agricultural-commodities-in-india

August 12, 2025

0.1 Price of Agricultural Commodities in India

- An EDA project

About:-

- The data refers to Daily prices of various commodities in India like Tomato, Potato, Brinjal, Wheat etc. It has the wholesale maximum price, minimum price and modal price on daily basis. the prices in the dataset refer to the wholesale prices of various commodities per quintal (100 kg) in Indian rupees. The wholesale price is the price at which goods are sold in large quantities to retailers or distributors.

Let's imports useful libraries

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

Let's load the dataset

[18]:
```
price_agr = pd.read_csv(r"C:
↪\Users\taigk\OneDrive\Documents\Desktop\new_journey_786.
↪py\Price_Agriculture_commodities_Week.csv")
```

let's start with the first few steps

[3]:
```
price_agr                    #to get the first view of the dataset
```

[3]:

	State	District	Market	Commodity	Variety	\
0	Gujarat	Amreli	Damnagar	Bhindi(Ladies Finger)	Bhindi	
1	Gujarat	Amreli	Damnagar	Brinjal	Other	
2	Gujarat	Amreli	Damnagar	Cabbage	Cabbage	
3	Gujarat	Amreli	Damnagar	Cauliflower	Cauliflower	
4	Gujarat	Amreli	Damnagar	Coriander(Leaves)	Coriander	
…	…	…	…	…	…	
23088	Uttrakhand	Haridwar	Roorkee	Mango	Other	
23089	Uttrakhand	Haridwar	Roorkee	Mousambi(Sweet Lime)	Other	
23090	Uttrakhand	Haridwar	Roorkee	Pear(Marasebu)	Other	
23091	Uttrakhand	Haridwar	Roorkee	Potato	Other	

```
23092  Uttrakhand  Haridwar   Roorkee                Pumpkin        Other

        Grade Arrival_Date  Min Price  Max Price  Modal Price
0         FAQ   27-07-2023     4100.0     4500.0       4350.0
1         FAQ   27-07-2023     2200.0     3000.0       2450.0
2         FAQ   27-07-2023     2350.0     3000.0       2700.0
3         FAQ   27-07-2023     7000.0     7500.0       7250.0
4         FAQ   27-07-2023     8400.0     9000.0       8850.0
…         …          …         …          …            …
23088  Medium   02-08-2023      800.0     1200.0       1000.0
23089  Medium   02-08-2023     1500.0     2500.0       2000.0
23090  Medium   02-08-2023     2000.0     3000.0       2500.0
23091     FAQ   02-08-2023      900.0     1800.0       1500.0
23092     FAQ   02-08-2023      500.0      700.0        600.0

[23093 rows x 10 columns]
```

[4]:
```
price_agr.shape              #to get to know about the shape of my dataset
```

[4]:
```
(23093, 10)
```

[19]:
```
price_agr.columns          #to get to know about the name of my columns name of␣
↪my dataset
```

[19]:
```
Index(['State', 'District', 'Market', 'Commodity', 'Variety', 'Grade',
       'Arrival_Date', 'Min Price', 'Max Price', 'Modal Price'],
      dtype='object')
```

[6]:
```
price_agr.head()              #to get to know about the first 5 row
```

[6]:
```
     State District    Market              Commodity      Variety Grade  \
0  Gujarat   Amreli  Damnagar  Bhindi(Ladies Finger)       Bhindi   FAQ
1  Gujarat   Amreli  Damnagar                Brinjal        Other   FAQ
2  Gujarat   Amreli  Damnagar                Cabbage      Cabbage   FAQ
3  Gujarat   Amreli  Damnagar             Cauliflower  Cauliflower   FAQ
4  Gujarat   Amreli  Damnagar      Coriander(Leaves)    Coriander   FAQ

  Arrival_Date  Min Price  Max Price  Modal Price
0   27-07-2023     4100.0     4500.0       4350.0
1   27-07-2023     2200.0     3000.0       2450.0
2   27-07-2023     2350.0     3000.0       2700.0
3   27-07-2023     7000.0     7500.0       7250.0
4   27-07-2023     8400.0     9000.0       8850.0
```

[7]:
```
price_agr.tail()              #to get to know about the last 5 row
```

[7]:

	State	District	Market	Commodity	Variety	Grade
23088	Uttrakhand	Haridwar	Roorkee	Mango	Other	Medium
23089	Uttrakhand	Haridwar	Roorkee	Mousambi(Sweet Lime)	Other	Medium
23090	Uttrakhand	Haridwar	Roorkee	Pear(Marasebu)	Other	Medium
23091	Uttrakhand	Haridwar	Roorkee	Potato	Other	FAQ
23092	Uttrakhand	Haridwar	Roorkee	Pumpkin	Other	FAQ

	Arrival_Date	Min Price	Max Price	Modal Price
23088	02-08-2023	800.0	1200.0	1000.0
23089	02-08-2023	1500.0	2500.0	2000.0
23090	02-08-2023	2000.0	3000.0	2500.0
23091	02-08-2023	900.0	1800.0	1500.0
23092	02-08-2023	500.0	700.0	600.0

[8]:
```
price_agr.info()              #to get to know about my data type of my dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23093 entries, 0 to 23092
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   State         23093 non-null  object
 1   District      23093 non-null  object
 2   Market        23093 non-null  object
 3   Commodity     23093 non-null  object
 4   Variety       23093 non-null  object
 5   Grade         23093 non-null  object
 6   Arrival_Date  23093 non-null  object
 7   Min Price     23093 non-null  float64
 8   Max Price     23093 non-null  float64
 9   Modal Price   23093 non-null  float64
dtypes: float64(3), object(7)
memory usage: 1.8+ MB
```

[9]:
```
price_agr.describe()          #to get done with the all kinda arithmetic␣
↪function
```

[9]:

	Min Price	Max Price	Modal Price
count	23093.000000	23093.000000	23093.000000
mean	4187.077045	4976.034260	4602.917742
std	5472.783385	6277.308057	5843.822711
min	0.000000	0.000000	0.830000
25%	1750.000000	2000.000000	1955.000000
50%	2725.000000	3400.000000	3000.000000
75%	5000.000000	6000.000000	5500.000000
max	223500.000000	227500.000000	225500.000000

Let's dealing with the values of dataset

[20]:
```
price_agr.columns.str.replace(' ' ,'_')
```

[20]:
```
Index(['State', 'District', 'Market', 'Commodity', 'Variety', 'Grade',
       'Arrival_Date', 'Min_Price', 'Max_Price', 'Modal_Price'],
      dtype='object')
```

- Insights:- To replace empty space with ***'_'*** from columns name so it'll be ease for working and analysing

[15]:
```
price_agr.isnull().sum()
```

[15]:
```
State           0
District        0
Market          0
Commodity       0
Variety         0
Grade           0
Arrival_Date    0
Min Price       0
Max Price       0
Modal Price     0
dtype: int64
```

- Insights:- No it's neet and clean while having no any null values

[16]:
```
price_agr.drop_duplicates(inplace=True)
```

- Inshights:- It'll drop all duplicates values if there will have so

Bivariate Analysis

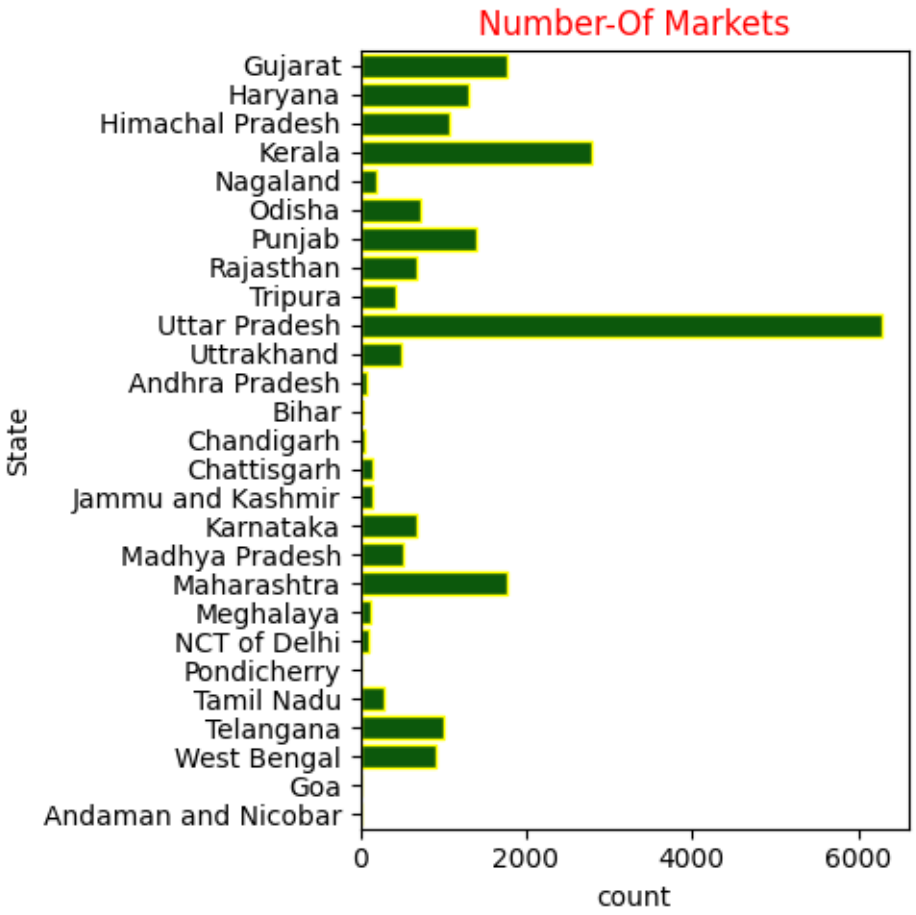
[22]:
```
price_agr.groupby('State')['District'].count().reset_index()          #grouping␣
↪with the 'State'
```

[22]:

	State	District
0	Andaman and Nicobar	7
1	Andhra Pradesh	79
2	Bihar	42
3	Chandigarh	59
4	Chattisgarh	158
5	Goa	9
6	Gujarat	1782
7	Haryana	1300
8	Himachal Pradesh	1087
9	Jammu and Kashmir	142
10	Karnataka	674
11	Kerala	2795

```
12       Madhya Pradesh       519
13          Maharashtra      1770
14            Meghalaya       136
15          NCT of Delhi        96
16             Nagaland       197
17               Odisha       726
18           Pondicherry        10
19                Punjab      1406
20            Rajasthan       674
21           Tamil Nadu       295
22            Telangana      1004
23              Tripura       418
24        Uttar Pradesh      6298
25           Uttrakhand       490
26          West Bengal       920
```

[6]:
```
#This graph will stand for 'Number of Markets'
plt.figure(figsize=(5,5))
sns.countplot(data=price_agr , y='State',color='darkgreen',edgecolor='yellow')
plt.title('Number-Of Markets',c='r')
plt.tight_layout()
plt.show()
```



- Insights :- It shows us the Uttar Pradesh is the state with the most market 6298

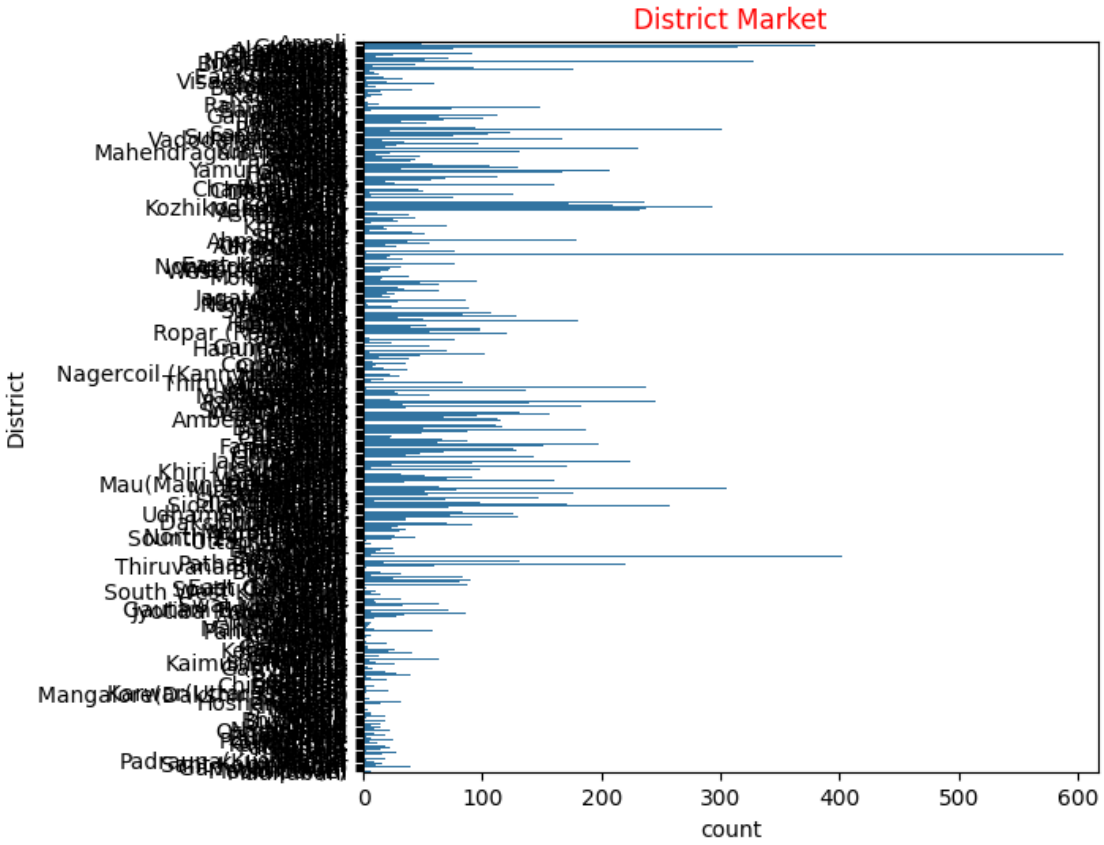
[4]:
```
price_agr.groupby('District')['State'].count().reset_index()
 ↪#grouping with the 'District'
```

[4]:

	District	State
0	Adilabad	26
1	Agra	131
2	Ahmedabad	28
3	Ahmednagar	179
4	Ajmer	22
..	…	…
398	Wokha	14
399	Yamuna Nagar	130
400	Yavatmal	18
401	Zunheboto	3
402	kapurthala	5

[403 rows x 2 columns]

[5]:
```
#this graph will stand for 'Number of district'
plt.figure(figsize=(6,6))
sns.countplot(data=price_agr, y='District')
plt.title('District Market',c='r')
plt.show()
```



- Insights :- It show us we've approximately 403 districts markets

[30]:
```
price_agr.groupby('Market')['Commodity'].count().reset_index()  #grouping with␣
↪the 'Market'
```

[30]:

	Market	Commodity
0	A lot	2
1	Abhanpur	2
2	Abohar	19
3	Achalda	59
4	Achalpur	1
…	…	…

```
1284   Yellapur          8
1285    Yeotmal          5
1286       Zira         19
1287  Zunheboto          3
1288  kalanwali         12

[1289 rows x 2 columns]
```

- Insights :- It shows us we've 1289 local markets in all over india

[31]:
```
price_agr.groupby('Commodity')['State'].count().reset_index()          #grouping␣
 ↪with the 'Commodity'
```

[31]:
```
          Commodity  State
0             Ajwan      7
1    Alasande Gram      1
2    Almond(Badam)      1
3       Alsandikai      6
4       Amaranthus     84
..             …    …
229     White Peas     51
230  White Pumpkin      4
231           Wood     21
232            Yam      4
233   Yam (Ratalu)     22

[234 rows x 2 columns]
```

- Insights:- It shows us in this dataset we've 234 commodity in entire data

[7]:
```
price_agr.groupby('Variety')['State'].count().reset_index()          #grouping␣
 ↪with the 'Variety'
```

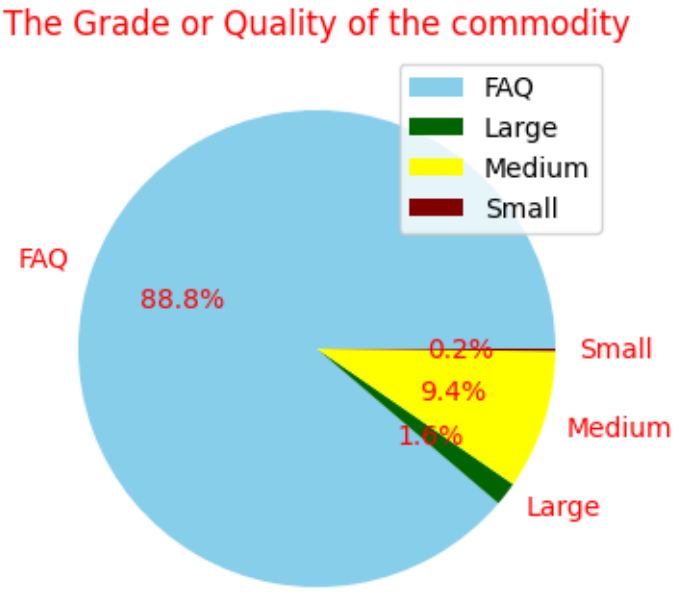
[7]:
```
             Variety  State
0     (Red Nanital)     49
1              1001     49
2          1009 Kar     10
3              1121      4
4       147 Average     18
..              …    …
447  Zinga(Zambo-A)      2
448  Zinga(Zambo-B)      2
449  Zinga(Zambo-C)      2
450             api      5
451           other     24

[452 rows x 2 columns]
```

- Insights:- In this data set we have 452 kind of variety

[29]:
```
price_agr_grade =price_agr.groupby('Grade')['State'].count().reset_index()
 ↪ #Grouping with the 'Grade'
```

[38]:
```
#This circle will stand for 'The Grade'
plt.figure(figsize=(4,5))
values= price_agr_grade['State'].values
labels=price_agr_grade['Grade'].values
plt.pie(values, labels=labels , autopct='%1.
 ↪1f%%',colors=['skyblue','darkgreen','yellow','maroon'], textprops={'color':
 ↪'red'})
plt.title('The Grade or Quality of the commodity',c='r')
plt.legend()
plt.show()
```



- Insights:- This graph shows us the FAQ is the brand of all commodities

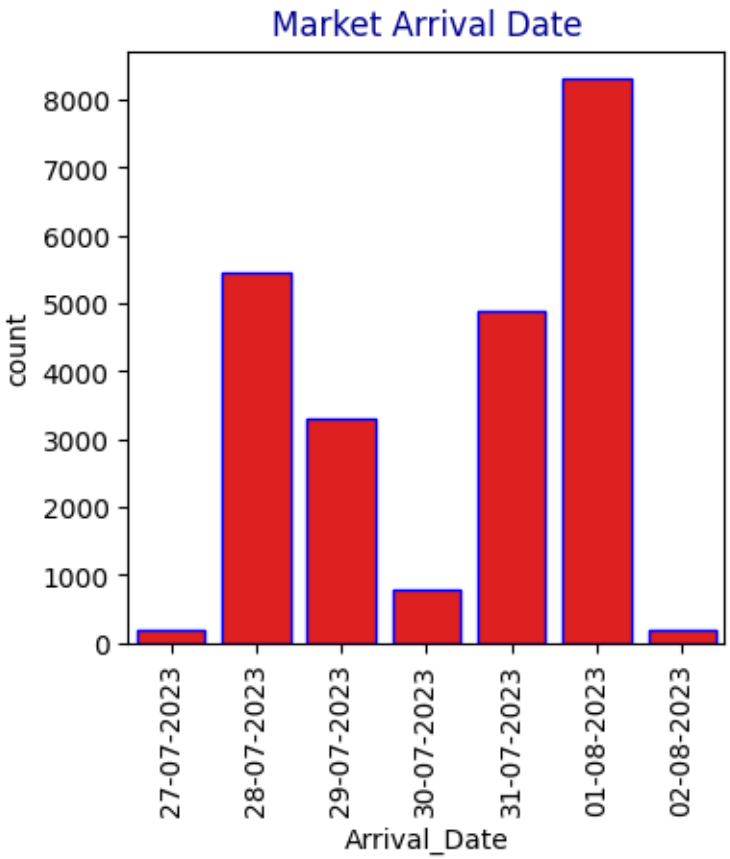
[10]:
```
price_agr.groupby('Arrival_Date')['State'].count().reset_index()
 ↪#grouping with the 'Arrival_Date'
```

[10]:

	Arrival_Date	State
0	01-08-2023	8304
1	02-08-2023	189
2	27-07-2023	181
3	28-07-2023	5454

```
4   29-07-2023   3312
5   30-07-2023    774
6   31-07-2023   4879
```

[43]:
```
#this graph will stand for 'Market Arrival Date ' of all Goods and Products'
plt.figure(figsize=(4,4))
sns.countplot(data=price_agr, x='Arrival_Date', color='red',edgecolor='blue')
plt.title('Market Arrival Date', c='darkblue')
plt.xticks(rotation=90)
plt.show()
```



- Insights :- When it comes to the date so 1-aug-2023 was the perfect day for market

[24]:
```
price_agr.groupby('Min Price')['State'].count().reset_index() #grouping with␣
↪the 'Max_Price'
```

[24]:

	Min Price	State
0	0.00	20
1	0.41	2
2	0.62	1

```
3          1.00      3
4          1.50      2
…           …    …
1788  100000.00      1
1789  110000.00      2
1790  136000.00      1
1791  140000.00      1
1792  223500.00      1

[1793 rows x 2 columns]
```

- Insights:- In this datasetwe have 1793 unique Min_price

[23]:
```
price_agr.groupby('Max Price')['State'].count().reset_index() #grouping with␣
↪the 'Max_Price'
```

[23]:
```
      Max Price  State
0          0.00     28
1          1.00      1
2          1.25      1
3          1.45      2
4          2.50      1
…           …    …
2068  115000.00      1
2069  130000.00      1
2070  140000.00      1
2071  150000.00      1
2072  227500.00      1

[2073 rows x 2 columns]
```

- Insights:- In this datasetwe have 2073 unique Max_price'

[27]:
```
price_agr.groupby('Modal Price')['State'].count().reset_index()
```

[27]:
```
      Modal Price  State
0            0.83      1
1            1.00      1
2            1.04      2
3            2.00      1
4            2.20      1
…             …    …
2066    112500.00      1
2067    120000.00      1
2068    138000.00      1
2069    150000.00      1
2070    225500.00      1
```

```
[2071 rows x 2 columns]
```

- Insights:- In this dataset we have 2071 unique 'Modal_price'

Conclusion – Price of Agricultural Commodities in India (EDA Project) This exploratory data analysis provided valuable insights into the wholesale price trends of agricultural commodities across India. The dataset, comprising daily maximum, minimum, and modal prices, revealed several important patterns:

- Market Distribution: Uttar Pradesh emerged as the state with the highest number of markets (6,298), with over 400 districts contributing to the agricultural trade network.
- Commodity Presence: Certain commodities, such as tomato, potato, and brinjal, dominated the market in terms of frequency and distribution across multiple states.
- Price Variations: Significant fluctuations were observed in the maximum and minimum prices of key commodities, highlighting regional pricing differences and possible seasonal influences.
- Outlier Detection: Boxplot analysis revealed price outliers, suggesting instances of unusually high or low wholesale rates that may be linked to supply-demand shifts or market disruptions.

This analysis demonstrates the power of data cleaning, grouping, and visualisation in uncovering patterns within large-scale agricultural datasets. By leveraging Python libraries such as Pandas, Matplotlib, and Seaborn, it was possible to transform raw price data into actionable insights that can inform decision-making for traders, policymakers, and market analysts.

-

[ ]: