In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


# Load the CSV file
file_path = 'C:/Users/91876/Desktop/CODING/PW Data Science/Data Analysis/Crop Produ
crop_data = pd.read_csv(file_path)
```

In [2]:
```
# Display basic information and the first few rows
print(crop_data.info())
print(crop_data.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246091 entries, 0 to 246090
Data columns (total 7 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   State_Name     246091 non-null  object 
 1   District_Name  246091 non-null  object 
 2   Crop_Year      246091 non-null  int64  
 3   Season         246091 non-null  object 
 4   Crop           246091 non-null  object 
 5   Area           246091 non-null  float64
 6   Production     242361 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 13.1+ MB
None
                   State_Name District_Name  Crop_Year       Season  \
0  Andaman and Nicobar Islands      NICOBARS       2000  Kharif       
1  Andaman and Nicobar Islands      NICOBARS       2000  Kharif       
2  Andaman and Nicobar Islands      NICOBARS       2000  Kharif       
3  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year   
4  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year   

                 Crop    Area  Production  
0             Arecanut  1254.0      2000.0  
1  Other Kharif pulses     2.0         1.0  
2                 Rice   102.0       321.0  
3               Banana   176.0       641.0  
4            Cashewnut   720.0       165.0  
```

In [3]:
```
# Check for missing values
missing_values = crop_data.isnull().sum()
print("Missing values in each column:\n", missing_values)
```

```
Missing values in each column:
 State_Name          0
District_Name       0
Crop_Year           0
Season              0
Crop                0
Area                0
Production       3730
dtype: int64
```

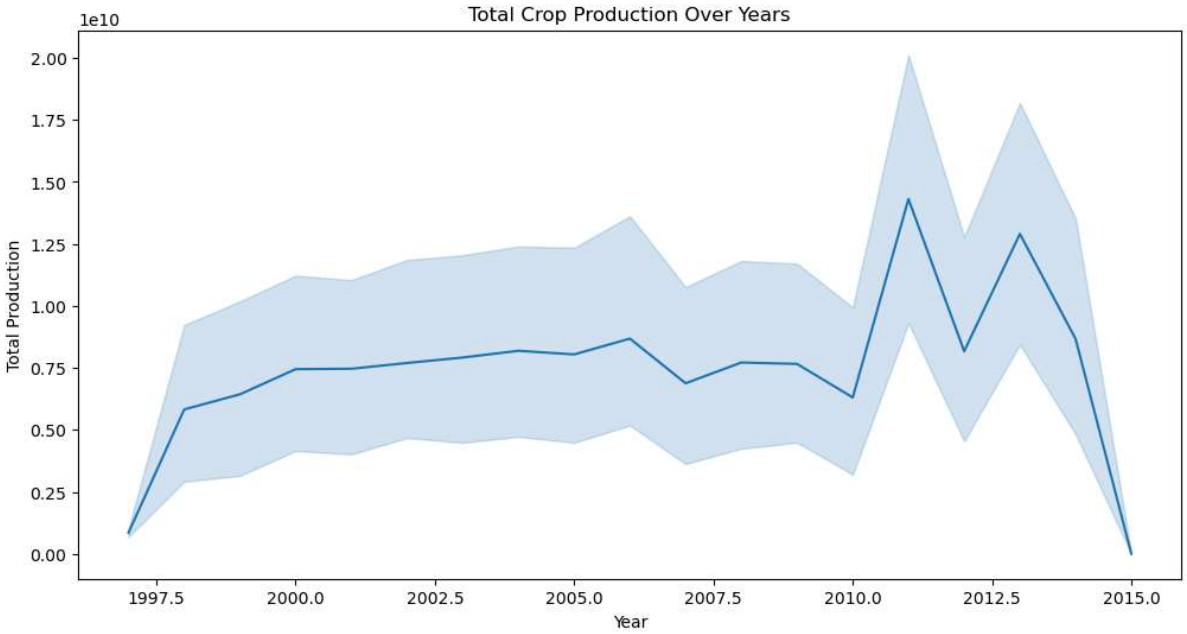
In [4]:
```
# Handling missing values in 'Production'
crop_data_clean = crop_data.dropna(subset=['Production'])
print("Data after handling missing values:\n", crop_data_clean.info())
# Summary statistics
print(crop_data_clean.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 242361 entries, 0 to 246090
Data columns (total 7 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   State_Name     242361 non-null  object 
 1   District_Name  242361 non-null  object 
 2   Crop_Year      242361 non-null  int64  
 3   Season         242361 non-null  object 
 4   Crop           242361 non-null  object 
 5   Area           242361 non-null  float64
 6   Production     242361 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 14.8+ MB
Data after handling missing values:
 None
           Crop_Year          Area    Production
count  242361.000000  2.423610e+05  2.423610e+05
mean     2005.625773  1.216741e+04  5.825034e+05
std         4.958285  5.085744e+04  1.706581e+07
min      1997.000000  1.000000e-01  0.000000e+00
25%      2002.000000  8.700000e+01  8.800000e+01
50%      2006.000000  6.030000e+02  7.290000e+02
75%      2010.000000  4.545000e+03  7.023000e+03
max      2015.000000  8.580100e+06  1.250800e+09
```

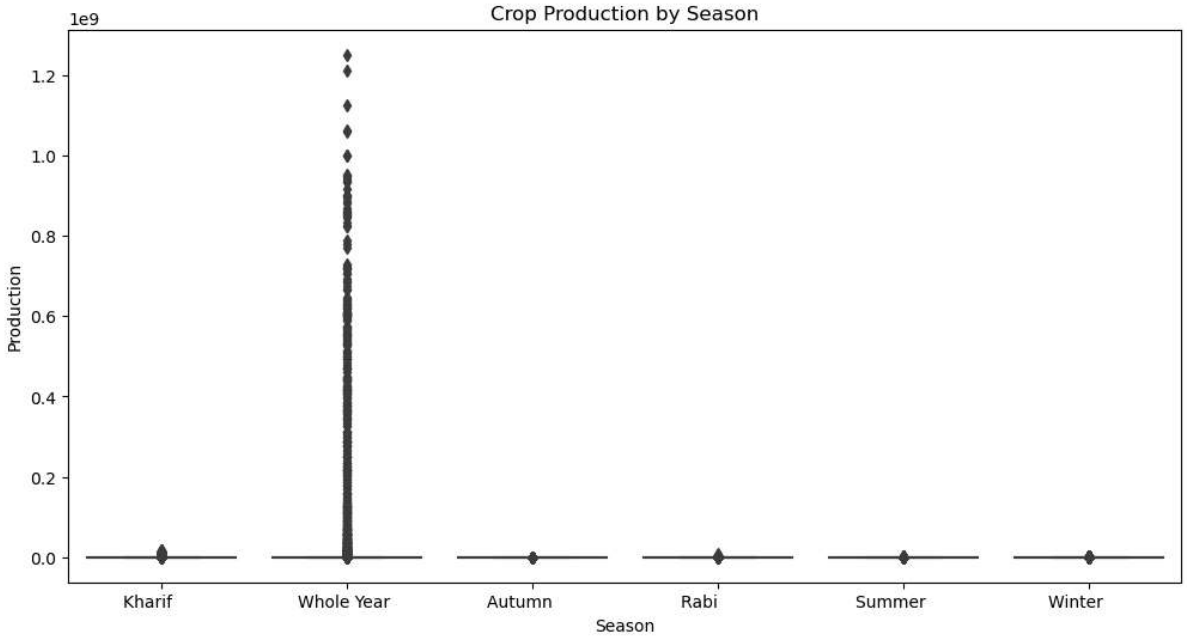
In [5]:
```
# Unique values in categorical columns
print("Unique values in 'State_Name':", crop_data_clean['State_Name'].nunique())
print("Unique values in 'District_Name':", crop_data_clean['District_Name'].nunique
print("Unique values in 'Crop_Year':", crop_data_clean['Crop_Year'].nunique())
print("Unique values in 'Season':", crop_data_clean['Season'].unique())
print("Unique values in 'Crop':", crop_data_clean['Crop'].nunique())
```

```
Unique values in 'State_Name': 33
Unique values in 'District_Name': 646
Unique values in 'Crop_Year': 19
Unique values in 'Season': ['Kharif     ' 'Whole Year ' 'Autumn     ' 'Rabi
' 'Summer     '
 'Winter     ']
Unique values in 'Crop': 124
```

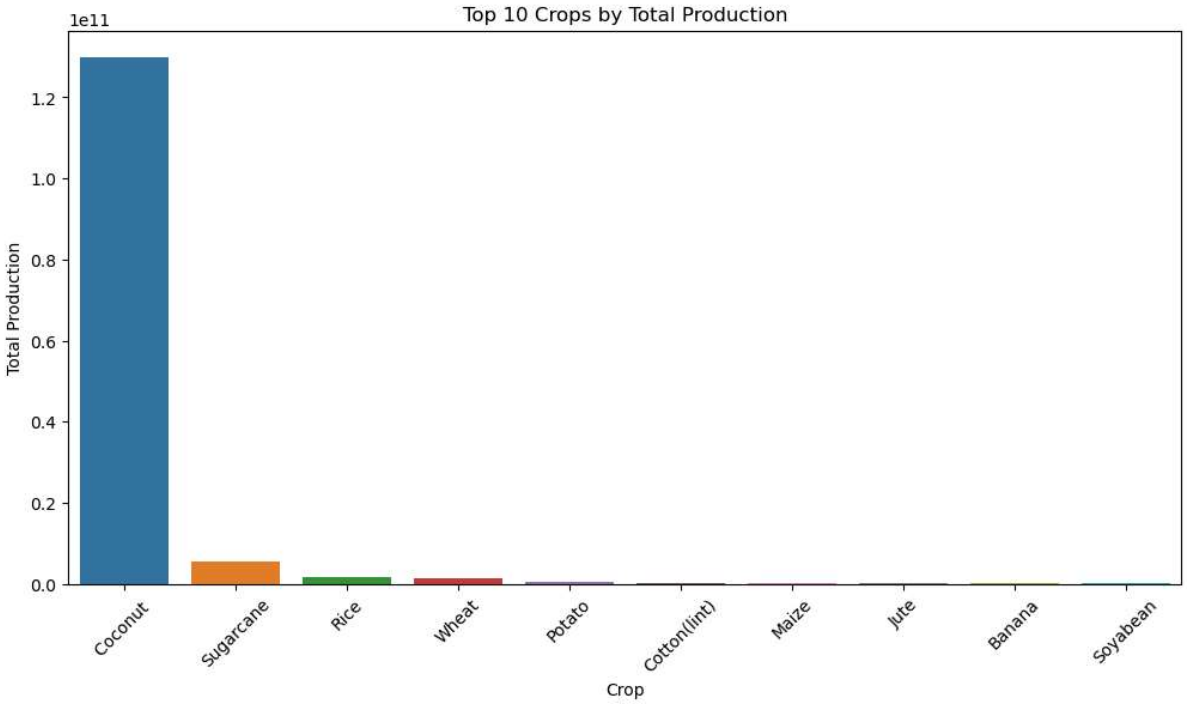
In [6]:
```
# Production over the years
plt.figure(figsize=(12, 6))
sns.lineplot(data=crop_data_clean, x='Crop_Year', y='Production', estimator='sum')
plt.title('Total Crop Production Over Years')
plt.xlabel('Year')
plt.ylabel('Total Production')
plt.show()
```



