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

```
faostat_df = pd.read_csv("/content/FAOSTAT_data_en_11-19-2025 .csv")
crop_df = pd.read_csv("/content/Crop_recommendation .csv")
```

```
faostat_india = faostat_df[faostat_df["Area"] == "India"].copy()
```

```
faostat_india["crop"] = (
faostat_india["Item"]
.str.split(",")
.apply(lambda lst: [x.strip() for x in lst])
)


faostat_exploded = faostat_india.explode("crop").reset_index(drop=True)
```

```
faostat_exploded = faostat_exploded.rename(columns={"Item": "item"})
crop_df = crop_df.rename(columns={"label": "crop"})


print("Crop columns:", crop_df.columns)
print("FAOSTAT columns:", faostat_exploded.columns)
```

```
Crop columns: Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'crop'], dtype='object')
FAOSTAT columns: Index(['Domain Code', 'Domain', 'Area Code (M49)', 'Area', 'Element Code',
       'Element', 'Item Code (CPC)', 'item', 'Year Code', 'Year', 'Unit',
       'Value', 'Flag', 'Flag Description', 'Note', 'crop'],
      dtype='object')
```

```
label_to_fao = {
"apple": "Apples",
"banana": "Bananas",
"chickpea": "Chick peas",
"coconut": "Coconuts",
"coffee": "Coffee",
"cotton": "Seed cotton",
"grapes": "Grapes",
"jute": "Jute",
"lentil": "Lentils",
"maize": "Maize (corn)",
"mango": "Mangoes",
"mothbeans": "Beans",
"muskmelon": "Cantaloupes and other melons",
"orange": "Oranges",
"papaya": "Papayas",
"pigeonpeas": "Pigeon peas",
"rice": "Rice",
"watermelon": "Watermelons",
}


crop_df["FAO_name"] = crop_df["crop"].map(label_to_fao)
faostat_exploded["FAO_name"] = faostat_exploded["crop"]
```

```
crop_mapped = crop_df[~crop_df["FAO_name"].isna()].copy()
print("Mapped crops:", sorted(crop_mapped["crop"].unique()))
print("Number of mapped crops:", crop_mapped["crop"].nunique())
```

```
Mapped crops: ['apple', 'banana', 'chickpea', 'coconut', 'coffee', 'cotton', 'grapes', 'jute', 'lentil', 'maize', 'mango', 'moth
Number of mapped crops: 18
```

```
merged = faostat_exploded.merge(
crop_mapped,
on="FAO_name",
how="inner",
suffixes=("_fao", "_ml")
)


merged["crop"] = merged["crop_ml"]


print("Merged shape:", merged.shape)
print("Unique crops after merge:", merged["crop"].nunique())
print(sorted(merged["crop"].unique()))
```

```
Merged shape: (129600, 26)
Unique crops after merge: 18
['apple', 'banana', 'chickpea', 'coconut', 'coffee', 'cotton', 'grapes', 'jute', 'lentil', 'maize', 'mango', 'mothbeans', 'muskm
```

```
merged_prod = merged[merged["Element"] == "Production"].copy()
print("Production shape:", merged_prod.shape)
```

```
Production shape: (43200, 26)
```

```
cols_keep = [
"Element", "Value", "N", "P", "K",
"temperature", "humidity", "ph", "rainfall", "crop"
]


final_df = merged_prod[cols_keep].copy()
print("Final DF shape:", final_df.shape)
print(final_df.head())
```

```
Final DF shape: (43200, 10)
        Element      Value   N    P    K  temperature   humidity        ph  \
200  Production  1050000.0  24  128  196    22.750888  90.694892  5.521467
201  Production  1050000.0   7  144  197    23.849401  94.348150  6.133221
202  Production  1050000.0  14  128  205    22.608010  94.589006  6.226290
203  Production  1050000.0   8  120  201    21.186674  91.134357  6.321152
204  Production  1050000.0  20  129  201    23.410447  91.699133  5.587906

       rainfall   crop
200  110.431786  apple
201  114.051249  apple
202  116.039659  apple
203  122.233323  apple
204  116.077793  apple
```

```
print("Unique crops in final_df:")
print(sorted(final_df["crop"].unique()))
print("Number of unique crops:", final_df["crop"].nunique())
```

```
Unique crops in final_df:
['apple', 'banana', 'chickpea', 'coconut', 'coffee', 'cotton', 'grapes', 'jute', 'lentil', 'maize', 'mango', 'mothbeans', 'muskm
Number of unique crops: 18
```

```
final_df
```

	Element	Value	N	P	K	temperature	humidity	ph	rainfall	crop
200	Production	1050000.0	24	128	196	22.750888	90.694892	5.521467	110.431786	apple
201	Production	1050000.0	7	144	197	23.849401	94.348150	6.133221	114.051249	apple
202	Production	1050000.0	14	128	205	22.608010	94.589006	6.226290	116.039659	apple
203	Production	1050000.0	8	120	201	21.186674	91.134357	6.321152	122.233323	apple
204	Production	1050000.0	20	129	201	23.410447	91.699133	5.587906	116.077793	apple
...	...	...	...	...	...	...	...	...	...	...
129595	Production	3626000.0	97	12	47	25.287846	89.636679	6.765095	58.286977	watermelon
129596	Production	3626000.0	110	7	45	26.638386	84.695469	6.189214	48.324286	watermelon
129597	Production	3626000.0	96	18	50	25.331045	84.305338	6.904242	41.532187	watermelon
129598	Production	3626000.0	83	23	55	26.897502	83.892415	6.463271	43.971937	watermelon
129599	Production	3626000.0	120	24	47	26.986037	89.413849	6.260839	58.548767	watermelon

43200 rows × 10 columns

Next steps: Generate code with `final_df` New interactive sheet

```
print("Null counts:")
print(final_df.isna().sum())
```

```
Null counts:
Element        0
Value          0
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
crop           0
dtype: int64
```

```
print("Duplicate rows:", final_df.duplicated().sum())
final_df = final_df.drop_duplicates().reset_index(drop=True)
print("Shape after removing duplicates:", final_df.shape)
```

```
Duplicate rows: 600
Shape after removing duplicates: (42600, 10)
```

```
numeric_cols = ["Value", "N", "P", "K", "temperature", "humidity", "ph", "rainfall"]


for col in numeric_cols:
  Q1 = final_df[col].quantile(0.25)
  Q3 = final_df[col].quantile(0.75)
  IQR = Q3 - Q1
  low = Q1 - 1.5 * IQR
  high = Q3 + 1.5 * IQR
  final_df = final_df[(final_df[col] >= low) & (final_df[col] <= high)]


print("Shape after outlier removal:", final_df.shape)
```

```
Shape after outlier removal: (31037, 10)
```

```
print("Unique crops:", sorted(final_df["crop"].unique()))
print("Number of unique crops:", final_df["crop"].nunique())
print(final_df["crop"].value_counts())
```

```
Unique crops: ['banana', 'chickpea', 'coconut', 'coffee', 'cotton', 'jute', 'lentil', 'maize', 'mango', 'mothbeans', 'muskmelon'
Number of unique crops: 15
crop
coconut       2400
jute          2400
lentil        2400
watermelon    2400
cotton        2400
```

```
coffee        2300
muskmelon     2300
mango         2232
chickpea      2088
pigeonpeas    2016
maize         2000
banana        1700
orange        1617
papaya        1488
mothbeans     1296
Name: count, dtype: int64
```

```
final_df = final_df.drop(columns=["Element"], errors="ignore")
```

```
numeric_features = ['N','P','K','temperature','humidity','ph','rainfall']
ranges = {feature: (final_df[feature].min(), final_df[feature].max()) for feature in numeric_features}
print(ranges)
```

```
{'N': (0, 140), 'P': (5, 95), 'K': (5, 85), 'temperature': (16.39624284, 36.32268069), 'humidity': (14.25803981, 99.98187601), '
```

```
final_df.to_csv("Clean_dataset_15.csv", index=False)
print("Dataset saved successfully")
```

```
Dataset saved successfully
```

```
print("Rows:", final_df.shape[0])
print("Columns:", final_df.shape[1])
```

```
Rows: 31037
Columns: 9
```

```
print("Null counts:")
print(final_df.isna().sum())
```

```
Null counts:
Value          0
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
crop           0
dtype: int64
```