

crop-production-project

August 30, 2024

```
[1]: from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

0.1 Importing the Necessary Libraries

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

0.1.1 Function for Data Preprocessing

```
[3]: def data_preprocessing(path):
    # Load the dataset
    data = pd.read_csv(path)
    # Remove null values
    data_cleaned = data.dropna()
    # Display missing values
    print(data_cleaned.isnull().sum())
    # Display summary statistics
    print(data_cleaned.describe())
    return data_cleaned
```

0.1.2 Function for plotting year wise crop production trend

```
[4]: def Year_wise_crop_production_trend(data):
    plt.figure(figsize=(10, 5))
    sns.lineplot(x='Crop_Year', y='Production', data=data)
    plt.title('Year wise crop production trend')
    plt.xlabel('Year')
    plt.ylabel('Production')
    plt.show()
```

0.1.3 Function for plotting crop wise crop production

```
[5]: def crop_wise_crop_production(data):  
    plt.figure(figsize=(20, 5))  
    sns.barplot(x='Crop', y='Production', data=data, estimator=sum)  
    plt.title('Crop wise crop production')  
    plt.xlabel('Crop')  
    plt.ylabel('Production')  
    plt.xticks(rotation=90)  
    plt.show()
```

0.1.4 Function for plotting state wise crop production

```
[6]: def state_wise_crop_production(data):  
    plt.figure(figsize=(10, 5))  
    sns.barplot(x='State_Name', y='Production', data=data, estimator=sum)  
    plt.title('State wise crop production')  
    plt.xlabel('State')  
    plt.ylabel('Production')  
    plt.xticks(rotation=90)  
    plt.show()
```

0.1.5 Function for plotting season wise crop production

```
[7]: def season_wise_crop_production(data):  
    plt.figure(figsize=(10, 5))  
    sns.barplot(x='Season', y='Production', data=data, estimator=sum)  
    plt.title('season_wise_crop_production')  
    plt.xlabel('Season')  
    plt.ylabel('Production')  
    plt.show()
```

0.1.6 Function for plotting distribution of crop production

```
[8]: def distribution_of_crop_production(data):  
    plt.figure(figsize=(10, 5))  
    sns.histplot(data['Production'], bins=30, kde=True)  
    plt.title('distribution_of_crop_production')  
    plt.xlabel('Production')  
    plt.ylabel('Count')  
    plt.show()
```

```
[9]: path = "/content/drive/MyDrive/Unified Mentor/Crop Production data.csv"  
    data_cleaned = data_preprocessing(path)
```

```
State_Name      0  
District_Name   0
```

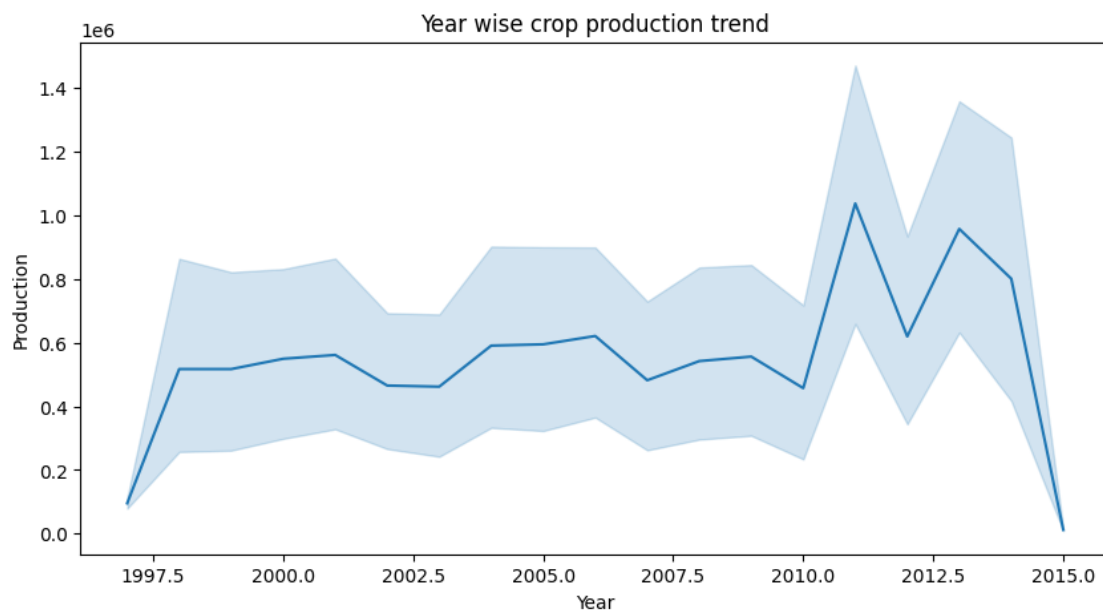
```

Crop_Year      0
Season          0
Crop            0
Area            0
Production      0
dtype: int64

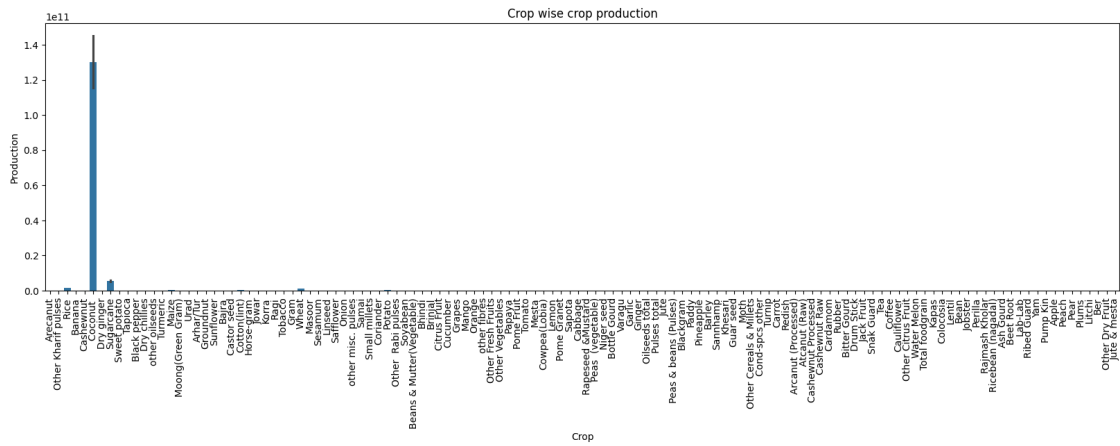
```

	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

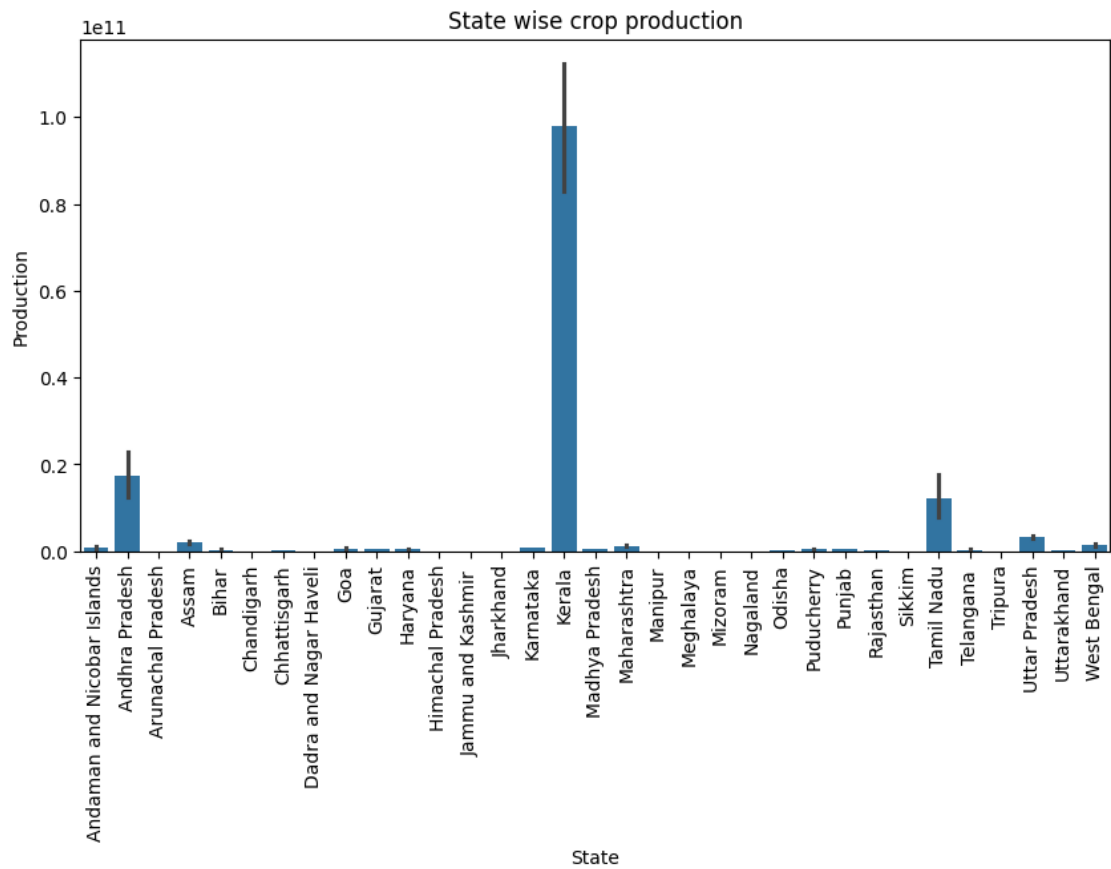
```
[10]: Year_wise_crop_production_trend(data_cleaned)