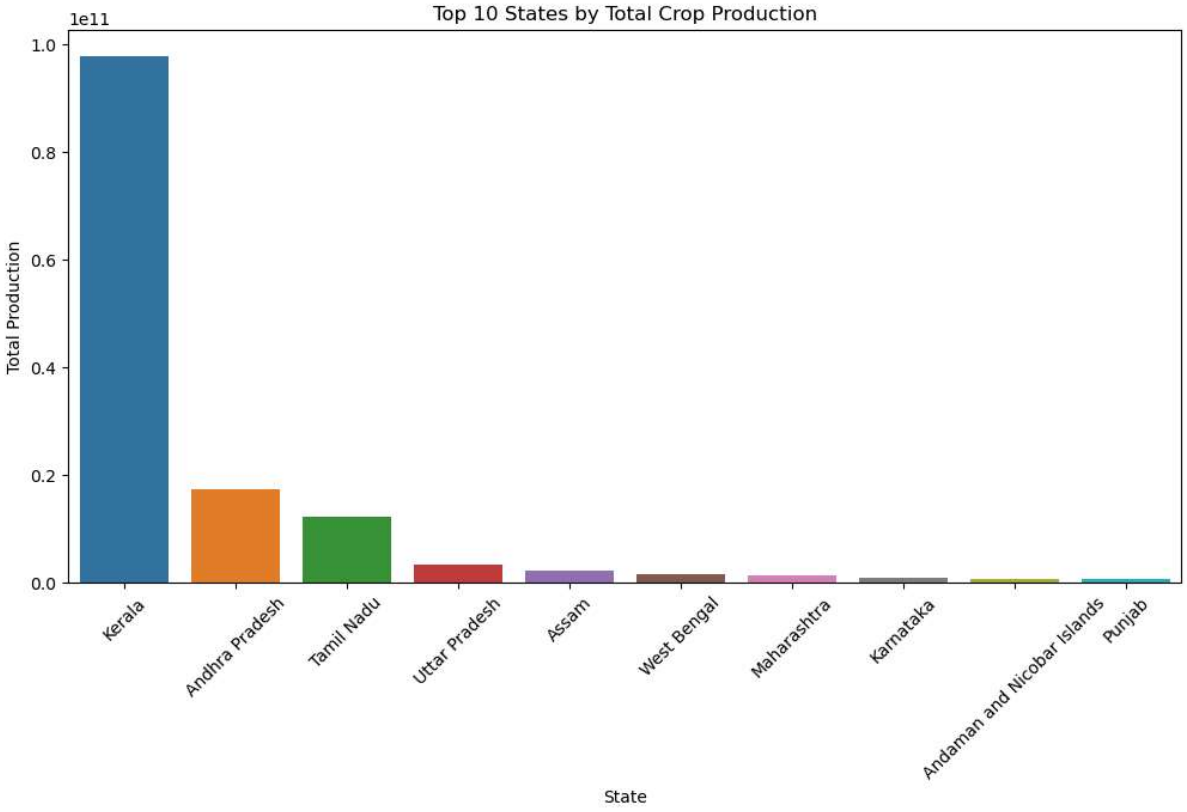
In [7]:
```
# Production by season
plt.figure(figsize=(12, 6))
sns.boxplot(data=crop_data_clean, x='Season', y='Production')
plt.title('Crop Production by Season')
plt.xlabel('Season')
plt.ylabel('Production')
plt.show()
```



In [8]:
```
# Top 10 crops by production
top_crops = crop_data_clean.groupby('Crop')['Production'].sum().nlargest(10).reset_
plt.figure(figsize=(12, 6))
sns.barplot(data=top_crops, x='Crop', y='Production')
plt.title('Top 10 Crops by Total Production')
plt.xlabel('Crop')
plt.ylabel('Total Production')
plt.xticks(rotation=45)
plt.show()
```



In [9]:
```
# Production by state
top_states = crop_data_clean.groupby('State_Name')['Production'].sum().nlargest(10)
plt.figure(figsize=(12, 6))
sns.barplot(data=top_states, x='State_Name', y='Production')
plt.title('Top 10 States by Total Crop Production')
plt.xlabel('State')
plt.ylabel('Total Production')
plt.xticks(rotation=45)
plt.show()
```



In [10]:
```
# Scatter plot of Area vs Production
plt.figure(figsize=(12, 6))
sns.scatterplot(data=crop_data_clean, x='Area', y='Production', hue='Season', alpha
plt.title('Area vs Production by Season')
plt.xlabel('Area (hectares)')
```

```
plt.ylabel('Production (tonnes)')
plt.legend(loc='upper right')
plt.show()

# Save the cleaned data to a new CSV file
crop_data_clean.to_csv('Crop_Production_Data_Cleaned.csv', index=False)
```