In [1]:
```python
import pandas as pd
import numpy as np

import plotly.express as px
import matplotlib.pyplot as plt
```

In [2]:
```python
#Reading the csv file
df_tarkari = pd.read_csv('kalimati_tarkari_dataset_cleaned.csv')
```

In [3]:
```python
#Checking the units of tarkari
df_tarkari['Unit'].value_counts()
```

Out[3]:
```
Kg      184408
KG        6889
1 Pc      3120
Doz       2744
Name: Unit, dtype: int64
```

In [4]:
```python
#Replacing KG with Kg
df_tarkari['Unit'].replace('KG','Kg',inplace = True)
```

In [5]:
```python
df_tarkari['Unit'].value_counts()
```

Out[5]:
```
Kg      191297
1 Pc      3120
Doz       2744
Name: Unit, dtype: int64
```

In [6]:
```python
#Taking tarkari with units Kg
df_tarkari = df_tarkari[df_tarkari['Unit']=='Kg']
```

In [7]:
```python
df_tarkari['Unit'].value_counts()
```

Out[7]:
```
Kg    191297
Name: Unit, dtype: int64
```

In [8]:
```python
df_tarkari
```

Out[8]:

	SN	Commodity	Date	Unit	Minimum	Maximum	Average
0	0	Tomato Big(Nepali)	2013-06-16	Kg	35.0	40.0	37.5
1	1	Tomato Small(Local)	2013-06-16	Kg	26.0	32.0	29.0
2	2	Potato Red	2013-06-16	Kg	20.0	21.0	20.5
3	3	Potato White	2013-06-16	Kg	15.0	16.0	15.5
4	4	Onion Dry (Indian)	2013-06-16	Kg	28.0	30.0	29.0
...	...	...	...	...	...	...	...
197156	197156	Garlic Dry Nepali	2021-05-13	Kg	100.0	120.0	110.0
197157	197157	Fish Fresh(Rahu)	2021-05-13	Kg	270.0	280.0	275.0
197158	197158	Fish Fresh(Bachuwa)	2021-05-13	Kg	225.0	235.0	230.0
197159	197159	Fish Fresh(Chhadi)	2021-05-13	Kg	220.0	230.0	225.0
197160	197160	Fish Fresh(Mungari)	2021-05-13	Kg	240.0	250.0	245.0

191297 rows × 7 columns

In [9]:
```python
df_valuecounts = df_tarkari['Commodity'].value_counts().head(10).reset_index()
```

In [10]:
```python
df_valuecounts.rename(columns = {'index': 'Commodity', 'Commodity': 'Value Counts'},inplace = True)
```

In [11]:
```python
df_valuecounts
```

Out[11]:

	Commodity	Value Counts
0	Ginger	2751
1	Cauli Local	2750
2	Cabbage(Local)	2749
3	Chilli Dry	2748
4	Raddish White(Local)	2747
5	Potato Red	2746
6	Bamboo Shoot	2744
7	Brd Leaf Mustard	2742
8	Onion Dry (Indian)	2742
9	Coriander Green	2741

In [12]:
```python
df_tarkari.sort_values(by = 'Average')
```

Out[12]:

	SN	Commodity	Date	Unit	Minimum	Maximum	Average
27521	27521	Christophine	2014-10-10	Kg	4.0	6.0	5.0
29230	29230	Christophine	2014-11-07	Kg	4.0	6.0	5.0
41347	41347	Squash(Long)	2015-05-02	Kg	5.0	6.0	5.5
27591	27591	Christophine	2014-10-12	Kg	5.0	6.0	5.5
41690	41690	Cabbage(Local)	2015-05-07	Kg	5.0	6.0	5.5
...	...	...	...	...	...	...	...
113258	113258	Lime	2018-03-22	Kg	1450.0	1500.0	1475.0
113188	113188	Lime	2018-03-21	Kg	1450.0	1500.0	1475.0
114449	114449	Lime	2018-04-08	Kg	1450.0	1500.0	1475.0
114517	114517	Lime	2018-04-09	Kg	1450.0	1500.0	1475.0
177595	177595	Asparagus	2020-10-22	Kg	1800.0	2000.0	1900.0

191297 rows × 7 columns

In [13]:
```python
df_tarkari_increase = df_tarkari.groupby('Commodity')
```

In [14]:
```python
df_tarkari_increase.get_group('Lime')
```

Out[14]:

	SN	Commodity	Date	Unit	Minimum	Maximum	Average
52	52	Lime	2013-06-16	Kg	200.0	400.0	300.0
125	125	Lime	2013-06-17	Kg	200.0	400.0	300.0
198	198	Lime	2013-06-18	Kg	200.0	400.0	300.0
271	271	Lime	2013-06-19	Kg	200.0	400.0	300.0
345	345	Lime	2013-06-20	Kg	200.0	400.0	300.0
...	...	...	...	...	...	...	...
196766	196766	Lime	2021-05-09	Kg	200.0	220.0	210.0
196857	196857	Lime	2021-05-10	Kg	200.0	220.0	210.0
196946	196946	Lime	2021-05-11	Kg	200.0	220.0	210.0
197039	197039	Lime	2021-05-12	Kg	200.0	220.0	210.0
197129	197129	Lime	2021-05-13	Kg	200.0	220.0	210.0

2725 rows × 7 columns

In [15]:
```python
#Price difference
df_tarkari_change = df_tarkari_increase.last()['Average'] - df_tarkari_increase.first()['Average']
```

In [16]:
```python
df_tarkari_change.reset_index().sort_values(by = 'Average')
```

Out[16]:

	Commodity	Average
26	Chilli Green(Akbare)	-200.0
89	Pear(Local)	-112.5
66	Litchi(Local)	-110.0
80	Onion Dry (Chinese)	-91.5
64	Lime	-90.0
...	...	...
73	Mombin	153.0
1	Apple(Jholey)	155.0
109	Strawberry	285.0
60	Kiwi	305.0
3	Asparagus	900.0

128 rows × 2 columns

In [17]:
```python
pd.options.mode.chained_assignment = None
```

In [18]:
```python
df_tarkari_increase.get_group('Apple(Fuji)')
```

Out[18]:

	SN	Commodity	Date	Unit	Minimum	Maximum	Average
141053	141053	Apple(Fuji)	2019-05-31	Kg	250.0	260.0	255.0
141130	141130	Apple(Fuji)	2019-06-01	Kg	250.0	260.0	255.0
141203	141203	Apple(Fuji)	2019-06-02	Kg	250.0	260.0	255.0
141278	141278	Apple(Fuji)	2019-06-03	Kg	250.0	260.0	255.0
141358	141358	Apple(Fuji)	2019-06-04	Kg	340.0	350.0	345.0
...	...	...	...	...	...	...	...
196587	196587	Apple(Fuji)	2021-05-07	Kg	280.0	300.0	290.0
196764	196764	Apple(Fuji)	2021-05-09	Kg	250.0	260.0	255.0
196855	196855	Apple(Fuji)	2021-05-10	Kg	260.0	280.0	270.0
196944	196944	Apple(Fuji)	2021-05-11	Kg	250.0	280.0	265.0
197127	197127	Apple(Fuji)	2021-05-13	Kg	250.0	270.0	260.0

395 rows × 7 columns

In [19]:
```python
#Top 10 price decrease (-ve)
df_tarkari_decrease = df_tarkari_change.reset_index().sort_values(by = 'Average').head(10)
```

In [20]:
```python
#Top 10 price increase (+ve)
df_tarkari_badyo = df_tarkari_change.reset_index().sort_values(by = 'Average').tail(10)
```

In [21]:
```python
#Absolute value of Top 10 price decrease
df_tarkari_decrease['Average'] = np.abs(df_tarkari_decrease['Average'])
```

In [22]:
```python
df_tarkari_decrease
```

Out[22]:

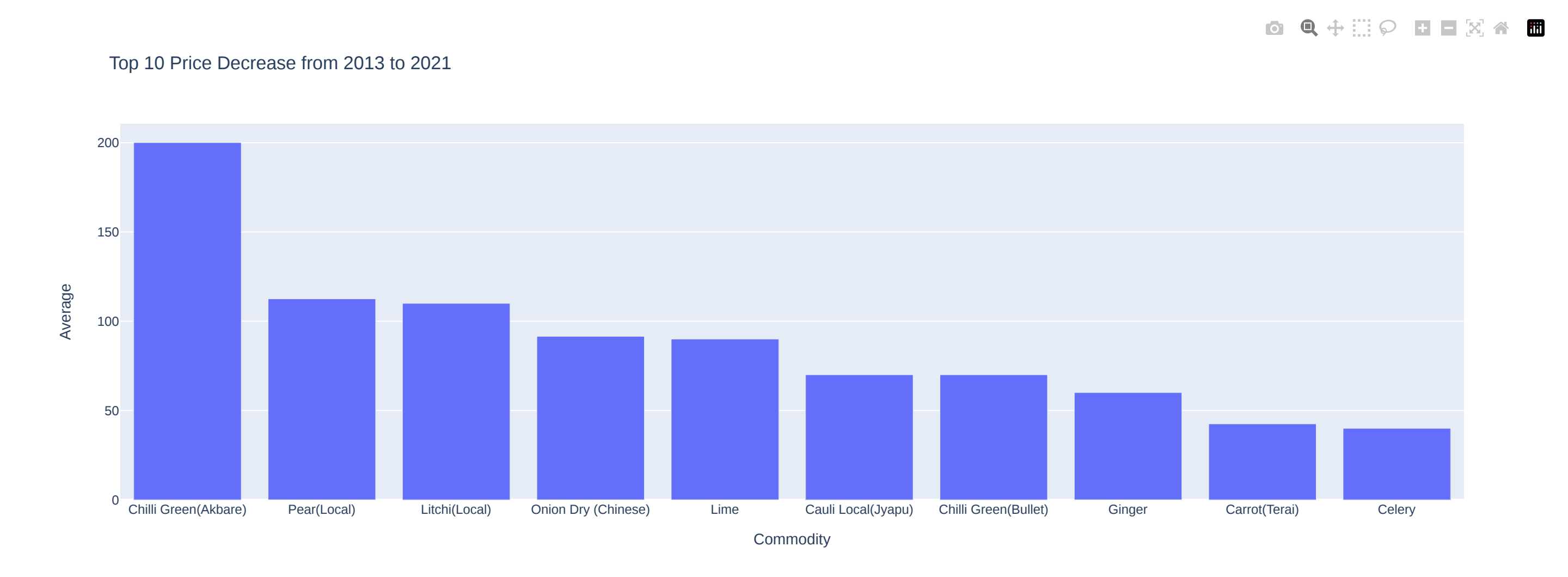
	Commodity	Average
26	Chilli Green(Akbare)	200.0
89	Pear(Local)	112.5
66	Litchi(Local)	110.0
80	Onion Dry (Chinese)	91.5
64	Lime	90.0
21	Cauli Local(Jyapu)	70.0
27	Chilli Green(Bullet)	70.0
52	Ginger	60.0
19	Carrot(Terai)	42.5
23	Celery	40.0

In [23]:
```python
df_tarkari_badyo
```

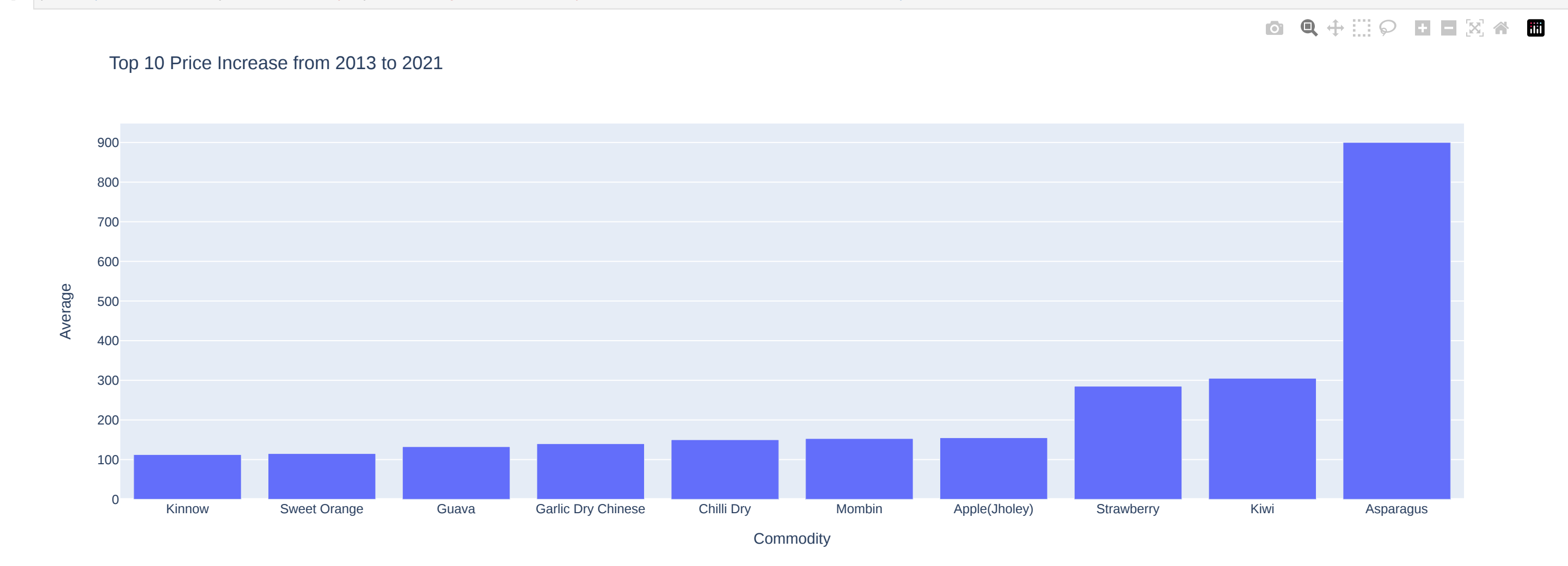
Out[23]:

	Commodity	Average
59	Kinnow	112.5
112	Sweet Orange	115.0
56	Guava	132.5
49	Garlic Dry Chinese	140.0
24	Chilli Dry	150.0
73	Mombin	153.0
1	Apple(Jholey)	155.0
109	Strawberry	285.0
60	Kiwi	305.0
3	Asparagus	900.0

In [24]:
```python
px.bar(df_tarkari_decrease,x='Commodity',y = 'Average',title = 'Top 10 Price Decrease from 2013 to 2021')
```



In [25]:
```python
px.bar(df_tarkari_badyo,x='Commodity',y = 'Average',title = 'Top 10 Price Increase from 2013 to 2021')
```



In [26]:
```python
#Taking the highest price of tarkari
df_tarkari_maximum = df_tarkari.groupby(['Commodity','Date'])['Average'].max().reset_index()
```

In [27]:
```python
df_tarkari_maximum
```

Out[27]:

	Commodity	Date	Average
0	Apple(Fuji)	2019-05-31	255.0
1	Apple(Fuji)	2019-06-01	255.0
2	Apple(Fuji)	2019-06-02	255.0
3	Apple(Fuji)	2019-06-03	255.0
4	Apple(Fuji)	2019-06-04	345.0
...	...	...	...
191292	Yam	2021-04-16	75.0
191293	Yam	2021-04-17	75.0
191294	Yam	2021-04-18	75.0
191295	Yam	2021-04-19	75.0
191296	Yam	2021-04-20	75.0

191297 rows × 3 columns

In [28]:
```python
#df_tarkari_maximum = df_tarkari_maximum.groupby('Commodity')['Average'].max().reset_index()
```

In [29]:
```python
#df_tarkari_maximum
```

In [30]:
```python
df_date_included = df_tarkari_maximum.sort_values(by = 'Average',ascending = False)
```

In [31]:
```python
df_date_included.groupby('Commodity').nth(0)
```

Out[31]:

	Date	Average
Commodity		
Apple(Fuji)	2019-06-04	345.0
Apple(Jholey)	2021-05-08	290.0
Arum	2019-07-26	67.5
Asparagus	2020-10-22	1900.0
Bakula	2021-02-07	125.0
...	...	...
Turnip	2021-02-09	85.0
Turnip A	2019-10-29	145.0
Water Melon(Dotted)	2019-07-08	52.5
Water Melon(Green)	2020-11-16	175.0
Yam	2015-04-23	125.0

128 rows × 2 columns

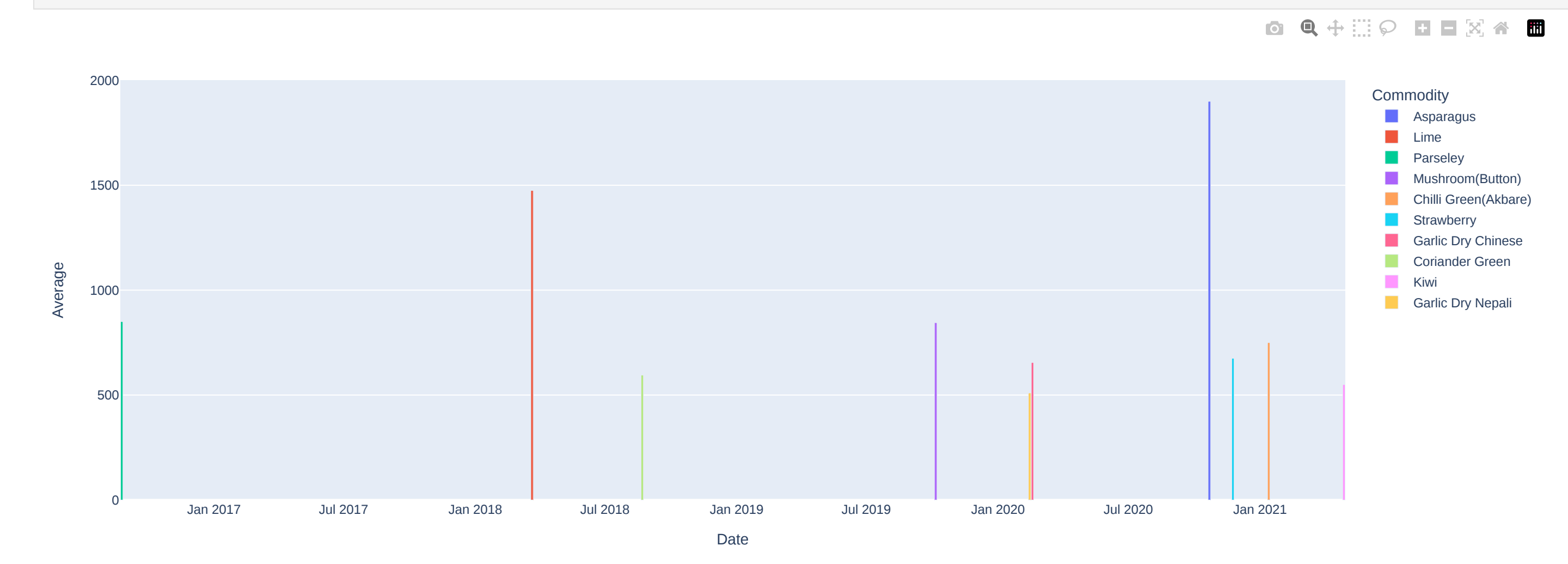
In [32]:
```python
#Maximum price of tarkari on which date and how much
df_withdate_max = df_date_included.groupby('Commodity').nth(0).reset_index().sort_values(by = 'Average',ascending = False).head(10)
```

In [33]:
```python
df_withdate_max
#includes date
```

Out[33]:

	Commodity	Date	Average
3	Asparagus	2020-10-22	1900.0
64	Lime	2018-03-21	1475.0
87	Parseley	2016-08-25	850.0
74	Mushroom(Button)	2019-10-06	845.0
26	Chilli Green(Akbare)	2021-01-13	750.0
109	Strawberry	2020-11-24	675.0
49	Garlic Dry Chinese	2020-02-18	655.0
32	Coriander Green	2018-08-22	595.0
60	Kiwi	2021-04-28	550.0
50	Garlic Dry Nepali	2020-02-14	510.0

In [34]:
```python
#Graph of maximum price of tarkari on which date and how much
px.bar(df_withdate_max,x = 'Date',y = 'Average',color = 'Commodity')
```



In [35]:
```python
import plotly.express as px
```

In [36]:
```python
df_bubble = df_tarkari.loc[df_tarkari['Commodity'].isin(df_withdate_max['Commodity'])]
```

In [37]:
```python
df_bubble['date_year'] = pd.to_datetime(df_bubble['Date']).dt.year
```

In [38]:
```python
df_bubble[df_bubble['Commodity'] == 'Asparagus'].sort_values(by = 'Average')
```

Out[38]:

	SN	Commodity	Date	Unit	Minimum	Maximum	Average	date_year
24604	24604	Asparagus	2014-08-17	Kg	60.0	70.0	65.0	2014
22057	22057	Asparagus	2014-07-09	Kg	100.0	110.0	105.0	2014
180	180	Asparagus	2013-06-18	Kg	100.0	120.0	110.0	2013
327	327	Asparagus	2013-06-20	Kg	100.0	120.0	110.0	2013
401	401	Asparagus	2013-06-21	Kg	100.0	120.0	110.0	2013
...	...	...	...	...	...	...	...	...
191329	191329	Asparagus	2021-03-16	Kg	950.0	1000.0	975.0	2021
197114	197114	Asparagus	2021-05-13	Kg	1000.0	1050.0	1025.0	2021
196931	196931	Asparagus	2021-05-11	Kg	1000.0	1200.0	1100.0	2021
197023	197023	Asparagus	2021-05-12	Kg	1000.0	1200.0	1100.0	2021
177595	177595	Asparagus	2020-10-22	Kg	1800.0	2000.0	1900.0	2020

1055 rows × 8 columns

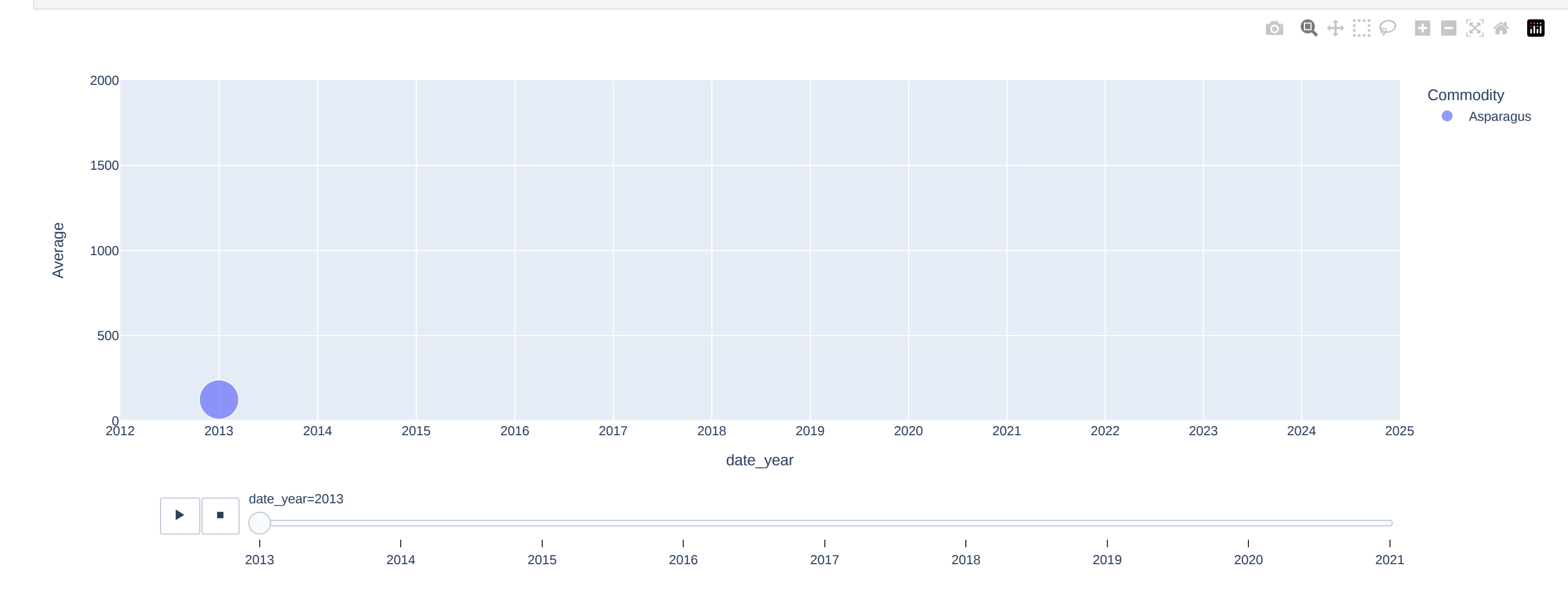
In [39]:
```python
df_bubble
```

Out[39]:

	SN	Commodity	Date	Unit	Minimum	Maximum	Average	date_year
34	34	Asparagus	2013-06-16	Kg	100.0	150.0	125.0	2013
42	42	Parseley	2013-06-16	Kg	180.0	190.0	185.0	2013
52	52	Lime	2013-06-16	Kg	200.0	400.0	300.0	2013
68	68	Coriander Green	2013-06-16	Kg	90.0	110.0	100.0	2013
69	69	Garlic Dry Chinese	2013-06-16	Kg	100.0	110.0	105.0	2013
...	...	...	...	...	...	...	...	...
197145	197145	Strawberry	2021-05-13	Kg	450.0	500.0	475.0	2021
197151	197151	Chilli Green(Akbare)	2021-05-13	Kg	140.0	150.0	145.0	2021
197154	197154	Coriander Green	2021-05-13	Kg	80.0	90.0	85.0	2021
197155	197155	Garlic Dry Chinese	2021-05-13	Kg	240.0	250.0	245.0	2021
197156	197156	Garlic Dry Nepali	2021-05-13	Kg	100.0	120.0	110.0	2021

15803 rows × 8 columns

In [40]:
```python
#Showing the change in Asparagus price from 2013 to 2021
px.scatter(df_bubble[df_bubble['Commodity'] == 'Asparagus'], x="date_year", y="Average",
           size="Average", color="Commodity", hover_name="Commodity",animation_frame="date_year", animation_group="Commodity",range_x=[2012,2025],range_y=[0,2000],log_x=True,size_ma
```



In [ ]: