competition

April 5, 2024

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

[4]:
```
from google.colab import files
files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving State_wise_rice_production_in_India.csv to
State_wise_rice_production_in_India.csv
```

[4]:
```
{'State_wise_rice_production_in_India.csv': b'"TABLE 60: STATE-WISE PRODUCTION
OF FOODGRAINS - RICE\n(Thousand Tonnes)",,,,,,,,,\nState/Union Territory,2004-05
,2005-06,2006-07,2007-08,2008-09,2009-10,2010-11,2011-12,2012-13\nAndhra Pradesh
,9601.0,11704.0,11872.0,13324.0,14241.0,10538.0,7882.4,7746.2,6862.4\nArunachal
Pradesh,135.0,146.2,146.2,158.1,163.9,215.8,234.0,255.0,263.0\nAssam,3470.7,3552
.5,2916.0,3319.0,4008.5,4335.9,4736.6,4516.3,5128.5\nBihar,2472.2,3495.5,4989.3,
4418.1,5590.3,3599.3,3102.1,7162.6,7529.3\nChhattisgarh,4383.3,5011.6,5041.4,542
6.6,4391.8,4110.4,6159.0,6028.4,6608.8\nNCT of Delhi,14.3,24.0,31.1,31.4,31.4,19
.3,19.6,19.8,19.7\nGoa,145.2,147.3,130.3,121.6,123.3,100.6,115.0,121.8,122.8\nGu
jarat,1238.2,1298.0,1390.0,1474.0,1303.0,1292.0,1496.6,1790.0,1541.0\nHaryana,30
23.0,3210.0,3371.0,3613.0,3298.0,3625.0,3472.0,3759.0,3976.0\nHimachal
Pradesh,122.0,112.1,123.5,121.5,118.3,105.9,128.9,131.6,125.3\nJammu & Kashmir,4
92.2,556.8,554.0,561.3,563.1,497.4,507.7,544.7,818.1\nJharkhand,1677.0,1558.0,29
67.8,3336.4,3420.2,1538.4,1110.0,3130.6,3164.9\nKarnataka,3547.0,5744.0,3446.0,3
717.0,3802.0,3691.0,4188.0,3955.0,3364.0\nKerala,667.1,629.9,631.0,528.5,590.3,5
98.3,522.7,569.0,508.3\nMadhya Pradesh,1169.0,1656.3,1368.4,1461.9,1559.7,1260.6
,1772.1,2227.3,2775.0\nMaharashtra,2164.0,2695.0,2569.0,2996.0,2284.0,2183.0,269
6.0,2841.0,3057.0\nManipur,435.9,386.1,386.1,406.2,397.0,319.9,521.7,591.0,257.6
\nMeghalaya,193.7,151.9,200.2,200.0,203.9,206.7,207.0,216.5,232.0\nMizoram,104.1
,99.2,29.5,15.7,46.0,44.3,47.2,54.3,30.5\nNagaland,259.8,263.1,263.5,290.6,345.1
,240.3,381.4,382.4,405.2\nOdisha,6466.0,6859.0,6824.7,7540.7,6812.7,6917.5,6827.
7,5807.0,7295.5\nPuducherry,65.7,59.9,59.9,53.4,50.8,52.4,52.0,42.1,46.5\nPunjab
,10437.0,10193.0,10138.0,10489.0,11000.0,11236.0,10837.0,10542.0,11374.0\nRajast
han,150.4,153.0,169.8,259.6,241.1,228.3,265.5,253.4,222.5\nSikkim,21.6,21.5,21.5
,22.9,21.7,24.3,21.0,20.9,21.3\nTamil Nadu,5062.2,5220.0,6610.6,5040.2,5182.7,56
```

```
65.2,5792.4,7458.7,4049.9\nTelangana,.,.,.,.,.,.,.,6535.6,5148.8,4647.6\nTripura,5
45.1,552.9,620.5,624.6,627.1,640.0,702.5,718.3,713.2\nUttar Pradesh,9555.6,11133
.7,11124.0,11780.0,13097.0,10807.1,11992.0,14022.0,14416.0\nUttarakhand,572.0,59
0.0,556.0,593.0,582.0,608.0,550.4,594.0,579.8\nWest Bengal,14884.8,14510.8,14745
.9,14719.5,15037.3,14340.7,13045.9,14605.8,15023.7\nALL INDIA,83131.7,91793.4,93
355.3,96692.9,99182.5,89092.9,95979.8,105310.9,105231.6\n'}
```

[5]:
```
# Data
df = pd.read_csv("State_wise_rice_production_in_India.csv")
```

[6]:
```
df.head()
```

[6]:
```
  TABLE 60: STATE-WISE PRODUCTION OF FOODGRAINS - RICE\n(Thousand Tonnes)  \
0                              State/Union Territory
1                                     Andhra Pradesh
2                                  Arunachal Pradesh
3                                              Assam
4                                              Bihar

  Unnamed: 1 Unnamed: 2 Unnamed: 3 Unnamed: 4 Unnamed: 5 Unnamed: 6  \
0    2004-05    2005-06    2006-07    2007-08    2008-09    2009-10
1     9601.0    11704.0    11872.0    13324.0    14241.0    10538.0
2      135.0      146.2      146.2      158.1      163.9      215.8
3     3470.7     3552.5     2916.0     3319.0     4008.5     4335.9
4     2472.2     3495.5     4989.3     4418.1     5590.3     3599.3

  Unnamed: 7 Unnamed: 8 Unnamed: 9
0    2010-11    2011-12    2012-13
1     7882.4     7746.2     6862.4
2      234.0      255.0      263.0
3     4736.6     4516.3     5128.5
4     3102.1     7162.6     7529.3
```

[7]:
```
df.describe
```

[7]:
```
<bound method NDFrame.describe of    TABLE 60: STATE-WISE PRODUCTION OF
FOODGRAINS - RICE\n(Thousand Tonnes)  \
0                              State/Union Territory
1                                     Andhra Pradesh
2                                  Arunachal Pradesh
3                                              Assam
4                                              Bihar
5                                       Chhattisgarh
6                                       NCT of Delhi
7                                                Goa
8                                            Gujarat
9                                            Haryana
```

10	Himachal Pradesh
11	Jammu & Kashmir
12	Jharkhand
13	Karnataka
14	Kerala
15	Madhya Pradesh
16	Maharashtra
17	Manipur
18	Meghalaya
19	Mizoram
20	Nagaland
21	Odisha
22	Puducherry
23	Punjab
24	Rajasthan
25	Sikkim
26	Tamil Nadu
27	Telangana
28	Tripura
29	Uttar Pradesh
30	Uttarakhand
31	West Bengal
32	ALL INDIA

	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6
0	2004-05	2005-06	2006-07	2007-08	2008-09	2009-10
1	9601.0	11704.0	11872.0	13324.0	14241.0	10538.0
2	135.0	146.2	146.2	158.1	163.9	215.8
3	3470.7	3552.5	2916.0	3319.0	4008.5	4335.9
4	2472.2	3495.5	4989.3	4418.1	5590.3	3599.3
5	4383.3	5011.6	5041.4	5426.6	4391.8	4110.4
6	14.3	24.0	31.1	31.4	31.4	19.3
7	145.2	147.3	130.3	121.6	123.3	100.6
8	1238.2	1298.0	1390.0	1474.0	1303.0	1292.0
9	3023.0	3210.0	3371.0	3613.0	3298.0	3625.0
10	122.0	112.1	123.5	121.5	118.3	105.9
11	492.2	556.8	554.0	561.3	563.1	497.4
12	1677.0	1558.0	2967.8	3336.4	3420.2	1538.4
13	3547.0	5744.0	3446.0	3717.0	3802.0	3691.0
14	667.1	629.9	631.0	528.5	590.3	598.3
15	1169.0	1656.3	1368.4	1461.9	1559.7	1260.6
16	2164.0	2695.0	2569.0	2996.0	2284.0	2183.0
17	435.9	386.1	386.1	406.2	397.0	319.9
18	193.7	151.9	200.2	200.0	203.9	206.7
19	104.1	99.2	29.5	15.7	46.0	44.3
20	259.8	263.1	263.5	290.6	345.1	240.3
21	6466.0	6859.0	6824.7	7540.7	6812.7	6917.5

```
22       65.7       59.9       59.9       53.4       50.8       52.4
23    10437.0    10193.0    10138.0    10489.0    11000.0    11236.0
24      150.4      153.0      169.8      259.6      241.1      228.3
25       21.6       21.5       21.5       22.9       21.7       24.3
26     5062.2     5220.0     6610.6     5040.2     5182.7     5665.2
27          .          .          .          .          .          .
28      545.1      552.9      620.5      624.6      627.1      640.0
29     9555.6    11133.7    11124.0    11780.0    13097.0    10807.1
30      572.0      590.0      556.0      593.0      582.0      608.0
31    14884.8    14510.8    14745.9    14719.5    15037.3    14340.7
32    83131.7    91793.4    93355.3    96692.9    99182.5    89092.9

   Unnamed: 7 Unnamed: 8 Unnamed: 9
0     2010-11    2011-12    2012-13
1      7882.4     7746.2     6862.4
2       234.0      255.0      263.0
3      4736.6     4516.3     5128.5
4      3102.1     7162.6     7529.3
5      6159.0     6028.4     6608.8
6        19.6       19.8       19.7
7       115.0      121.8      122.8
8      1496.6     1790.0     1541.0
9      3472.0     3759.0     3976.0
10      128.9      131.6      125.3
11      507.7      544.7      818.1
12     1110.0     3130.6     3164.9
13     4188.0     3955.0     3364.0
14      522.7      569.0      508.3
15     1772.1     2227.3     2775.0
16     2696.0     2841.0     3057.0
17      521.7      591.0      257.6
18      207.0      216.5      232.0
19       47.2       54.3       30.5
20      381.4      382.4      405.2
21     6827.7     5807.0     7295.5
22       52.0       42.1       46.5
23    10837.0    10542.0    11374.0
24      265.5      253.4      222.5
25       21.0       20.9       21.3
26     5792.4     7458.7     4049.9
27     6535.6     5148.8     4647.6
28      702.5      718.3      713.2
29    11992.0    14022.0    14416.0
30      550.4      594.0      579.8
31    13045.9    14605.8    15023.7
32    95979.8   105310.9   105231.6  >
```

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 33 entries, 0 to 32
Data columns (total 10 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   TABLE 60: STATE-WISE PRODUCTION OF FOODGRAINS - RICE
(Thousand Tonnes)  33 non-null     object
 1   Unnamed: 1                                                      33
non-null     object
 2   Unnamed: 2                                                      33
non-null     object
 3   Unnamed: 3                                                      33
non-null     object
 4   Unnamed: 4                                                      33
non-null     object
 5   Unnamed: 5                                                      33
non-null     object
 6   Unnamed: 6                                                      33
non-null     object
 7   Unnamed: 7                                                      33
non-null     object
 8   Unnamed: 8                                                      33
non-null     object
 9   Unnamed: 9                                                      33
non-null     object
dtypes: object(10)
memory usage: 2.7+ KB
```

[9]:
```
print("Shape of the dataframe: ",df.shape)
df.isna().sum()
```

```
Shape of the dataframe:  (33, 10)
```

[9]:
```
TABLE 60: STATE-WISE PRODUCTION OF FOODGRAINS - RICE\n(Thousand Tonnes)    0
Unnamed: 1                                                                 0
Unnamed: 2                                                                 0
Unnamed: 3                                                                 0
Unnamed: 4                                                                 0
Unnamed: 5                                                                 0
Unnamed: 6                                                                 0
Unnamed: 7                                                                 0
Unnamed: 8                                                                 0
Unnamed: 9                                                                 0
```

```
dtype: int64
```

[10]:
```
#Heatmap to check null/missing values
sns.heatmap(df.isnull(),cmap="coolwarm")
plt.show()
```

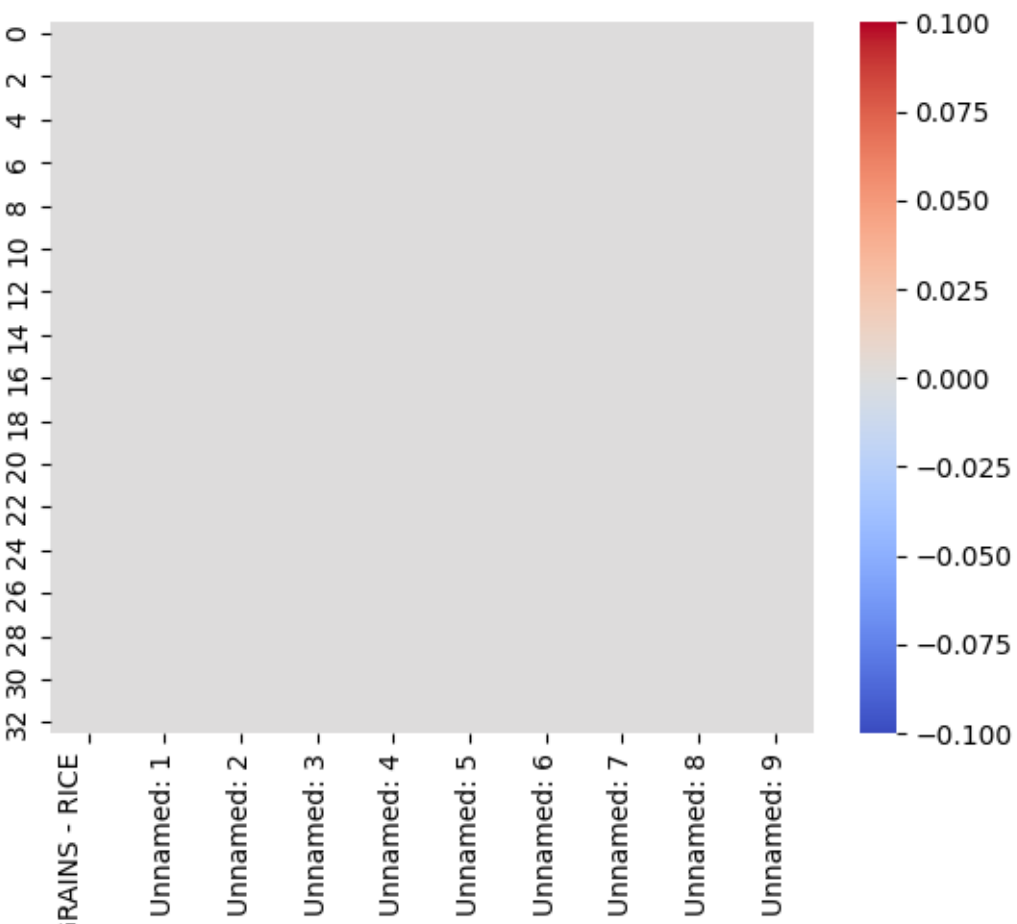


TABLE 60: STATE-WISE PRODUCTION OF FOODGRAINS - RICE
(Thousand Tonnes)
Unnamed: 1
Unnamed: 2
Unnamed: 3
Unnamed: 4
Unnamed: 5
Unnamed: 6
Unnamed: 7
Unnamed: 8
Unnamed: 9
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100

[63]:
```
# Drop 'ALL INDIA' row if present
df_no_all_india = df.drop(index='ALL INDIA', errors='ignore')

# Display the DataFrame
print("Table: Rice Production by State/Union Territory (2004-2013)")
print(df_no_all_india)
```

```
Table: Rice Production by State/Union Territory (2004-2013)
                      2004-05  2005-06  2006-07  2007-08  2008-09  2009-10  \
State/Union Territory
Andhra Pradesh         9601.0  11704.0  11872.0  13324.0  14241.0  10538.0
Arunachal Pradesh       135.0    146.2    146.2    158.1    163.9    215.8
Assam                  3470.7   3552.5   2916.0   3319.0   4008.5   4335.9
Bihar                  2472.2   3495.5   4989.3   4418.1   5590.3   3599.3
Chhattisgarh           4383.3   5011.6   5041.4   5426.6   4391.8   4110.4
NCT of Delhi             14.3     24.0     31.1     31.4     31.4     19.3
Goa                     145.2    147.3    130.3    121.6    123.3    100.6
Gujarat                1238.2   1298.0   1390.0   1474.0   1303.0   1292.0
Haryana                3023.0   3210.0   3371.0   3613.0   3298.0   3625.0
Himachal Pradesh        122.0    112.1    123.5    121.5    118.3    105.9
Jammu & Kashmir         492.2    556.8    554.0    561.3    563.1    497.4
Jharkhand              1677.0   1558.0   2967.8   3336.4   3420.2   1538.4
Karnataka              3547.0   5744.0   3446.0   3717.0   3802.0   3691.0
Kerala                  667.1    629.9    631.0    528.5    590.3    598.3
Madhya Pradesh         1169.0   1656.3   1368.4   1461.9   1559.7   1260.6
Maharashtra            2164.0   2695.0   2569.0   2996.0   2284.0   2183.0
Manipur                 435.9    386.1    386.1    406.2    397.0    319.9
Meghalaya               193.7    151.9    200.2    200.0    203.9    206.7
Mizoram                 104.1     99.2     29.5     15.7     46.0     44.3
Nagaland                259.8    263.1    263.5    290.6    345.1    240.3
Odisha                 6466.0   6859.0   6824.7   7540.7   6812.7   6917.5
Puducherry               65.7     59.9     59.9     53.4     50.8     52.4
Punjab                10437.0  10193.0  10138.0  10489.0  11000.0  11236.0
Rajasthan               150.4    153.0    169.8    259.6    241.1    228.3
Sikkim                   21.6     21.5     21.5     22.9     21.7     24.3
Tamil Nadu             5062.2   5220.0   6610.6   5040.2   5182.7   5665.2
Telangana                 NaN      NaN      NaN      NaN      NaN      NaN
Tripura                 545.1    552.9    620.5    624.6    627.1    640.0
Uttar Pradesh          9555.6  11133.7  11124.0  11780.0  13097.0  10807.1
Uttarakhand             572.0    590.0    556.0    593.0    582.0    608.0
West Bengal           14884.8  14510.8  14745.9  14719.5  15037.3  14340.7

                      2010-11  2011-12  2012-13  Cluster
State/Union Territory
Andhra Pradesh         7882.4   7746.2   6862.4        1
Arunachal Pradesh       234.0    255.0    263.0        0
Assam                  4736.6   4516.3   5128.5        2
```

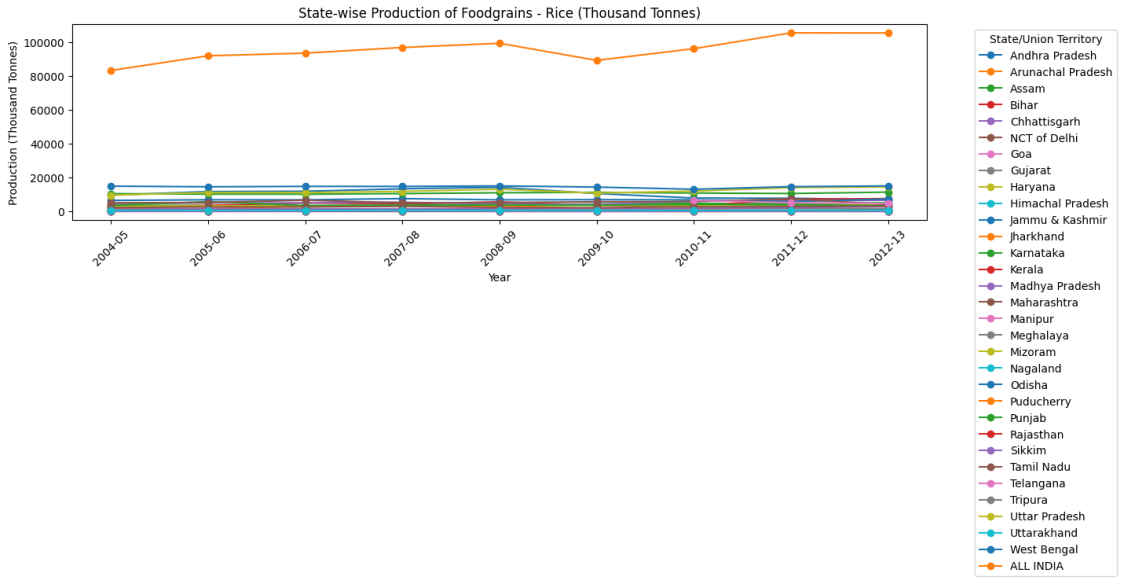
Bihar	3102.1	7162.6	7529.3	2
Chhattisgarh	6159.0	6028.4	6608.8	2
NCT of Delhi	19.6	19.8	19.7	0
Goa	115.0	121.8	122.8	0
Gujarat	1496.6	1790.0	1541.0	0
Haryana	3472.0	3759.0	3976.0	2
Himachal Pradesh	128.9	131.6	125.3	0
Jammu & Kashmir	507.7	544.7	818.1	0
Jharkhand	1110.0	3130.6	3164.9	0
Karnataka	4188.0	3955.0	3364.0	2
Kerala	522.7	569.0	508.3	0
Madhya Pradesh	1772.1	2227.3	2775.0	0
Maharashtra	2696.0	2841.0	3057.0	0
Manipur	521.7	591.0	257.6	0
Meghalaya	207.0	216.5	232.0	0
Mizoram	47.2	54.3	30.5	0
Nagaland	381.4	382.4	405.2	0
Odisha	6827.7	5807.0	7295.5	2
Puducherry	52.0	42.1	46.5	0
Punjab	10837.0	10542.0	11374.0	1
Rajasthan	265.5	253.4	222.5	0
Sikkim	21.0	20.9	21.3	0
Tamil Nadu	5792.4	7458.7	4049.9	2
Telangana	6535.6	5148.8	4647.6	2
Tripura	702.5	5148.8	4647.6	0
Uttar Pradesh	11992.0	14022.0	14416.0	1
Uttarakhand	550.4	594.0	579.8	0
West Bengal	13045.9	14605.8	15023.7	1

sns.displot(x=df[‘2004-05’], bins=20,kde=True,edgecolor=“black”,color=‘black’,facecolor=‘#ffb03b’) plt.title(“2004-05”,size=20) plt.show()

[19]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data into a DataFrame
data = {
    'State/Union Territory': ['Andhra Pradesh', 'Arunachal Pradesh', 'Assam',␣
 ↪'Bihar', 'Chhattisgarh', 'NCT of Delhi', 'Goa', 'Gujarat', 'Haryana',␣
 ↪'Himachal Pradesh', 'Jammu & Kashmir', 'Jharkhand', 'Karnataka', 'Kerala',␣
 ↪'Madhya Pradesh', 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram',␣
 ↪'Nagaland', 'Odisha', 'Puducherry', 'Punjab', 'Rajasthan', 'Sikkim', 'Tamil␣
 ↪Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West␣
 ↪Bengal', 'ALL INDIA'],
    '2004-05': [9601.0, 135.0, 3470.7, 2472.2, 4383.3, 14.3, 145.2, 1238.2,␣
 ↪3023.0, 122.0, 492.2, 1677.0, 3547.0, 667.1, 1169.0, 2164.0, 435.9, 193.7,␣
 ↪104.1, 259.8, 6466.0, 65.7, 10437.0, 150.4, 21.6, 5062.2, None, 545.1, 9555.
 ↪6, 572.0, 14884.8, 83131.7],
```

```
    '2005-06': [11704.0, 146.2, 3552.5, 3495.5, 5011.6, 24.0, 147.3, 1298.0,␣
 ↪3210.0, 112.1, 556.8, 1558.0, 5744.0, 629.9, 1656.3, 2695.0, 386.1, 151.9,␣
 ↪99.2, 263.1, 6859.0, 59.9, 10193.0, 153.0, 21.5, 5220.0, None, 552.9, 11133.
 ↪7, 590.0, 14510.8, 91793.4],
    '2006-07': [11872.0, 146.2, 2916.0, 4989.3, 5041.4, 31.1, 130.3, 1390.0,␣
 ↪3371.0, 123.5, 554.0, 2967.8, 3446.0, 631.0, 1368.4, 2569.0, 386.1, 200.2,␣
 ↪29.5, 263.5, 6824.7, 59.9, 10138.0, 169.8, 21.5, 6610.6, None, 620.5, 11124.
 ↪0, 556.0, 14745.9, 93355.3],
    '2007-08': [13324.0, 158.1, 3319.0, 4418.1, 5426.6, 31.4, 121.6, 1474.0,␣
 ↪3613.0, 121.5, 561.3, 3336.4, 3717.0, 528.5, 1461.9, 2996.0, 406.2, 200.0,␣
 ↪15.7, 290.6, 7540.7, 53.4, 10489.0, 259.6, 22.9, 5040.2, None, 624.6, 11780.
 ↪0, 593.0, 14719.5, 96692.9],
    '2008-09': [14241.0, 163.9, 4008.5, 5590.3, 4391.8, 31.4, 123.3, 1303.0,␣
 ↪3298.0, 118.3, 563.1, 3420.2, 3802.0, 590.3, 1559.7, 2284.0, 397.0, 203.9,␣
 ↪46.0, 345.1, 6812.7, 50.8, 11000.0, 241.1, 21.7, 5182.7, None, 627.1, 13097.
 ↪0, 582.0, 15037.3, 99182.5],
    '2009-10': [10538.0, 215.8, 4335.9, 3599.3, 4110.4, 19.3, 100.6, 1292.0,␣
 ↪3625.0, 105.9, 497.4, 1538.4, 3691.0, 598.3, 1260.6, 2183.0, 319.9, 206.7,␣
 ↪44.3, 240.3, 6917.5, 52.4, 11236.0, 228.3, 24.3, 5665.2, None, 640.0, 10807.
 ↪1, 608.0, 14340.7, 89092.9],
    '2010-11': [7882.4, 234.0, 4736.6, 3102.1, 6159.0, 19.6, 115.0, 1496.6,␣
 ↪3472.0, 128.9, 507.7, 1110.0, 4188.0, 522.7, 1772.1, 2696.0, 521.7, 207.0,␣
 ↪47.2, 381.4, 6827.7, 52.0, 10837.0, 265.5, 21.0, 5792.4, 6535.6, 702.5,␣
 ↪11992.0, 550.4, 13045.9, 95979.8],
    '2011-12': [7746.2, 255.0, 4516.3, 7162.6, 6028.4, 19.8, 121.8, 1790.0,␣
 ↪3759.0, 131.6, 544.7, 3130.6, 3955.0, 569.0, 2227.3, 2841.0, 591.0, 216.5,␣
 ↪54.3, 382.4, 5807.0, 42.1, 10542.0, 253.4, 20.9, 7458.7, 5148.8, 718.3,␣
 ↪14022.0, 594.0, 14605.8, 105310.9],
    '2012-13': [6862.4, 263.0, 5128.5, 7529.3, 6608.8, 19.7, 122.8, 1541.0,␣
 ↪3976.0, 125.3, 818.1, 3164.9, 3364.0, 508.3, 2775.0, 3057.0, 257.6, 232.0,␣
 ↪30.5, 405.2, 7295.5, 46.5, 11374.0, 222.5, 21.3, 4049.9, 4647.6, 713.2,␣
 ↪14416.0, 579.8, 15023.7, 105231.6]
}

df = pd.DataFrame(data)

# Set 'State/Union Territory' column as index
df.set_index('State/Union Territory', inplace=True)

# Plotting
plt.figure(figsize=(12, 6))

# Change the colors and marker style as per your preference
df.T.plot(marker='o', ax=plt.gca())
plt.title('State-wise Production of Foodgrains - Rice (Thousand Tonnes)')
plt.xlabel('Year')
```

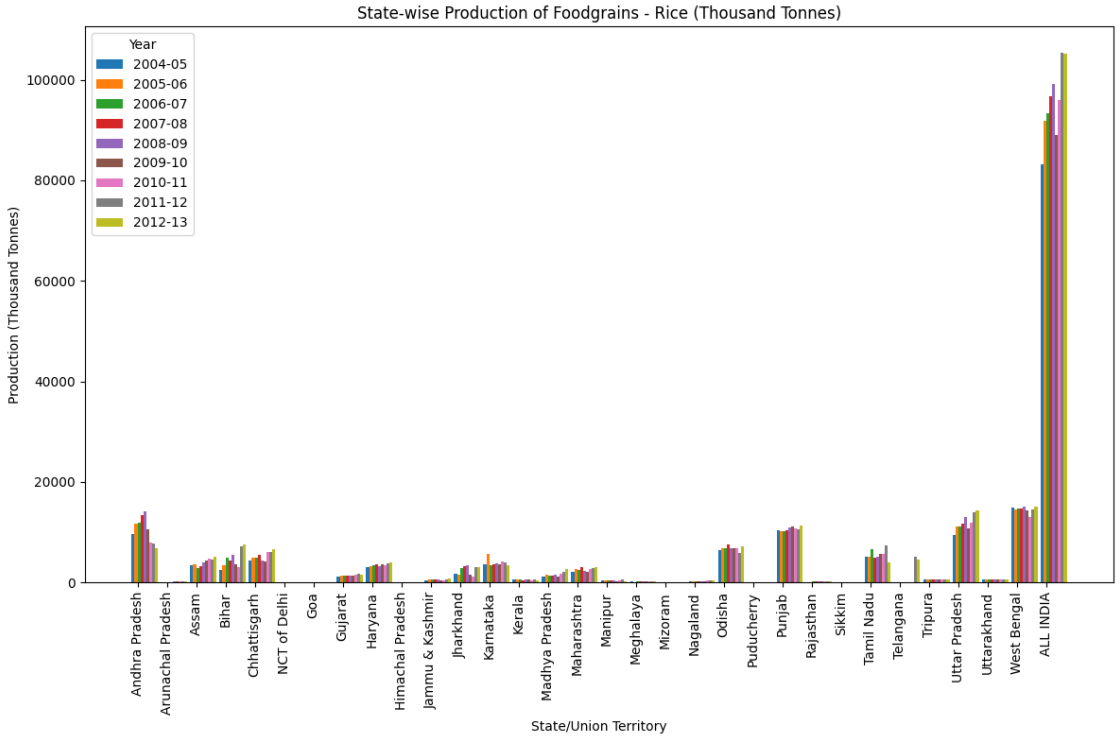
```
plt.ylabel('Production (Thousand Tonnes)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.legend(title='State/Union Territory', bbox_to_anchor=(1.05, 1), loc='upper␣
 ↪left')
plt.show()
```



[21]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data into a DataFrame
data = {
    'State/Union Territory': ['Andhra Pradesh', 'Arunachal Pradesh', 'Assam',␣
 ↪'Bihar', 'Chhattisgarh', 'NCT of Delhi', 'Goa', 'Gujarat', 'Haryana',␣
 ↪'Himachal Pradesh', 'Jammu & Kashmir', 'Jharkhand', 'Karnataka', 'Kerala',␣
 ↪'Madhya Pradesh', 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram',␣
 ↪'Nagaland', 'Odisha', 'Puducherry', 'Punjab', 'Rajasthan', 'Sikkim', 'Tamil␣
 ↪Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West␣
 ↪Bengal', 'ALL INDIA'],
    '2004-05': [9601.0, 135.0, 3470.7, 2472.2, 4383.3, 14.3, 145.2, 1238.2,␣
 ↪3023.0, 122.0, 492.2, 1677.0, 3547.0, 667.1, 1169.0, 2164.0, 435.9, 193.7,␣
 ↪104.1, 259.8, 6466.0, 65.7, 10437.0, 150.4, 21.6, 5062.2, None, 545.1, 9555.
 ↪6, 572.0, 14884.8, 83131.7],
    '2005-06': [11704.0, 146.2, 3552.5, 3495.5, 5011.6, 24.0, 147.3, 1298.0,␣
 ↪3210.0, 112.1, 556.8, 1558.0, 5744.0, 629.9, 1656.3, 2695.0, 386.1, 151.9,␣
 ↪99.2, 263.1, 6859.0, 59.9, 10193.0, 153.0, 21.5, 5220.0, None, 552.9, 11133.
 ↪7, 590.0, 14510.8, 91793.4],
```

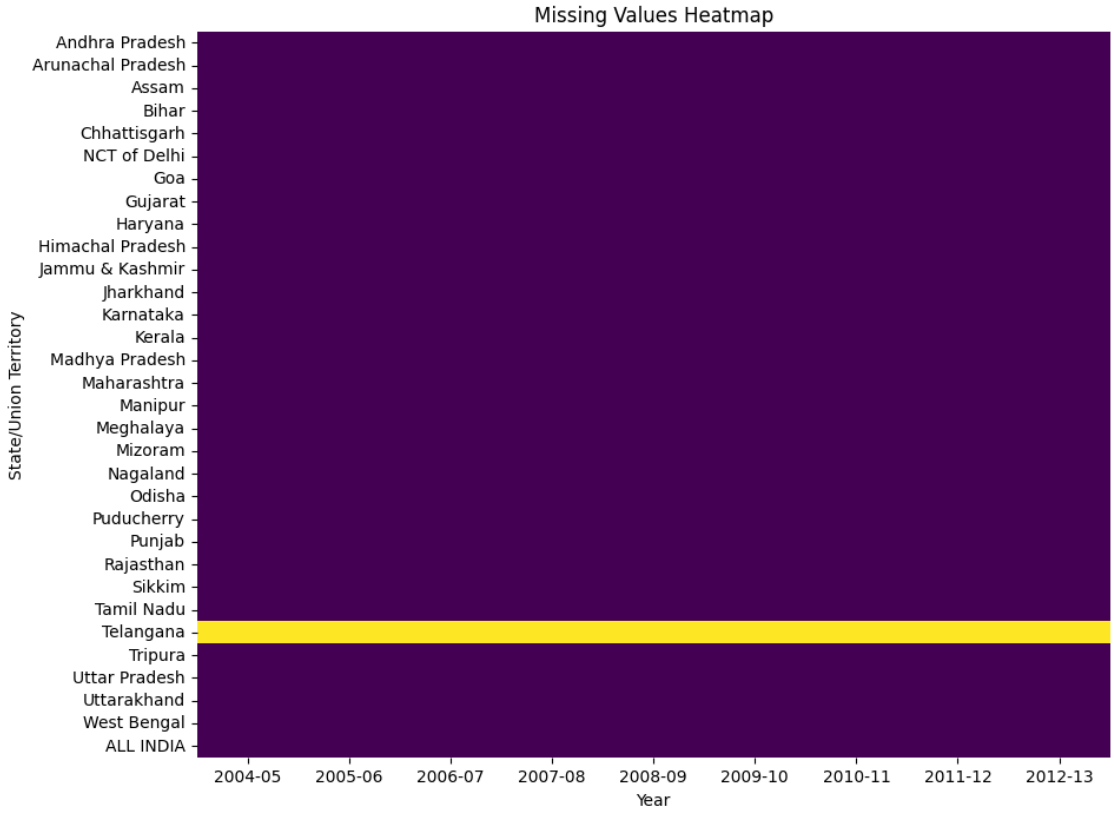
```
    '2006-07': [11872.0, 146.2, 2916.0, 4989.3, 5041.4, 31.1, 130.3, 1390.0,␣
 ↪3371.0, 123.5, 554.0, 2967.8, 3446.0, 631.0, 1368.4, 2569.0, 386.1, 200.2,␣
 ↪29.5, 263.5, 6824.7, 59.9, 10138.0, 169.8, 21.5, 6610.6, None, 620.5, 11124.
 ↪0, 556.0, 14745.9, 93355.3],
    '2007-08': [13324.0, 158.1, 3319.0, 4418.1, 5426.6, 31.4, 121.6, 1474.0,␣
 ↪3613.0, 121.5, 561.3, 3336.4, 3717.0, 528.5, 1461.9, 2996.0, 406.2, 200.0,␣
 ↪15.7, 290.6, 7540.7, 53.4, 10489.0, 259.6, 22.9, 5040.2, None, 624.6, 11780.
 ↪0, 593.0, 14719.5, 96692.9],
    '2008-09': [14241.0, 163.9, 4008.5, 5590.3, 4391.8, 31.4, 123.3, 1303.0,␣
 ↪3298.0, 118.3, 563.1, 3420.2, 3802.0, 590.3, 1559.7, 2284.0, 397.0, 203.9,␣
 ↪46.0, 345.1, 6812.7, 50.8, 11000.0, 241.1, 21.7, 5182.7, None, 627.1, 13097.
 ↪0, 582.0, 15037.3, 99182.5],
    '2009-10': [10538.0, 215.8, 4335.9, 3599.3, 4110.4, 19.3, 100.6, 1292.0,␣
 ↪3625.0, 105.9, 497.4, 1538.4, 3691.0, 598.3, 1260.6, 2183.0, 319.9, 206.7,␣
 ↪44.3, 240.3, 6917.5, 52.4, 11236.0, 228.3, 24.3, 5665.2, None, 640.0, 10807.
 ↪1, 608.0, 14340.7, 89092.9],
    '2010-11': [7882.4, 234.0, 4736.6, 3102.1, 6159.0, 19.6, 115.0, 1496.6,␣
 ↪3472.0, 128.9, 507.7, 1110.0, 4188.0, 522.7, 1772.1, 2696.0, 521.7, 207.0,␣
 ↪47.2, 381.4, 6827.7, 52.0, 10837.0, 265.5, 21.0, 5792.4, None, 702.5, 11992.
 ↪0, 550.4, 13045.9, 95979.8],
    '2011-12': [7746.2, 255.0, 4516.3, 7162.6, 6028.4, 19.8, 121.8, 1790.0,␣
 ↪3759.0, 131.6, 544.7, 3130.6, 3955.0, 569.0, 2227.3, 2841.0, 591.0, 216.5,␣
 ↪54.3, 382.4, 5807.0, 42.1, 10542.0, 253.4, 20.9, 7458.7, 5148.8, 718.3,␣
 ↪14022.0, 594.0, 14605.8, 105310.9],
    '2012-13': [6862.4, 263.0, 5128.5, 7529.3, 6608.8, 19.7, 122.8, 1541.0,␣
 ↪3976.0, 125.3, 818.1, 3164.9, 3364.0, 508.3, 2775.0, 3057.0, 257.6, 232.0,␣
 ↪30.5, 405.2, 7295.5, 46.5, 11374.0, 222.5, 21.3, 4049.9, 4647.6, 713.2,␣
 ↪14416.0, 579.8, 15023.7, 105231.6]
}

df = pd.DataFrame(data)

# Set 'State/Union Territory' column as index
df.set_index('State/Union Territory', inplace=True)

# Plotting
plt.figure(figsize=(12, 8))

# Plotting each state's production for each year side by side
for i, column in enumerate(df.columns):
    plt.bar([x + i * 0.1 for x in range(len(df.index))], df[column], width=0.1,␣
 ↪label=column)

plt.title('State-wise Production of Foodgrains - Rice (Thousand Tonnes)')
plt.xlabel('State/Union Territory')
plt.ylabel('Production (Thousand Tonnes)')
```

```
plt.xticks(ticks=[x + 0.15 for x in range(len(df.index))], labels=df.index,␣
 ↪rotation=90)
plt.legend(title='Year')
plt.tight_layout()
plt.show()
```



[25]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the data into a DataFrame
data = {
    'State/Union Territory': ['Andhra Pradesh', 'Arunachal Pradesh', 'Assam',␣
 ↪'Bihar', 'Chhattisgarh', 'NCT of Delhi', 'Goa', 'Gujarat', 'Haryana',␣
 ↪'Himachal Pradesh', 'Jammu & Kashmir', 'Jharkhand', 'Karnataka', 'Kerala',␣
 ↪'Madhya Pradesh', 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram',␣
 ↪'Nagaland', 'Odisha', 'Puducherry', 'Punjab', 'Rajasthan', 'Sikkim', 'Tamil␣
 ↪Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West␣
 ↪Bengal', 'ALL INDIA'],
    '2004-05': [9601.0, 135.0, 3470.7, 2472.2, 4383.3, 14.3, 145.2, 1238.2,␣
 ↪3023.0, 122.0, 492.2, 1677.0, 3547.0, 667.1, 1169.0, 2164.0, 435.9, 193.7,␣
 ↪104.1, 259.8, 6466.0, 65.7, 10437.0, 150.4, 21.6, 5062.2, None, 545.1, 9555.
 ↪6, 572.0, 14884.8, 83131.7],
```

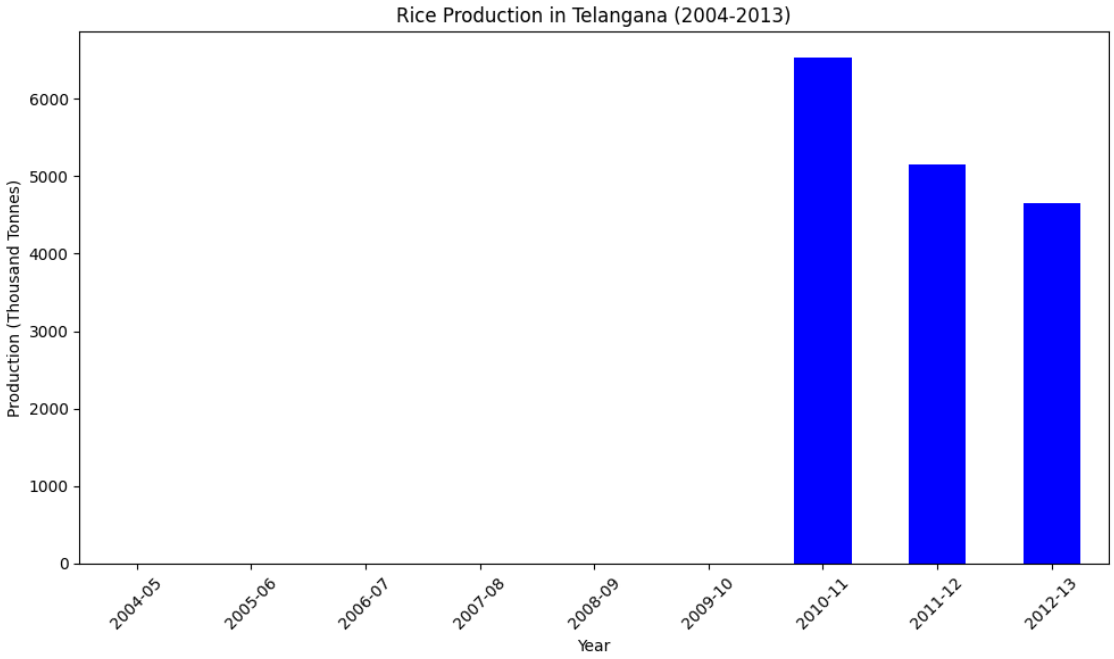
```
    '2005-06': [11704.0, 146.2, 3552.5, 3495.5, 5011.6, 24.0, 147.3, 1298.0,␣
 ↪3210.0, 112.1, 556.8, 1558.0, 5744.0, 629.9, 1656.3, 2695.0, 386.1, 151.9,␣
 ↪99.2, 263.1, 6859.0, 59.9, 10193.0, 153.0, 21.5, 5220.0, None, 552.9, 11133.
 ↪7, 590.0, 14510.8, 91793.4],
    '2006-07': [11872.0, 146.2, 2916.0, 4989.3, 5041.4, 31.1, 130.3, 1390.0,␣
 ↪3371.0, 123.5, 554.0, 2967.8, 3446.0, 631.0, 1368.4, 2569.0, 386.1, 200.2,␣
 ↪29.5, 263.5, 6824.7, 59.9, 10138.0, 169.8, 21.5, 6610.6, None, 620.5, 11124.
 ↪0, 556.0, 14745.9, 93355.3],
    '2007-08': [13324.0, 158.1, 3319.0, 4418.1, 5426.6, 31.4, 121.6, 1474.0,␣
 ↪3613.0, 121.5, 561.3, 3336.4, 3717.0, 528.5, 1461.9, 2996.0, 406.2, 200.0,␣
 ↪15.7, 290.6, 7540.7, 53.4, 10489.0, 259.6, 22.9, 5040.2, None, 624.6, 11780.
 ↪0, 593.0, 14719.5, 96692.9],
    '2008-09': [14241.0, 163.9, 4008.5, 5590.3, 4391.8, 31.4, 123.3, 1303.0,␣
 ↪3298.0, 118.3, 563.1, 3420.2, 3802.0, 590.3, 1559.7, 2284.0, 397.0, 203.9,␣
 ↪46.0, 345.1, 6812.7, 50.8, 11000.0, 241.1, 21.7, 5182.7, None, 627.1, 13097.
 ↪0, 582.0, 15037.3, 99182.5],
    '2009-10': [10538.0, 215.8, 4335.9, 3599.3, 4110.4, 19.3, 100.6, 1292.0,␣
 ↪3625.0, 105.9, 497.4, 1538.4, 3691.0, 598.3, 1260.6, 2183.0, 319.9, 206.7,␣
 ↪44.3, 240.3, 6917.5, 52.4, 11236.0, 228.3, 24.3, 5665.2, None, 640.0, 10807.
 ↪1, 608.0, 14340.7, 89092.9],
    '2010-11': [7882.4, 234.0, 4736.6, 3102.1, 6159.0, 19.6, 115.0, 1496.6,␣
 ↪3472.0, 128.9, 507.7, 1110.0, 4188.0, 522.7, 1772.1, 2696.0, 521.7, 207.0,␣
 ↪47.2, 381.4, 6827.7, 52.0, 10837.0, 265.5, 21.0, 5792.4, None, 702.5, 11992.
 ↪0, 550.4, 13045.9, 95979.8],
    '2011-12': [7746.2, 255.0, 4516.3, 7162.6, 6028.4, 19.8, 121.8, 1790.0,␣
 ↪3759.0, 131.6, 544.7, 3130.6, 3955.0, 569.0, 2227.3, 2841.0, 591.0, 216.5,␣
 ↪54.3, 382.4, 5807.0, 42.1, 10542.0, 253.4, 20.9, 7458.7, None, 5148.8, 14022.
 ↪0, 594.0, 14605.8, 105310.9],
    '2012-13': [6862.4, 263.0, 5128.5, 7529.3, 6608.8, 19.7, 122.8, 1541.0,␣
 ↪3976.0, 125.3, 818.1, 3164.9, 3364.0, 508.3, 2775.0, 3057.0, 257.6, 232.0,␣
 ↪30.5, 405.2, 7295.5, 46.5, 11374.0, 222.5, 21.3, 4049.9, None, 4647.6, 14416.
 ↪0, 579.8, 15023.7, 105231.6]
}

df = pd.DataFrame(data)

# Set 'State/Union Territory' column as index
df.set_index('State/Union Territory', inplace=True)

# Create a heatmap to visualize missing values
plt.figure(figsize=(10, 8))
sns.heatmap(df.isnull(), cmap='viridis', cbar=False)
plt.title('Missing Values Heatmap')
plt.xlabel('Year')
plt.ylabel('State/Union Territory')
plt.show()
```



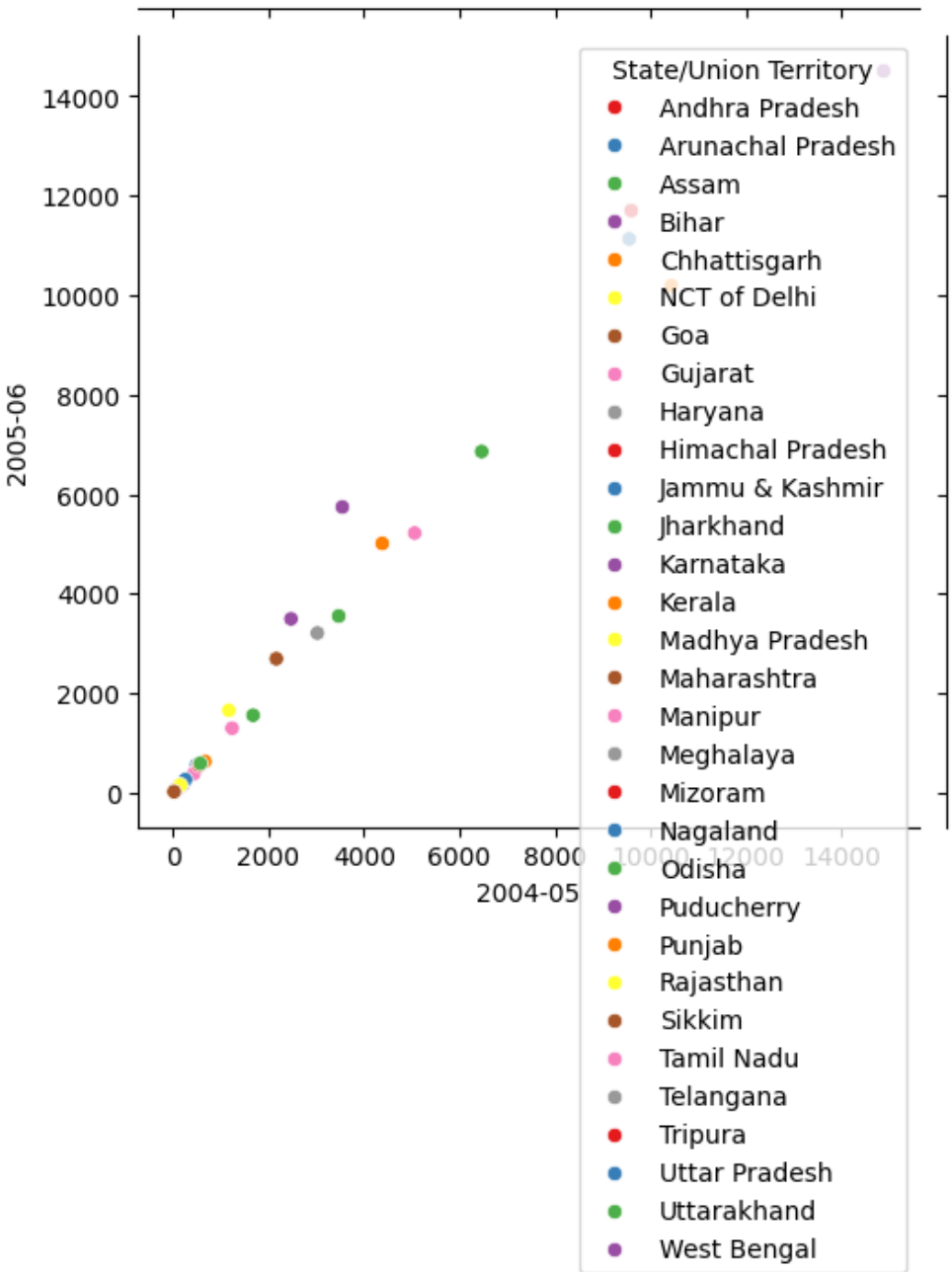
[29]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data into a DataFrame
data = {
    'State/Union Territory': ['Andhra Pradesh', 'Arunachal Pradesh', 'Assam',␣
 ↪'Bihar', 'Chhattisgarh', 'NCT of Delhi', 'Goa', 'Gujarat', 'Haryana',␣
 ↪'Himachal Pradesh', 'Jammu & Kashmir', 'Jharkhand', 'Karnataka', 'Kerala',␣
 ↪'Madhya Pradesh', 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram',␣
 ↪'Nagaland', 'Odisha', 'Puducherry', 'Punjab', 'Rajasthan', 'Sikkim', 'Tamil␣
 ↪Nadu', 'Telangana', 'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West␣
 ↪Bengal', 'ALL INDIA'],
    '2004-05': [9601.0, 135.0, 3470.7, 2472.2, 4383.3, 14.3, 145.2, 1238.2,␣
 ↪3023.0, 122.0, 492.2, 1677.0, 3547.0, 667.1, 1169.0, 2164.0, 435.9, 193.7,␣
 ↪104.1, 259.8, 6466.0, 65.7, 10437.0, 150.4, 21.6, 5062.2, None, 545.1, 9555.
 ↪6, 572.0, 14884.8, 83131.7],
    '2005-06': [11704.0, 146.2, 3552.5, 3495.5, 5011.6, 24.0, 147.3, 1298.0,␣
 ↪3210.0, 112.1, 556.8, 1558.0, 5744.0, 629.9, 1656.3, 2695.0, 386.1, 151.9,␣
 ↪99.2, 263.1, 6859.0, 59.9, 10193.0, 153.0, 21.5, 5220.0, None, 552.9, 11133.
 ↪7, 590.0, 14510.8, 91793.4],
```

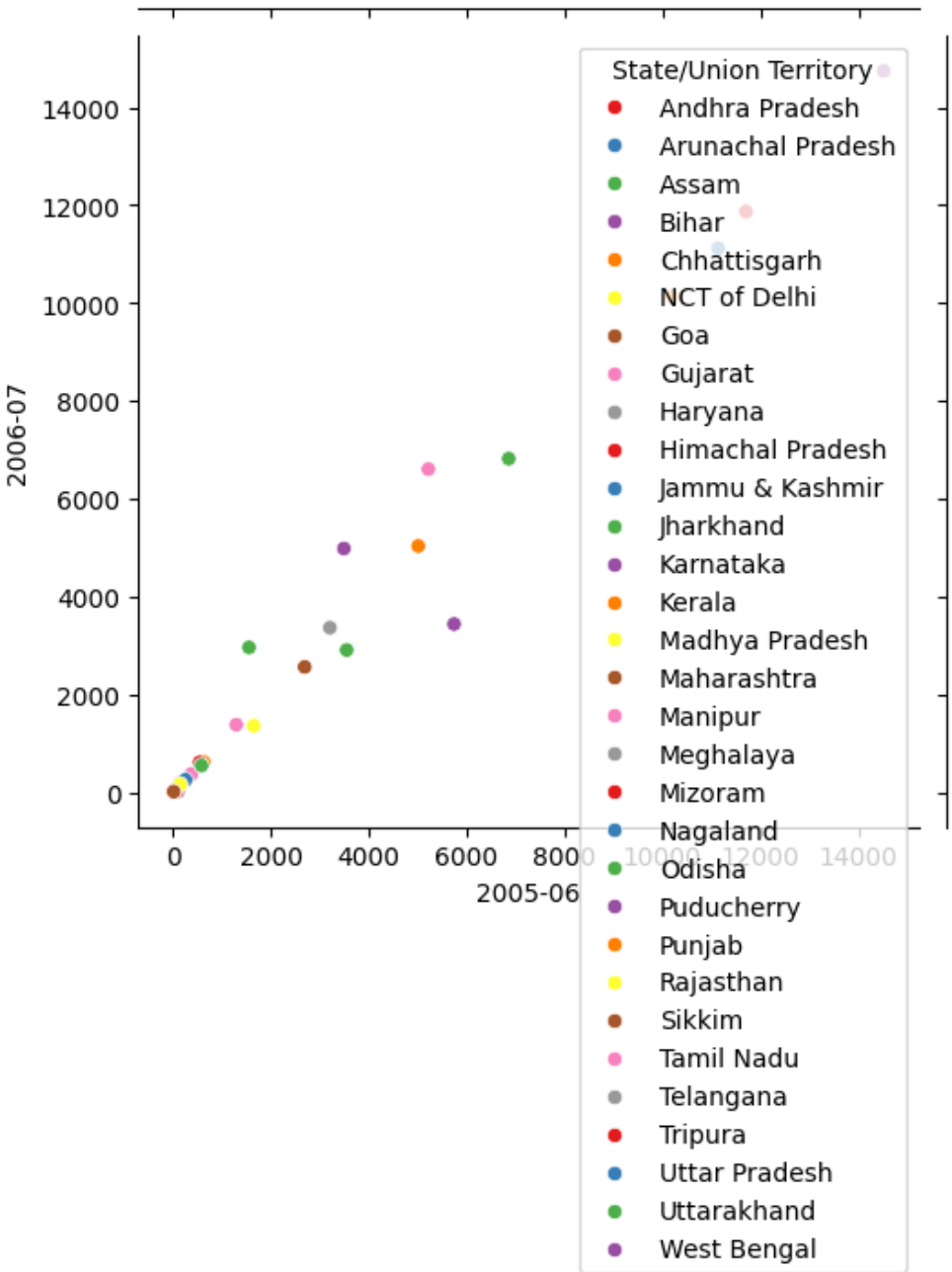
```
    '2006-07': [11872.0, 146.2, 2916.0, 4989.3, 5041.4, 31.1, 130.3, 1390.0,␣
 ↪3371.0, 123.5, 554.0, 2967.8, 3446.0, 631.0, 1368.4, 2569.0, 386.1, 200.2,␣
 ↪29.5, 263.5, 6824.7, 59.9, 10138.0, 169.8, 21.5, 6610.6, None, 620.5, 11124.
 ↪0, 556.0, 14745.9, 93355.3],
    '2007-08': [13324.0, 158.1, 3319.0, 4418.1, 5426.6, 31.4, 121.6, 1474.0,␣
 ↪3613.0, 121.5, 561.3, 3336.4, 3717.0, 528.5, 1461.9, 2996.0, 406.2, 200.0,␣
 ↪15.7, 290.6, 7540.7, 53.4, 10489.0, 259.6, 22.9, 5040.2, None, 624.6, 11780.
 ↪0, 593.0, 14719.5, 96692.9],
    '2008-09': [14241.0, 163.9, 4008.5, 5590.3, 4391.8, 31.4, 123.3, 1303.0,␣
 ↪3298.0, 118.3, 563.1, 3420.2, 3802.0, 590.3, 1559.7, 2284.0, 397.0, 203.9,␣
 ↪46.0, 345.1, 6812.7, 50.8, 11000.0, 241.1, 21.7, 5182.7, None, 627.1, 13097.
 ↪0, 582.0, 15037.3, 99182.5],
    '2009-10': [10538.0, 215.8, 4335.9, 3599.3, 4110.4, 19.3, 100.6, 1292.0,␣
 ↪3625.0, 105.9, 497.4, 1538.4, 3691.0, 598.3, 1260.6, 2183.0, 319.9, 206.7,␣
 ↪44.3, 240.3, 6917.5, 52.4, 11236.0, 228.3, 24.3, 5665.2, None, 640.0, 10807.
 ↪1, 608.0, 14340.7, 89092.9],
    '2010-11': [7882.4, 234.0, 4736.6, 3102.1, 6159.0, 19.6, 115.0, 1496.6,␣
 ↪3472.0, 128.9, 507.7, 1110.0, 4188.0, 522.7, 1772.1, 2696.0, 521.7, 207.0,␣
 ↪47.2, 381.4, 6827.7, 52.0, 10837.0, 265.5, 21.0, 5792.4, 6535.6, 702.5,␣
 ↪11992.0, 550.4, 13045.9, 95979.8],
    '2011-12': [7746.2, 255.0, 4516.3, 7162.6, 6028.4, 19.8, 121.8, 1790.0,␣
 ↪3759.0, 131.6, 544.7, 3130.6, 3955.0, 569.0, 2227.3, 2841.0, 591.0, 216.5,␣
 ↪54.3, 382.4, 5807.0, 42.1, 10542.0, 253.4, 20.9, 7458.7, 5148.8        ,␣
 ↪5148.8, 14022.0, 594.0, 14605.8, 105310.9],
    '2012-13': [6862.4, 263.0, 5128.5, 7529.3, 6608.8, 19.7, 122.8, 1541.0,␣
 ↪3976.0, 125.3, 818.1, 3164.9, 3364.0, 508.3, 2775.0, 3057.0, 257.6, 232.0,␣
 ↪30.5, 405.2, 7295.5, 46.5, 11374.0, 222.5, 21.3, 4049.9, 4647.6, 4647.6,␣
 ↪14416.0, 579.8, 15023.7, 105231.6]
}

df = pd.DataFrame(data)

# Set 'State/Union Territory' column as index
df.set_index('State/Union Territory', inplace=True)

# Select Telangana data
telangana_data = df.loc['Telangana']

# Check if there are non-null values for Telangana
if telangana_data.notnull().any():
    # Plotting
    plt.figure(figsize=(10, 6))
    telangana_data.plot(kind='bar', color='blue')
    plt.title('Rice Production in Telangana (2004-2013)')
    plt.xlabel('Year')
    plt.ylabel('Production (Thousand Tonnes)')
```

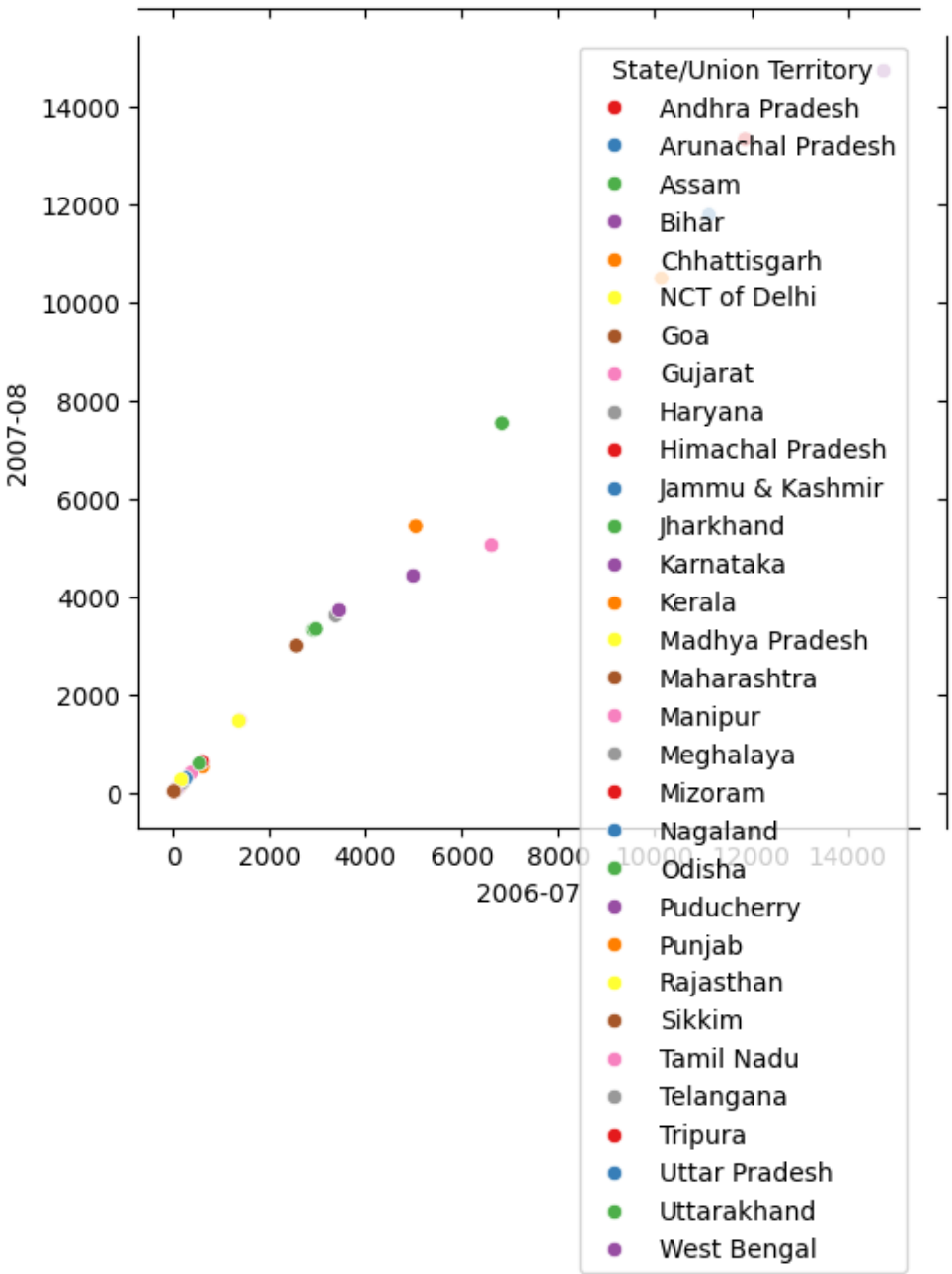
```
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
else:
    print("No data available for Telangana.")
```

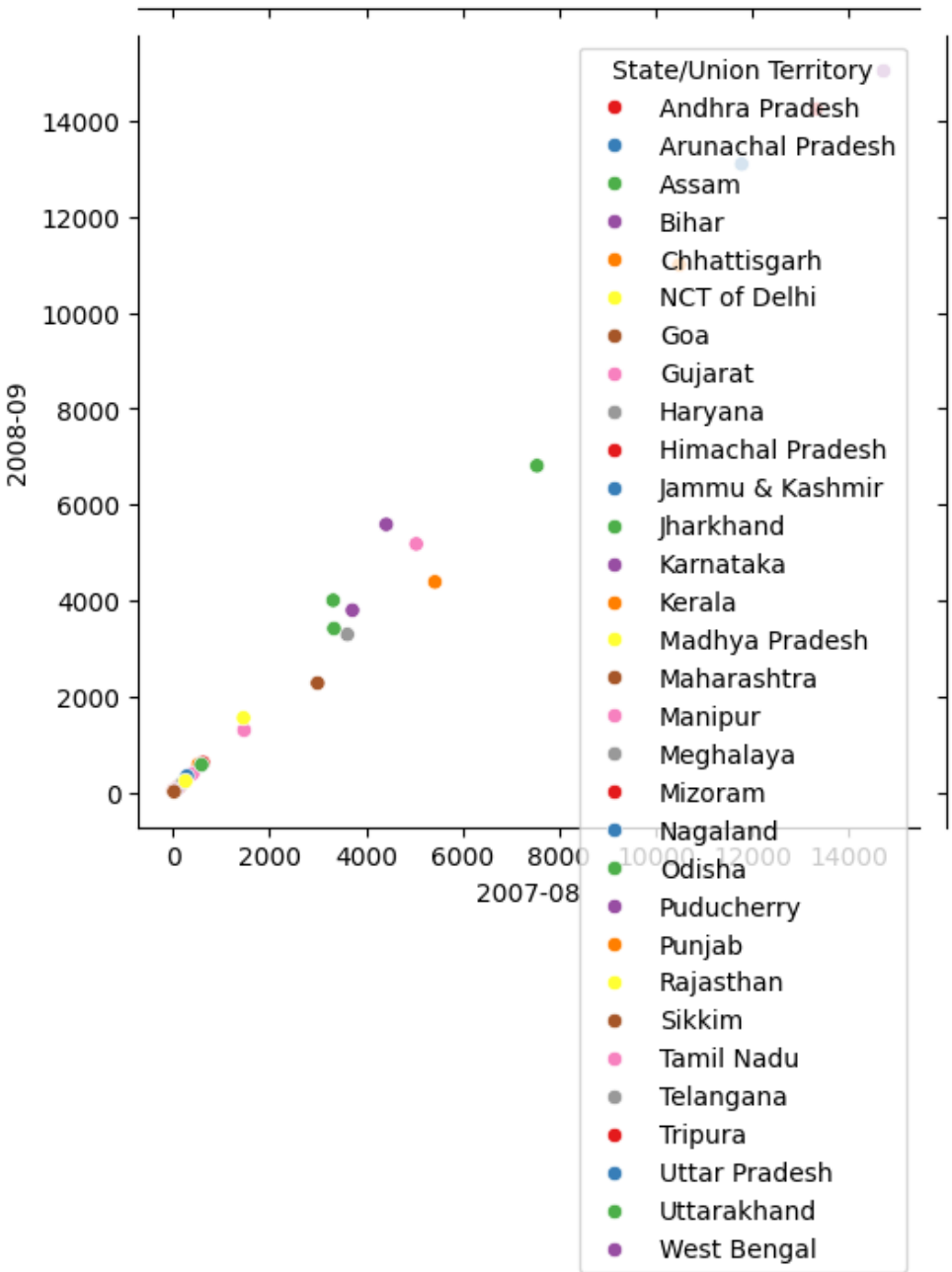


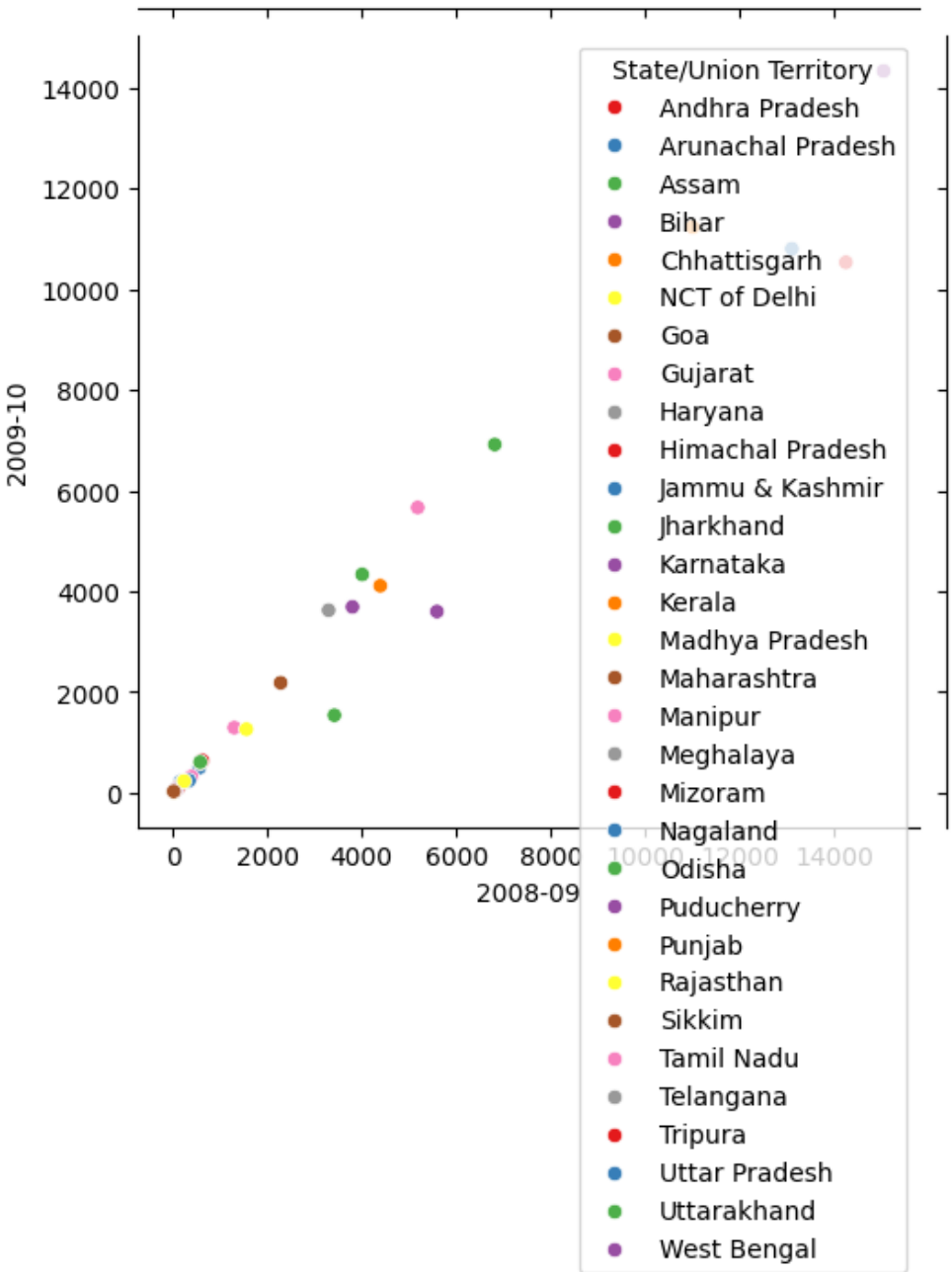
[38]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# List of columns representing the years
years = ['2004-05', '2005-06', '2006-07', '2007-08', '2008-09', '2009-10',␣
 ↪'2010-11', '2011-12', '2012-13']

# Iterate over each pair of years and create a jointplot
for i in range(len(years)-1):
    sns.jointplot(data=df, x=years[i], y=years[i+1], kind='scatter', hue='State/
 ↪Union Territory', palette='Set1').set_axis_labels(years[i], years[i+1])

# Display the plot
plt.show()
```

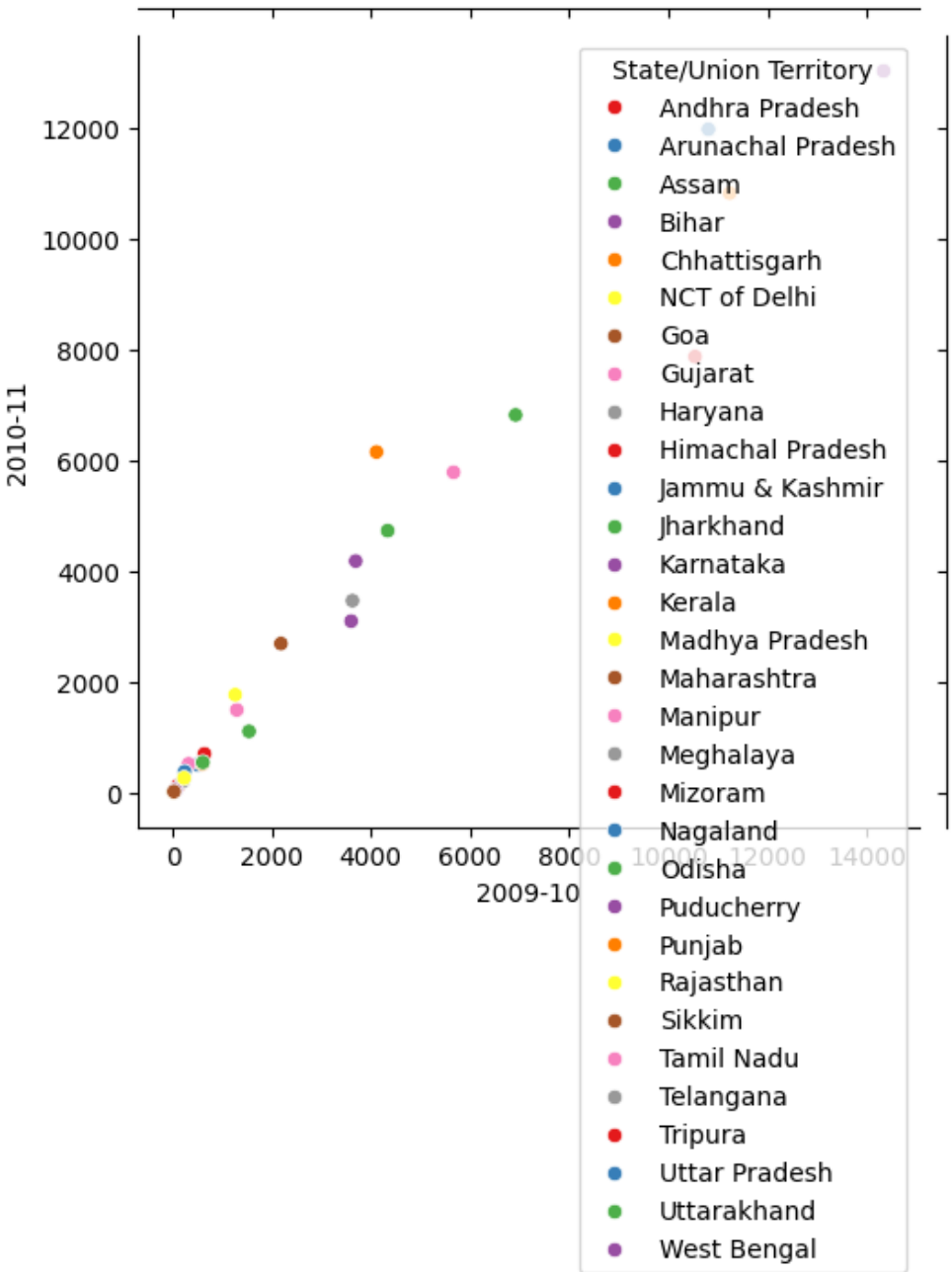

2005-06
2004-05
0
2000
4000
6000
8000
10000
12000
14000
State/Union Territory
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
NCT of Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu & Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal

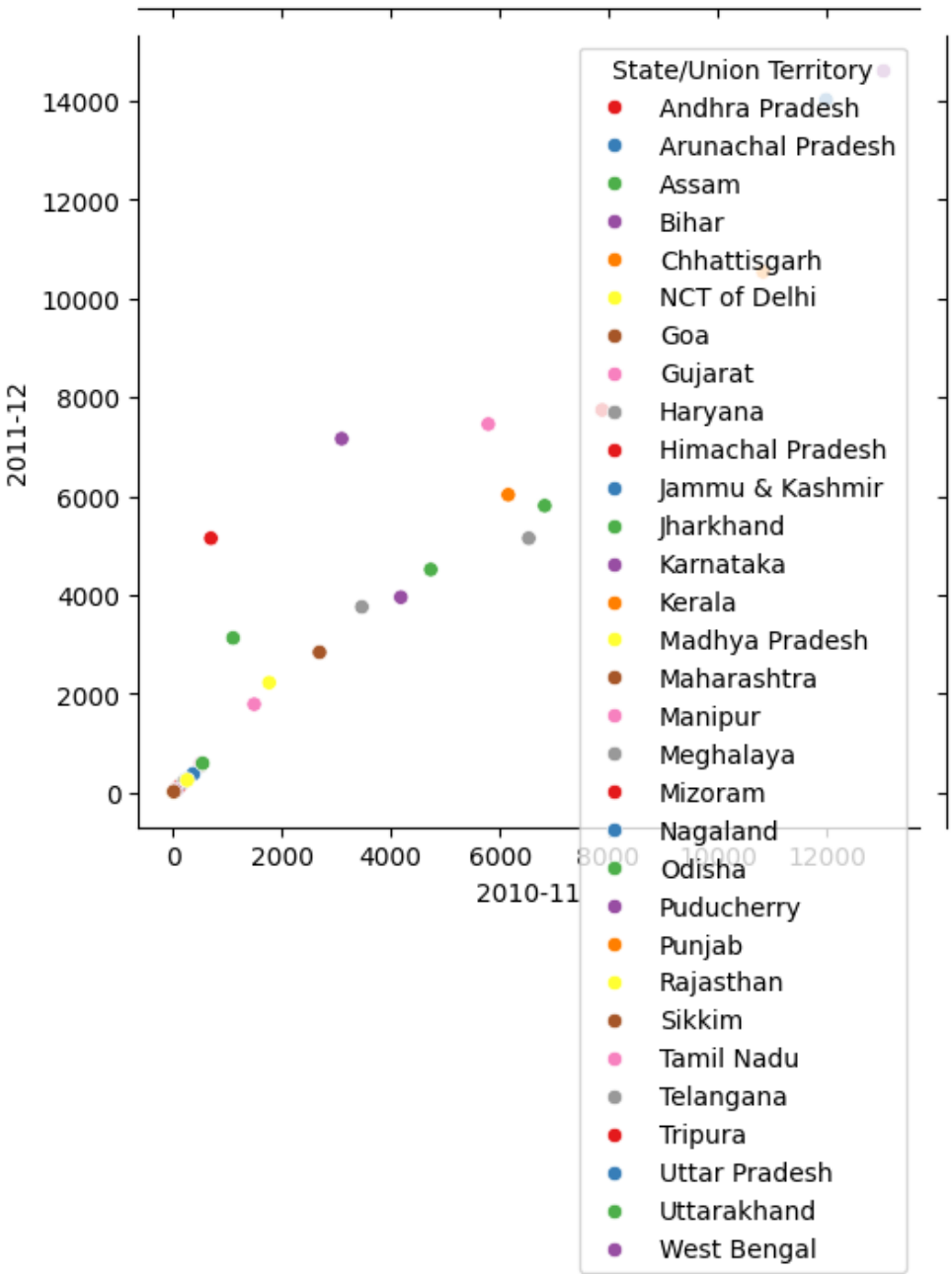

State/Union Territory
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
NCT of Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu & Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
2006-07
2005-06
0
2000
4000
6000
8000
10000
12000
14000

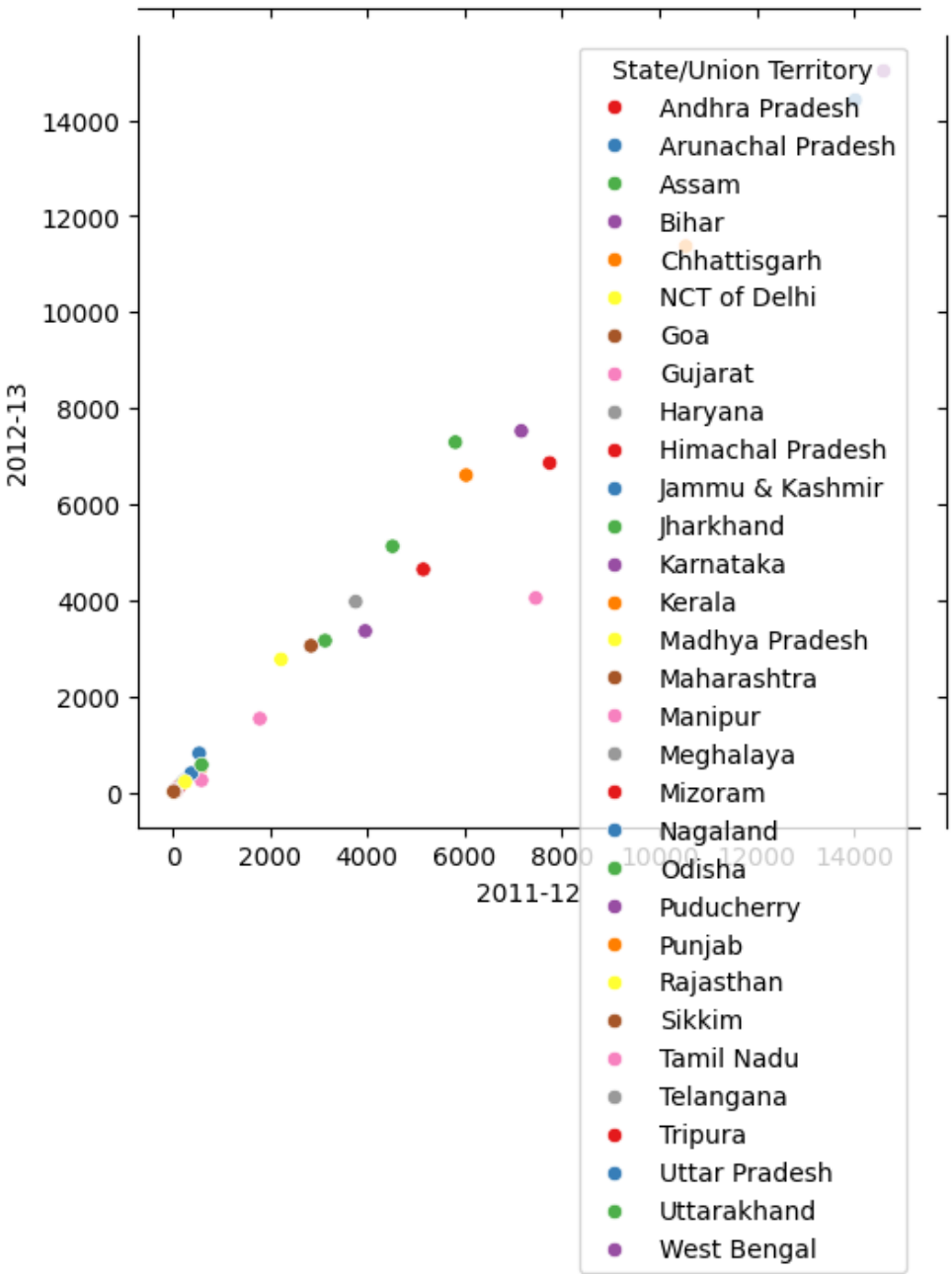

State/Union Territory
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
NCT of Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu & Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
2007-08
2006-07
0
2000
4000
6000
8000
10000
12000
14000


2008-09
2007-08
0
2000
4000
6000
8000
10000
12000
14000
State/Union Territory
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
NCT of Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu & Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal


2009-10
2008-09
0
2000
4000
6000
8000
10000
12000
14000
State/Union Territory
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
NCT of Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu & Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal

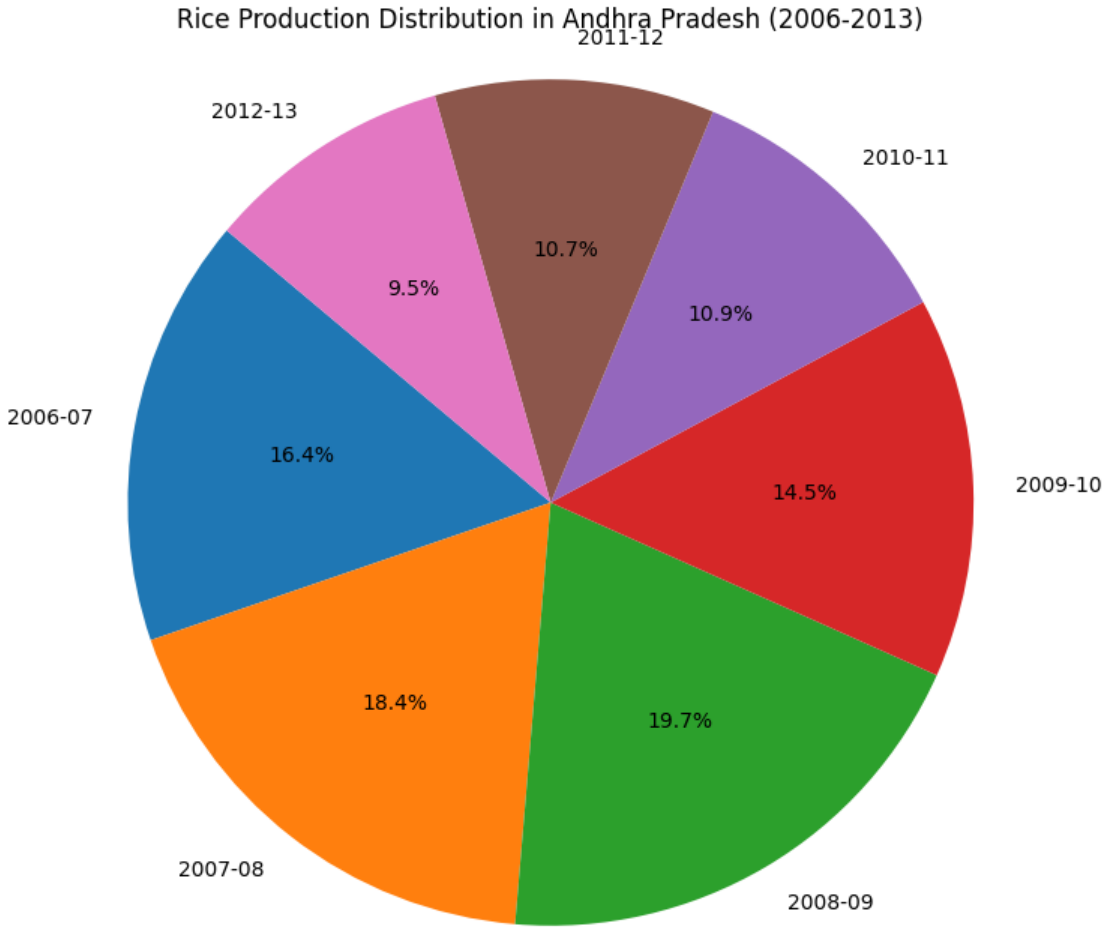

State/Union Territory
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
NCT of Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu & Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
2010-11
2009-10
0
2000
4000
6000
8000
10000
12000
14000

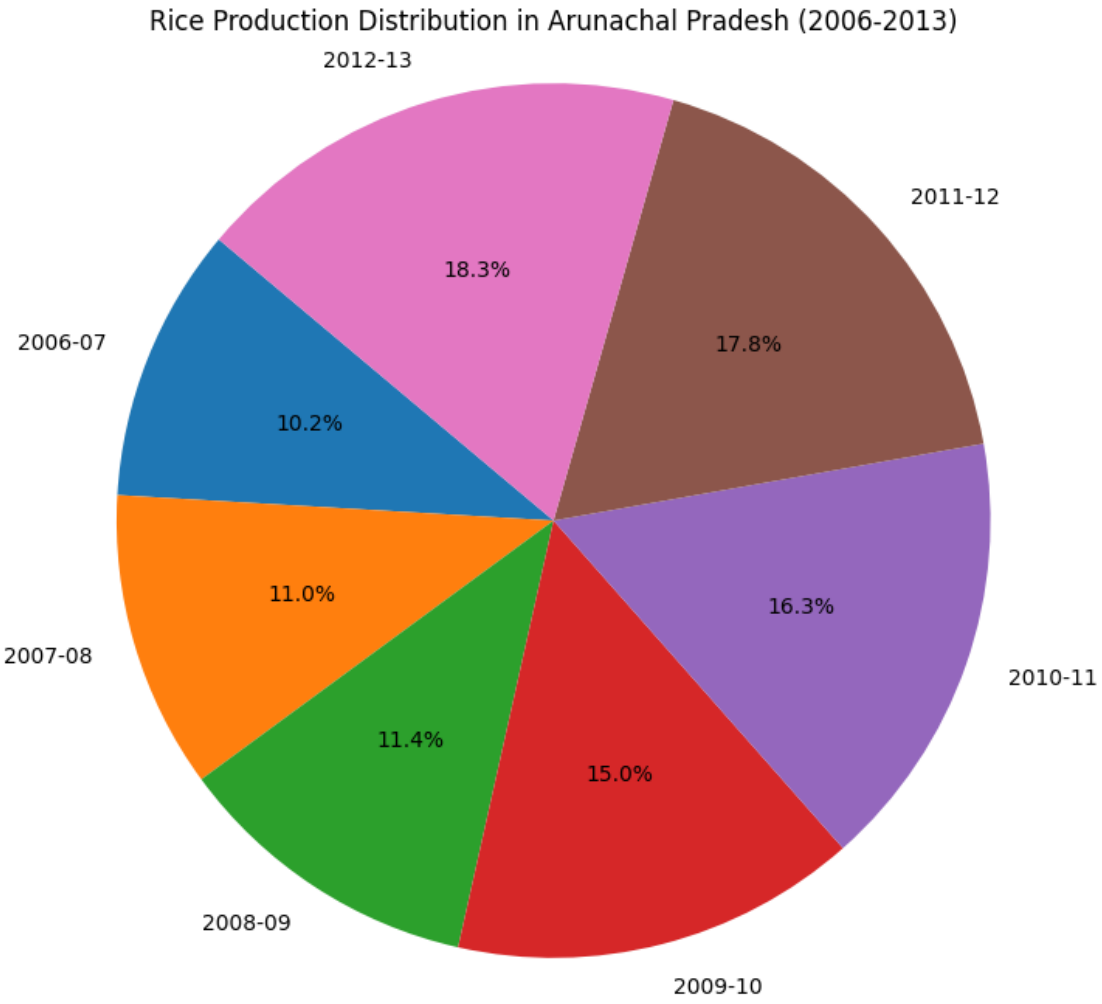

2011-12
2010-11
0
2000
4000
6000
8000
10000
12000
14000
State/Union Territory
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
NCT of Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu & Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal

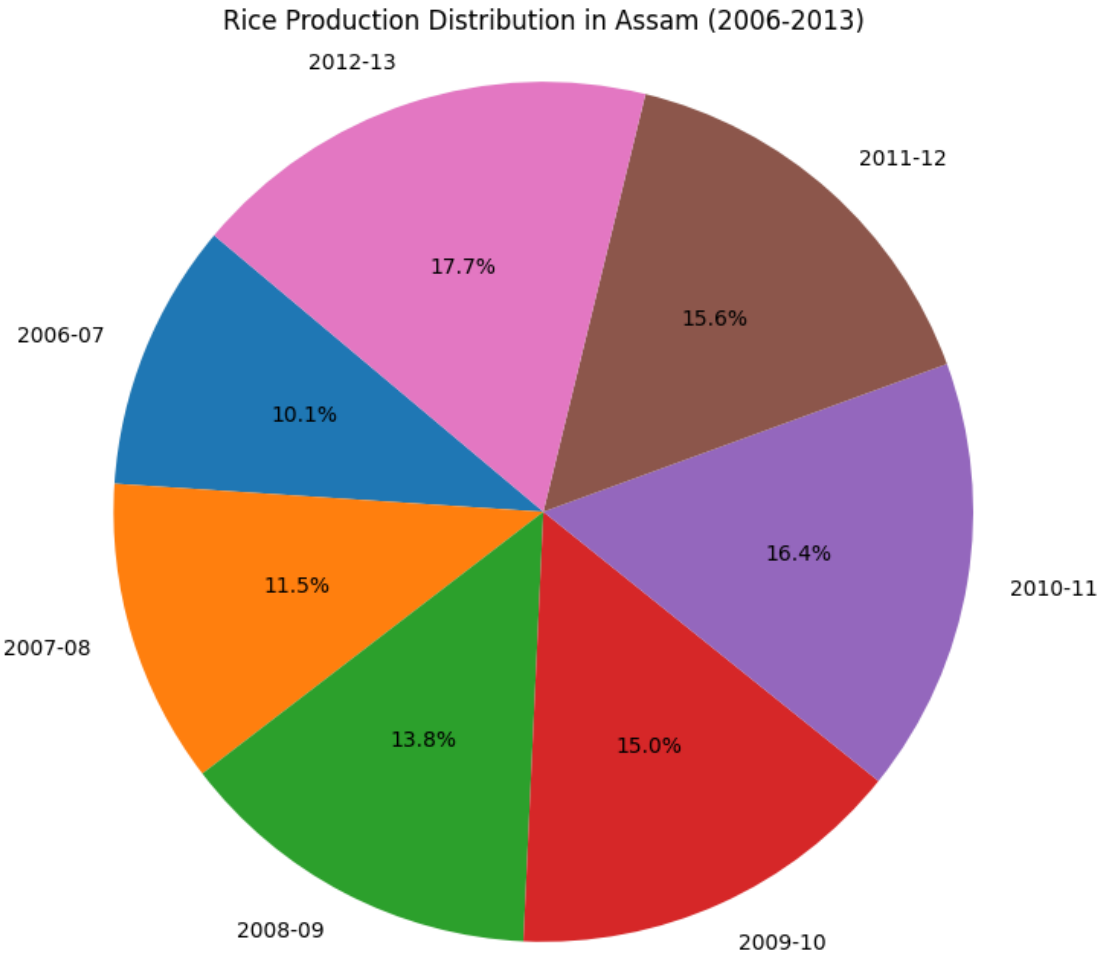


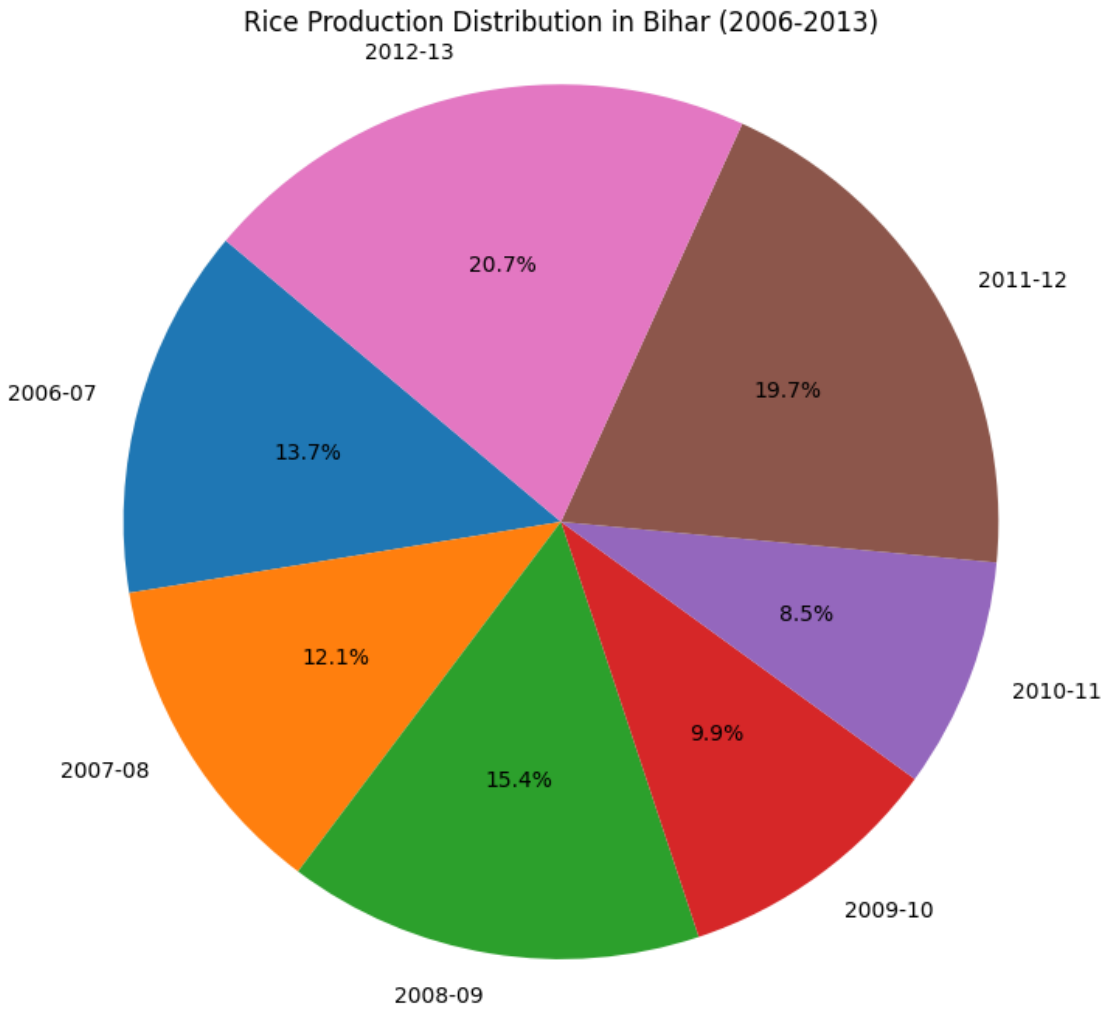
[45]:
```
import matplotlib.pyplot as plt

# Iterate over each state except Telangana
for state in df.index:
```

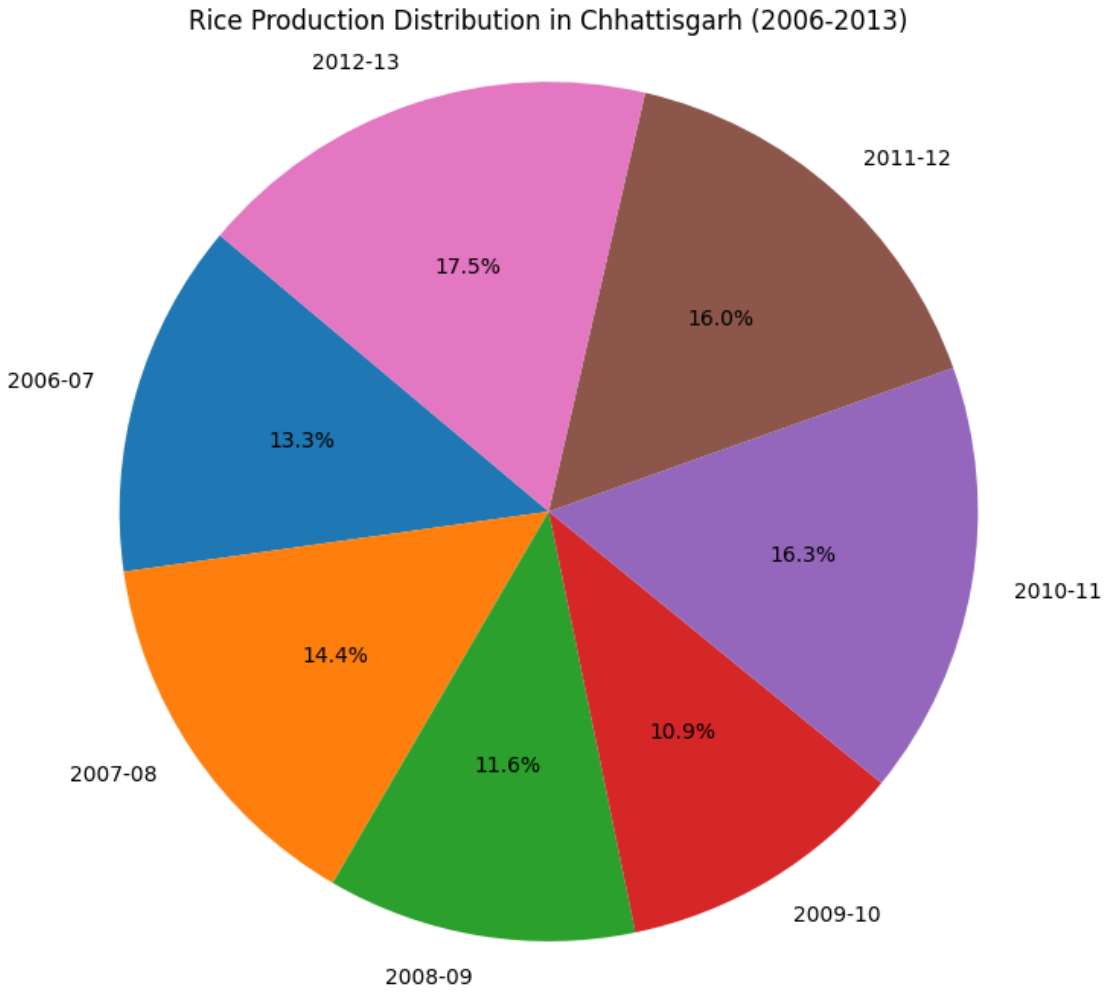
```
    if state != 'Telangana':
        # Get the production data for the current state
        sizes = df.loc[state, ['2006-07', '2007-08', '2008-09', '2009-10',␣
↪'2010-11', '2011-12', '2012-13']]

        # Create a pie chart for the current state
        plt.figure(figsize=(8, 8))
        plt.pie(sizes, labels=sizes.index, autopct='%1.1f%%', startangle=140)
        plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a␣
↪circle
        plt.title(f'Rice Production Distribution in {state} (2006-2013)')
        plt.show()
```

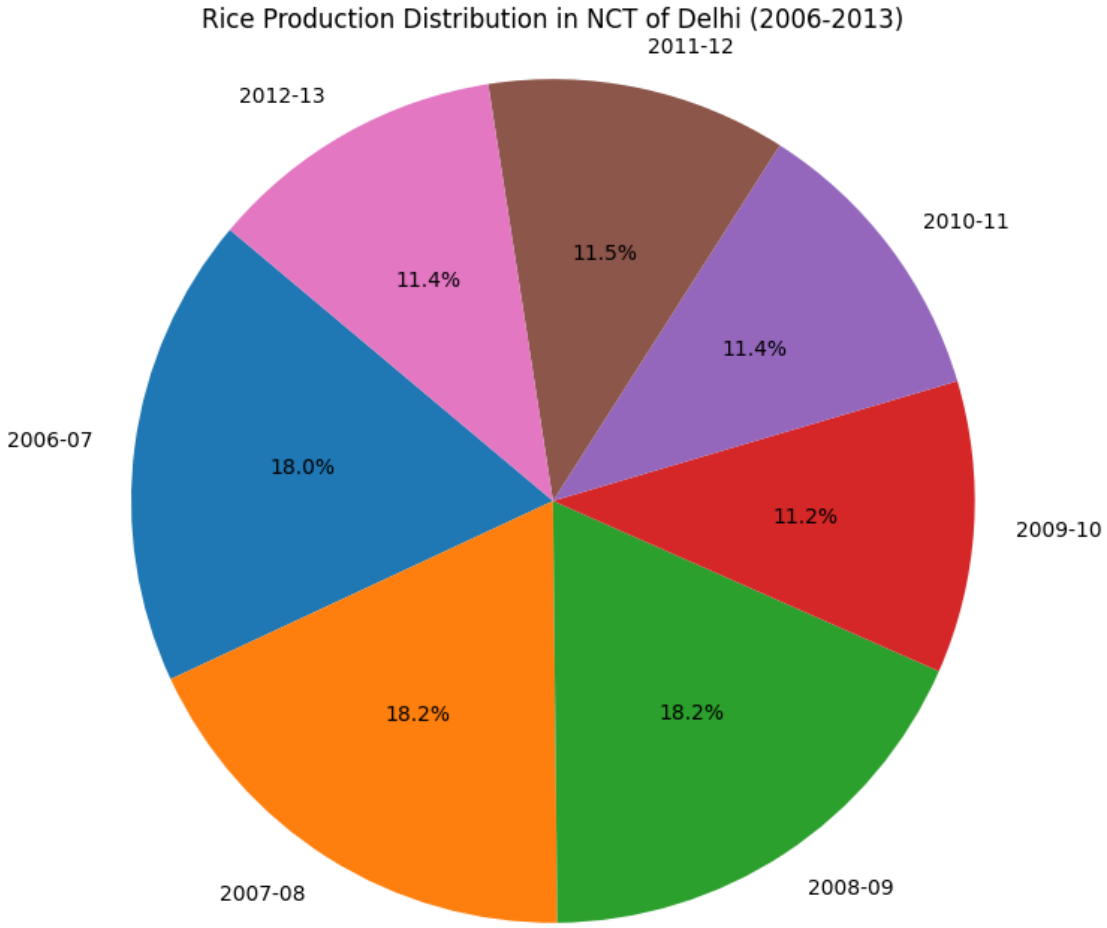


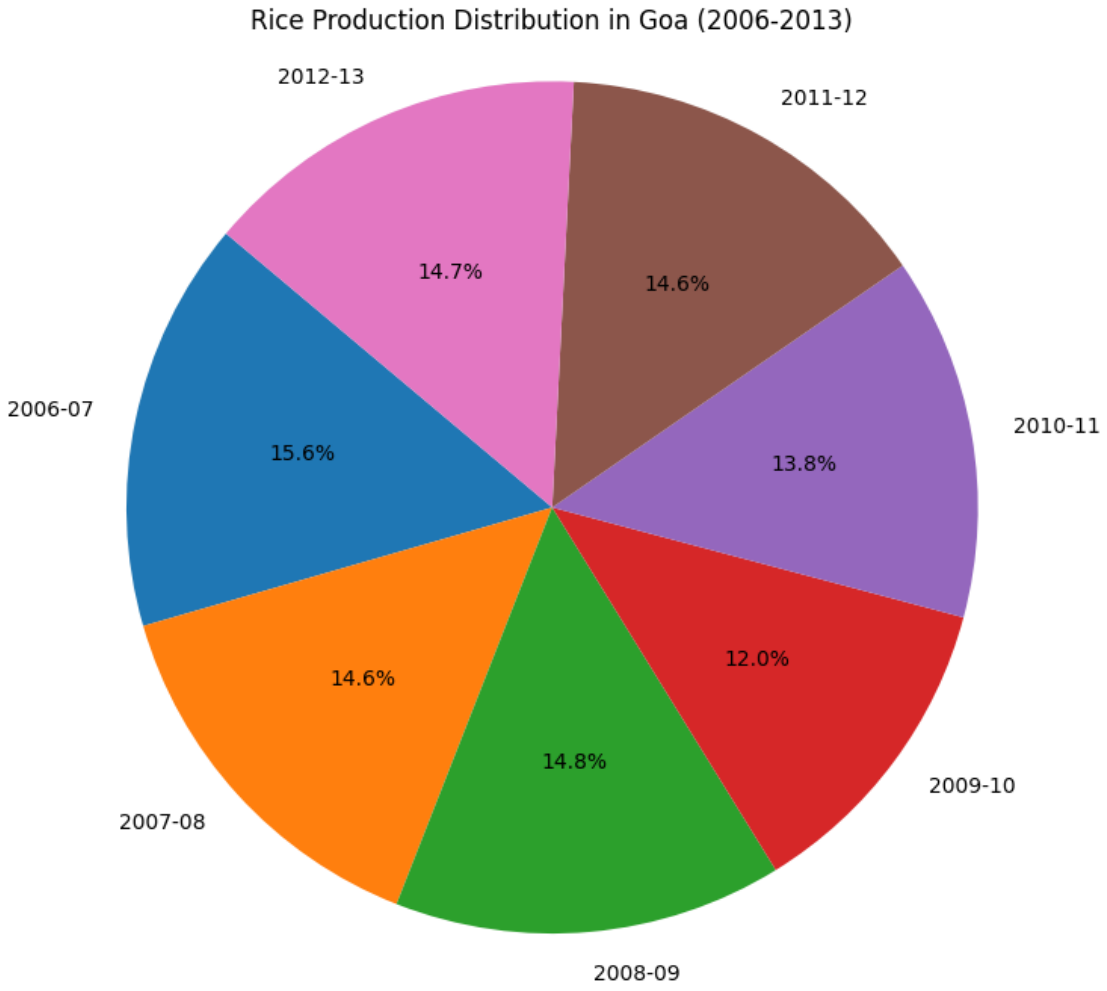

Rice Production Distribution in Arunachal Pradesh (2006-2013)
2012-13
18.3%
2011-12
17.8%
2006-07
10.2%
16.3%
2010-11
11.0%
2007-08
11.4%
15.0%
2008-09
2009-10

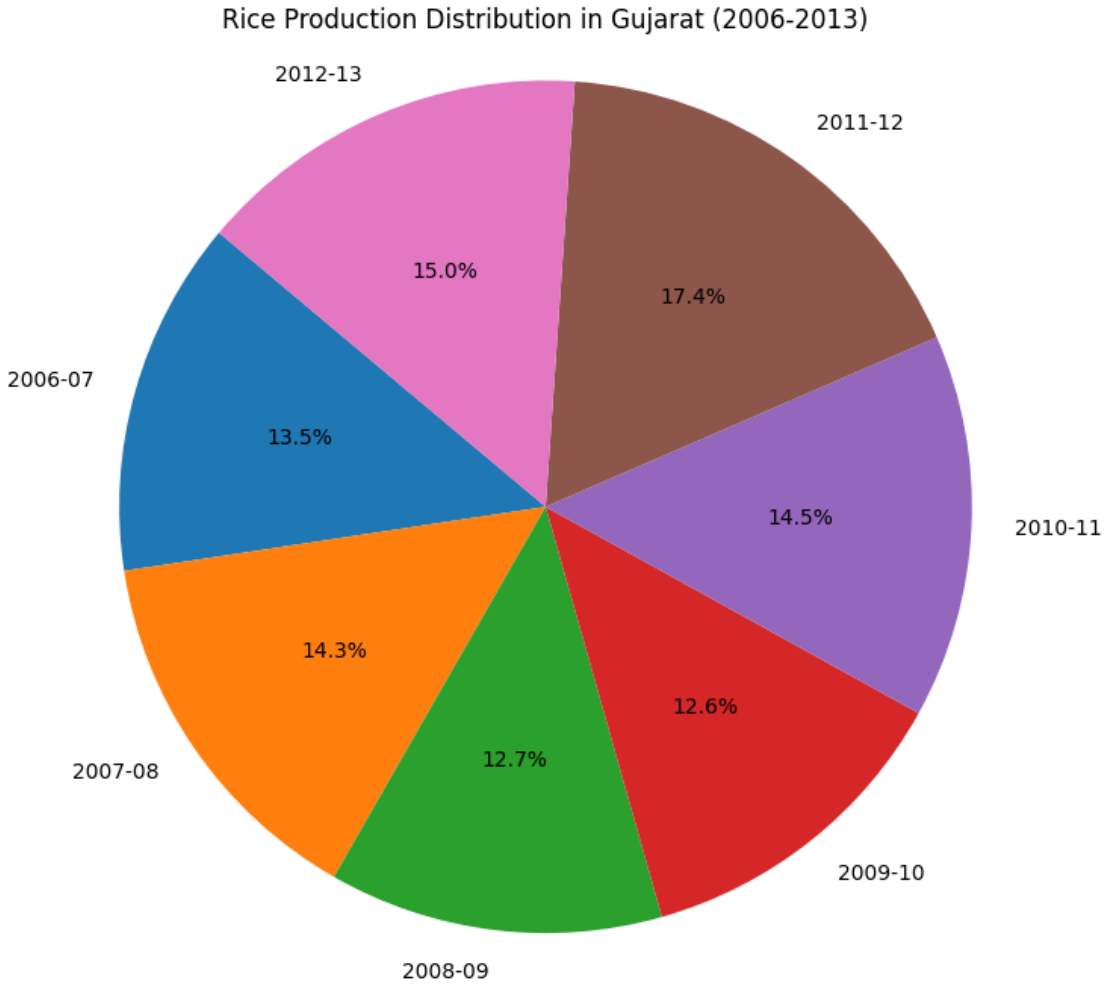

Rice Production Distribution in Assam (2006-2013)
2012-13
17.7%
2011-12
15.6%
2006-07
10.1%
16.4%
2010-11
11.5%
2007-08
13.8%
15.0%
2008-09
2009-10

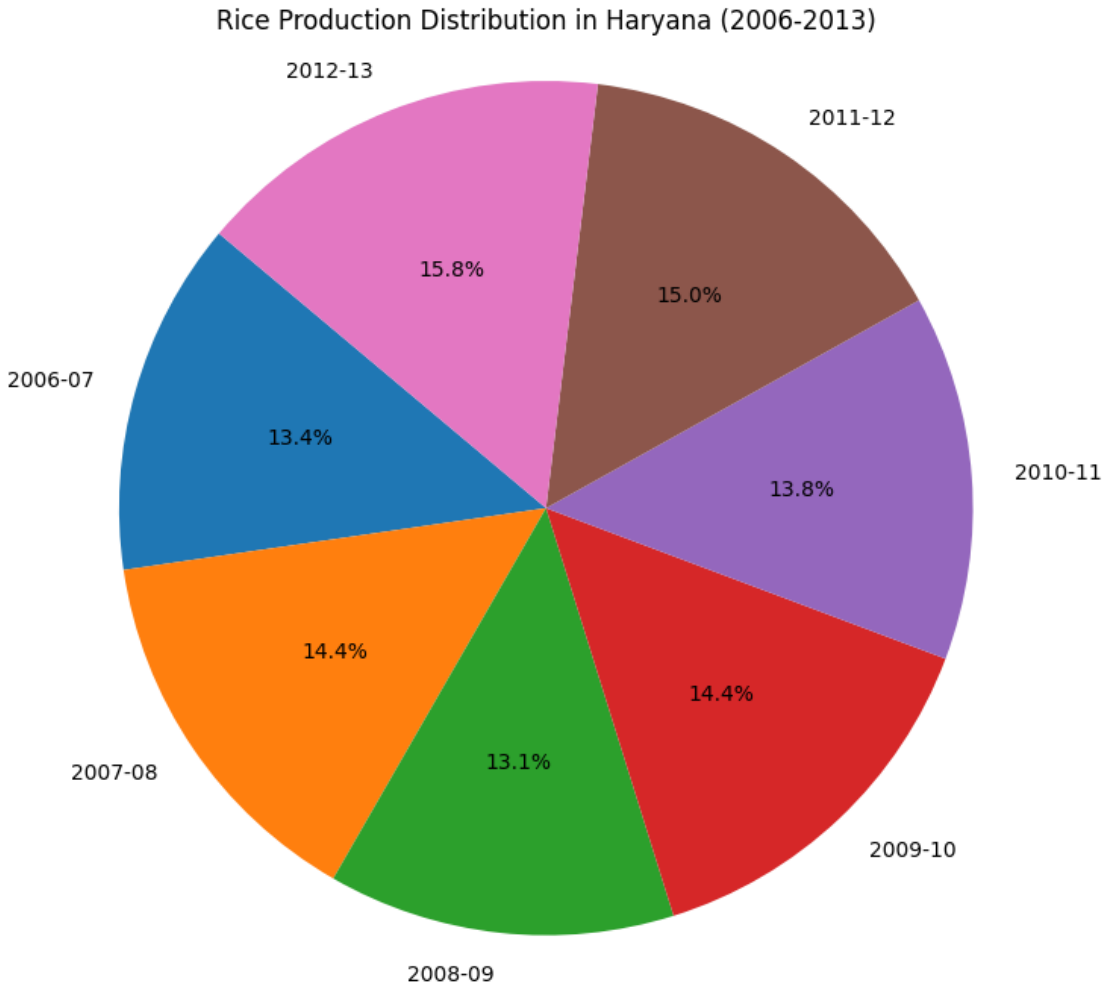

Rice Production Distribution in Bihar (2006-2013)
2012-13
20.7%
2011-12
19.7%
2006-07
13.7%
8.5%
2010-11
12.1%
2007-08
9.9%
15.4%
2009-10
2008-09

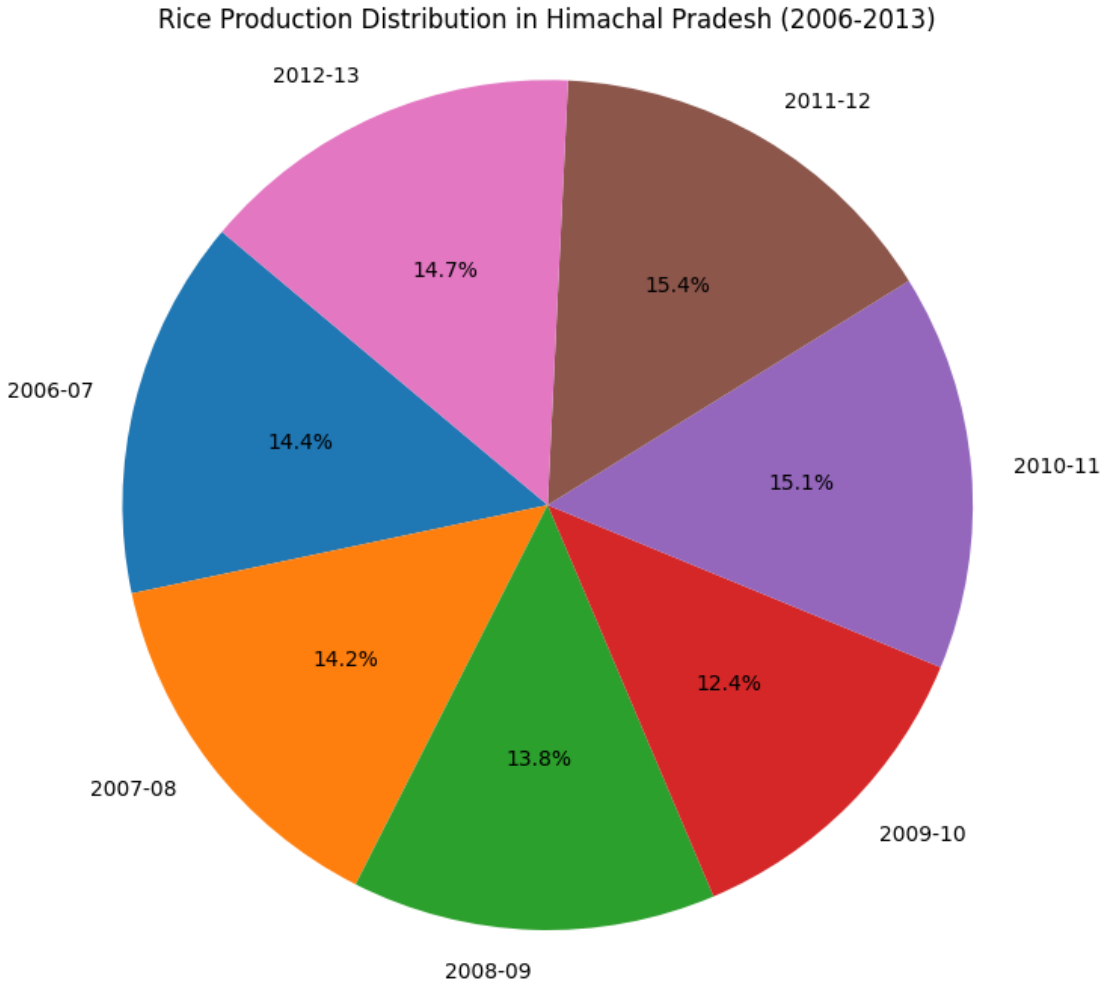

Rice Production Distribution in Chhattisgarh (2006-2013)
2012-13
17.5%
2011-12
16.0%
2006-07
13.3%
16.3%
2010-11
14.4%
2007-08
11.6%
10.9%
2009-10
2008-09

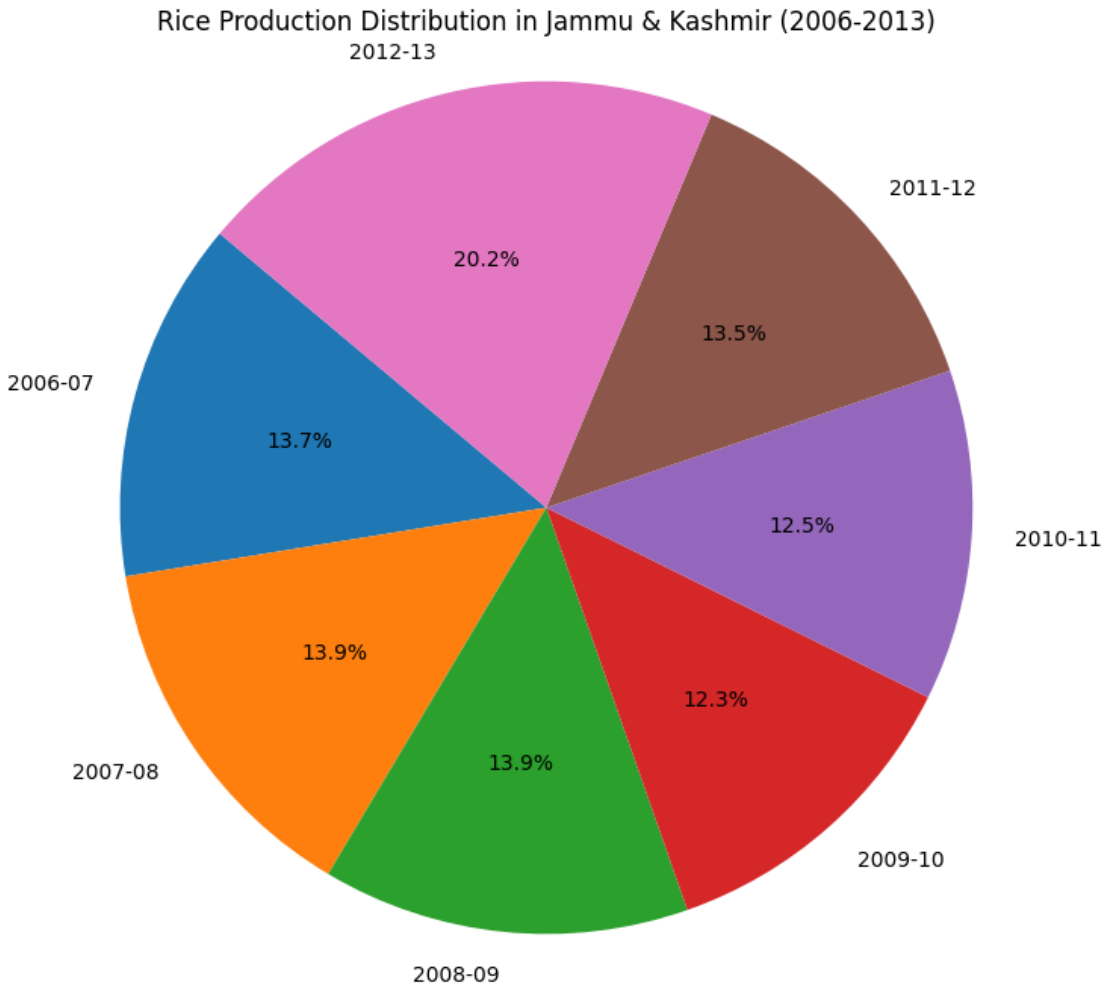

Rice Production Distribution in NCT of Delhi (2006-2013)
2011-12
2012-13
2010-11
11.5%
11.4%
11.4%
2006-07
18.0%
11.2%
2009-10
18.2%
18.2%
2007-08
2008-09

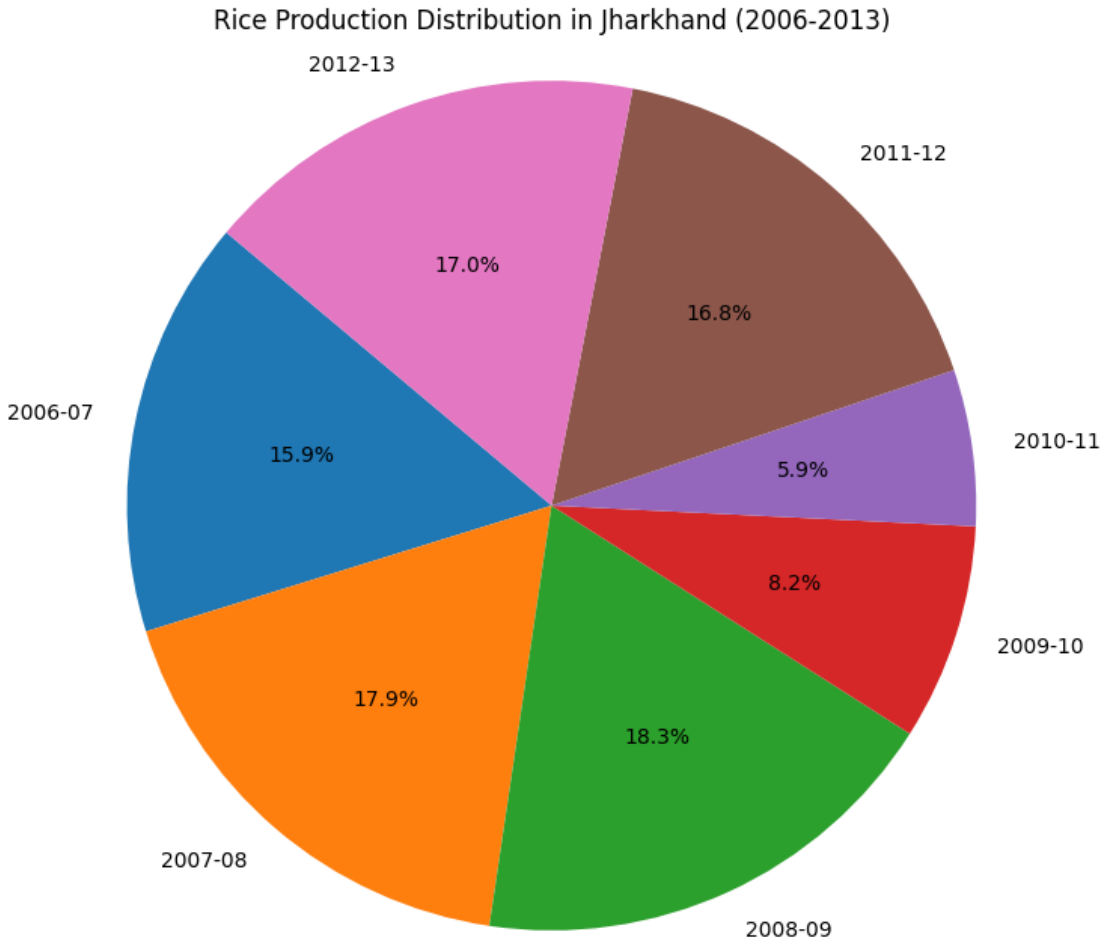

Rice Production Distribution in Goa (2006-2013)
2012-13
2011-12
14.7%
14.6%
2006-07
2010-11
15.6%
13.8%
12.0%
14.6%
14.8%
2009-10
2007-08
2008-09

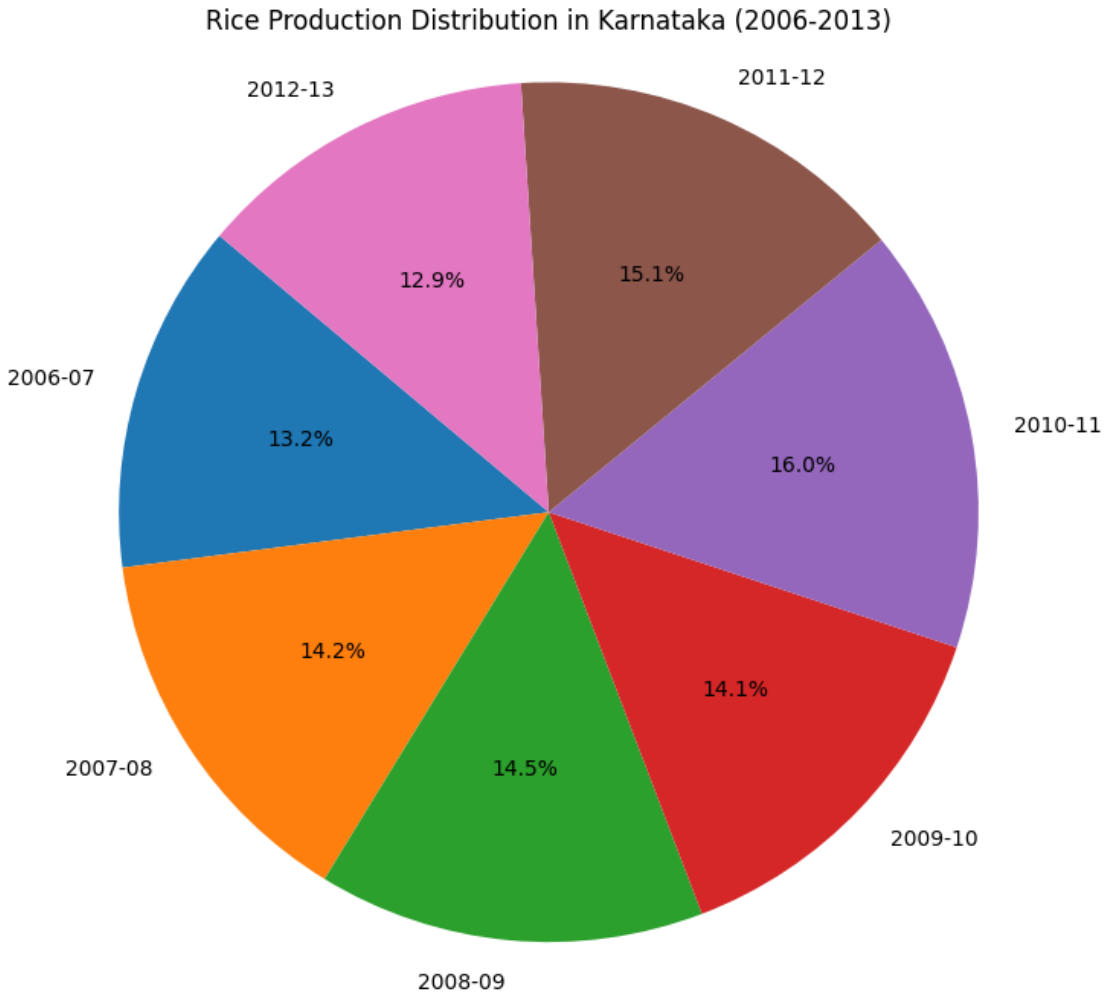

Rice Production Distribution in Gujarat (2006-2013)
2012-13
2011-12
15.0%
17.4%
2006-07
13.5%
14.5%
2010-11
14.3%
12.6%
2007-08
12.7%
2009-10
2008-09

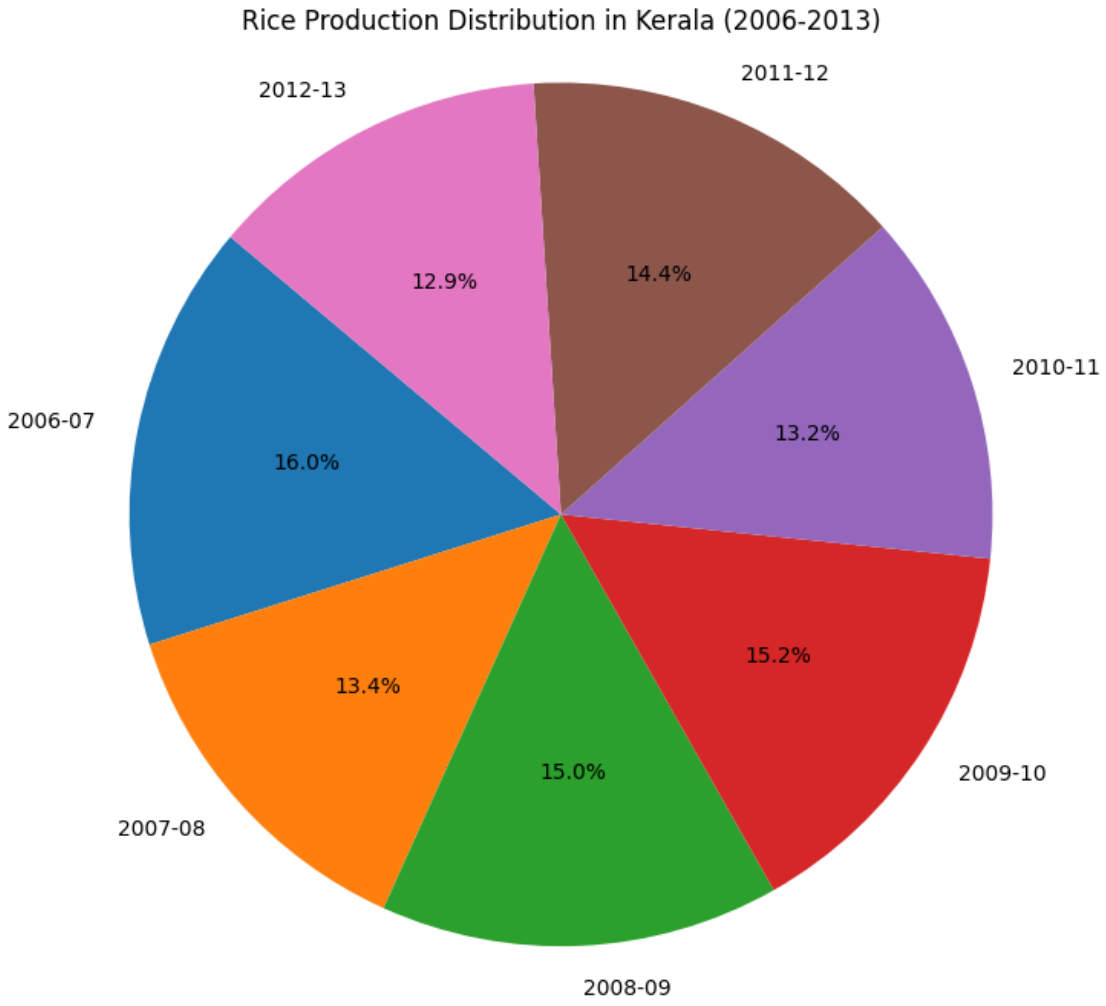

Rice Production Distribution in Haryana (2006-2013)
2012-13
2011-12
15.8%
15.0%
2006-07
13.4%
2010-11
13.8%
14.4%
14.4%
2007-08
13.1%
2009-10
2008-09

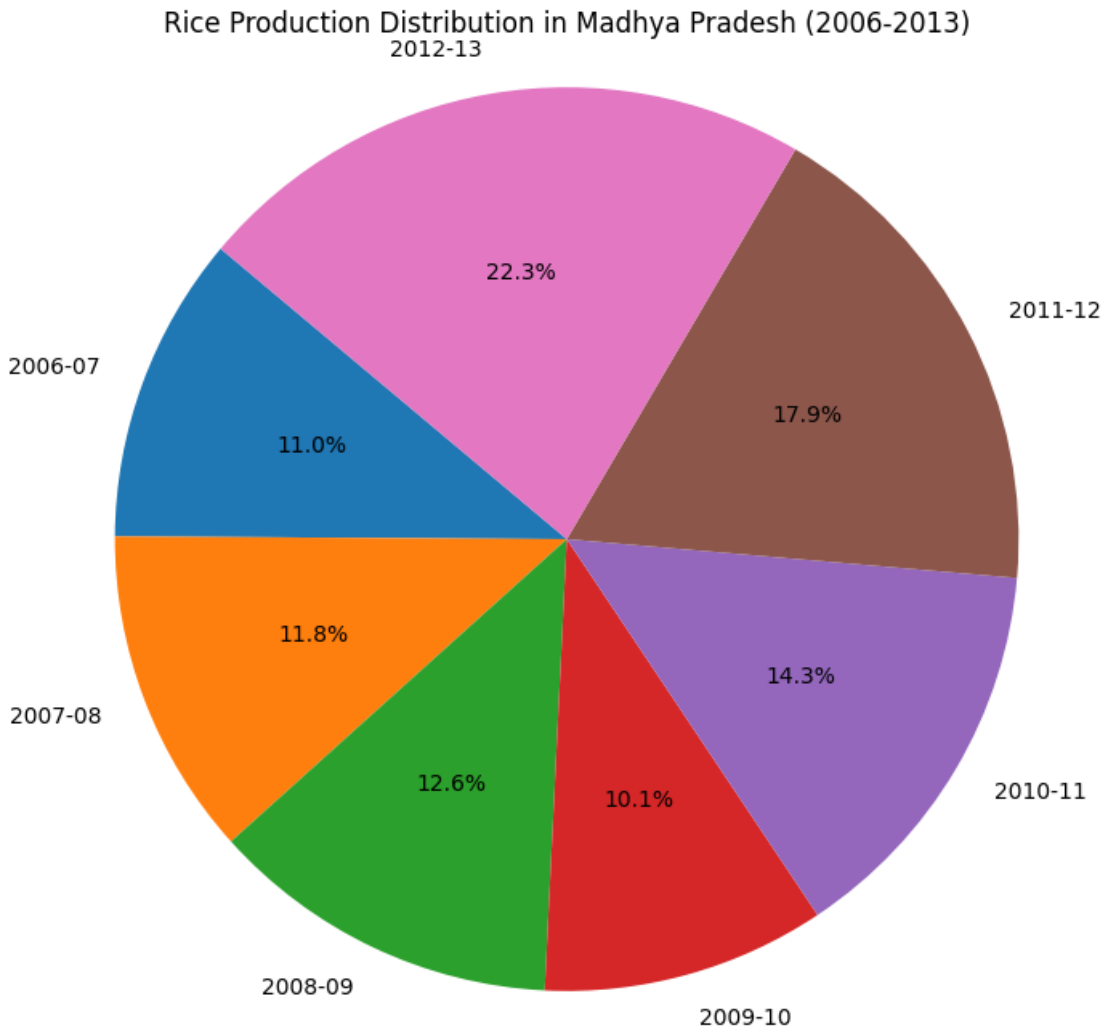

Rice Production Distribution in Himachal Pradesh (2006-2013)
2012-13
2011-12
14.7%
15.4%
2006-07
14.4%
2010-11
15.1%
14.2%
12.4%
13.8%
2007-08
2009-10
2008-09

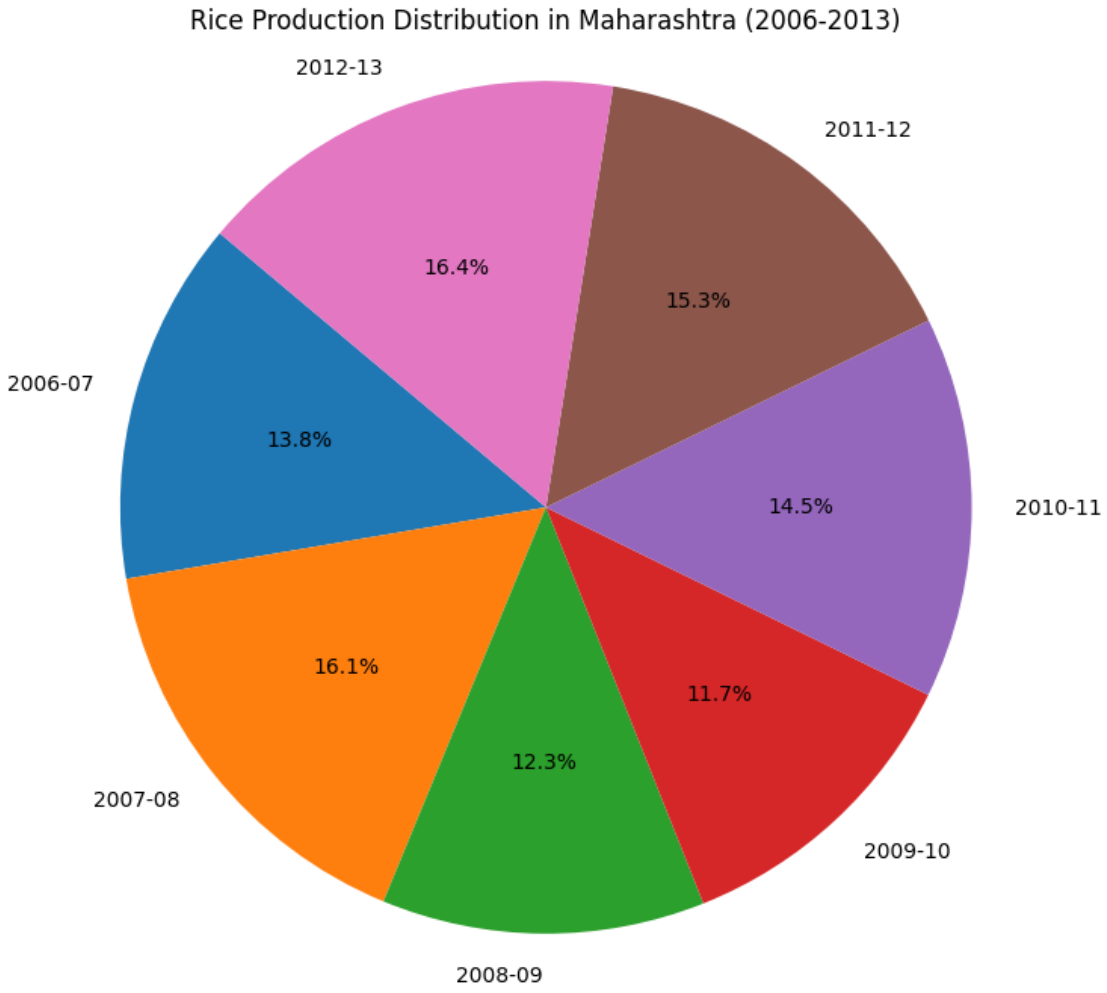

Rice Production Distribution in Jammu & Kashmir (2006-2013)
2012-13
20.2%
2011-12
13.5%
2006-07
13.7%
12.5%
2010-11
13.9%
12.3%
2007-08
13.9%
2009-10
2008-09

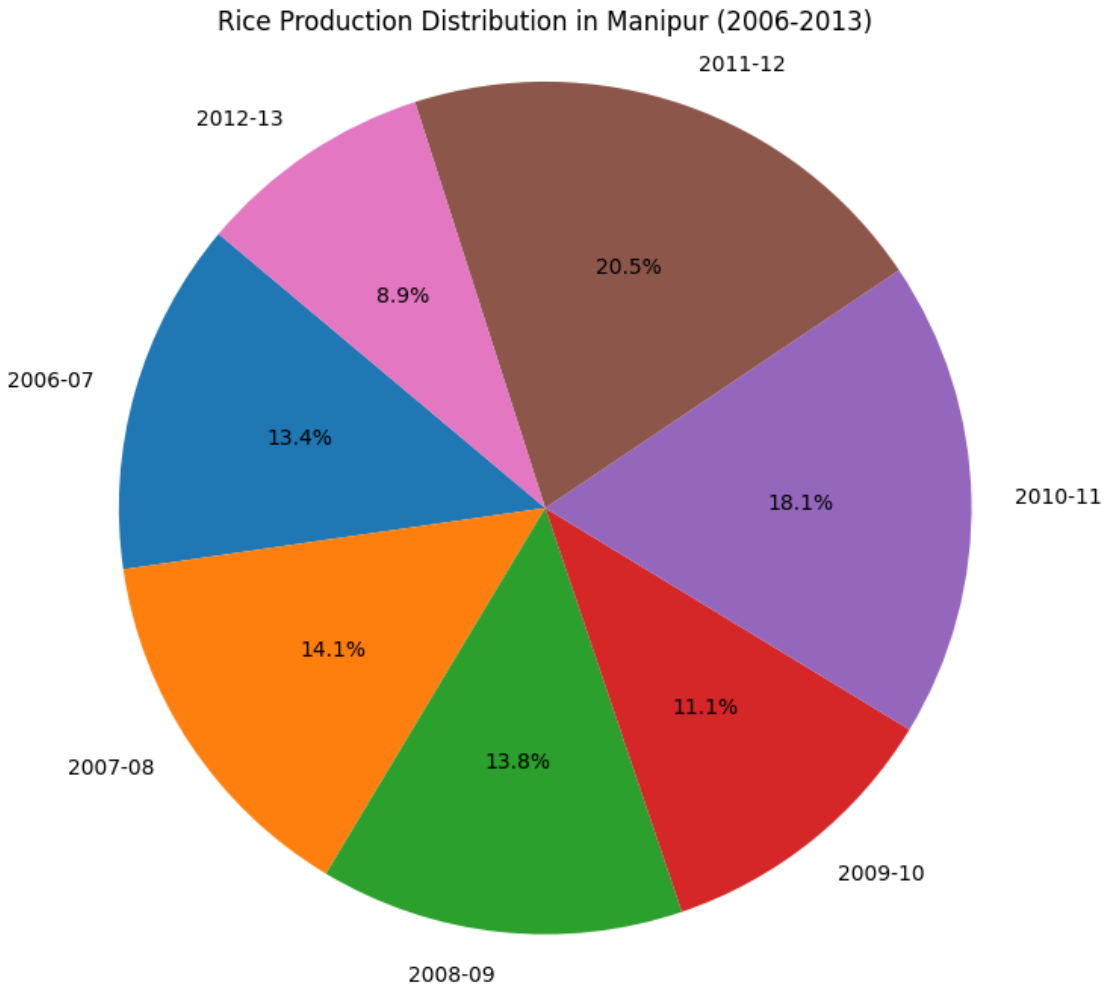

Rice Production Distribution in Jharkhand (2006-2013)
2012-13
17.0%
2011-12
16.8%
2010-11
5.9%
2009-10
8.2%
2008-09
18.3%
2007-08
17.9%
2006-07
15.9%

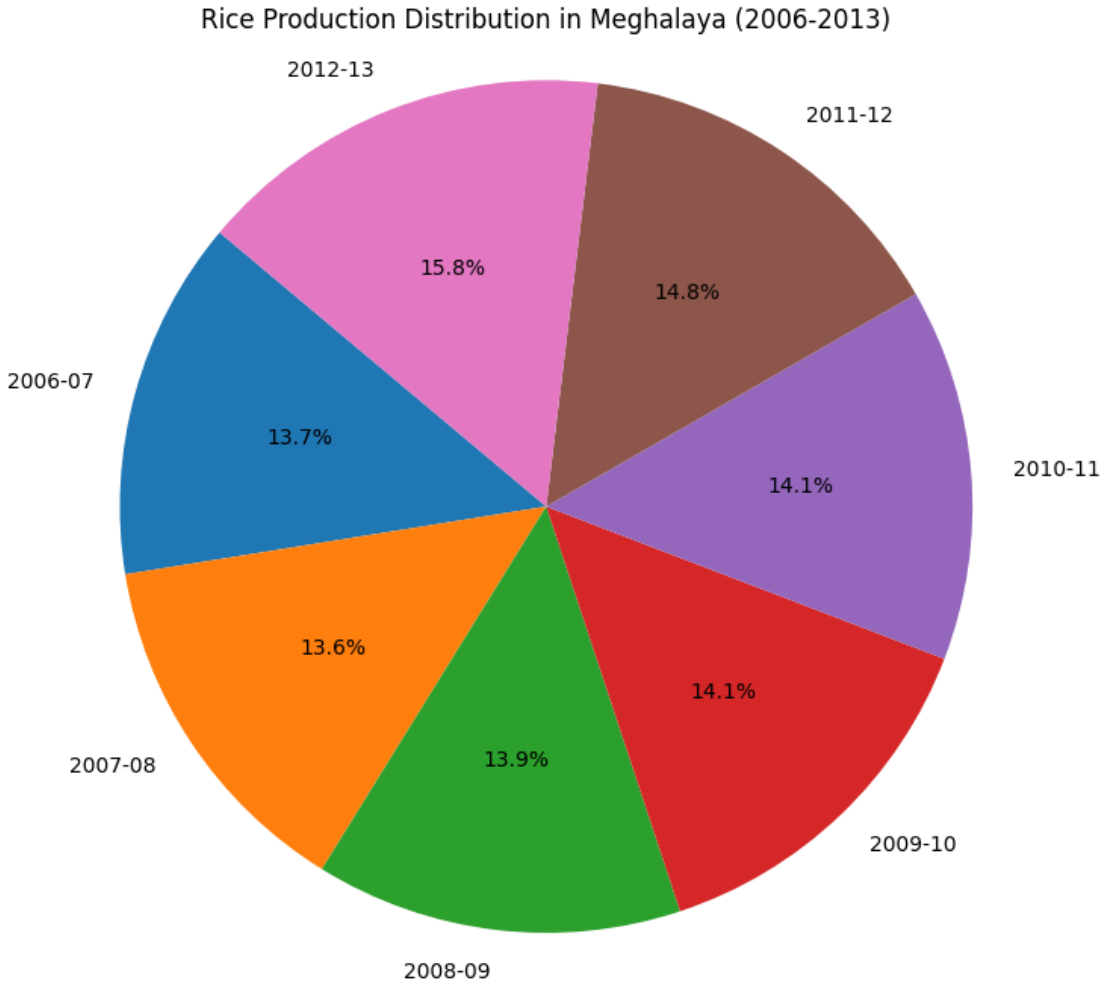

Rice Production Distribution in Karnataka (2006-2013)
2012-13
2011-12
12.9%
15.1%
2006-07
13.2%
2010-11
16.0%
14.2%
14.1%
2007-08
14.5%
2009-10
2008-09

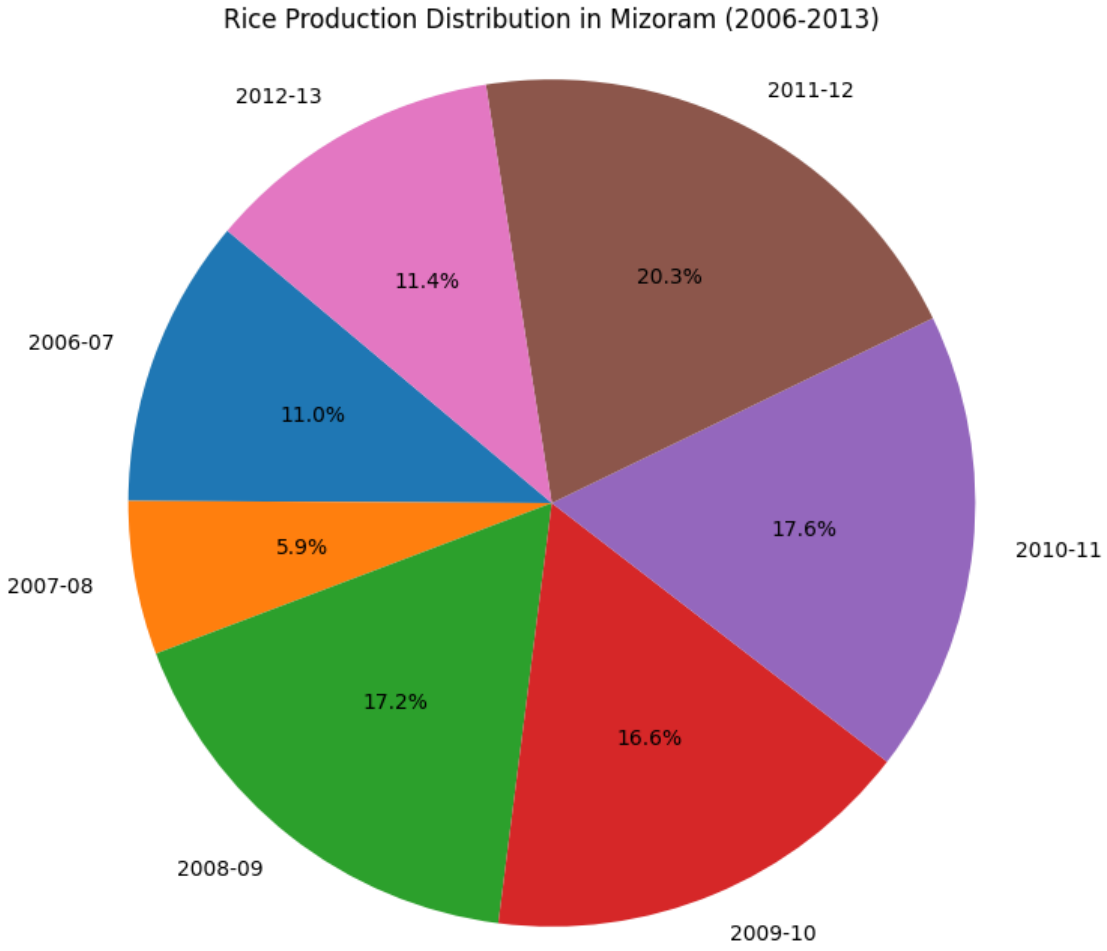

Rice Production Distribution in Kerala (2006-2013)
2011-12
2012-13
14.4%
12.9%
2010-11
2006-07
13.2%
16.0%
15.2%
13.4%
2009-10
15.0%
2007-08
2008-09

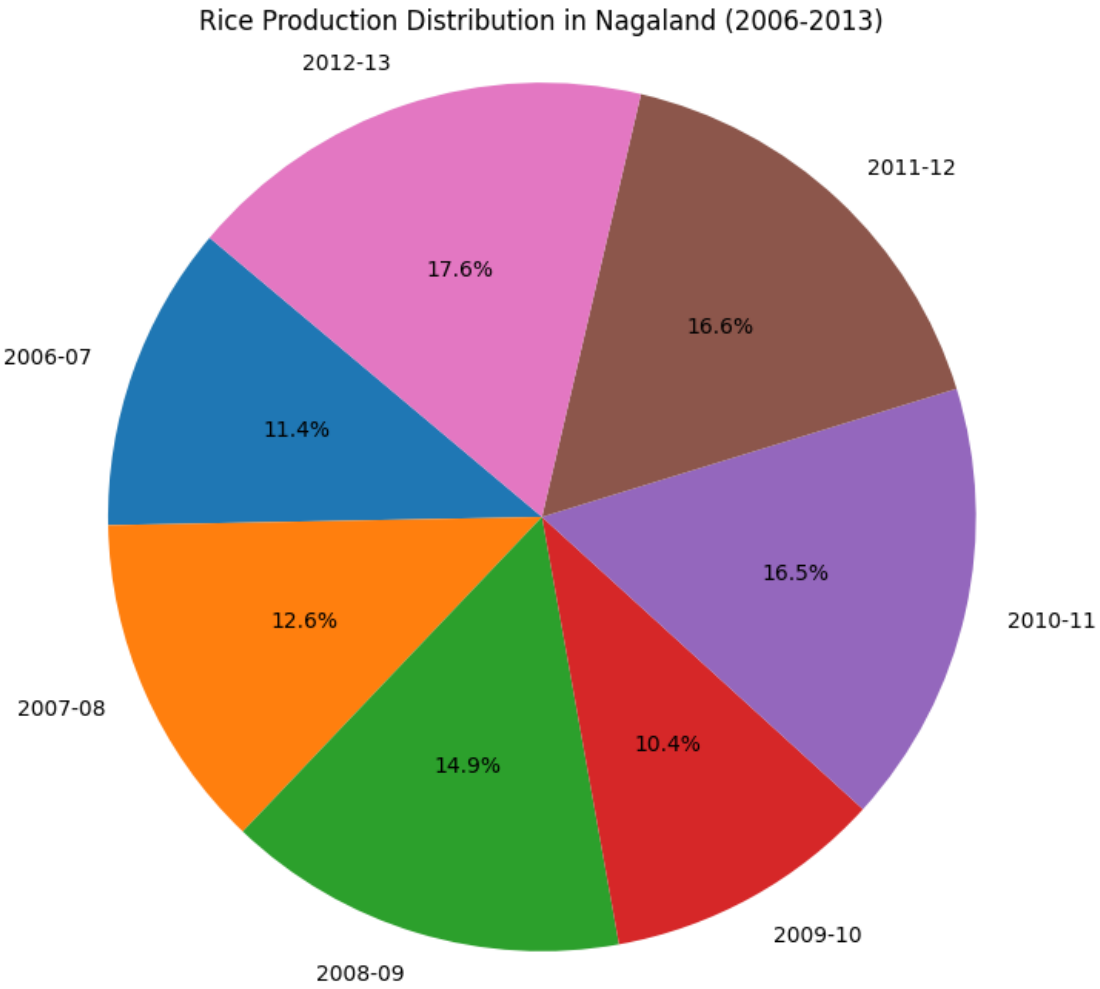

Rice Production Distribution in Madhya Pradesh (2006-2013)
2012-13
22.3%
2011-12
17.9%
2006-07
11.0%
11.8%
2007-08
14.3%
2010-11
12.6%
10.1%
2008-09
2009-10

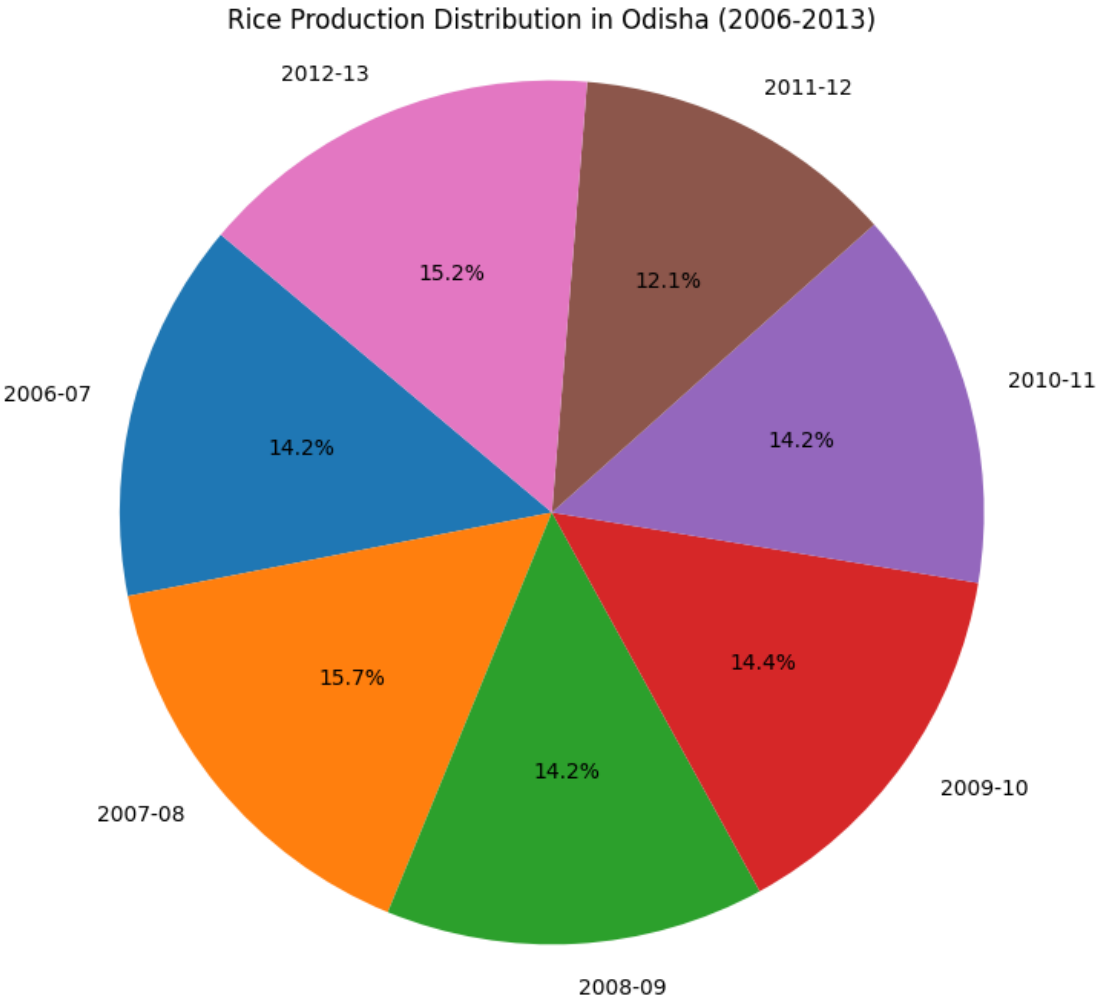

Rice Production Distribution in Maharashtra (2006-2013)
2012-13
2011-12
16.4%
15.3%
2006-07
13.8%
14.5%
2010-11
16.1%
11.7%
12.3%
2007-08
2009-10
2008-09

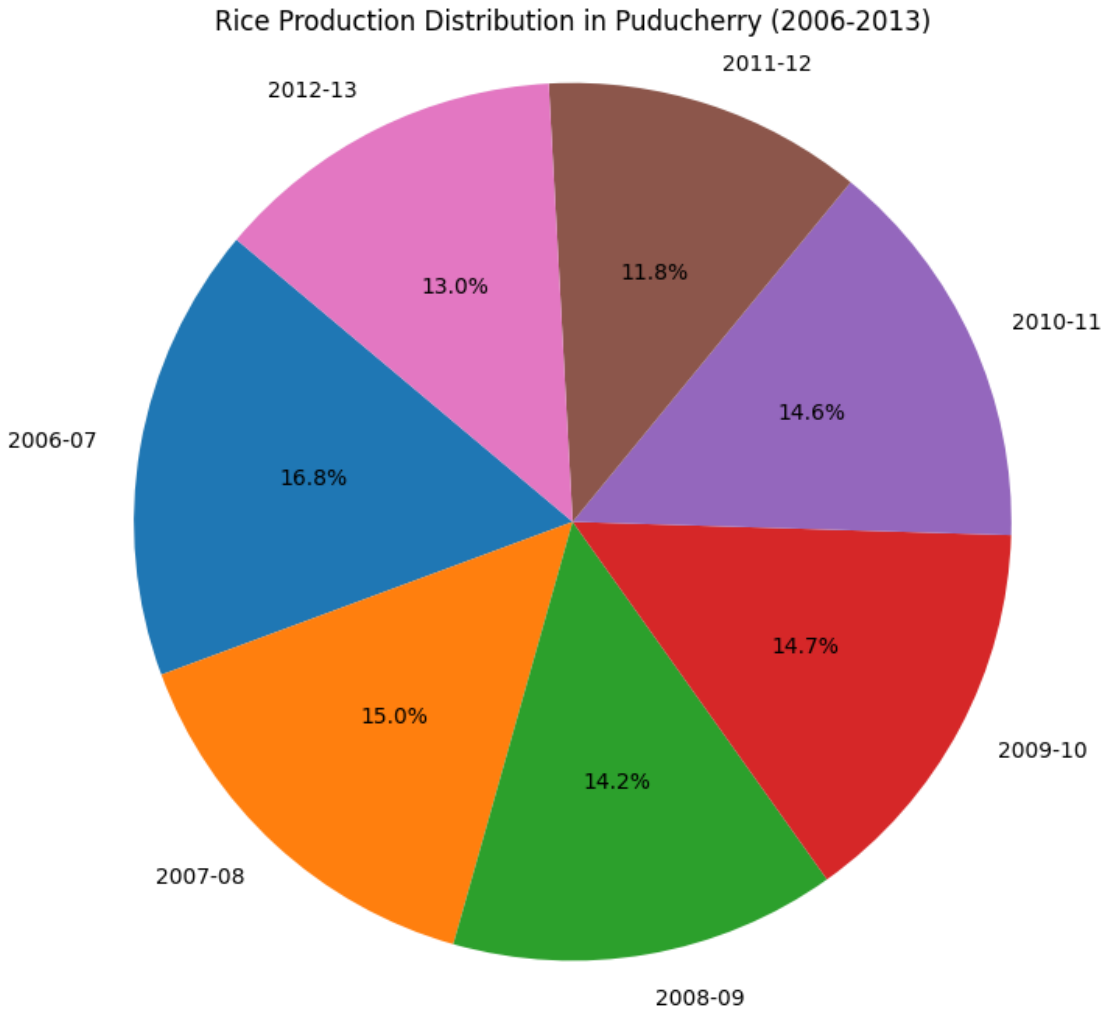

Rice Production Distribution in Manipur (2006-2013)
2011-12
2012-13
20.5%
8.9%
2006-07
13.4%
18.1%
2010-11
14.1%
11.1%
2007-08
13.8%
2009-10
2008-09

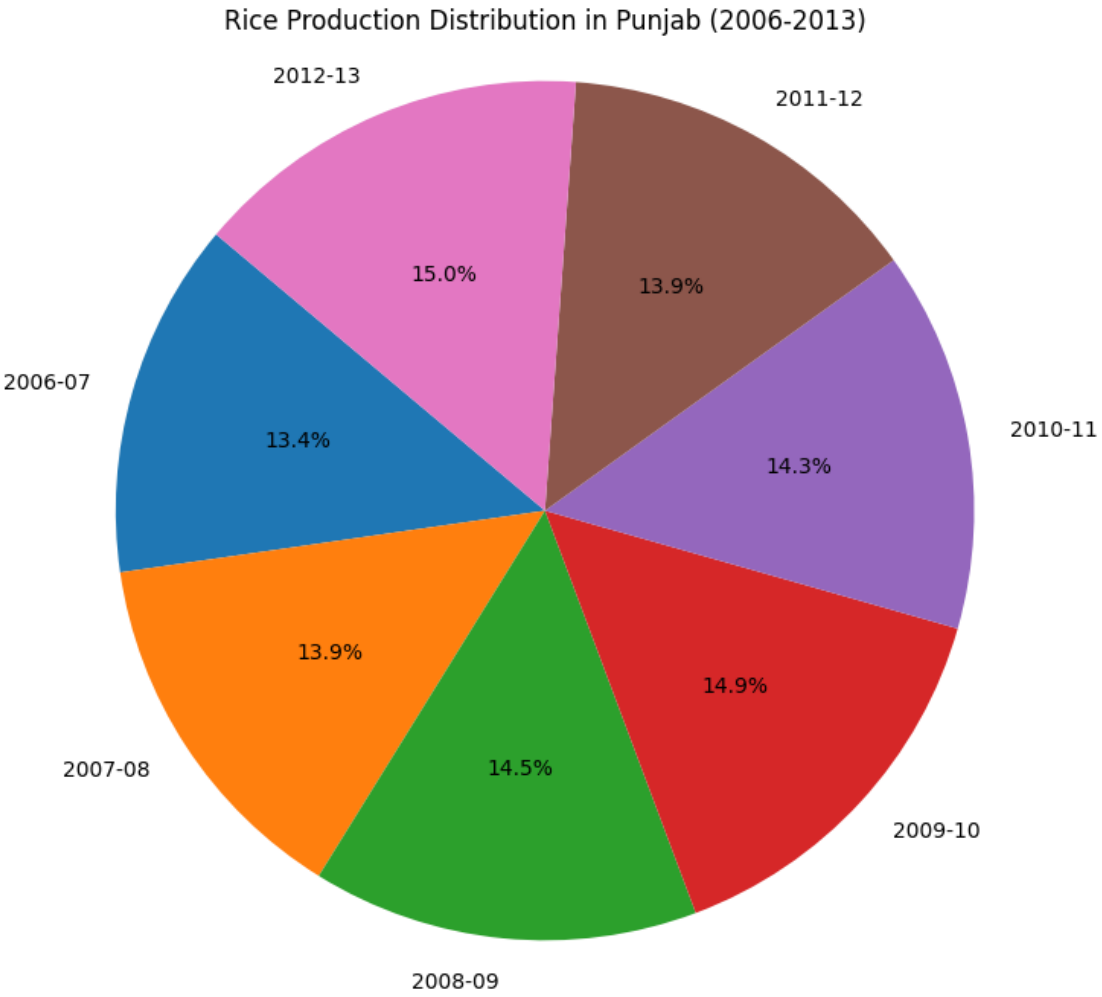

Rice Production Distribution in Meghalaya (2006-2013)
2012-13
2011-12
15.8%
14.8%
2006-07
13.7%
2010-11
14.1%
13.6%
14.1%
2007-08
13.9%
2009-10
2008-09

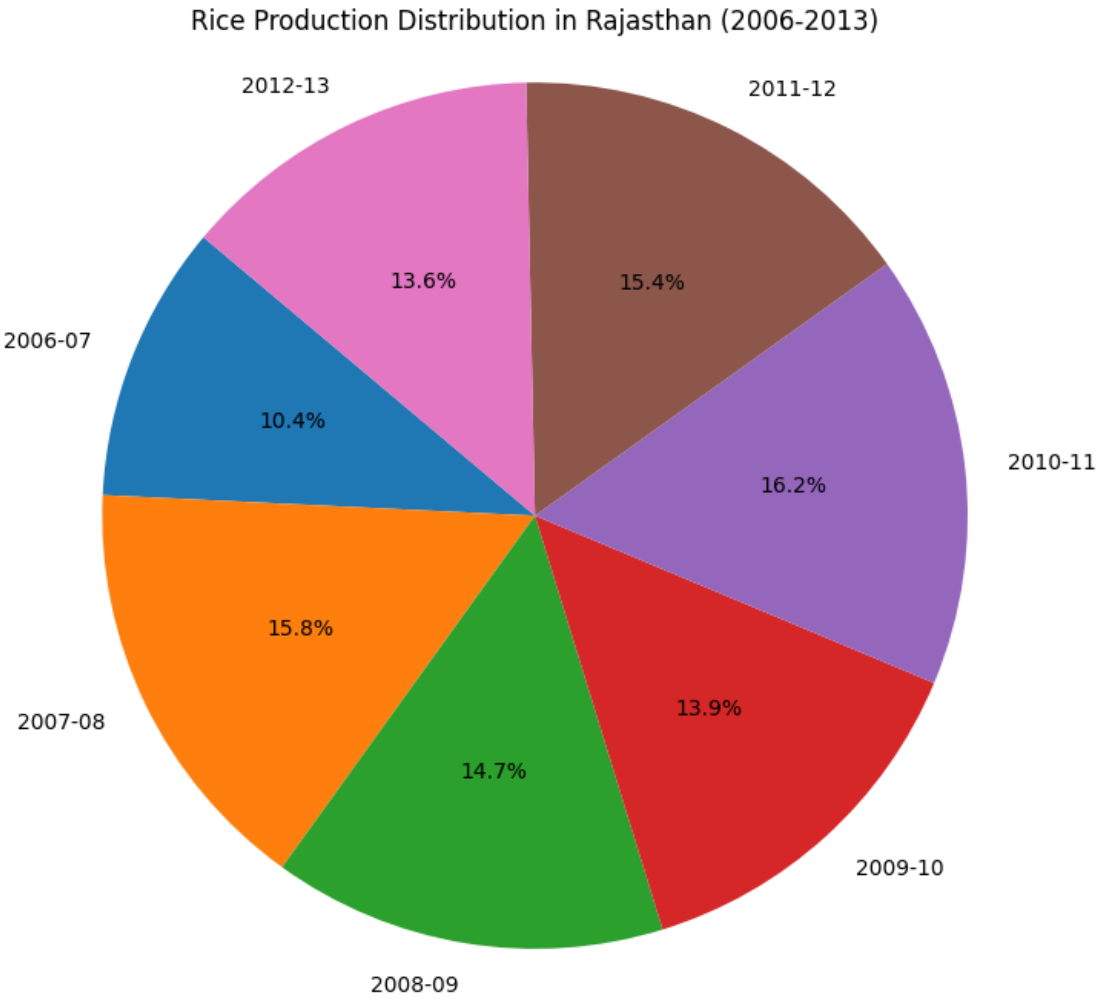

Rice Production Distribution in Mizoram (2006-2013)
2012-13
2011-12
11.4%
20.3%
2006-07
11.0%
17.6%
2010-11
5.9%
2007-08
17.2%
16.6%
2008-09
2009-10

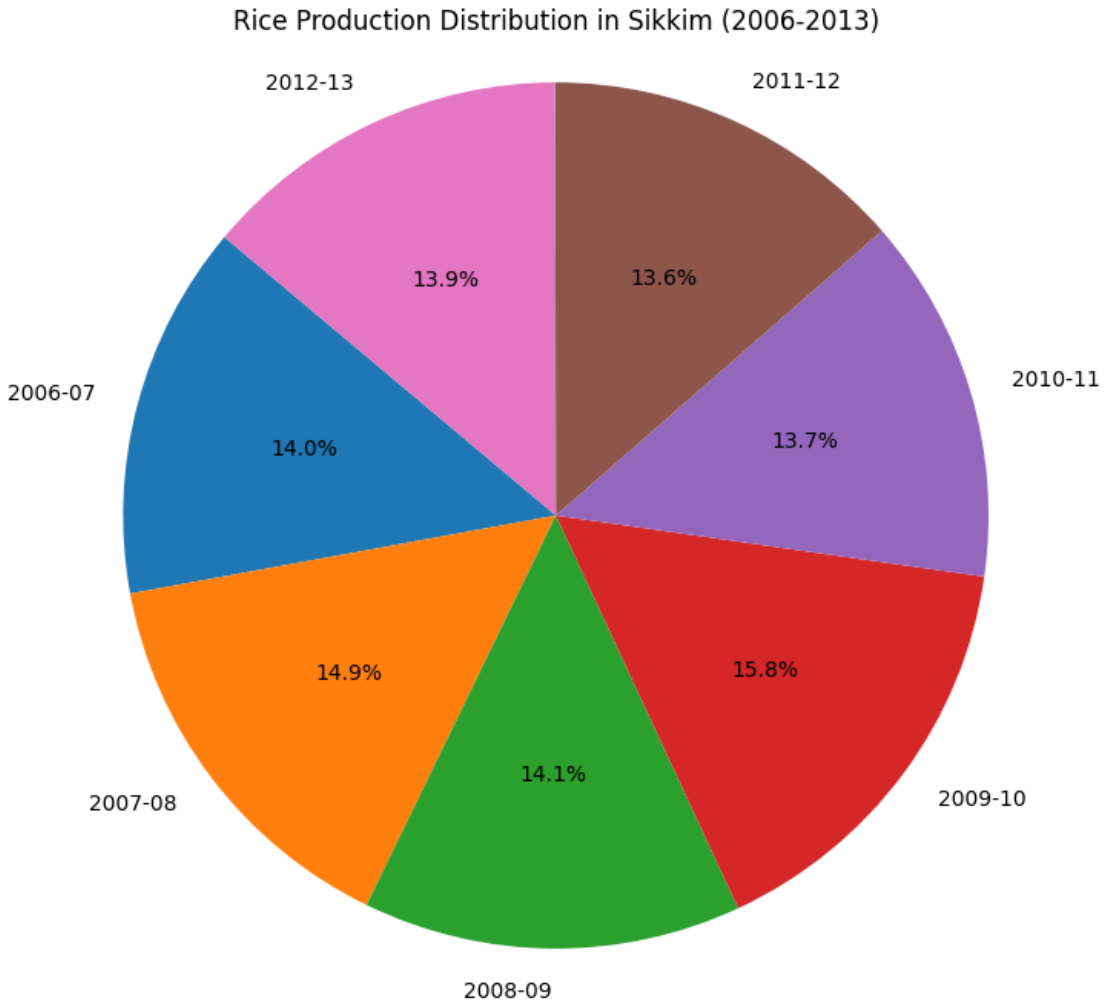

Rice Production Distribution in Nagaland (2006-2013)
2012-13
17.6%
2011-12
16.6%
2006-07
11.4%
16.5%
2010-11
12.6%
2007-08
14.9%
10.4%
2009-10
2008-09

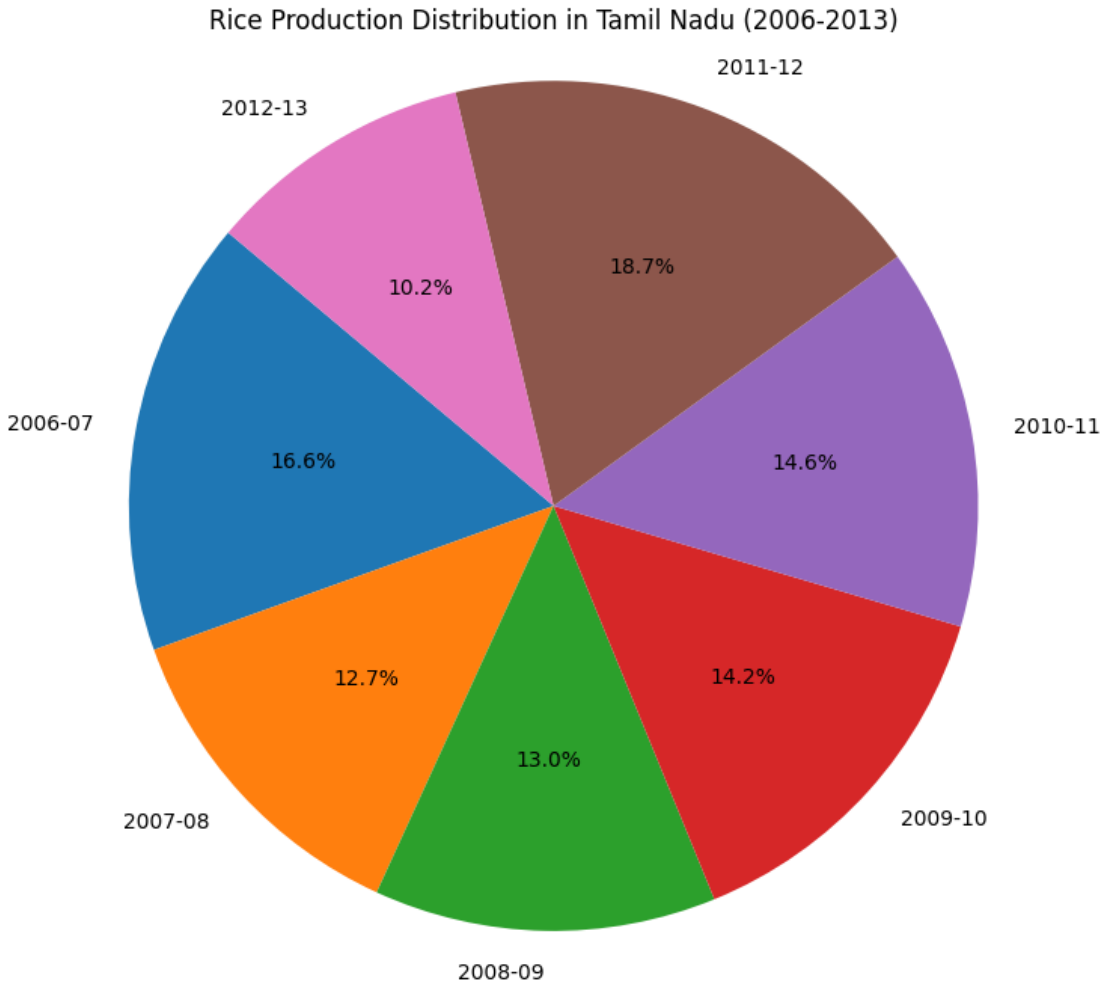

Rice Production Distribution in Odisha (2006-2013)
2012-13
2011-12
15.2%
12.1%
2006-07
2010-11
14.2%
14.2%
15.7%
14.4%
14.2%
2007-08
2009-10
2008-09

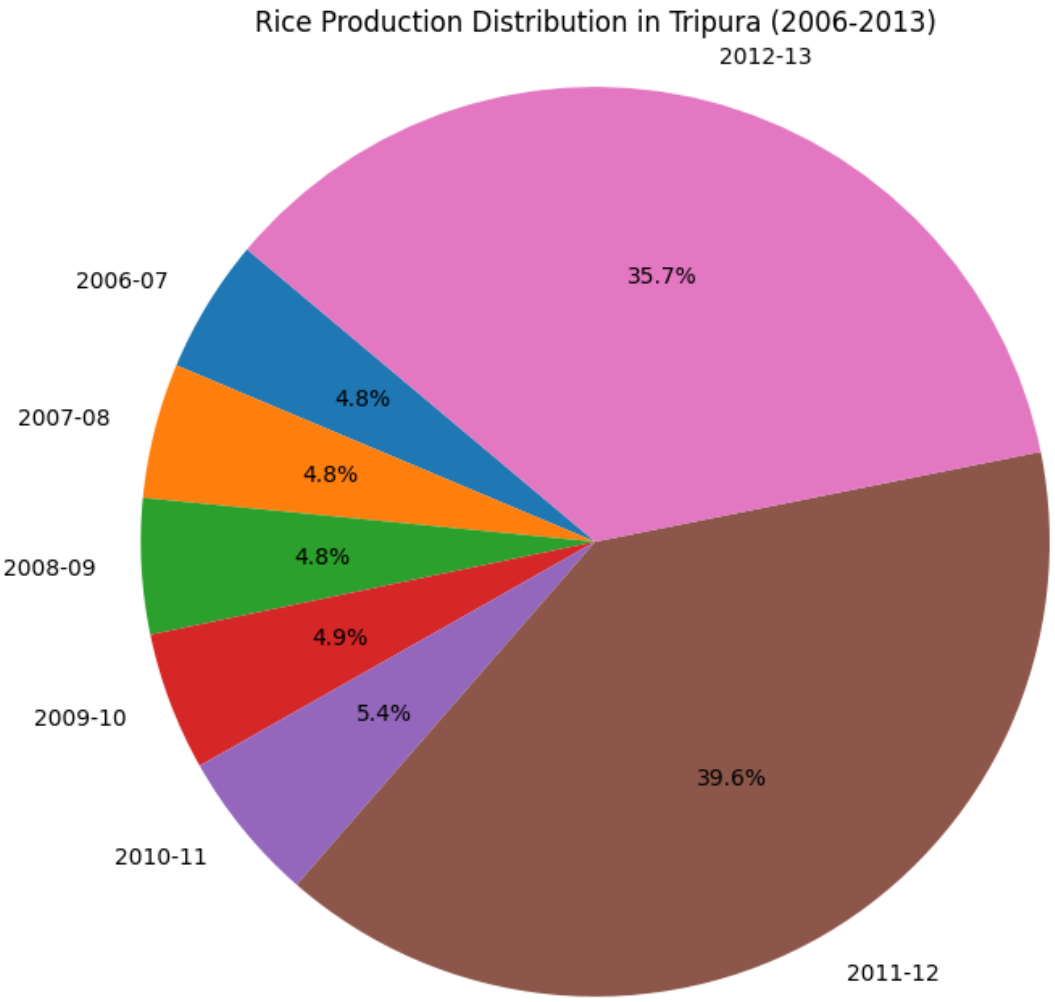

Rice Production Distribution in Puducherry (2006-2013)
2011-12
2012-13
2010-11
2006-07
2009-10
2007-08
2008-09
11.8%
13.0%
14.6%
16.8%
14.7%
15.0%
14.2%

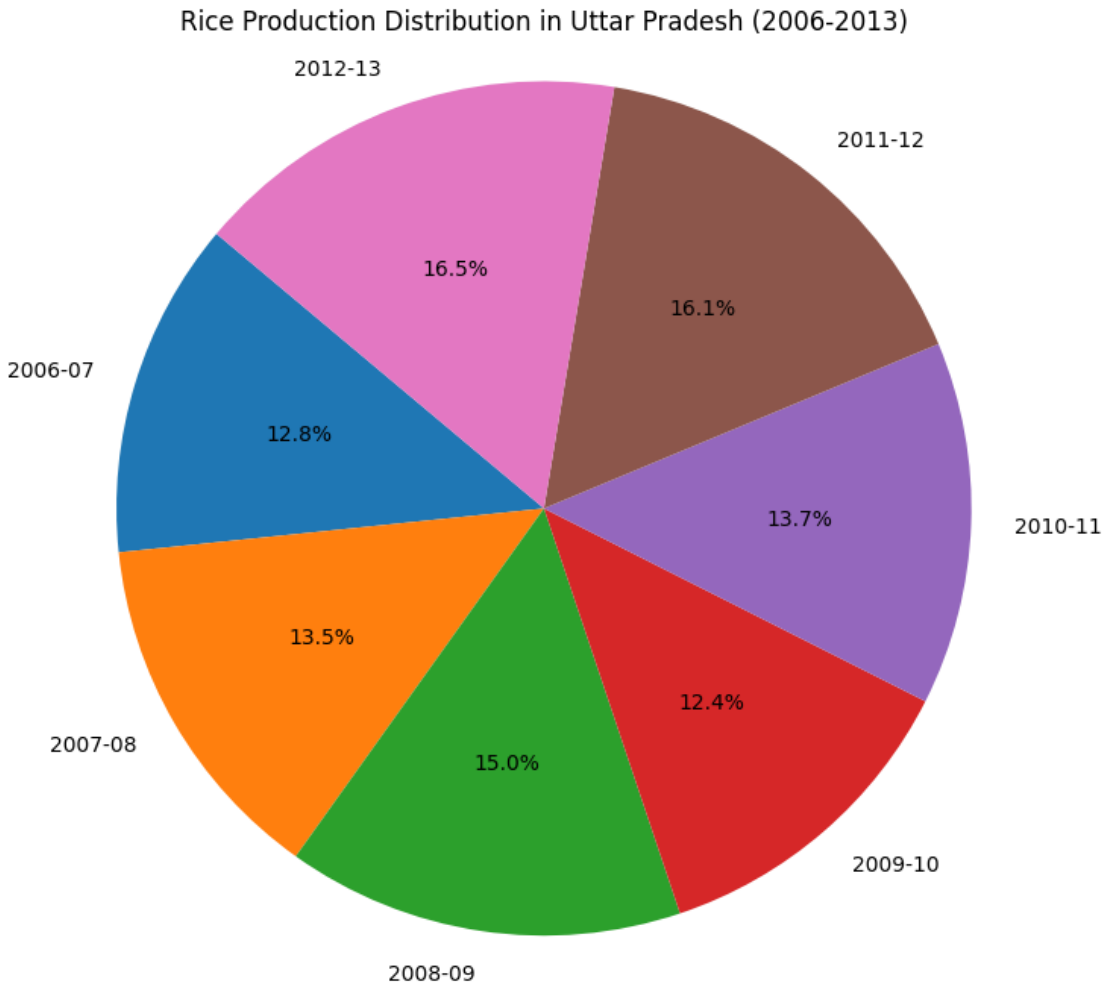

Rice Production Distribution in Punjab (2006-2013)
2012-13
2011-12
15.0%
13.9%
2006-07
13.4%
2010-11
14.3%
13.9%
14.9%
2007-08
14.5%
2009-10
2008-09

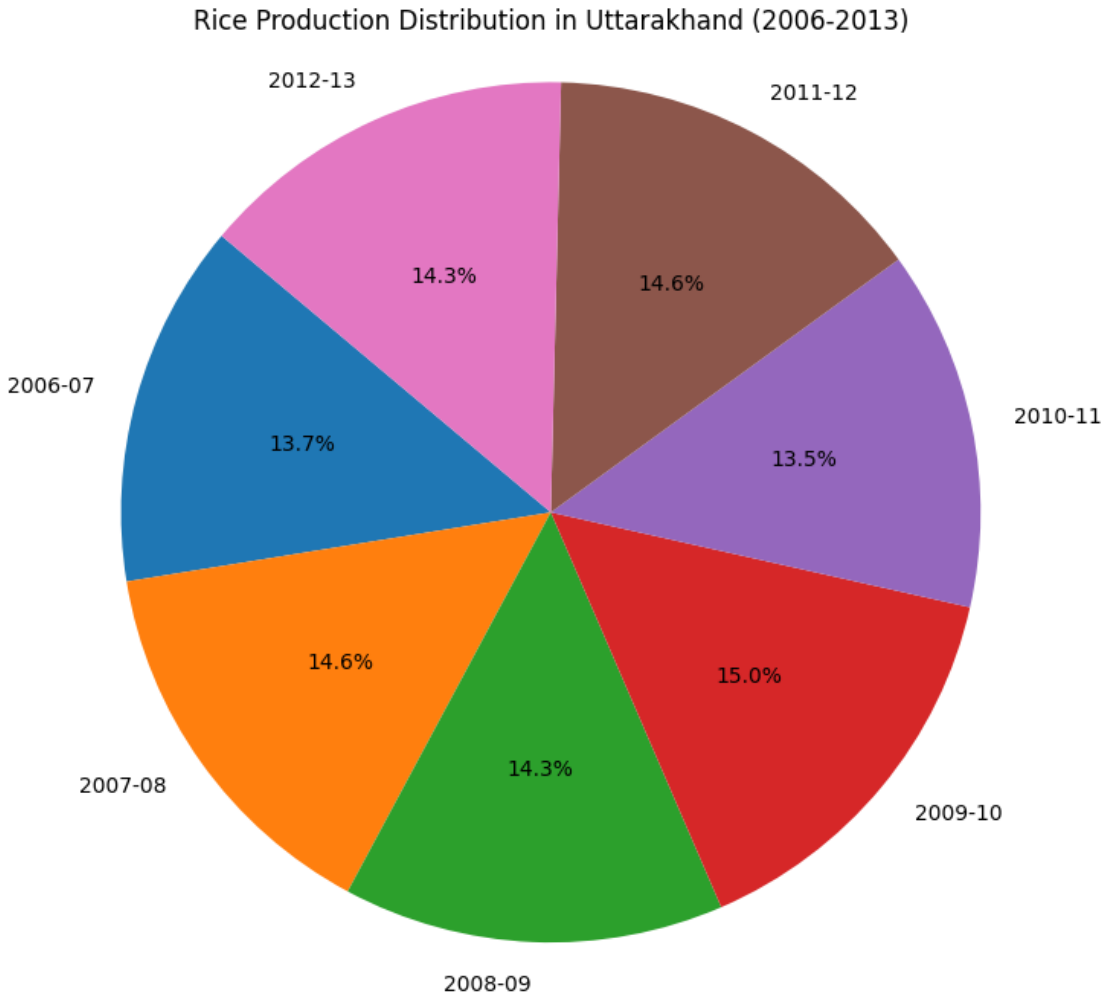

Rice Production Distribution in Rajasthan (2006-2013)
2012-13
2011-12
13.6%
15.4%
2006-07
10.4%
2010-11
16.2%
15.8%
2007-08
13.9%
14.7%
2009-10
2008-09

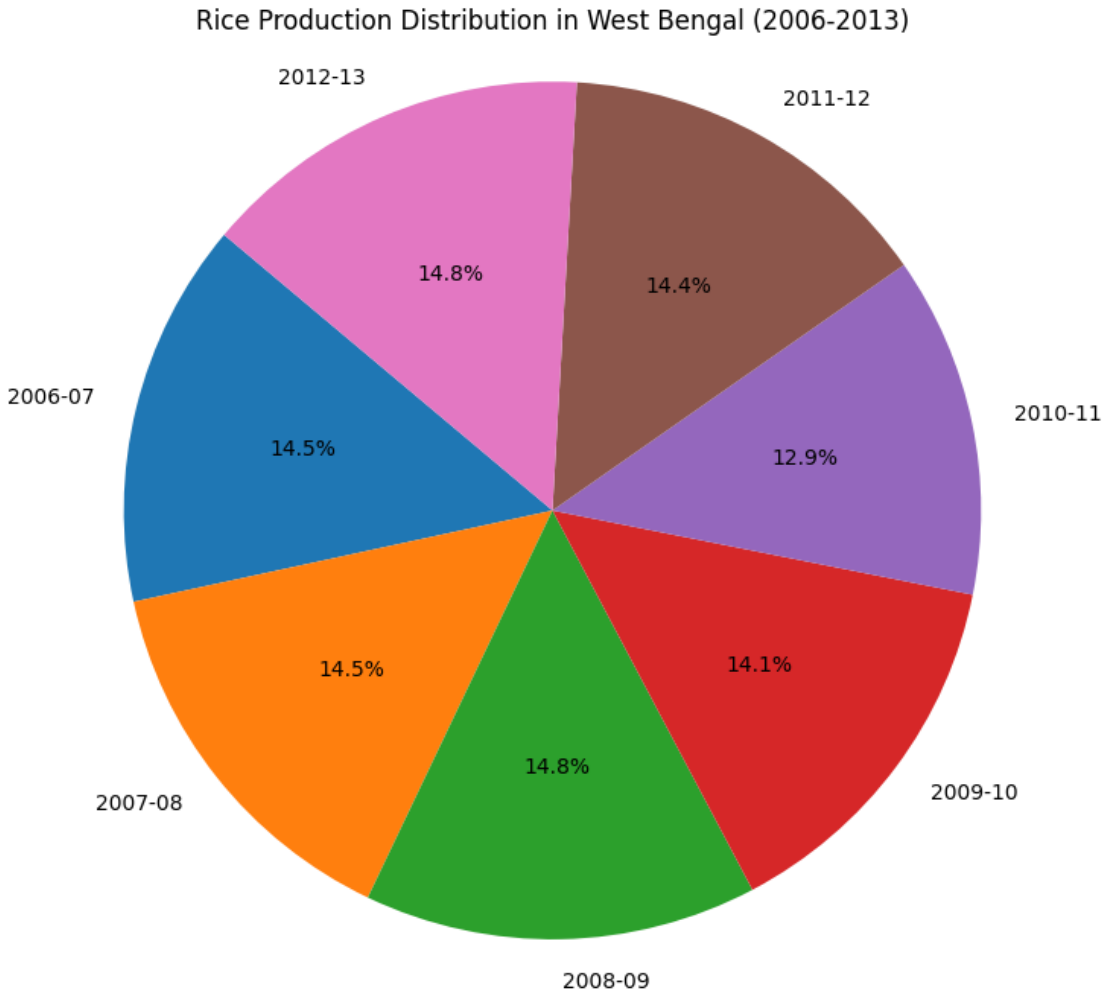

Rice Production Distribution in Sikkim (2006-2013)
2012-13
2011-12
13.9%
13.6%
2006-07
2010-11
14.0%
13.7%
14.9%
15.8%
2007-08
14.1%
2009-10
2008-09


Rice Production Distribution in Tamil Nadu (2006-2013)
2011-12
2012-13
18.7%
10.2%
2006-07
2010-11
16.6%
14.6%
12.7%
14.2%
13.0%
2007-08
2009-10
2008-09


Rice Production Distribution in Tripura (2006-2013)
2012-13
35.7%
2006-07
4.8%
2007-08
4.8%
2008-09
4.8%
2009-10
4.9%
2010-11
5.4%
2011-12
39.6%

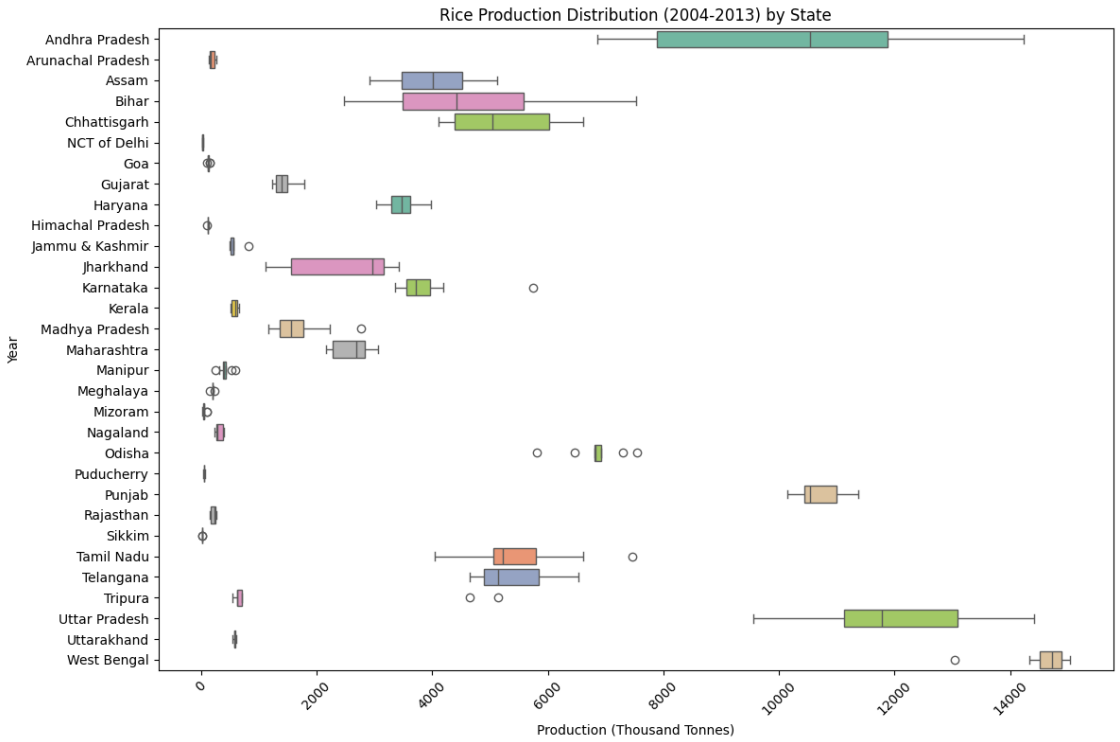

Rice Production Distribution in Uttar Pradesh (2006-2013)
2012-13
16.5%
2011-12
16.1%
2006-07
12.8%
2010-11
13.7%
2007-08
13.5%
2009-10
12.4%
2008-09
15.0%


Rice Production Distribution in Uttarakhand (2006-2013)
2012-13
2011-12
14.3%
14.6%
2006-07
2010-11
13.7%
13.5%
14.6%
15.0%
2007-08
14.3%
2009-10
2008-09

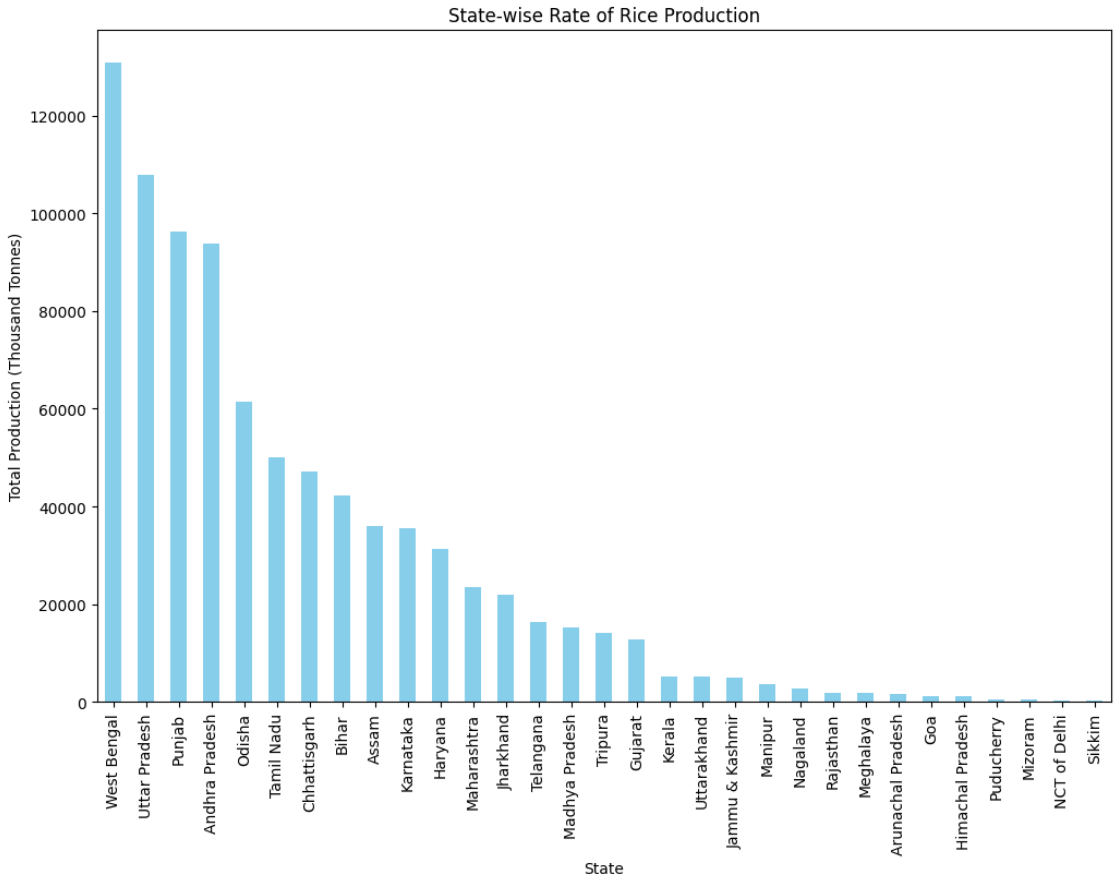


[46]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Transpose the DataFrame to have years as rows and states as columns
df_transposed = df.T

# Set up the figure and axis
plt.figure(figsize=(12, 8))

# Create the boxplot
sns.boxplot(data=df_transposed, orient="h", palette="Set2")

# Set the title and labels
plt.title('Rice Production Distribution (2004-2013) by State')
plt.xlabel('Production (Thousand Tonnes)')
plt.ylabel('Year')
```

```
# Rotate the y-axis labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```

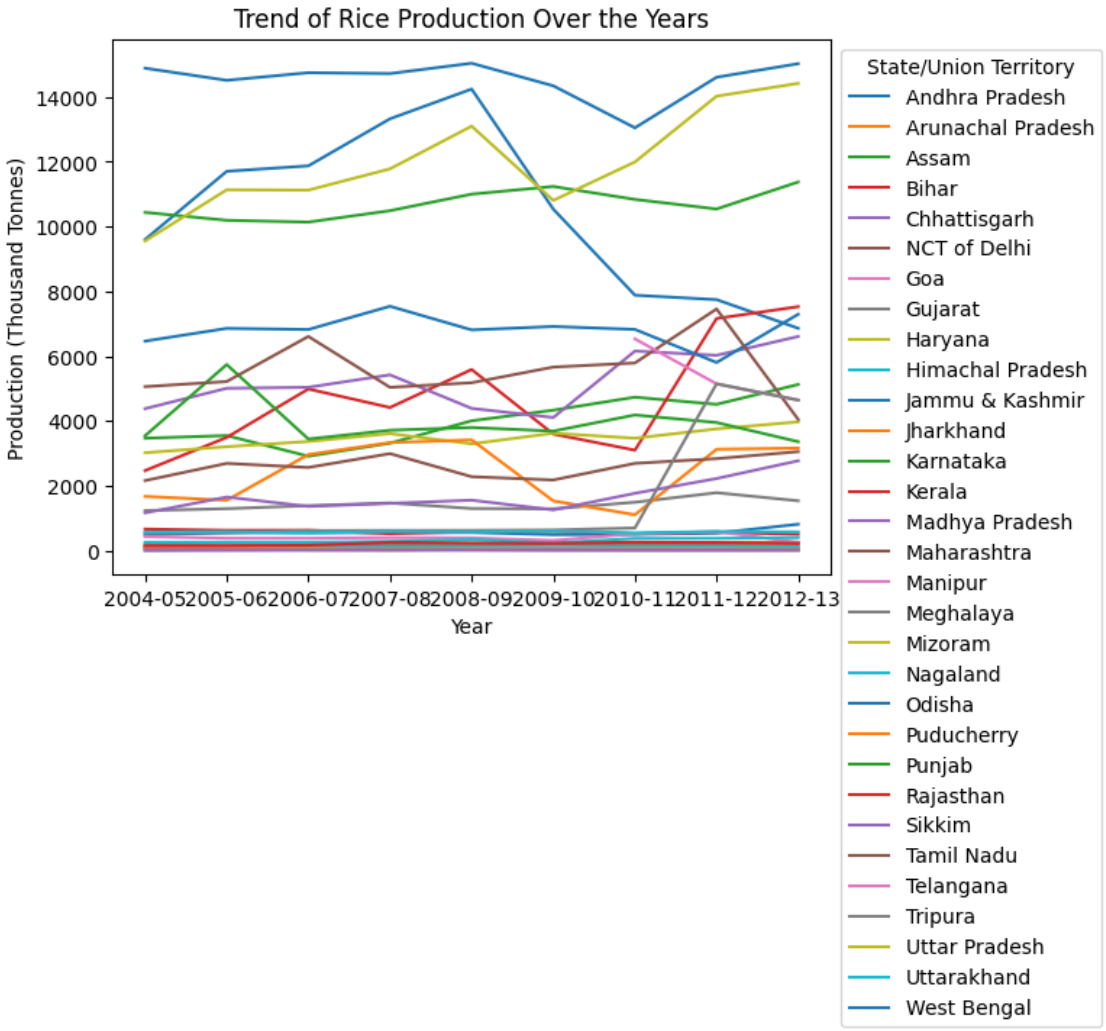


[47]:
```
# Calculate the total production for each state
total_production = df.sum(axis=1)

# Sort the total production in descending order
total_production_sorted = total_production.sort_values(ascending=False)

# Plot the state-wise rate of production
plt.figure(figsize=(12, 8))
total_production_sorted.plot(kind='bar', color='skyblue')
plt.title('State-wise Rate of Rice Production')
plt.xlabel('State')
plt.ylabel('Total Production (Thousand Tonnes)')
plt.xticks(rotation=90)
plt.show()
```



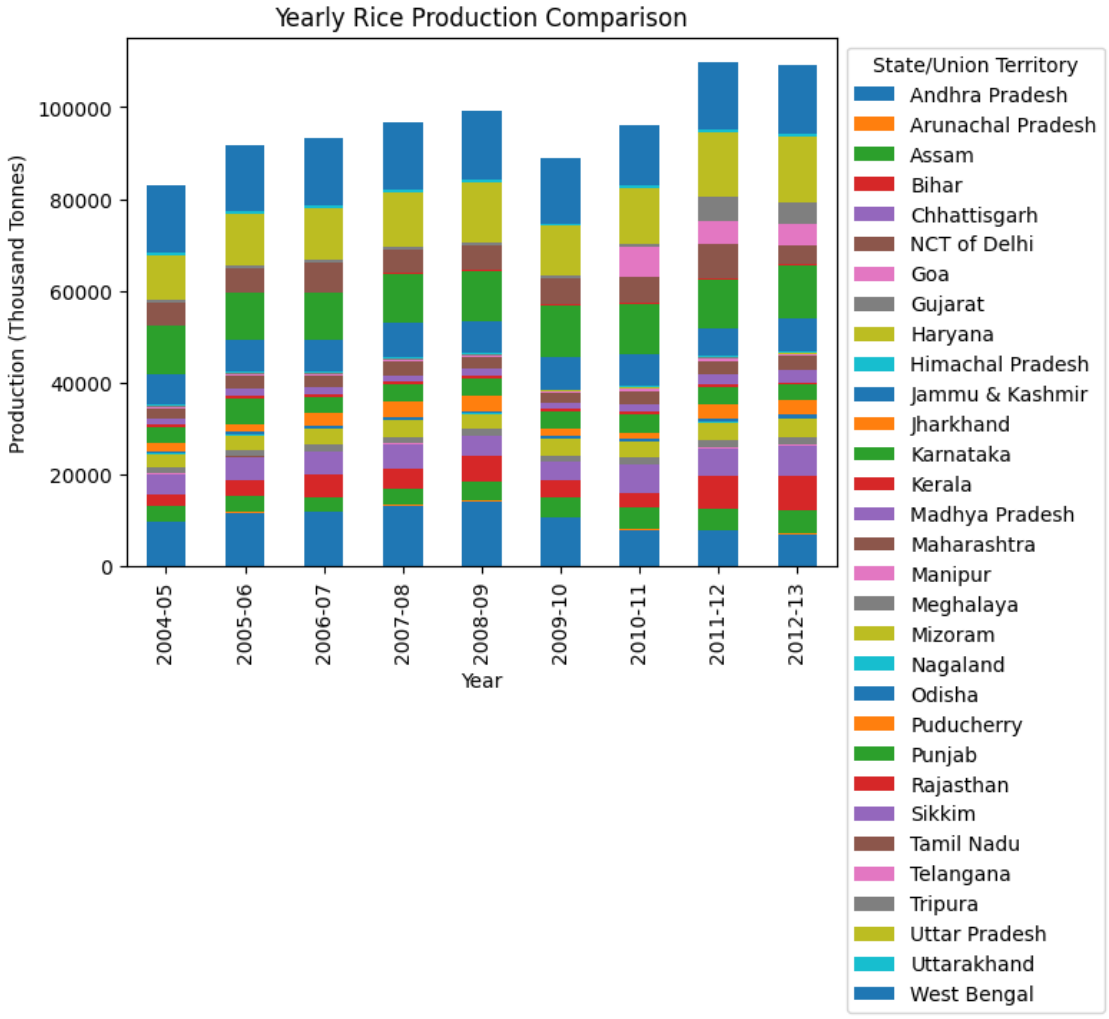
[48]:
```
#Line Plot for Trend Analysis:
plt.figure(figsize=(12, 8))
df.T.plot(kind='line')
plt.title('Trend of Rice Production Over the Years')
plt.xlabel('Year')
plt.ylabel('Production (Thousand Tonnes)')
plt.legend(title='State/Union Territory', bbox_to_anchor=(1, 1))
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```

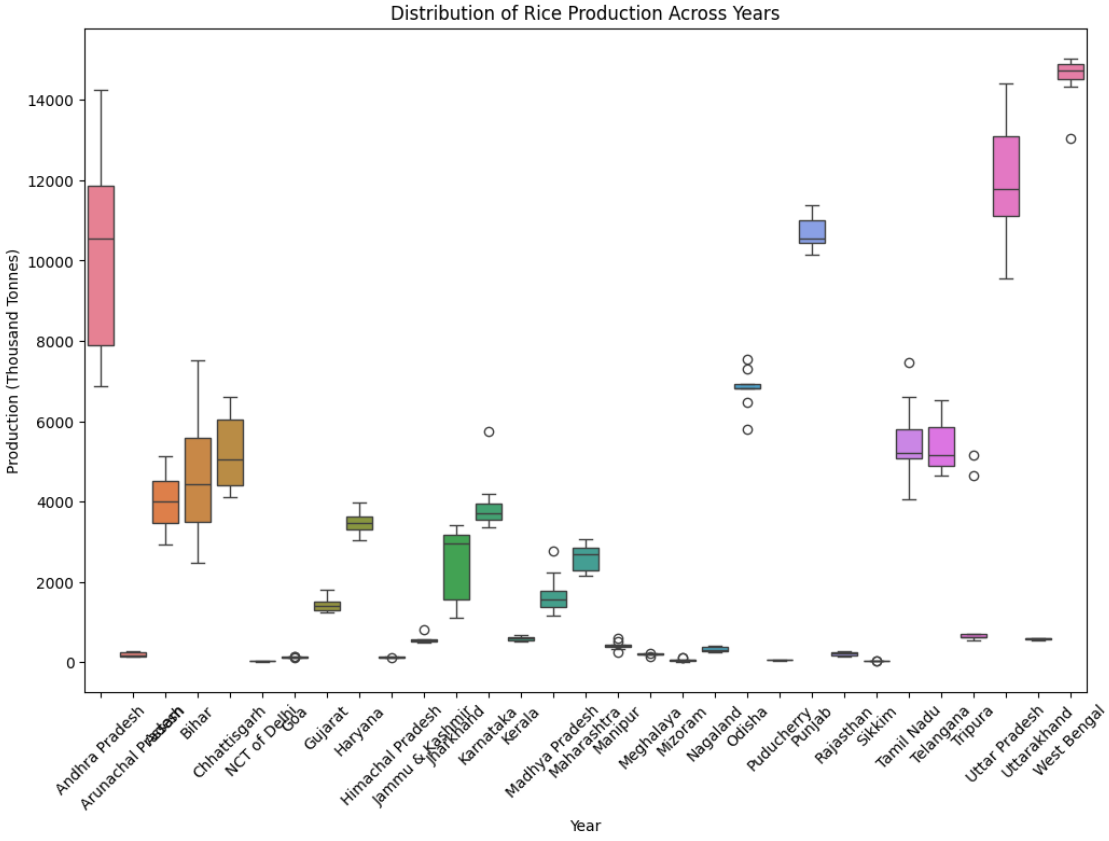


[49]:
```
#bar plot
plt.figure(figsize=(12, 8))
df.T.plot(kind='bar', stacked=True)
plt.title('Yearly Rice Production Comparison')
plt.xlabel('Year')
plt.ylabel('Production (Thousand Tonnes)')
plt.legend(title='State/Union Territory', bbox_to_anchor=(1, 1))
plt.show()
```

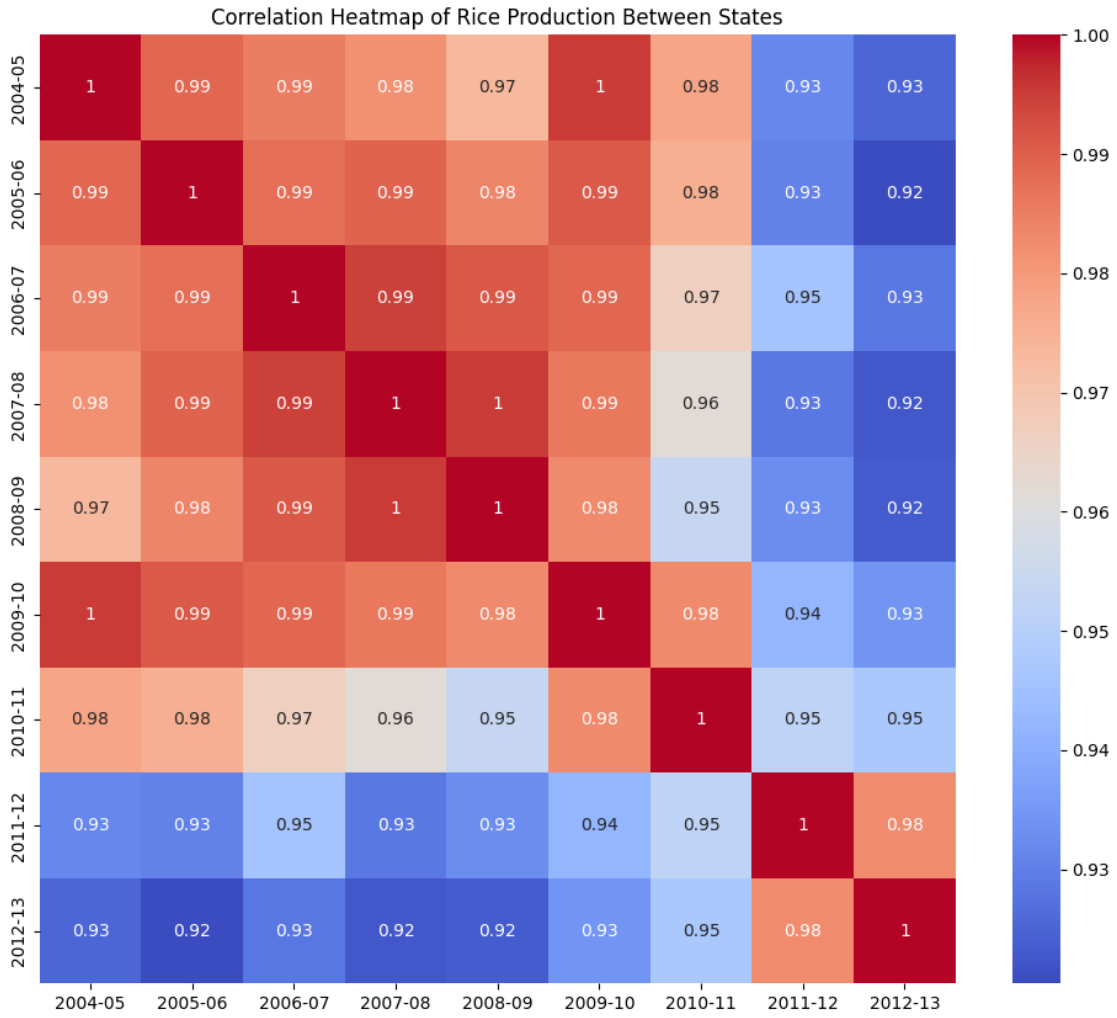
```
<Figure size 1200x800 with 0 Axes>
```



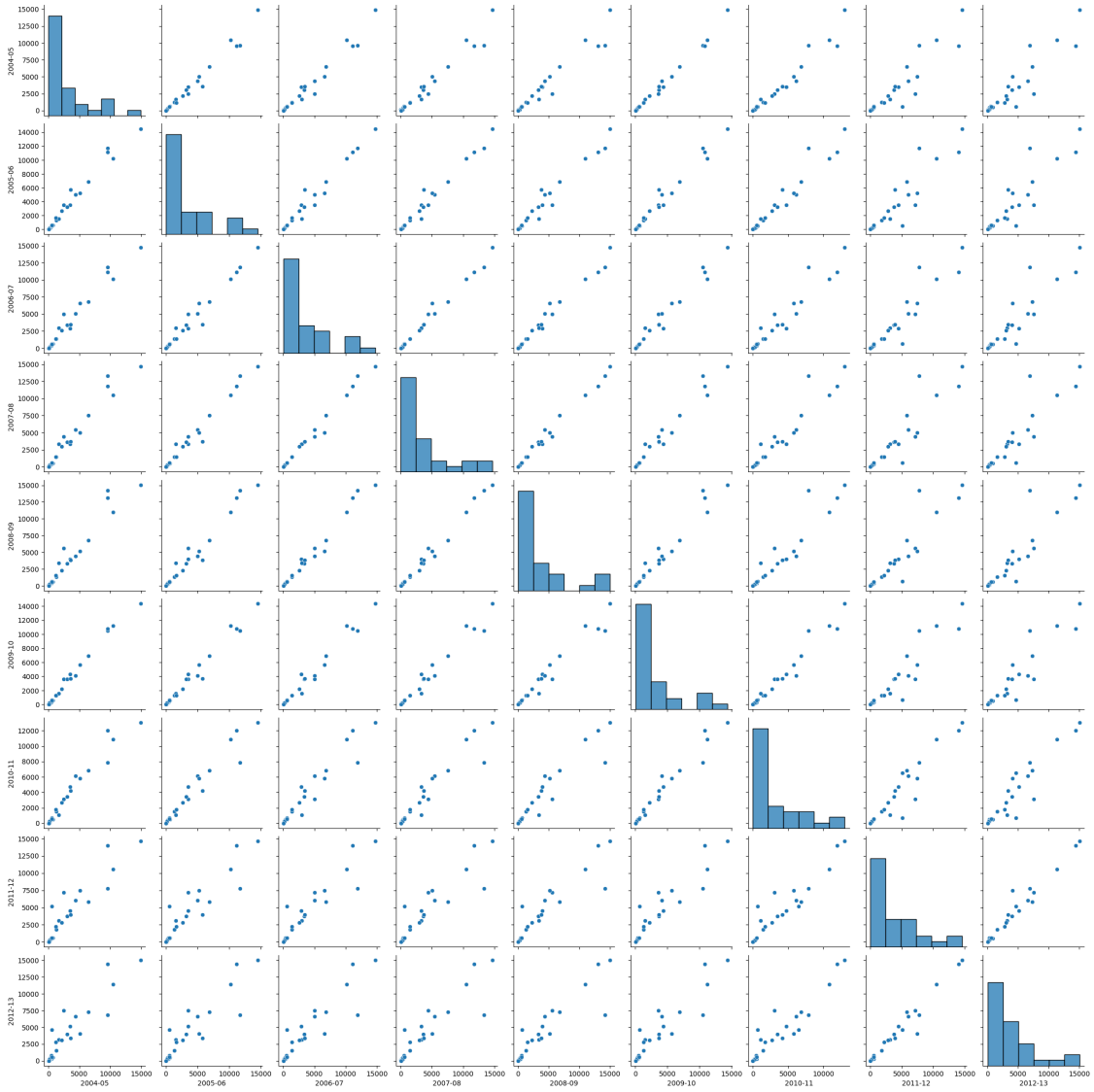
[51]:
```
#boxplot
plt.figure(figsize=(12, 8))
sns.boxplot(data=df.T)
plt.title('Distribution of Rice Production Across Years')
plt.xlabel('Year')
plt.ylabel('Production (Thousand Tonnes)')
plt.xticks(rotation=45)
plt.show()
```



[52]:
```
#heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap of Rice Production Between States')
plt.show()
```

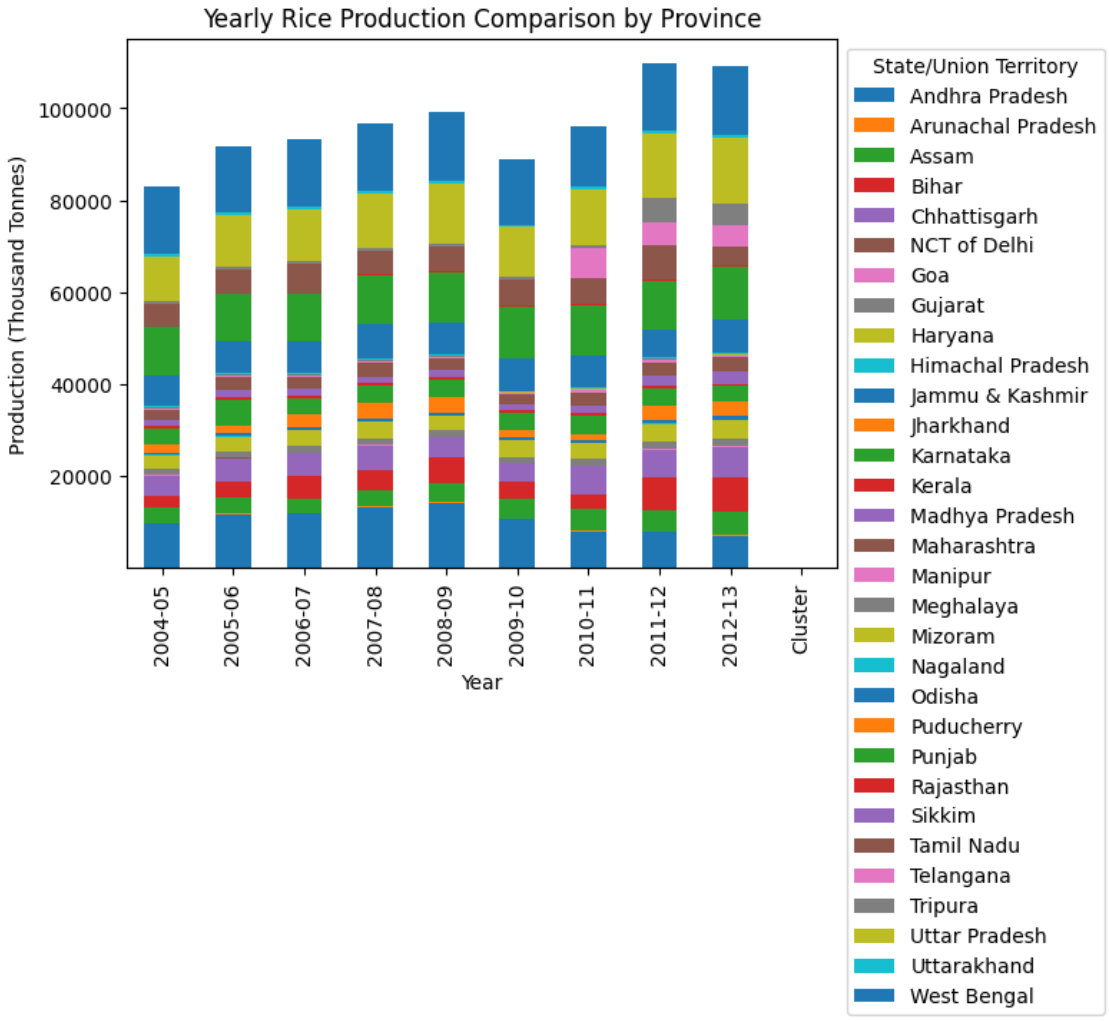


[53]:
```
import seaborn as sns

# Selecting specific columns for pairplot
columns_of_interest = ['2004-05', '2005-06', '2006-07', '2007-08', '2008-09',␣
↪'2009-10', '2010-11', '2011-12', '2012-13']
data_for_pairplot = df[columns_of_interest]

# Plotting pairplot
sns.pairplot(data_for_pairplot)
plt.show()
```



[61]:
```
import matplotlib.pyplot as plt

# Exclude 'ALL INDIA' row if present
df_no_all_india = df.drop(index='ALL INDIA', errors='ignore')

# Plotting
plt.figure(figsize=(12, 8))
df_no_all_india.T.plot(kind='bar', stacked=True)
plt.title('Yearly Rice Production Comparison by Province')
plt.xlabel('Year')
plt.ylabel('Production (Thousand Tonnes)')
plt.legend(title='State/Union Territory', bbox_to_anchor=(1, 1))
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



[64]:
```
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA

# Assuming df_no_all_india contains the prepared dataset with year-wise rice␣
 ↪production for each state/union territory

# Choose a state/union territory for prediction (e.g., 'Andhra Pradesh')
selected_state = 'Andhra Pradesh'

# Select data for the selected state
state_data = df_no_all_india.loc[selected_state]

# Define ARIMA model parameters (p, d, q)
p = 1  # Autoregression (AR) order
```

```
d = 1  # Differencing (I) order
q = 1  # Moving Average (MA) order

# Fit ARIMA model
model = ARIMA(state_data, order=(p, d, q))
model_fit = model.fit()

# Make predictions for the next 5 years
forecast = model_fit.forecast(steps=5)

# Display forecasted production
print("Forecasted Production for", selected_state, "for the Next 5 Years:")
print(forecast)
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
```

```
Forecasted Production for Andhra Pradesh for the Next 5 Years:
10   -3138.189244
11   -3115.641638
12   -3115.803589
13   -3115.802426
14   -3115.802434
Name: predicted_mean, dtype: float64
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
```

[65]:
```
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA

# Load the dataset
# Assuming df_no_all_india contains the prepared dataset with year-wise rice␣
 ↪production for each state/union territory
# Ensure the dataset is in a format where each row represents a state/union␣
 ↪territory and each column represents a year
# Convert the 'State/Union Territory' column to index
# Clean the data (handle missing values, outliers, etc.)

# Define ARIMA model parameters (p, d, q)
p = 1  # Autoregression (AR) order
d = 1  # Differencing (I) order
q = 1  # Moving Average (MA) order

# Initialize an empty DataFrame to store forecasts for all states
forecast_all_states = pd.DataFrame()

# Loop through each state/union territory
for state in df_no_all_india.index:
    # Select data for the current state
    state_data = df_no_all_india.loc[state]

    # Fit ARIMA model
    model = ARIMA(state_data, order=(p, d, q))
    model_fit = model.fit()

    # Make predictions for the next 5 years
    forecast = model_fit.forecast(steps=5)

    # Add forecasted production to the DataFrame
    forecast_all_states[state] = forecast

# Display forecasted production for all states/union territories for the next 5␣
 ↪years
print("Forecasted Production for All States/Union Territories for the Next 5␣
 ↪Years:")
print(forecast_all_states)
```

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
```

```
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
```

```
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:866: UserWarning: Too few
observations to estimate starting parameters for ARMA and trend. All parameters
except for variances will be set to zeros.
  warn('Too few observations to estimate starting parameters%s.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary
starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
```

```
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
```

```
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:473:
ValueWarning: An unsupported index was provided and will be ignored when e.g.
forecasting.
  self._init_dates(dates, freq)
```

Forecasted Production for All States/Union Territories for the Next 5 Years:

	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhattisgarh	\
10	-3138.189244	98.746318	401.790940	2893.220849	1990.403667	
11	-3115.641638	136.577512	2.047837	3533.201258	2936.254522	
12	-3115.803589	151.071210	401.743109	3674.862842	3386.180185	
13	-3115.802426	156.623963	2.095663	3706.220058	3600.202408	
14	-3115.802434	158.751306	401.695289	3713.161072	3702.009252	

	NCT of Delhi	Goa	Gujarat	Haryana	Himachal Pradesh	…	\
10	-5.013116	-4.123489	-1814.937191	570.929178	-21.987911	…	
11	-6.075089	-8.246104	-3039.396004	988.737607	-24.015305	…	
12	-6.300056	-12.367846	-3865.484858	1295.566454	-24.202241	…	
13	-6.347713	-16.488714	-4422.810909	1520.894461	-24.219477	…	
14	-6.357808	-20.608709	-4798.814441	1686.370138	-24.221066	…	

	Puducherry	Punjab	Rajasthan	Sikkim	Tamil Nadu	Telangana	\
10	19.928047	1025.520154	-63.374678	5.962385	-746.537321	652.763029	
11	4.648278	1.281070	-25.388155	4.294905	-861.279386	1110.353645	
12	16.363994	1025.239162	-48.157121	4.761243	-878.868010	1432.113122	
13	7.381004	1.561985	-34.509496	4.630824	-881.564141	1658.361558	
14	14.268684	1024.958324	-42.689827	4.667298	-881.977427	1817.450437	

	Tripura	Uttar Pradesh	Uttarakhand	West Bengal
10	-330.596975	-2159.379393	-317.098012	1118.554171
11	-179.267931	-1717.614887	-252.423936	1.005588
12	-248.538013	-1807.948964	-265.614609	1118.548583
13	-216.829994	-1789.477028	-262.924291	1.011176
14	-231.344174	-1793.254256	-263.472997	1118.542995

[5 rows x 31 columns]

```
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
ValueWarning: No supported index is available. Prediction results will be given
with an integer index beginning at `start`.
  return get_prediction_index(
/usr/local/lib/python3.10/dist-packages/statsmodels/tsa/base/tsa_model.py:836:
FutureWarning: No supported index is available. In the next version, calling
this method in a model without a supported index will result in an exception.
```

```
  return get_prediction_index(
```

[86]:
```
import matplotlib.pyplot as plt

# Assuming forecast_all_states contains the forecasted production for all␣
 ↪states/union territories for the next 5 years

# Define the years for the x-axis labels
years = ['2014-2015', '2015-2016', '2016-2017', '2017-2018', '2018-2019']

# Plotting
plt.figure(figsize=(12, 8))
forecast_all_states.T.plot(kind='bar')
plt.title('Forecasted Rice Production for the Next 5 Years by State/Union␣
 ↪Territory')
plt.xlabel('State/Union Territory')
plt.ylabel('Production')
plt.legend(title='Year', labels=years, bbox_to_anchor=(1, 1))
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```