



RABIN ANTO

PRESENT

COMMODITY PRICE ANALYSING

DATASCIENCE STUDENT

@ GUVI GEEK
NETWORK

IMPORTING THE LIBRARY

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
import plotly.express as px
from plotly.subplots import make_subplots
import warnings

palette = sns.color_palette("rainbow", 8)
warnings.filterwarnings("ignore")
```

```
In [2]: df=pd.read_csv("D:\DATASET\csv\commodity_price.csv")
```

```
In [3]: df
```

```
Out[3]:
```

	s_no	state	district	market	pincode	country	commodity	variety	arrival_date	min_price	max_price	modal_price	update
0	0	Karnataka	Belgaum	Athani	100001	India	Bull	Bull	26-01-2020	20000.0	40000.0	40000.0	
1	1	Karnataka	Belgaum	Athani	100002	India	Bull	Bull	16-02-2020	10000.0	12000.0	11000.0	
2	2	Karnataka	Belgaum	Athani	100003	India	Bull	Bull	01-03-2020	32000.0	40000.0	35000.0	
3	3	Karnataka	Belgaum	Ramdurga	100004	India	Bull	Bull	05-01-2020	20000.0	25000.0	20000.0	
4	4	Karnataka	Belgaum	Ramdurga	100005	India	BULL	Bull	12-01-2020	20000.0	35000.0	25000.0	
...
78646	78598	West Bengal	Murshidabad	Jangipur	178599	India	Turmeric	Finger	26-12-2022	9440.0	9550.0	9500.0	
78647	78599	West Bengal	Murshidabad	Jangipur	178600	India	Turmeric	Finger	27-12-2022	9435.0	9550.0	9490.0	
78648	78600	West Bengal	Murshidabad	Jangipur	178601	India	Turmeric	Finger	28-12-2022	9410.0	9530.0	9475.0	
78649	78601	West Bengal	Murshidabad	Jangipur	178602	India	Turmeric	Finger	29-12-2022	9425.0	9550.0	9490.0	
78650	78602	West Bengal	Murshidabad	Jangipur	178603	India	Turmeric	Finger	30-12-2022	9425.0	9550.0	9490.0	

78651 rows × 13 columns

CHECKING FIRST FIVE AND LAST FIVE VALUES IN DATAFRAME

```
In [4]: df.head(5)
```

```
Out[4]:
```

	s_no	state	district	market	pincode	country	commodity	variety	arrival_date	min_price	max_price	modal_price	update_date
0	0	Karnataka	Belgaum	Athani	100001	India	Bull	Bull	26-01-2020	20000.0	40000.0	40000.0	NaN
1	1	Karnataka	Belgaum	Athani	100002	India	Bull	Bull	16-02-2020	10000.0	12000.0	11000.0	NaN
2	2	Karnataka	Belgaum	Athani	100003	India	Bull	Bull	01-03-2020	32000.0	40000.0	35000.0	NaN
3	3	Karnataka	Belgaum	Ramdurga	100004	India	Bull	Bull	05-01-2020	20000.0	25000.0	20000.0	NaN
4	4	Karnataka	Belgaum	Ramdurga	100005	India	BULL	Bull	12-01-2020	20000.0	35000.0	25000.0	NaN

```
In [5]: df.tail()
```

```
Out[5]:
```

	s_no	state	district	market	pincode	country	commodity	variety	arrival_date	min_price	max_price	modal_price	update_d
78646	78598	West Bengal	Murshidabad	Jangipur	178599	India	Turmeric	Finger	26-12-2022	9440.0	9550.0	9500.0	N
78647	78599	West Bengal	Murshidabad	Jangipur	178600	India	Turmeric	Finger	27-12-2022	9435.0	9550.0	9490.0	N
78648	78600	West Bengal	Murshidabad	Jangipur	178601	India	Turmeric	Finger	28-12-2022	9410.0	9530.0	9475.0	N
78649	78601	West Bengal	Murshidabad	Jangipur	178602	India	Turmeric	Finger	29-12-2022	9425.0	9550.0	9490.0	N
78650	78602	West Bengal	Murshidabad	Jangipur	178603	India	Turmeric	Finger	30-12-2022	9425.0	9550.0	9490.0	N

TO CHECK THE SHAPE OF DATAFRAME

In [6]: df.shape

Out[6]: (78651, 13)

In [7]: print("The number of rows",df.shape[0])
print("The number of columns",df.shape[1])

The number of rows 78651
The number of columns 13

TO KNOW INFO ABOUT DATA

In [8]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 78651 entries, 0 to 78650
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   s_no        78651 non-null   int64  
 1   state       78651 non-null   object  
 2   district    78651 non-null   object  
 3   market      78651 non-null   object  
 4   pincode     78651 non-null   int64  
 5   country     78651 non-null   object  
 6   commodity   78651 non-null   object  
 7   variety     78651 non-null   object  
 8   arrival_date 78651 non-null   object  
 9   min_price   78483 non-null   float64 
 10  max_price   78467 non-null   float64 
 11  modal_price 78651 non-null   float64 
 12  update_date 928 non-null    object  
dtypes: float64(3), int64(2), object(8)
memory usage: 7.8+ MB
```

HANDLING WITH MISSING DATA

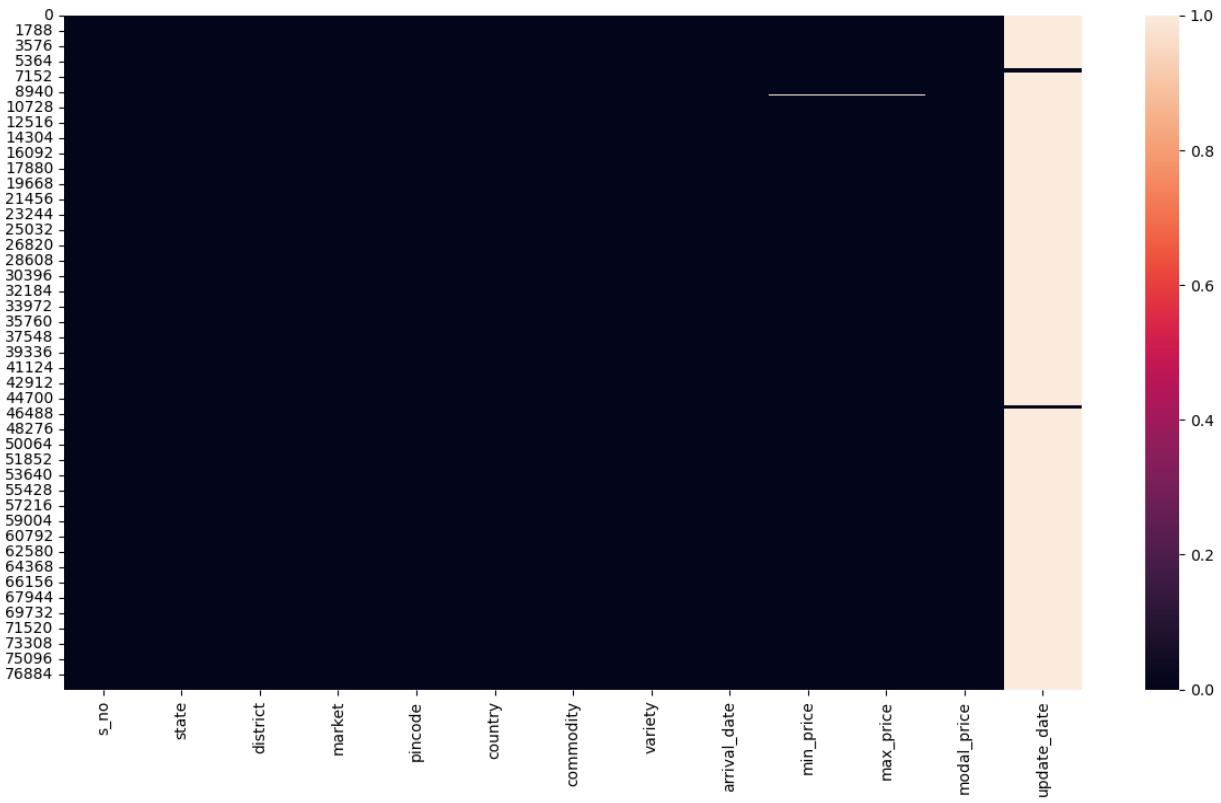
In [9]: df.isna()

Out[9]:

	s_no	state	district	market	pincode	country	commodity	variety	arrival_date	min_price	max_price	modal_price	update_date
0	False	False	False	False	False	False	False	False	False	False	False	False	True
1	False	False	False	False	False	False	False	False	False	False	False	False	True
2	False	False	False	False	False	False	False	False	False	False	False	False	True
3	False	False	False	False	False	False	False	False	False	False	False	False	True
4	False	False	False	False	False	False	False	False	False	False	False	False	True
...
78646	False	False	False	False	False	False	False	False	False	False	False	False	True
78647	False	False	False	False	False	False	False	False	False	False	False	False	True
78648	False	False	False	False	False	False	False	False	False	False	False	False	True
78649	False	False	False	False	False	False	False	False	False	False	False	False	True
78650	False	False	False	False	False	False	False	False	False	False	False	False	True

78651 rows × 13 columns

```
In [10]: plt.figure(figsize=(15, 8))
sns.heatmap(df.isna())
plt.savefig("heatmap.png", dpi=300)
plt.show()
```



```
In [11]: df.isna().sum()
```

```
Out[11]: s_no      0
state      0
district    0
market      0
pincode     0
country     0
commodity   0
variety     0
arrival_date 0
min_price   168
max_price   184
modal_price  0
update_date  77723
dtype: int64
```

```
In [12]: df[df.max_price.isna()]
```

```
Out[12]:
```

s_no	state	district	market	pincode	country	commodity	variety	arrival_date	min_price	max_price	modal_price	update_date
9256	9256	Haryana	Jhajjar	Bahadurgarh	109257	India	Grapes	Other	04-01-2022	NaN	NaN	6000.0
9257	9257	Haryana	Jhajjar	Bahadurgarh	109258	India	Grapes	Other	07-01-2022	NaN	NaN	8000.0
9258	9258	Haryana	Jhajjar	Bahadurgarh	109259	India	Grapes	Other	11-01-2022	NaN	NaN	7000.0
9259	9259	Haryana	Jhajjar	Bahadurgarh	109260	India	Grapes	Other	12-01-2022	NaN	NaN	7000.0
9260	9260	Haryana	Jhajjar	Bahadurgarh	109261	India	Grapes	Other	14-01-2022	NaN	NaN	9000.0
...
58857	58857	Punjab	Fatehgarh		158858	India	Grapes	Other	10-01-2022	NaN	NaN	6000.0
58858	58858	Punjab	Fatehgarh	Sirhind	158859	India	Grapes	Other	11-01-2022	NaN	NaN	6000.0
58860	58860	Punjab	Fatehgarh	Sirhind	158861	India	Grapes	Other	17-01-2022	NaN	NaN	6000.0
58863	58863	Punjab	Fatehgarh	Sirhind	158864	India	Grapes	Other	24-01-2022	NaN	NaN	6000.0

```
In [13]: df.dropna(how='all', axis=0, inplace=True)
```

```
In [14]: df.max_price.fillna(df.modal_price,inplace=True)
```

```
In [15]: #checking missing value
df.max_price.isna().sum()
```

Out[15]: 0

```
In [16]: df.min_price.fillna(df.modal_price,inplace=True)
```

```
In [17]: df.isna().sum()
```

```
Out[17]: s_no          0
state          0
district       0
market         0
pincode        0
country        0
commodity      0
variety        0
arrival_date   0
min_price      0
max_price      0
modal_price    0
update_date    77723
dtype: int64
```

```
In [18]: df.tail()
```

Out[18]:

	s_no	state	district	market	pincode	country	commodity	variety	arrival_date	min_price	max_price	modal_price	update_d
78646	78598	West Bengal	Murshidabad	Jangipur	178599	India	Turmeric	Finger	26-12-2022	9440.0	9550.0	9500.0	N
78647	78599	West Bengal	Murshidabad	Jangipur	178600	India	Turmeric	Finger	27-12-2022	9435.0	9550.0	9490.0	N
78648	78600	West Bengal	Murshidabad	Jangipur	178601	India	Turmeric	Finger	28-12-2022	9410.0	9530.0	9475.0	N
78649	78601	West Bengal	Murshidabad	Jangipur	178602	India	Turmeric	Finger	29-12-2022	9425.0	9550.0	9490.0	N
78650	78602	West Bengal	Murshidabad	Jangipur	178603	India	Turmeric	Finger	30-12-2022	9425.0	9550.0	9490.0	N

◀ ▶

```
In [19]: df.drop(['update_date'],axis=1,inplace=True)
```

```
In [20]: import pandas as pd
```

```
# Convert 'arrival_date' column to datetime format
df['arrival_date'] = pd.to_datetime(df['arrival_date'])

# Create 'arrival_day' column using the .dt accessor
df['arrival_day'] = df['arrival_date'].dt.day_name()
```

In [21]: df

Out[21]:

	s_no	state	district	market	pincode	country	commodity	variety	arrival_date	min_price	max_price	modal_price	arriv:
0	0	Karnataka	Belgaum	Athani	100001	India	Bull	Bull	2020-01-26	20000.0	40000.0	40000.0	S
1	1	Karnataka	Belgaum	Athani	100002	India	Bull	Bull	2020-02-16	10000.0	12000.0	11000.0	S
2	2	Karnataka	Belgaum	Athani	100003	India	Bull	Bull	2020-01-03	32000.0	40000.0	35000.0	
3	3	Karnataka	Belgaum	Ramdurga	100004	India	Bull	Bull	2020-05-01	20000.0	25000.0	20000.0	
4	4	Karnataka	Belgaum	Ramdurga	100005	India	BULL	Bull	2020-12-01	20000.0	35000.0	25000.0	Tu
...
78646	78598	West Bengal	Murshidabad	Jangipur	178599	India	Turmeric	Finger	2022-12-26	9440.0	9550.0	9500.0	N
78647	78599	West Bengal	Murshidabad	Jangipur	178600	India	Turmeric	Finger	2022-12-27	9435.0	9550.0	9490.0	Tu
78648	78600	West Bengal	Murshidabad	Jangipur	178601	India	Turmeric	Finger	2022-12-28	9410.0	9530.0	9475.0	Wedr
78649	78601	West Bengal	Murshidabad	Jangipur	178602	India	Turmeric	Finger	2022-12-29	9425.0	9550.0	9490.0	Th
78650	78602	West Bengal	Murshidabad	Jangipur	178603	India	Turmeric	Finger	2022-12-30	9425.0	9550.0	9490.0	

78651 rows × 13 columns



In [22]: df.state.unique()

Out[22]: 30

In [23]: df.market.unique()

Out[23]: 459

In [24]: df.district.unique()

Out[24]: 247

In [25]: df.state.unique()

```
Out[25]: array(['Karnataka', 'Karnatak', 'Karnatka', 'Tripura', 'Goa', 'Kerala',
       'kerala', 'Karala', 'Tamil Nadu', 'Andhra Pradesh', 'Chattisgarh',
       'Gujarat', 'Madhya Pradesh', 'Manipur', 'Odisha', 'Rajasthan',
       'Telangana', 'Uttar Pradesh', 'Bihar', 'Chandigarh', 'Haryana',
       'Himachal Pradesh', 'Jammu and Kashmir', 'Maharashtra',
       'NCT of Delhi', 'Punjab', 'Uttrakhand', 'Meghalaya', 'Nagaland',
       'West Bengal'], dtype=object)
```

In [26]: #TO REMOVE THE DUPLICATES

In [27]: df.state=df.state.str.lower()

In [28]: df.state.unique()

```
Out[28]: array(['karnataka', 'karnatak', 'karnatka', 'tripura', 'goa', 'kerala',
       'karala', 'tamil nadu', 'andhra pradesh', 'chattisgarh', 'gujarat',
       'madhya pradesh', 'manipur', 'odisha', 'rajasthan', 'telangana',
       'uttar pradesh', 'bihar', 'chandigarh', 'haryana',
       'himachal pradesh', 'jammu and kashmir', 'maharashtra',
       'nct of delhi', 'punjab', 'uttrakhand', 'meghalaya', 'nagaland',
       'west bengal'], dtype=object)
```

In [29]: df.state.replace({'karnatak':'Karnataka','karnatka':'Karnataka','karala':'Kerala'},inplace=True)

In [30]: df.state.unique()

```
Out[30]: array(['karnataka', 'tripura', 'goa', 'kerala', 'tamil nadu',
       'andhra pradesh', 'chattisgarh', 'gujarat', 'madhya pradesh',
       'manipur', 'odisha', 'rajasthan', 'telangana', 'uttar pradesh',
       'bihar', 'chandigarh', 'haryana', 'himachal pradesh',
       'jammu and kashmir', 'maharashtra', 'nct of delhi', 'punjab',
       'uttrakhand', 'meghalaya', 'nagaland', 'west bengal'], dtype=object)
```

In [31]: df.district.unique()

```
Out[31]: array(['Belgaum', 'Davangere', 'Dharwad', 'Hassan', 'Haveri', 'Havri',
   'Khowai', 'Nrth Goa', 'North Goa', 'North Goaa', 'Nort Goa',
   'Bangalore', 'Bengalore', 'Bangalor', 'Chikmagalur', 'Haassan',
   'Hassann', 'Mandy', 'Maandya', 'Mandyah',
   'Mangalore(Dakshin Kannad)', 'Tumkur', 'Tumkoor', 'Idukky',
   'Idukki', 'Iddukki', 'Kannur', 'Kasargod', 'Ariyalur',
   'Coimbatore', 'Cuddalore', 'Dindigul', 'Erode', 'Namakkal',
   'Salem', 'Vellore', 'Guntur', 'Kurnool', 'Bastar', 'Dahod',
   'Rajkot', 'Bellary', 'Gadag', 'Kolar', 'Mysore', 'Raichur',
   'Udupi', 'Alappuzha', 'Malappuram', 'Dhar', 'Bishnupur', 'Chandel',
   'Imphal East', 'Imphal West', 'Thoubal', 'Ganjam', 'Jodhpur',
   'Adilabad', 'Hyderabad', 'Khammam', 'Warangal', 'Sultanpur',
   'Dhalai', 'Ernakulam', 'Angul', 'Mayurbhanja', 'Nowrangpur',
   'Chitrakut', 'Jhansi', 'Araria', 'Gopalgang', 'Kishanganj',
   'Patna', 'Rohtas', 'Siwan', 'Supaul', 'West Chambaran',
   'Chandigarh', 'Bilaspur', 'Durg', 'Rajnandgaon', 'Banaskanth',
   'Porbandar', 'Ambala', 'Faridabad', 'Fatehabad', 'Gurgaon',
   'Hissar', 'Jhajjar', 'Karnal', 'Kurukshestra',
   'Mahendragarh-Narnaul', 'Mewat', 'Palwal', 'Panchkula', 'Panipat',
   'Rohtak', 'Sirsia', 'Sonipat', 'Yamuna Nagar', 'Chamba', 'Hamirpur',
   'Kangra', 'Kullu', 'Mandi', 'Shimla', 'Sirmore', 'Solan', 'Una',
   'Anantnag', 'Jammu', 'Kathua', 'Rajouri', 'Srinagar', 'Udhampur',
   'Gulbarga', 'Kozhikode(Calicut)', 'Thiruvananthapuram', 'Bhopal',
   'Gwalior', 'Harda', 'Hoshangabad', 'Indore', 'Morena', 'Satna',
   'Sheopur', 'Ujjain', 'Ahmednagar', 'Amarawati', 'Aurangabad',
   'Chandrapur', 'Jalgaon', 'Kolhapur', 'Mumbai', 'Nagpur', 'Nashik',
   'Osmanabad', 'Pune', 'Sangli', 'Sholapur', 'Thane', 'Delhi',
   'Nuapada', 'Amritsar', 'Barnala', 'Bhatinda', 'Faridkot',
   'Fatehgarh', 'Fazilka', 'Ferozpur', 'Gurdaspur', 'Hoshiarpur',
   'Jalandhar', 'kapurthala', 'Ludhiana', 'Mansa', 'Moga', 'Muktsar',
   'Nawanshahr', 'Pathankot', 'Patiala', 'Ropar (Rupnagar)',
   'Sangrur', 'Tarntaran', 'Ajmer', 'Alwar', 'Bikaner', 'Chittorgarh',
   'Ganganagar', 'Jalore', 'Udaipur', 'Agra', 'Ambedkar Nagar',
   'Bharaich', 'Ballia', 'Barabanki', 'Basti',
   'Bhadohi(Sant Ravi Nagar)', 'Bulandshahar', 'Deoria', 'Etah',
   'Faizabad', 'Farukhabad', 'Firozabad', 'Gautam Budh Nagar',
   'Ghaziabad', 'Gorakhpur', 'Hathras', 'Jalaun (Orai)', 'Jaunpur',
   'Kanpur', 'Khiri (Lakhimpur)', 'Lucknow', 'Mainpuri', 'Mathura',
   'Mau(Maunathbhawanjan)', 'Meerut', 'Mirzapur', 'Muzaffarnagar',
   'Padrauna(Kusinagar)', 'Pratapgarh', 'Raebarelli', 'Rampur',
   'Saharanpur', 'Shahjahanpur', 'Siddharth Nagar', 'Sitapur',
   'Unnao', 'Varanasi', 'Dehradoon', 'Garhwal (Pauri)', 'Haridwar',
   'Nanital', 'UdhamSinghNagar', 'Palakad', 'Kottayam',
   'Pathanamthitta', 'Wayanad', 'Nagercoil (Kanniyakumari)',
   'Surajpur', 'Surguja', 'Balrampur', 'Kabirdham', 'Cuddapah',
   'Janjgir', 'Jashpur', 'Koria', 'Chamrajnagar', 'Anupur',
   'Chhindwara', 'Hingoli', 'Nanded', 'Parbhani', 'Vashim',
   'West Garo Hills', 'Longleng', 'Peren', 'Tuensang', 'Kandhamal',
   'Koraput', 'Rayagada', 'Karimnagar', 'Medak', 'Nizamabad',
   'Ranga Reddy', 'Maharajganj', 'Murshidabad'], dtype=object)
```

In [32]: df.district=df.district.str.lower()

```
In [33]: df.district.unique()
```

```
Out[33]: array(['belgaum', 'davangere', 'dharwad', 'hassan', 'haveri', 'havri',
   'khowai', 'nrth goa', 'north goa', 'north goaa', 'nort goa',
   'bangalore', 'bengalore', 'bangalor', 'chikmagalur', 'haassan',
   'hassann', 'mandya', 'maandya', 'mandyah',
   'mangalore(dakshin kannad)', 'tumkur', 'tumkoor', 'idukky',
   'idukki', 'iddukki', 'kannur', 'kasargod', 'ariyalur',
   'coimbatore', 'cuddalore', 'dindigul', 'erode', 'namakkal',
   'salem', 'vellore', 'guntur', 'kurnool', 'bastar', 'dahod',
   'rajkot', 'bellary', 'gadag', 'kolar', 'mysore', 'raichur',
   'udupi', 'alappuzha', 'malappuram', 'dhar', 'bishnupur', 'chandel',
   'imphal east', 'imphal west', 'thouabal', 'ganjam', 'jodhpur',
   'adilabad', 'hyderabad', 'khammam', 'warangal', 'sultanpur',
   'dhelai', 'ernakulam', 'angul', 'mayurbhanja', 'nowrangpur',
   'chitrakut', 'jhansi', 'araria', 'gopalgang', 'kishanganj',
   'patna', 'rohtas', 'siwan', 'supaul', 'west chambaran',
   'chandigarh', 'bilaspur', 'durg', 'rajnandgaon', 'banaskanth',
   'porbandar', 'ambala', 'faridabad', 'fatehabad', 'gurgaon',
   'hissar', 'jhajjar', 'karnal', 'kurukshetra',
   'mahendragarh-narnaul', 'mewat', 'palwal', 'panchkula', 'panipat',
   'rohtak', 'sirsia', 'sonipat', 'yamuna nagar', 'chamba', 'hamirpur',
   'kangra', 'kullu', 'mandi', 'shimla', 'sirmore', 'solan', 'una',
   'anantrnag', 'jammu', 'kathua', 'rajouri', 'srinagar', 'udhampur',
   'gulbarga', 'kozhikode(calicut)', 'thiruvananthapuram', 'bhopal',
   'gwalior', 'harda', 'hoshangabad', 'indore', 'morena', 'satna',
   'sheopur', 'ujjain', 'ahmednagar', 'amarawati', 'aurangabad',
   'chandrapur', 'jalgaon', 'kolhapur', 'mumbai', 'nagpur', 'nashik',
   'osmanabad', 'pune', 'sangli', 'sholapur', 'thane', 'delhi',
   'nuapada', 'amritsar', 'barnala', 'bhatinda', 'faridkot',
   'fatehgarh', 'fazilka', 'ferozpur', 'gurdaspur', 'hoshaipur',
   'jalandhar', 'kapurthala', 'ludhiana', 'mansa', 'moga', 'muktsar',
   'nawanshahr', 'pathankot', 'patiala', 'ropar (rupnagar)',
   'sangrur', 'tarntaran', 'ajmer', 'alwar', 'bikaner', 'chittorgarh',
   'ganganagar', 'jalore', 'udaipur', 'agra', 'ambedkarnagar',
   'babraich', 'ballia', 'barabanki', 'basti',
   'bhadohi(sant ravi nagar)', 'bulandshahar', 'deoria', 'etah',
   'faizabad', 'farukhabad', 'firozabad', 'gautam budh nagar',
   'ghaziabad', 'gorakhpur', 'hathras', 'jalaun (orai)', 'jaunpur',
   'kanpur', 'khiri (lakhimpur)', 'lucknow', 'mainpuri', 'mathura',
   'mau(maunathbhanjan)', 'meerut', 'mirzapur', 'muzaffarnagar',
   'padrauna(kusinagar)', 'pratapgarh', 'raebarelli', 'rampur',
   'saharanpur', 'shahjahanpur', 'siddharth nagar', 'sitapur',
   'unnao', 'varanasi', 'dehradoon', 'garhwal (pauri)', 'haridwar',
   'nanital', 'udhamsinghnagar', 'palakad', 'kottayam',
   'pathanamthitta', 'wayanad', 'nagercoil (kannyakumari)',
   'surajpur', 'surguja', 'balrampur', 'kabirdham', 'cuddapah',
   'janjgir', 'jashpur', 'koria', 'chamrajnagar', 'anupur',
   'chhindwara', 'hingoli', 'nanded', 'parbhani', 'vashim',
   'west garo hills', 'longleng', 'peren', 'tuensang', 'kandhamal',
   'koraput', 'rayagada', 'karimnagar', 'medak', 'nizamabad',
   'ranga reddy', 'maharajganj', 'murshidabad'], dtype=object)
```

```
In [34]: df.district.replace(['havri'], 'haveri', inplace=True)
goa={'nrth goa':'north goa', 'north goaa':'north goa', 'nort goa':'north goa'}
df=df.replace({'district':goa})
```

In [35]: df.district.unique()

```
Out[35]: array(['belgaum', 'davangere', 'dharwad', 'hassan', 'haveri', 'khawai',
   'north goa', 'bangalore', 'bengalore', 'bangalor', 'chikmagalur',
   'haassan', 'hassann', 'mandya', 'maandya', 'mandyah',
   'mangalore(dakshin kannad)', 'tumkur', 'tumkoor', 'idukky',
   'idukki', 'iddukki', 'kannur', 'kasargod', 'ariyalur',
   'coimbatore', 'cuddalore', 'dindigul', 'erode', 'namakkal',
   'salem', 'vellore', 'guntur', 'kurnool', 'bastar', 'dahod',
   'rajkot', 'bellary', 'gadag', 'kolar', 'mysore', 'raichur',
   'udipi', 'alappuzha', 'malappuram', 'dhar', 'bishnupur', 'chandel',
   'imphal east', 'imphal west', 'thoubal', 'ganjam', 'jodhpur',
   'adilabad', 'hyderabad', 'khammam', 'warangal', 'sultanpur',
   'dhala', 'ernakulam', 'angul', 'mayurbhanja', 'nowarangpur',
   'chitrakut', 'jhansi', 'araria', 'gopalgang', 'kishanganj',
   'patna', 'rohtas', 'siwan', 'supaul', 'west chambaran',
   'chandigarh', 'bilaspur', 'durg', 'rajinandgaon', 'banaskanth',
   'porbandar', 'ambala', 'faridabad', 'fatehabad', 'gurgaon',
   'hissar', 'jhajjar', 'karnal', 'kurukshetra',
   'mahendragarh-narnaul', 'mewat', 'palwal', 'panchkula', 'panipat',
   'rohtak', 'sirs', 'sonipat', 'yamuna nagar', 'chamba', 'hamirpur',
   'kangra', 'kullu', 'mandi', 'shimla', 'sirmore', 'solan', 'una',
   'anantrnag', 'jammu', 'kathua', 'rajouri', 'srinagar', 'udhampur',
   'gulbarga', 'kozhikode(calicut)', 'thiruvananthapuram', 'bhopal',
   'gwalior', 'harda', 'hoshangabad', 'indore', 'morena', 'satna',
   'sheopur', 'ujjain', 'ahmednagar', 'amarawati', 'aurangabad',
   'chandrapur', 'jalgaon', 'kolhapur', 'mumbai', 'nagpur', 'nashik',
   'osmanabad', 'pune', 'sangli', 'sholapur', 'thane', 'delhi',
   'nuapada', 'amritsar', 'barnala', 'bhatinda', 'faridkot',
   'fatehgarh', 'fazilka', 'ferozpur', 'gurdaspur', 'hoshiarpur',
   'jalandhar', 'kapurthal', 'ludhiana', 'mansa', 'moga', 'muktsar',
   'nawanshahr', 'pathankot', 'patiala', 'ropar (rupnagar)',
   'sangrur', 'tarntaran', 'ajmer', 'alwar', 'bikaner', 'chittorgarh',
   'ganganagar', 'jalore', 'udaipur', 'agra', 'ambedkarnagar',
   'babraich', 'ballia', 'barabanki', 'basti',
   'bhadohi(sant ravi nagar)', 'bulandshahar', 'deoria', 'etah',
   'faizabad', 'farukhabad', 'firozabad', 'gautam budh nagan',
   'ghaziabad', 'gorakhpur', 'hathras', 'jalaun (orai)', 'jaunpur',
   'kanpur', 'khiri (lakhimpur)', 'lucknow', 'mainpuri', 'mathura',
   'mau(maunathbhanjan)', 'meerut', 'mirzapur', 'muzaffarnagar',
   'padrauna(kusinagar)', 'pratapgarh', 'raebarelli', 'rampur',
   'saharanpur', 'shahjahanpur', 'siddharth nagar', 'sitapur',
   'unnao', 'varanasi', 'dehradoon', 'garhwal (pauri)', 'haridwar',
   'nanital', 'udhamsinghnagar', 'palakad', 'kottayam',
   'pathanamthitta', 'wayanad', 'nagercoil (kannyakumari)',
   'surajpur', 'suruja', 'balrampur', 'kabirdham', 'cuddapah',
   'janjgir', 'jashpur', 'koria', 'chamrajnagar', 'anupur',
   'chhindwara', 'hingoli', 'nanded', 'parbhani', 'vashim',
   'west garo hills', 'longleng', 'peren', 'tuensang', 'kandhamal',
   'koraput', 'rayagada', 'karimnagar', 'medak', 'nizamabad',
   'ranga reddy', 'maharajganj', 'murshidabad'], dtype=object)
```

In [36]: dist=list(df.district.unique())
dist.sort()
print(dist)

```
['adilabad', 'agra', 'ahmednagar', 'ajmer', 'alappuzha', 'alwar', 'amarawati', 'ambala', 'ambedkarnagar', 'amritsar', 'anantrnag', 'angul', 'anupur', 'araria', 'ariyalur', 'aurangabad', 'babraich', 'ballia', 'balrampur', 'banaskanth', 'bangalore', 'bangalore', 'barabanki', 'barnala', 'bastar', 'basti', 'belgaum', 'bellary', 'bengalore', 'bhadohi(sant ravi nagar)', 'bhatinda', 'bhopal', 'bikaner', 'bilaspur', 'bishnupur', 'bulandshahar', 'chamba', 'chamrajnagar', 'chandel', 'chandigarh', 'chandrapur', 'chhindwara', 'chikmagalur', 'chitrakut', 'chittorgarh', 'coimbatore', 'cuddalore', 'cuddapah', 'dahod', 'davangere', 'dehradoon', 'delhi', 'deoria', 'dhala', 'dharwad', 'dindigul', 'durg', 'ernakulam', 'erode', 'etah', 'faizabad', 'faridabad', 'faridkot', 'farukhabad', 'fatehabad', 'fatehgarh', 'fazilka', 'ferozpur', 'firozabad', 'gadag', 'ganganagar', 'ganjam', 'garhwal (pauri)', 'gautam budh nagan', 'ghaziabad', 'gopalgang', 'gorakhpur', 'gulbarga', 'guntur', 'gurdaspur', 'gurgaon', 'gwalior', 'haassan', 'hamirpur', 'harda', 'haridwar', 'hassan', 'hassann', 'hathras', 'haveri', 'hingoli', 'hissar', 'hoshangabad', 'hoshiarpur', 'hyderabad', 'iddukki', 'idukky', 'imphal east', 'imphal west', 'indore', 'jalandhar', 'jalau n (orai)', 'jalgaon', 'jalore', 'jammu', 'janjgir', 'jashpur', 'jaunpur', 'jhajjar', 'jodhpur', 'kabirdham', 'kandhamal', 'kangra', 'kannur', 'kanpur', 'kapurthal', 'karimnagar', 'karnal', 'kasargod', 'kathua', 'khammam', 'khiri (lakhimpur)', 'khawai', 'kishanganj', 'kolar', 'kolhapur', 'koraput', 'koria', 'kottayam', 'kozhikode(calicut)', 'kullu', 'kurnool', 'kurukshetra', 'longleng', 'lucknow', 'ludhiana', 'maandya', 'mahendragarh-narnaul', 'mainpuri', 'malappuram', 'mandi', 'mandya', 'mandyah', 'mangalore(dakshin kannad)', 'mansa', 'mathura', 'mau(maunathbhanjan)', 'mayurbhanja', 'medak', 'meerut', 'mewat', 'mirzapur', 'moga', 'morena', 'muktsar', 'mumbai', 'murshidabad', 'muzaffarnagar', 'mysore', 'nagercoil (kannyakumari)', 'nagpur', 'namakkal', 'nanded', 'nanital', 'nashik', 'nawanshahr', 'nizamabad', 'north goa', 'nowarangpur', 'nuapada', 'osmanabad', 'padrauna(kusinagar)', 'palakad', 'palwal', 'panchkula', 'panipat', 'parbhani', 'pathanamthitta', 'pathankot', 'patiala', 'patna', 'peren', 'porbandar', 'pratapgarh', 'pune', 'raebarelli', 'raichur', 'rajkot', 'rajinandgaon', 'rajouri', 'rampur', 'ranga reddy', 'rayagada', 'rohtak', 'rohtas', 'ropar (rupnagar)', 'saharanpur', 'salem', 'sangli', 'sangrur', 'satna', 'shahjahanpur', 'sheopur', 'shimla', 'sholapur', 'siddharth nagar', 'sirmore', 'sirs', 'sitapur', 'siwan', 'solan', 'sonipat', 'srinagar', 'sultapur', 'supaul', 'surajpur', 'suruja', 'tarntaran', 'thane', 'thiruvananthapuram', 'thoubal', 'tuensang', 'tumkoor', 'tumkur', 'udaipur', 'udhampur', 'udhamsinghnagar', 'udipi', 'ujjain', 'una', 'unnao', 'varanasi', 'vashim', 'vellore', 'warangal', 'wayanad', 'west chambaran', 'west garo hills', 'yamuna nagar']
```

```
In [37]: d={'bengalore':'bangalore','bangalor':'bangalore', 'maandya':'mandya','mandyah':'mandya','haassan':'hassan','hassa
In [38]: df=df.replace({'district':d})
```

```
In [39]: df.district.unique()
```

```
Out[39]: array(['belgaum', 'davangere', 'dharwad', 'hassan', 'haveri', 'khawai',
   'north goa', 'bangalore', 'chikmagalur', 'mandya',
   'mangalore(dakshin kannad)', 'tumkur', 'tumkoor', 'idukki',
   'kannur', 'kasargod', 'ariyalur', 'coimbatore', 'cuddalore',
   'dindigul', 'erode', 'namakkal', 'salem', 'vellore', 'guntur',
   'kurnool', 'bastar', 'dahod', 'rajkot', 'bellary', 'gadag',
   'kolar', 'mysore', 'raichur', 'udupi', 'alappuzha', 'malappuram',
   'dhar', 'bishnupur', 'chandel', 'imphal east', 'imphal west',
   'thoubal', 'ganjam', 'jodhpur', 'adilabad', 'hyderabad', 'khammam',
   'warangal', 'sultanpur', 'dhala', 'ernakulam', 'angul',
   'mayurbhanja', 'nowrangpur', 'chitrakut', 'jhansi', 'araria',
   'gopalgang', 'kishanganj', 'patna', 'rohtas', 'siwan', 'supaul',
   'west chambaran', 'chandigarh', 'bilaspur', 'durg', 'rajnandgaon',
   'banaskanth', 'porbandar', 'ambala', 'faridabad', 'fatehabad',
   'gurgaon', 'hissar', 'jhajjar', 'karnal', 'kurukshetra',
   'mahendragarh-narnaul', 'mewat', 'palwal', 'panchkula', 'panipat',
   'rohtak', 'sirsia', 'sonipat', 'yamuna nagar', 'chamba', 'hamirpur',
   'kangra', 'kullu', 'mandi', 'shimla', 'sirmore', 'solan', 'una',
   'ananthag', 'jammu', 'kathua', 'rajouri', 'srinagar', 'udhampur',
   'gulbarga', 'kozhikode(calicut)', 'thiruvananthapuram', 'bhopal',
   'gwalior', 'harda', 'hoshangabad', 'indore', 'morena', 'satna',
   'sheopur', 'ujjain', 'ahmednagar', 'amarawati', 'aurangabad',
   'chandrapur', 'jalgaon', 'kolhapur', 'mumbai', 'nagpur', 'nashik',
   'osmanabad', 'pune', 'sangli', 'sholapur', 'thane', 'delhi',
   'nuapada', 'amritsar', 'barnala', 'bhatinda', 'faridkot',
   'fatehgarh', 'fazilka', 'ferozpur', 'gurdaspur', 'hoshiarpur',
   'jalandhar', 'kapurthala', 'ludhiana', 'mansa', 'moga', 'muktsar',
   'nawanshahr', 'pathankot', 'patiala', 'ropar (rupnagar)',
   'sangrur', 'tarntaran', 'ajmer', 'alwar', 'bikaner', 'chittorgarh',
   'ganganagar', 'jalore', 'udaipur', 'agra', 'ambedkaragar',
   'bahraich', 'ballia', 'barabanki', 'basti',
   'bhadohi(sant ravi nagar)', 'bulandshahar', 'deoria', 'etah',
   'faizabad', 'farukhabad', 'firozabad', 'gautam budh nagar',
   'ghaziabad', 'gorakhpur', 'hathras', 'jalaun (orai)', 'jaunpur',
   'kanpur', 'khiri (lakhimpur)', 'lucknow', 'mainpuri', 'mathura',
   'mau(maunathbhanjan)', 'meerut', 'mirzapur', 'muzaffarnagar',
   'padrauna(kusinagar)', 'pratapgarh', 'raebarelli', 'rampur',
   'saharanpur', 'shahjahanpur', 'siddharth nagar', 'sitapur',
   'unnao', 'varanasi', 'dehradoon', 'garhwal (pauri)', 'haridwar',
   'nanital', 'udhamsinghnagar', 'palakad', 'kottayam',
   'pathanamthitta', 'wayanad', 'nagercoil (kannyakumari)',
   'surajpur', 'surguja', 'balrampur', 'kabirdham', 'cuddapah',
   'janjgir', 'jashpur', 'koria', 'chamrajnagar', 'anupur',
   'chhindwara', 'hingoli', 'nanded', 'parbhani', 'vashim',
   'west garo hills', 'longleng', 'peren', 'tuensang', 'kandhamal',
   'koraput', 'rayagada', 'karimnagar', 'medak', 'nizamabad',
   'ranga reddy', 'maharajganj', 'murshidabad'], dtype=object)
```

```
In [40]: df.commodity=df.commodity.str.lower()
```

```
In [41]: df.commodity.unique()
```

```
Out[41]: array(['bull', 'copra', 'copraa', 'copraa//', 'dry chillies', 'duck',
   'egg', 'eggs', 'grapes', 'grapee', 'green fodder', 'jasmine',
   'rubber', 'sugarcane', 'turmeric', 'turmerc', 'dry chilly',
   'drychilly', 'greenfodder', 'jasmin', 'jasmyne'], dtype=object)
```

```
In [42]: c={'copraa':'copra', 'copraa//':'copra','egg': 'eggs', 'grapee':'grapes', 'turmerc':'turmeric', 'jasmyne':'jasmine',
   'drychilly','dry chillies':'drychilly','green fodder': 'greenfodder'}
```

```
In [43]: df=df.replace({'commodity':c})
```

```
In [44]: df.commodity.unique()
```

```
Out[44]: array(['bull', 'copra', 'drychilly', 'duck', 'eggs', 'grapes',
   'greenfodder', 'jasmine', 'rubber', 'sugarcane', 'turmeric'],
   dtype=object)
```

```
In [45]: df.variety=df.variety.str.lower()
```

```
In [46]: df.variety.unique()
```

```
Out[46]: array(['bull', 'bul1l', 'copra', 'copraa', 'medium', 'other', 'ball',
       'small', 'milling copra', 'guntur', 'red', 'red new', 'red top',
       'white', 'local', 'dry', 'dry chillies', 'byadgi', 'mankattu',
       'dabbi', 'kaddi', '1st sort', '2nd sort', 'talu', 'desi', 'egg',
       'annabesahai', 'green', 'black', 'indian', 'natal', 'green fodder',
       'jasmine', 'rss-4', 'sugarcane', 'bulb', 'finger', 'turmeric',
       'turmeric stick', 'rajapuri'], dtype=object)
```

```
In [47]: vary={'bul1l':'bull','copraa':'copra'}
df=df.replace({'variety':vary})
```

```
In [48]: df.state.value_counts()
```

```
Out[48]: kerala          14360
tamil nadu        10516
punjab            8982
haryana           6880
uttar pradesh     6276
maharashtra      5742
telangana          5686
karnataka         4938
himachal pradesh 2906
rajasthan          1810
uttrakhand         1756
andhra pradesh    1278
jammu and kashmir 1250
madhya pradesh    1204
chattisgarh        1084
odisha              984
nct of delhi       742
west bengal        672
bihar                498
goa                  388
gujarat             262
tripura              129
manipur              126
chandigarh            88
nagaland              58
meghalaya             36
Name: state, dtype: int64
```

```
In [49]: df.district.value_counts()
```

```
Out[49]: erode          3586
idukki            2922
kannur            2518
coimbatore        2456
kasargod          2234
...
supaul              2
alappuzha          2
rampur              2
ariyalur            2
nuapada              2
Name: district, Length: 235, dtype: int64
```

```
In [50]: df.commodity.value_counts()
```

```
Out[50]: grapes          41948
turmeric          14520
copra            12300
rubber            5396
eggs              1176
greenfodder        996
drychilly          928
jasmine            704
sugarcane          510
duck                118
bull                 55
Name: commodity, dtype: int64
```

In [51]: df.unique()

```
Out[51]: s_no      78603
state       26
district    235
market      459
pincode     78603
country      1
commodity   11
variety     38
arrival_date 810
min_price   2540
max_price   3061
modal_price 3207
arrival_day   7
dtype: int64
```

In [52]: df.head()

Out[52]:

	s_no	state	district	market	pincode	country	commodity	variety	arrival_date	min_price	max_price	modal_price	arrival_day
0	0	karnataka	belgaum	Athani	100001	India	bull	bull	2020-01-26	20000.0	40000.0	40000.0	Sunday
1	1	karnataka	belgaum	Athani	100002	India	bull	bull	2020-02-16	10000.0	12000.0	11000.0	Sunday
2	2	karnataka	belgaum	Athani	100003	India	bull	bull	2020-01-03	32000.0	40000.0	35000.0	Friday
3	3	karnataka	belgaum	Ramdurga	100004	India	bull	bull	2020-05-01	20000.0	25000.0	20000.0	Friday
4	4	karnataka	belgaum	Ramdurga	100005	India	bull	bull	2020-12-01	20000.0	35000.0	25000.0	Tuesday

In [53]: print("IS THERE IS ANY DUPLICATES ",df.duplicated().any())

IS THERE IS ANY DUPLICATES True

ANALYSING

In [54]: df.describe()

Out[54]:

	s_no	pincode	min_price	max_price	modal_price
count	78651.000000	78651.000000	78651.000000	78651.000000	78651.000000
mean	39324.978030	139325.978030	6574.610323	8105.770534	7360.387507
std	22704.694333	22704.694333	16577.056585	19472.631902	17951.236404
min	0.000000	100001.000000	2.000000	0.000000	2.000000
25%	19662.500000	119663.500000	3550.000000	4850.000000	4200.000000
50%	39325.000000	139326.000000	5211.000000	6969.000000	6000.000000
75%	58987.500000	158988.500000	7600.000000	8840.000000	8210.000000
max	78602.000000	178603.000000	1000000.000000	1000000.000000	1000000.000000

```
In [55]: plt.figure(figsize=(15, 8))
sns.heatmap(df.corr(), annot=True)
plt.savefig("heatmap.png", dpi=300)
plt.show()
```



DATA VISUALIZATION

```
In [68]: max_states = df['state'].count()
print('Maximum number of states:', max_states)
```

Maximum number of states: 78651

TOP 10 AND LAST 10 COUNTS OF STATES

```
In [77]: state_counts = df['state'].value_counts()

# Get the top 10 states with the highest counts
top_states = state_counts.nlargest(10)

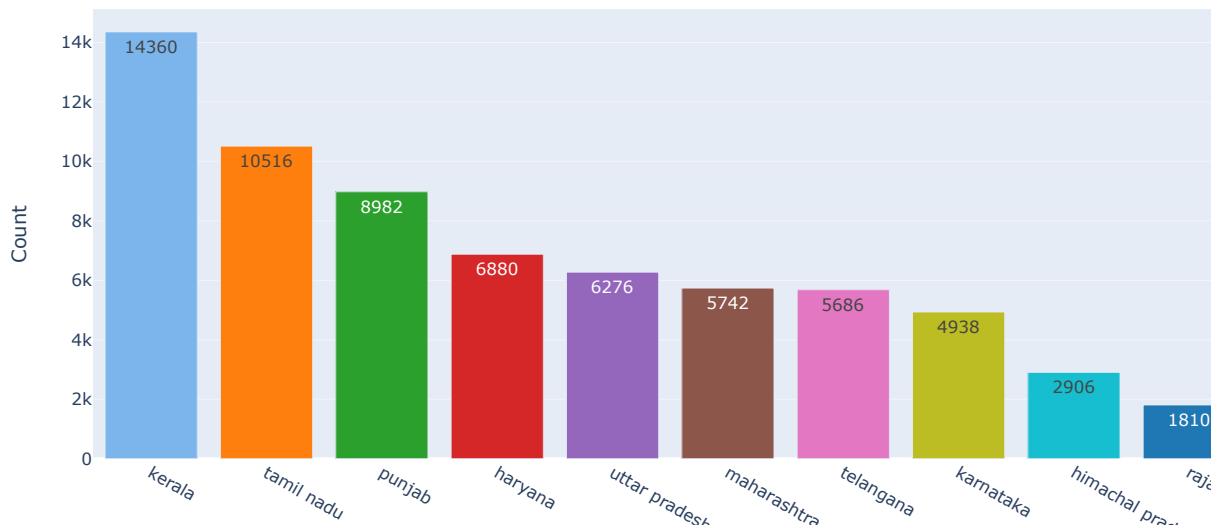
# Define a color scale for the bars
color_scale = ['#7CB5EC', '#FF7F0E', '#2CA02C', '#D62728', '#9467BD',
               '#8C564B', '#E377C2', '#BCBD22', '#17BECF', '#1F77B4']

# Create a bar chart using Plotly
fig = go.Figure([go.Bar(x=top_states.index, y=top_states.values,
                        text=top_states.values, marker=dict(color=color_scale))])

# Add a title and axis labels
fig.update_layout(title='Top 10 States by Count',
                  xaxis_title='State', yaxis_title='Count')

# Show the plot
fig.show()
```

Top 10 States by Count



```
In [123]: state_counts = df['state'].value_counts()

# Get the last 10 states with the lowest counts
last_states = state_counts.nsmallest(10)

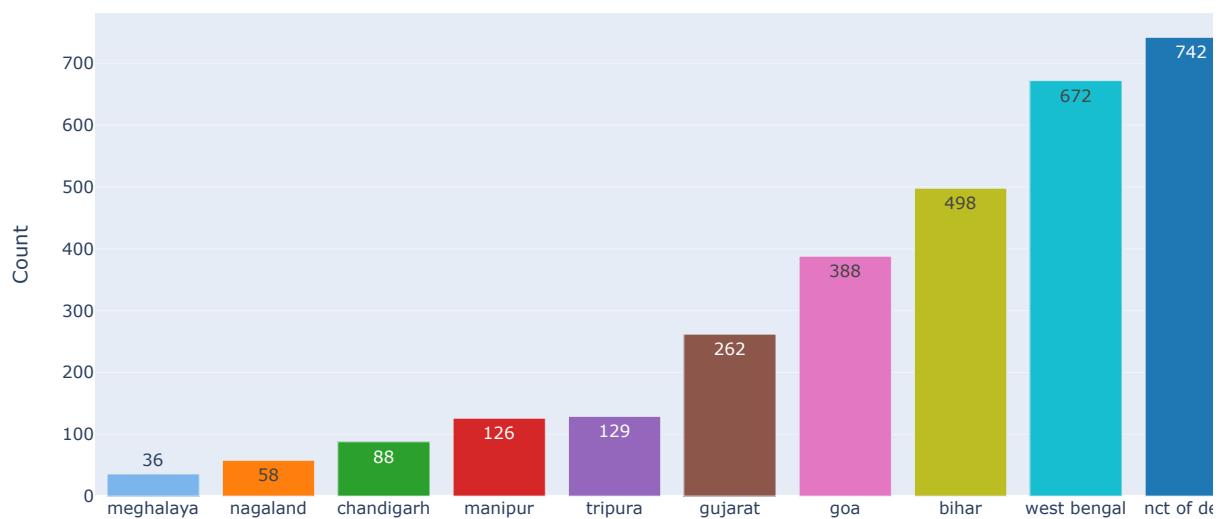
# Define a color scale for the bars
color_scale = ['#7CB5EC', '#FF7F0E', '#2CA02C', '#D62728', '#9467BD',
               '#8C564B', '#E377C2', '#BCBD22', '#17BECF', '#1F77B4']

# Create a bar chart using Plotly
fig = go.Figure([go.Bar(x=last_states.index, y=last_states.values,
                        text=last_states.values, marker=dict(color=color_scale))])

# Add a title and axis labels
fig.update_layout(title='Last 10 States by Count',
                  xaxis_title='State', yaxis_title='Count')

# Show the plot
fig.show()
```

Last 10 States by Count



TOP 10 AND LAST 10 COUNTS OF DISTRICT

```
In [79]: district_counts = df['district'].value_counts()

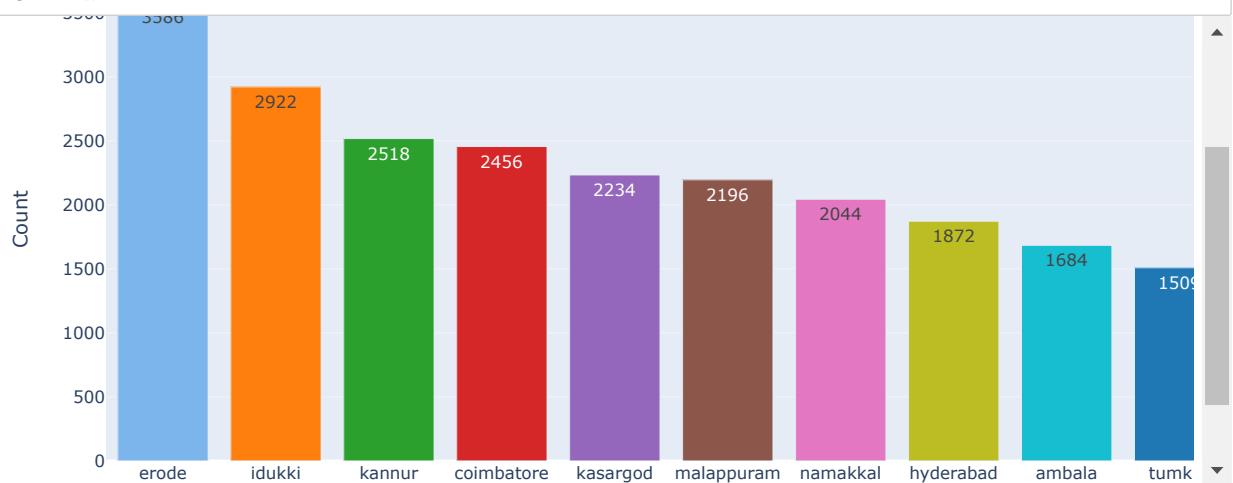
# Get the top 10 states with the highest counts
top_district = district_counts.nlargest(10)

# Define a color scale for the bars
color_scale = ['#7CB5EC', '#FF7F0E', '#2CA02C', '#D62728', '#9467BD',
               '#8C564B', '#E377C2', '#BCBD22', '#17BECF', '#1F77B4']

# Create a bar chart using Plotly
fig = go.Figure([go.Bar(x=top_district.index, y=top_district.values,
                        text=top_district.values, marker=dict(color=color_scale))])

# Add a title and axis labels
fig.update_layout(title='Top 10 district by Count',
                  xaxis_title='district', yaxis_title='Count')

# Show the plot
fig.show()
```



```
In [124]: district_counts = df['district'].value_counts()

# Get the last 10 districts with the lowest counts
last_district = district_counts.nsmallest(10)

# Define a color scale for the bars
color_scale = ['#7CB5EC', '#FF7F0E', '#2CA02C', '#D62728', '#9467BD',
               '#8C564B', '#E377C2', '#BCBD22', '#17BECF', '#1F77B4']

# Create a bar chart using Plotly
fig = go.Figure([go.Bar(x=last_district.index, y=last_district.values,
                        text=last_district.values, marker=dict(color=color_scale))])

# Add a title and axis labels
fig.update_layout(title='Last 10 district by Count',
                  xaxis_title='district', yaxis_title='Count')

# Show the plot
fig.show()
```

Last 10 district by Count

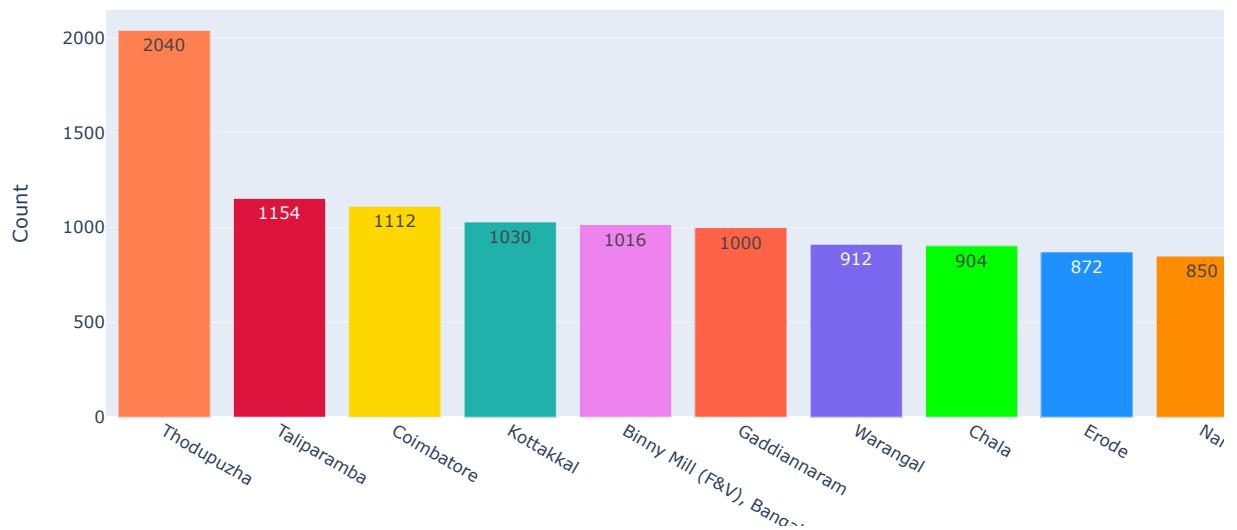


TOP TEN AND LAST TEN COUNTS OF MARKET

```
In [82]: market_counts = df['market'].value_counts()
top_market = market_counts.nlargest(10)

color_scale = ['#FF7F50', '#DC143C', '#FFD700', '#20B2AA', '#EE82EE',
               '#FF6347', '#7B68EE', '#00FF00', '#1E90FF', '#FF8C00']
fig = go.Figure([go.Bar(x=top_market.index, y=top_market.values,
                        text=top_market.values, marker=dict(color=color_scale))])
fig.update_layout(title='Top 10 market by Count',
                  xaxis_title='market', yaxis_title='Count')
fig.show()
```

Top 10 market by Count



In [126]:

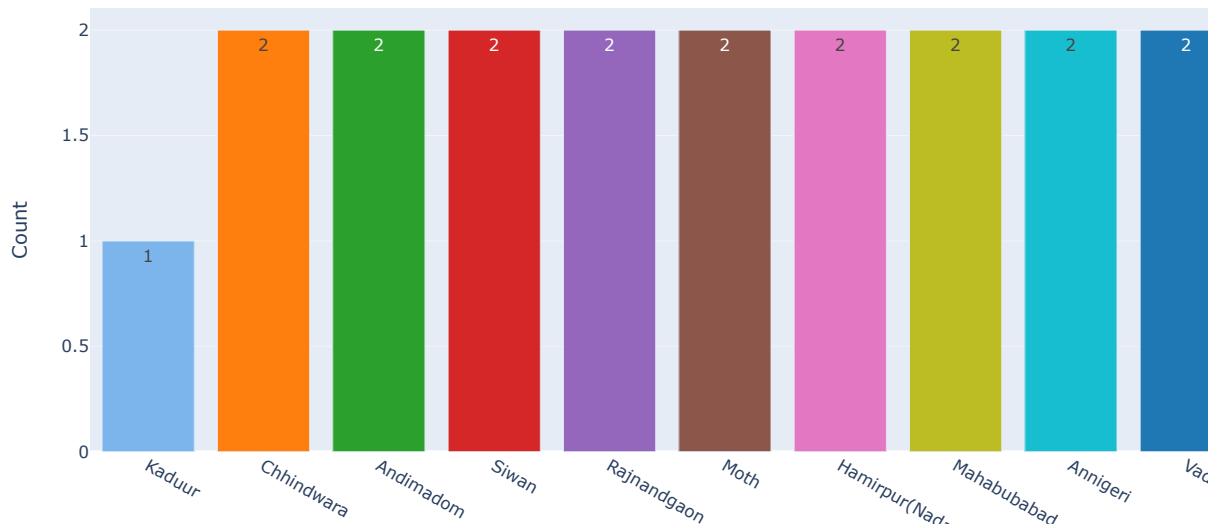
```
market_counts = df['market'].value_counts()
last_market = market_counts.nsmallest(10)
color_scales = ['#7CB5EC', '#FF7F0E', '#2CA02C', '#D62728', '#9467BD',
                '#8C564B', '#E377C2', '#BCBD22', '#17BECF', '#1F77B4']

fig = go.Figure([go.Bar(x=last_market.index, y=last_market.values,
                        text=last_market.values,
                        marker=dict(color=[color_scales[i] for i in range(len(last_market))]))])

fig.update_layout(title='Last 10 markets by Count',
                  xaxis_title='Market', yaxis_title='Count')

fig.show()
```

Last 10 markets by Count



TOP 10 AND LAST 10 COUNTS OF COMMODITY

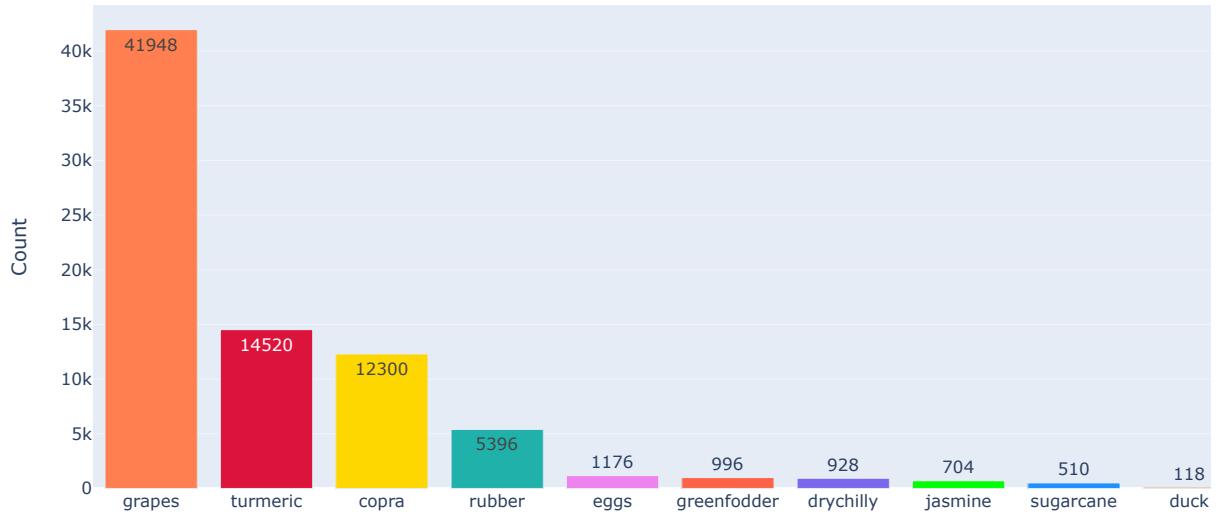
```
In [83]: commodity_counts = df['commodity'].value_counts()

top_commodity = commodity_counts.nlargest(10)
color_scale = ['#FF7F50', '#DC143C', '#FFD700', '#20B2AA', '#EE82EE',
               '#FF6347', '#7B68EE', '#00FF00', '#1E90FF', '#FF8C00']
fig = go.Figure([go.Bar(x=top_commodity.index, y=top_commodity.values,
                        text=top_commodity.values, marker=dict(color=color_scale))])

fig.update_layout(title='Top 10 commodity by Count',
                  xaxis_title='commodity', yaxis_title='Count')

fig.show()
```

Top 10 commodity by Count



In [127]:

```
commodity_counts = df['commodity'].value_counts()
last_commodity = commodity_counts.nsmallest(10)

color_scale = ['#FF7F50', '#DC143C', '#FFD700', '#20B2AA', '#EE82EE',
               '#FF6347', '#7B68EE', '#00FF00', '#1E90FF', '#FF8C00']

fig = go.Figure([go.Bar(x=last_commodity.index, y=last_commodity.values,
                        text=last_commodity.values, marker=dict(color=color_scale))])

fig.update_layout(title='Last 10 commodities by Count',
                  xaxis_title='Commodity', yaxis_title='Count')
fig.show()
```

Last 10 commodities by Count



TOP 10 AND LAST 10 COUNTS OF VARIETY

```
In [88]: variety_counts = df['variety'].value_counts()
top_variety = variety_counts.nlargest(10)

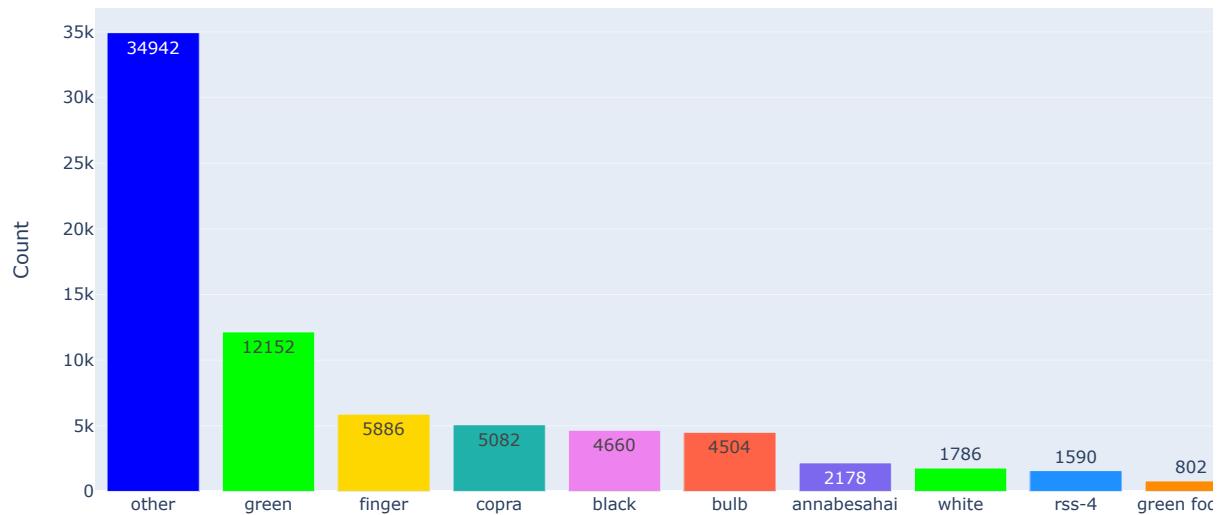
color_scale = ['#0000FF', '#00FF00', '#FFD700', '#20B2AA', '#EE82EE', '#FF6347', '#7B68EE', '#00FF00', '#1E90FF', '#800080']

fig = go.Figure([go.Bar(x=top_variety.index, y=top_variety.values,
                        text=top_variety.values, marker=dict(color=color_scale))])

fig.update_layout(title='Top 10 variety by Count',
                  xaxis_title='variety', yaxis_title='Count')

fig.show()
```

Top 10 variety by Count



```
In [128]: variety_counts = df['variety'].value_counts()
last_variety = variety_counts.nsmallest(10)

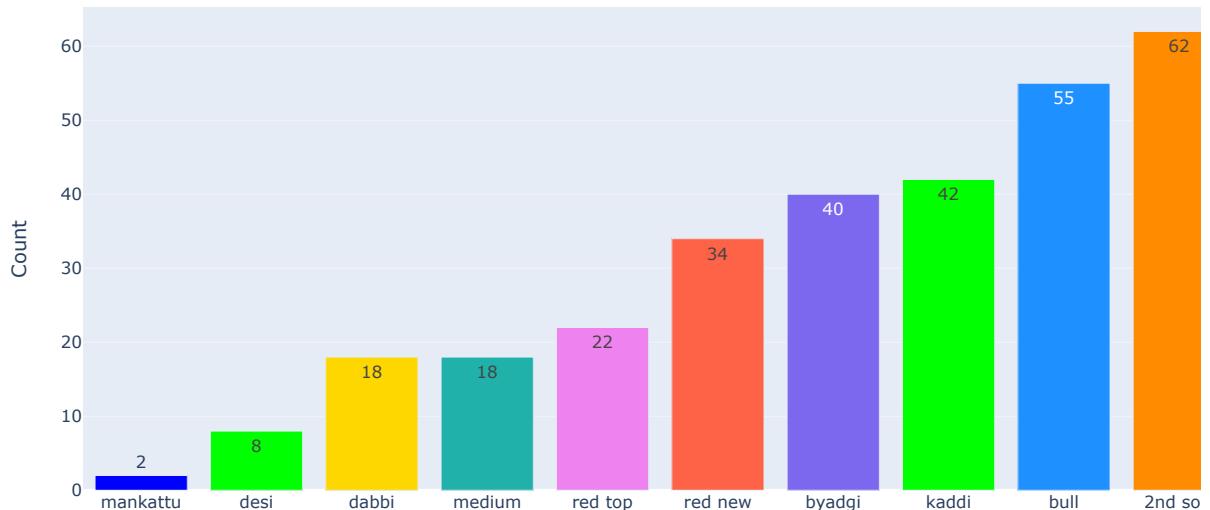
color_scale = ['#0000FF', '#00FF00', '#FFD700', '#20B2AA', '#EE82EE', '#FF6347', '#7B68EE', '#00FF00', '#1E90FF', '#FF0000']

fig = go.Figure([go.Bar(x=last_variety.index, y=last_variety.values,
                        text=last_variety.values, marker=dict(color=color_scale))])

fig.update_layout(title='Last 10 variety by Count',
                  xaxis_title='variety', yaxis_title='Count')

fig.show()
```

Last 10 variety by Count



FREQUENCY OF ANALYSING BY ARRIVAL STATES

```
In [99]: import plotly.graph_objects as go

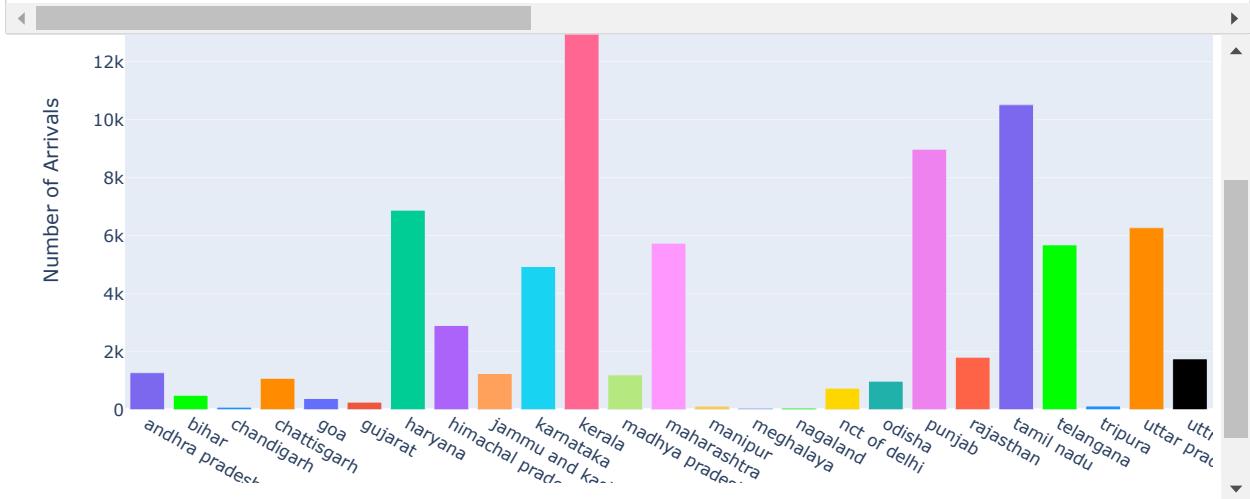
df['arrival_date'] = pd.to_datetime(df['arrival_date'])

df['Year'] = df['arrival_date'].dt.year

state_counts = df.groupby('state')['arrival_date'].count()

color_scale = ['#7B68EE', '#00FF00', '#1E90FF', '#FF8C00', '#636EFA', '#EF553B', '#00CC96', '#AB63FA', '#FFA15A', '#800080']

fig = go.Figure(
    data=go.Bar(x=state_counts.index, y=state_counts.values, marker=dict(color=color_scale)),
    layout=go.Layout(
        title='Frequency of Arrivals by State',
        xaxis=dict(title='State'),
        yaxis=dict(title='Number of Arrivals')
    )
)
fig.show()
```



TOP 10 DISTRICT BY MAXIMUM PRICE

```
In [100]:  
top_districts = df.groupby('district').max()['max_price'].sort_values(ascending=False)[:10]  
  
colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2']  
  
fig = go.Figure(data=[go.Bar(  
    x=top_districts.index,  
    y=top_districts.values,  
    text=top_districts.values,  
    textposition='auto',  
    marker_color=colors  
)])  
  
fig.update_layout(  
    title='Top 10 districts by max price',  
    xaxis_title='District',  
    yaxis_title='Max price',  
)  
pio.show(fig)
```

Top 10 districts by max price



TOP 10 DISTRICT BY MINIMUM PRICE

In [105]:

```
top_districts = df.groupby('district').min()['min_price'].sort_values(ascending=False)[:10]

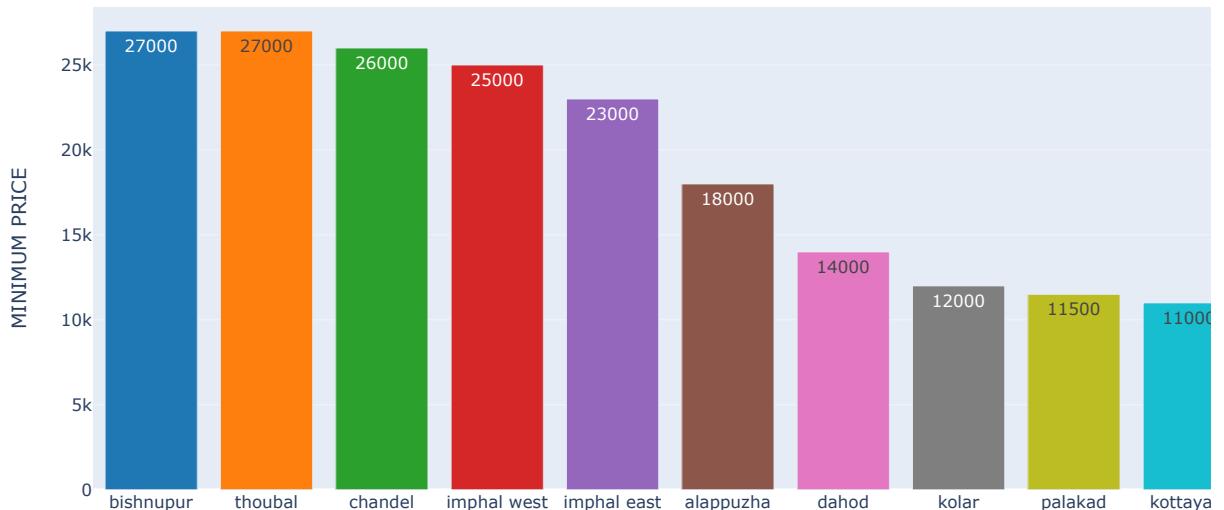
colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']

fig = go.Figure(data=[go.Bar(
    x=top_districts.index,
    y=top_districts.values,
    text=top_districts.values,
    textposition='auto',
    marker_color=colors
)])

fig.update_layout(
    title='Top 10 districts by min price',
    xaxis_title='District',
    yaxis_title='MINIMUM PRICE',
)

pio.show(fig)
```

Top 10 districts by min price



TOP 10 COMMODITY BY MAXIMUM PRICE

In [109]:

```
top_commodities = df.groupby('commodity').max()['max_price'].sort_values(ascending=False)[:10]

colors = ['#bcdbd2', '#17becf', '#FF7F50', '#DC143C', '#FFD700', '#20B2AA', '#EE82EE', '#7f7f7f', '#bcdbd2', '#17becf']

fig = go.Figure(data=[go.Bar(
    x=top_commodities.index,
    y=top_commodities.values,
    text=top_commodities.values,
    textposition='auto',
    marker_color=colors
)])

fig.update_layout(
    title='Top 10 commodities by max price',
    xaxis_title='Commodity',
    yaxis_title='MAXIMUM PRICE',
)
pio.show(fig)
```

Top 10 commodities by max price



TOP 10 COMMODITY BY MINIMUM PRICE

In [110]:

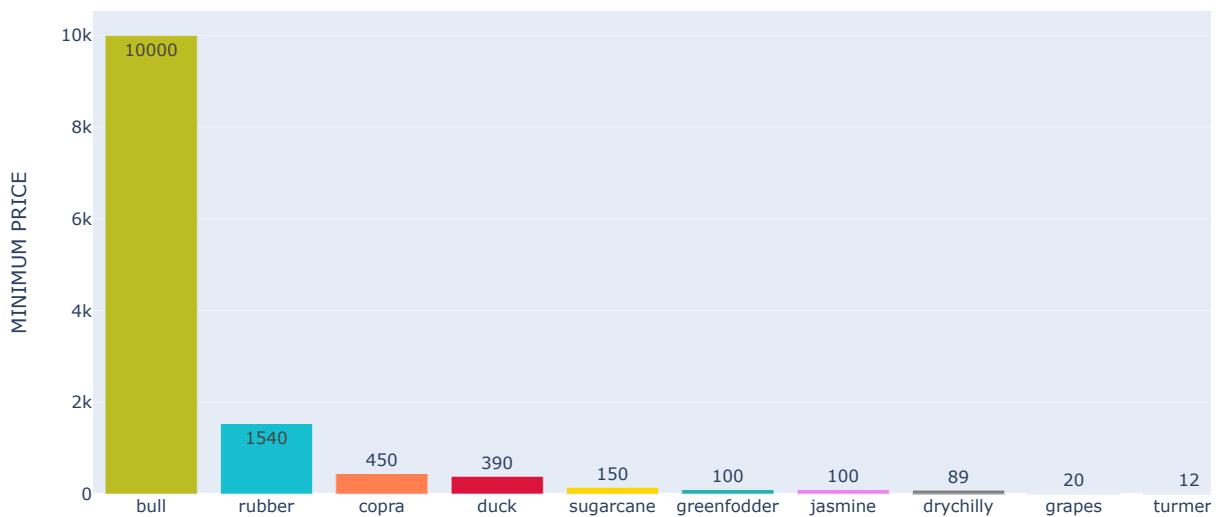
```
top_commodities = df.groupby('commodity').min()['min_price'].sort_values(ascending=False)[:10]

colors = ['#bcdbd2', '#17becf', '#FF7F50', '#DC143C', '#FFD700', '#20B2AA', '#EE82EE', '#7f7f7f', '#bcdbd2', '#17becf']

fig = go.Figure(data=[go.Bar(
    x=top_commodities.index,
    y=top_commodities.values,
    text=top_commodities.values,
    textposition='auto',
    marker_color=colors
)])

fig.update_layout(
    title='Top 10 commodities by min_price',
    xaxis_title='Commodity',
    yaxis_title='MINIMUM PRICE',
)
pio.show(fig)
```

Top 10 commodities by min_price



TOP 10 STATES BY MAXIMUM PRICE

```
In [120]: import plotly.graph_objs as go

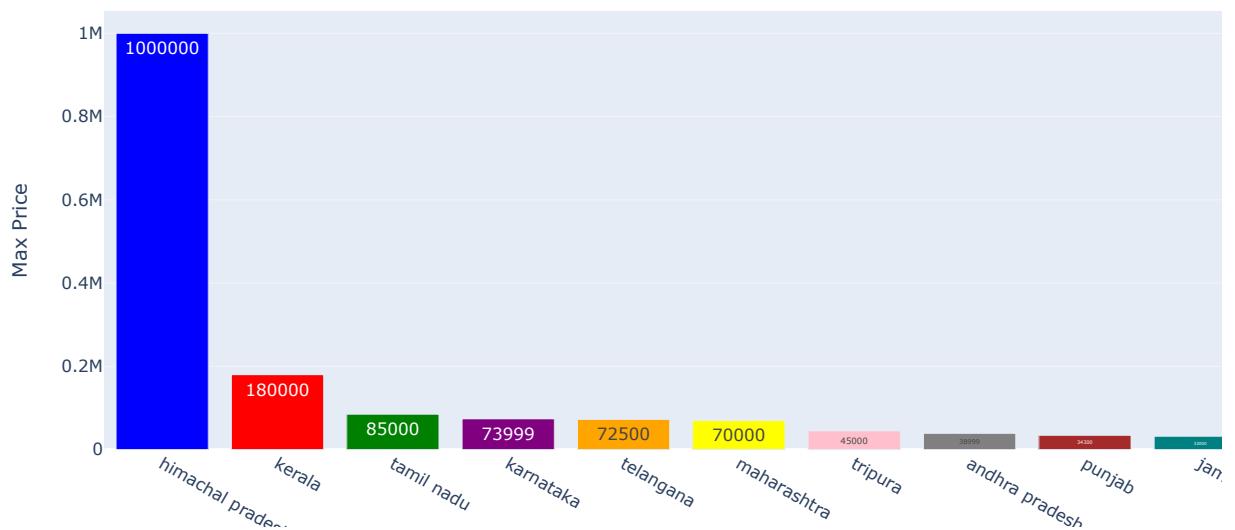
max_prices = df.groupby('state')['max_price'].max()
top_10_max_prices = max_prices.sort_values(ascending=False)[:10]
top_10_states = top_10_max_prices.index.tolist()

colors = ['blue', 'red', 'green', 'purple', 'orange', 'yellow', 'pink', 'gray', 'brown', 'teal']

fig = go.Figure(go.Bar(
    x=top_10_states,
    y=top_10_max_prices,
    marker=dict(color=colors),
    text=top_10_max_prices,
    textposition='inside',
))

fig.update_layout(
    title='Top 10 States by Max Price',
    xaxis_title='State',
    yaxis_title='Max Price',
)
fig.show()
```

Top 10 States by Max Price



TOP 10 STATE BY MINIMUM PRICE

```
In [122]: import plotly.graph_objs as go

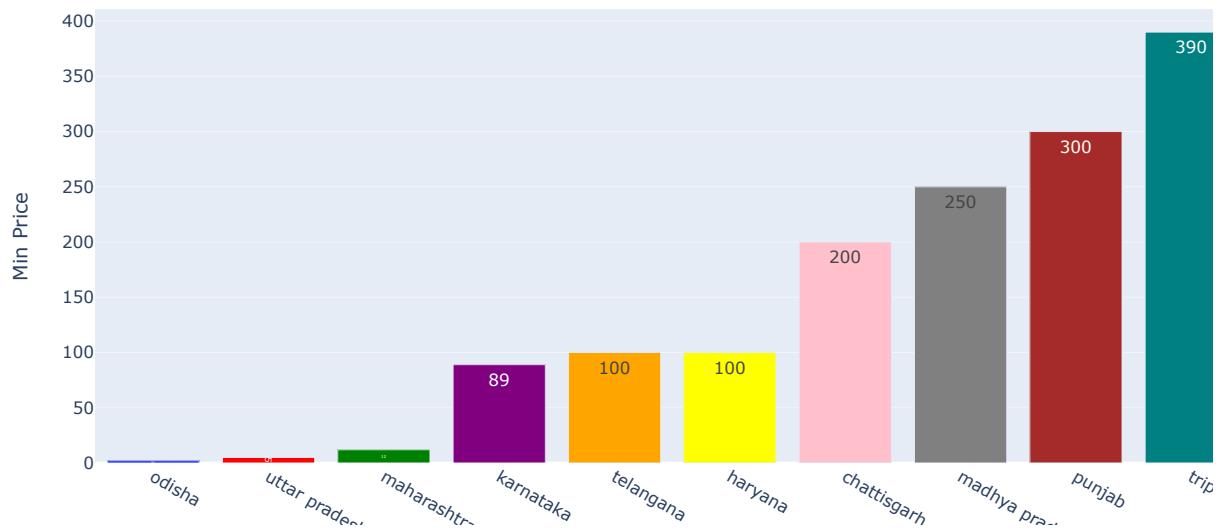
min_prices = df.groupby('state')['min_price'].min()
top_10_min_prices = min_prices.sort_values()[:10]
top_10_states = top_10_min_prices.index.tolist()

colors = ['blue', 'red', 'green', 'purple', 'orange', 'yellow', 'pink', 'gray', 'brown', 'teal']

fig = go.Figure(go.Bar(
    x=top_10_states,
    y=top_10_min_prices,
    marker=dict(color=colors),
    text=top_10_min_prices,
    textposition='inside',
))

fig.update_layout(
    title='Top 10 States by Min Price',
    xaxis_title='State',
    yaxis_title='Min Price',
)
fig.show()
```

Top 10 States by Min Price



In []: