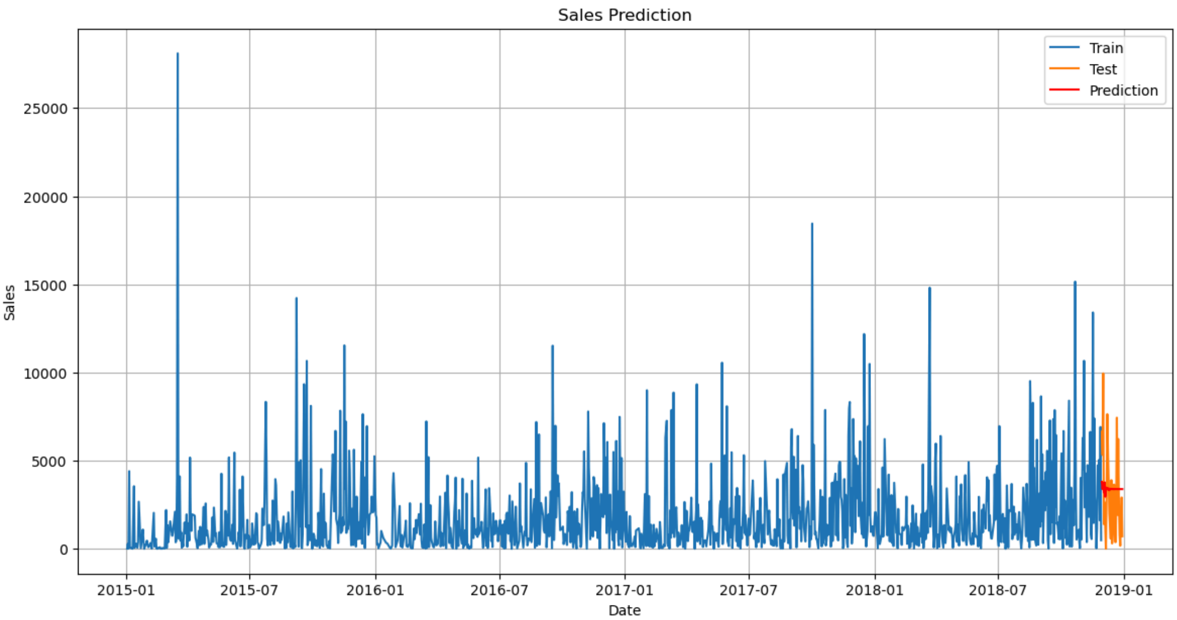
```
E:\anaconda3\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarn
ing: A date index has been provided, but it has no associated frequency informa
tion and so will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
E:\anaconda3\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarn
ing: A date index has been provided, but it has no associated frequency informa
tion and so will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
E:\anaconda3\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarn
ing: A date index has been provided, but it has no associated frequency informa
tion and so will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

PREDICTION AND VISUALIZATION

In [57]:
```
# 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:\anaconda3\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:836: ValueWarn
ing: 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 [59]:
```
# Evaluate the model
mse = mean_squared_error(test_data,pred)
print(f' Mean Squared Error: {mse}')
```

```
 Mean Squared Error: 6261646.15971704
```

In [ ]:
```

```