```



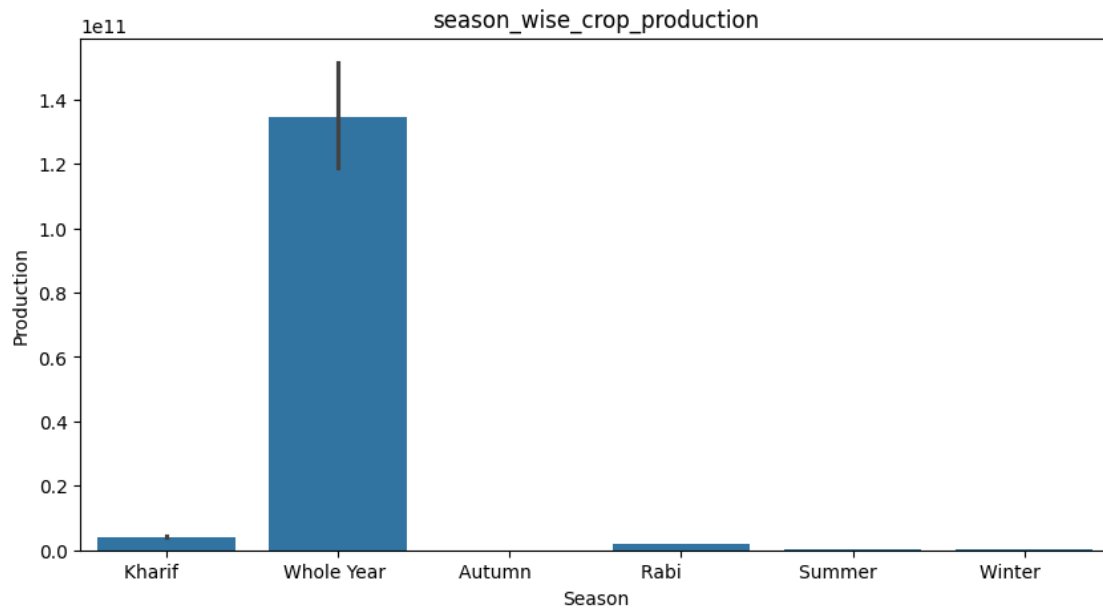
```
[11]: crop_wise_crop_production(data_cleaned)
```



```
[12]: state_wise_crop_production(data_cleaned)
```



```
[13]: season_wise_crop_production(data_cleaned)
```



```
[14]: distribution_of_crop_production(data_cleaned)
```

