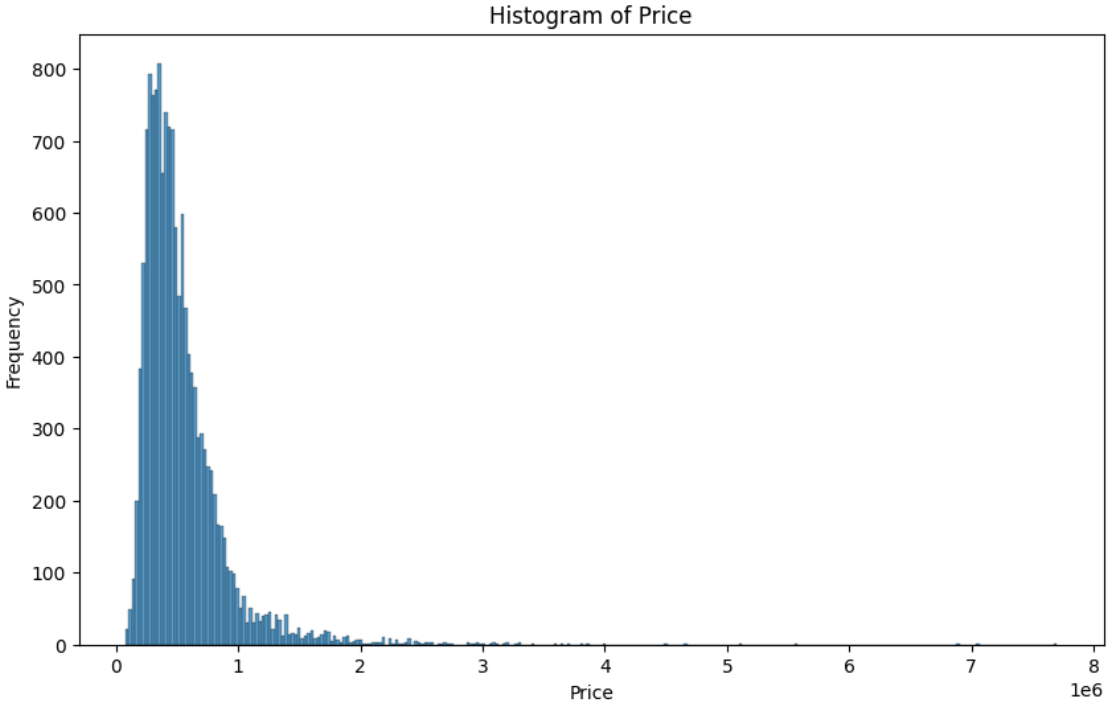
# untitled1

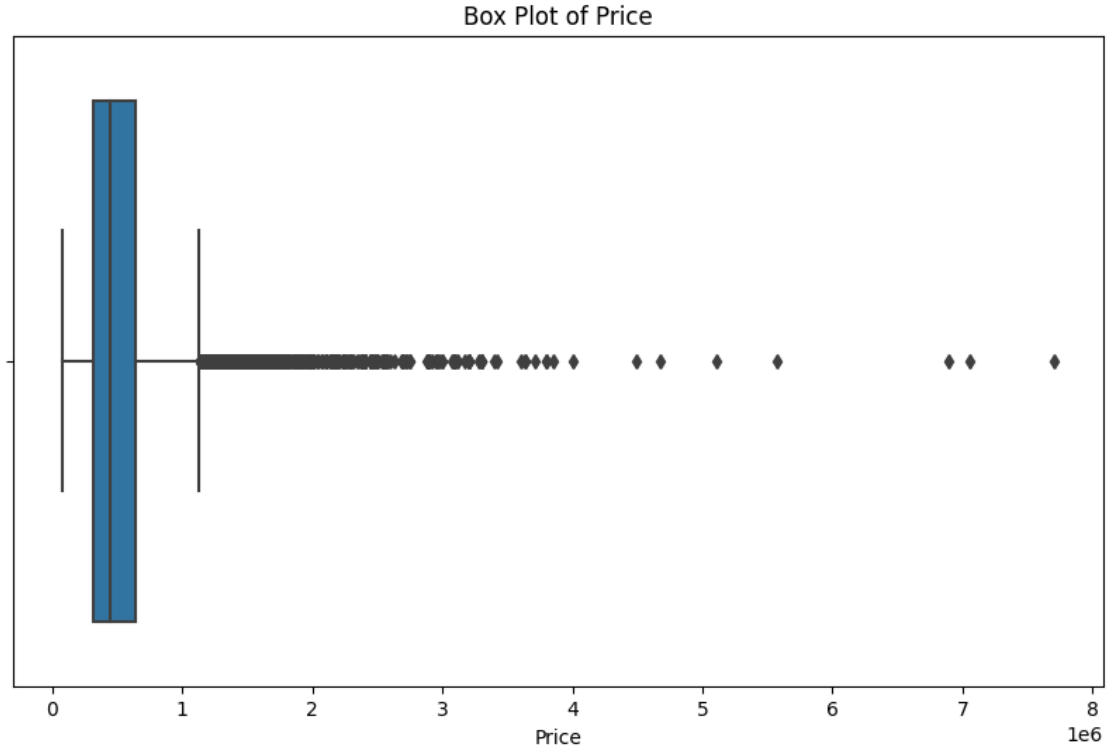
September 19, 2023

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

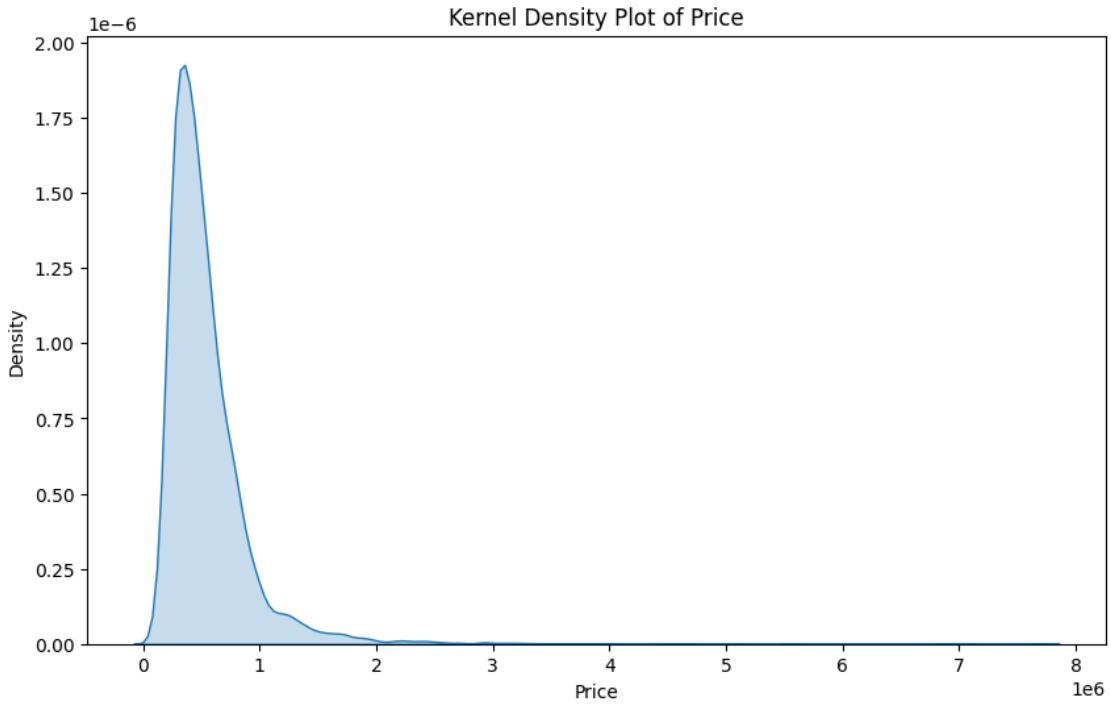
[5]:
```
file_path = "/content/House Price India.csv"
df = pd.read_csv(file_path)
```

[6]:
```
# Univariate Analysis
# Histogram
plt.figure(figsize=(10, 6))
sns.histplot(df['Price'])
plt.title('Histogram of Price')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()

# Box Plot (Box-and-Whisker Plot)
plt.figure(figsize=(10, 6))
sns.boxplot(x=df['Price'])
plt.title('Box Plot of Price')
plt.xlabel('Price')
plt.show()

# Kernel Density Plot
plt.figure(figsize=(10, 6))
sns.kdeplot(df['Price'], shade=True)
plt.title('Kernel Density Plot of Price')
plt.xlabel('Price')
plt.show()
```

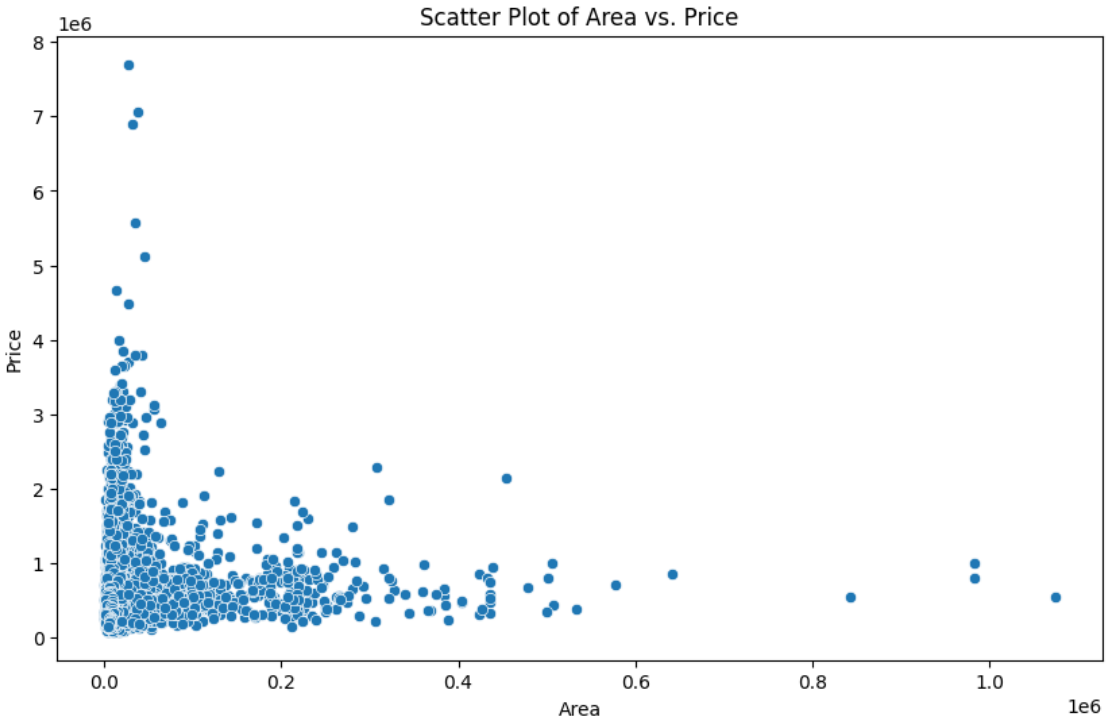

Histogram of Price
Frequency
Price
1e6
0
100
200
300
400
500
600
700
800
0
1
2
3
4
5
6
7
8

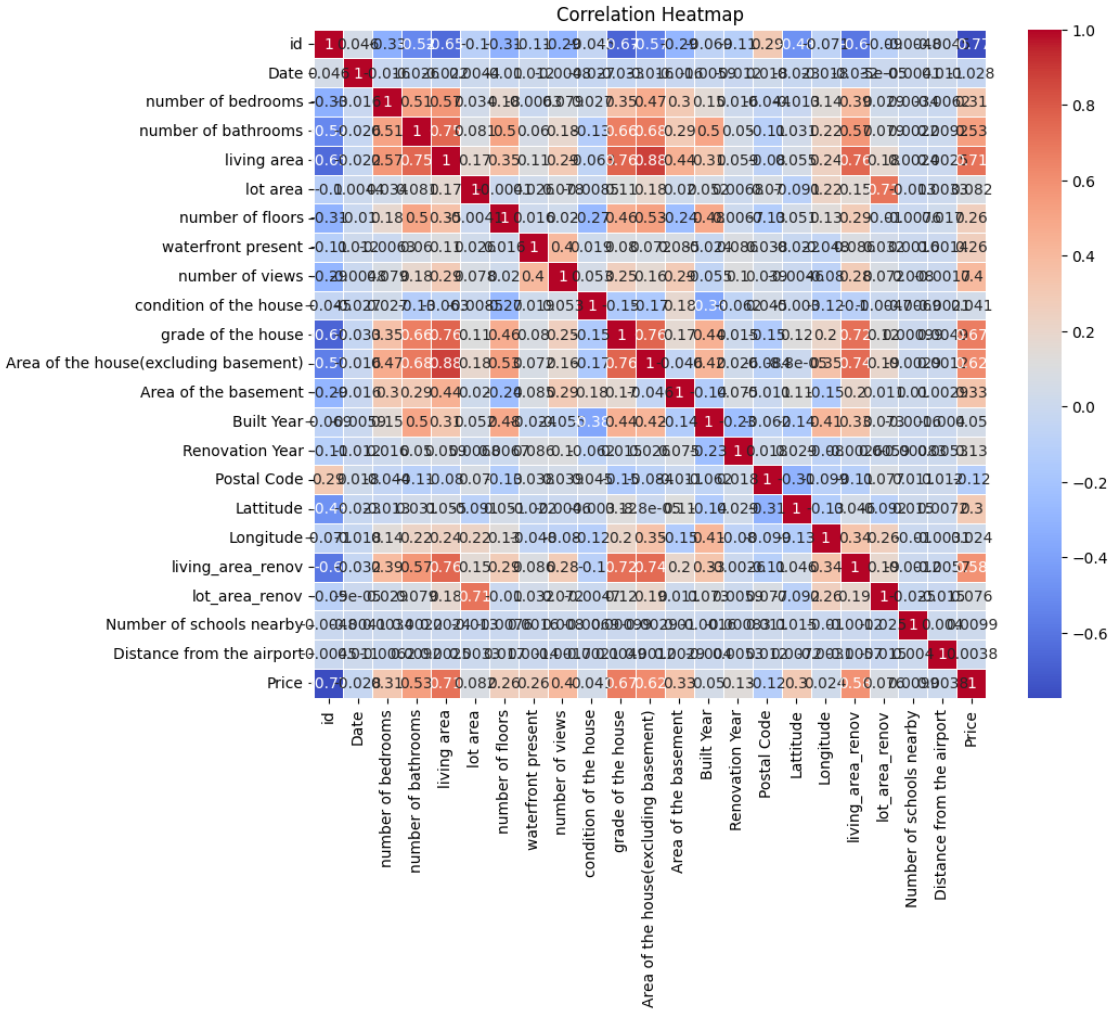

Box Plot of Price
Price
1e6
0
1
2
3
4
5
6
7
8

```
<ipython-input-6-4c6ca09cffb4>:19: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df['Price'], shade=True)
```

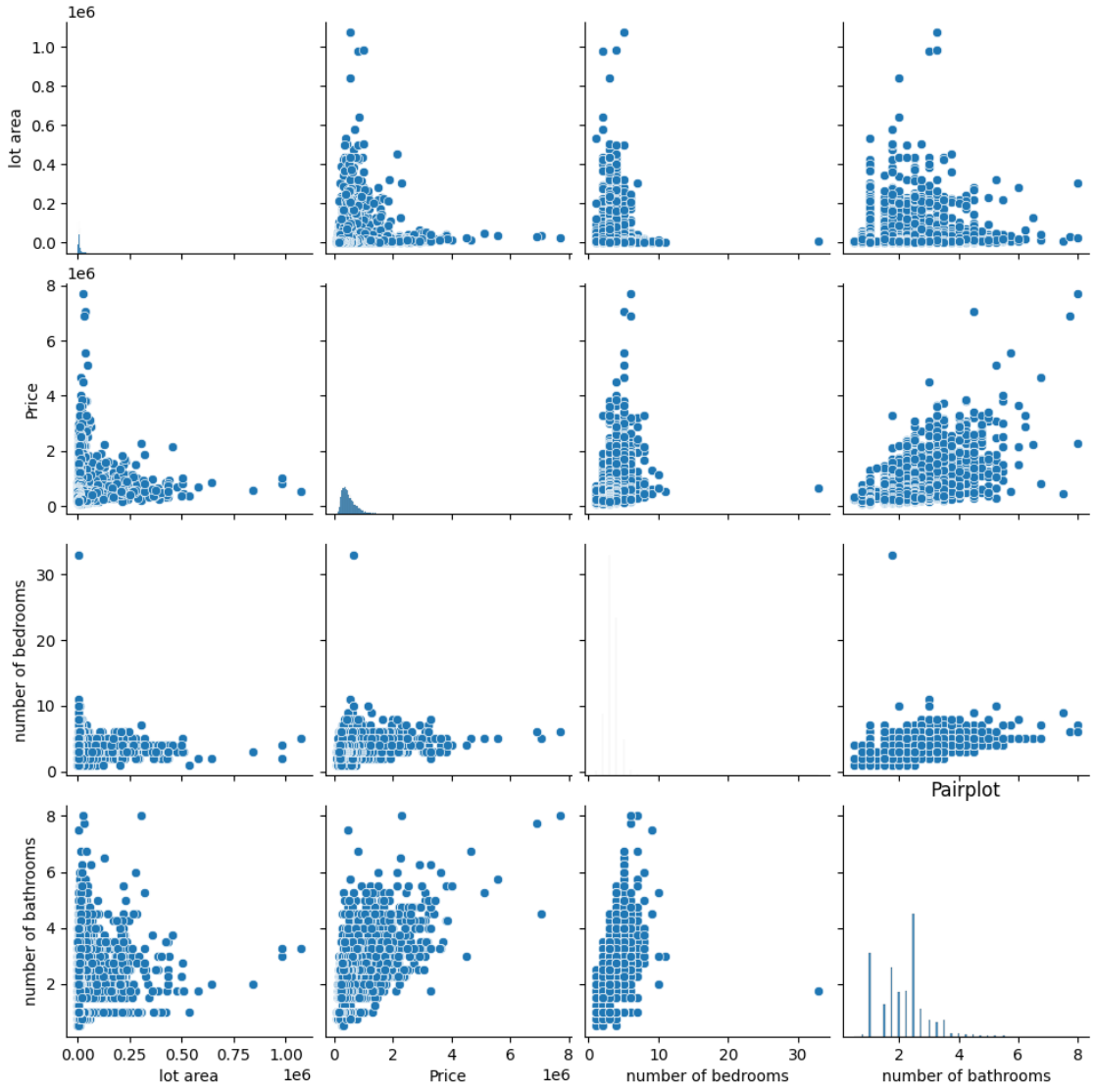



[18]:
```python
# Bivariate Analysis
# Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df['lot area'], y=df['Price'])
plt.title('Scatter Plot of Area vs. Price')
plt.xlabel('Area')
plt.ylabel('Price')
plt.show()

# Heatmap
correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=.5)
plt.title('Correlation Heatmap')
plt.show()
```


Scatter Plot of Area vs. Price
1e6
8
7
6
5
4
3
2
1
0
Price
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Area




[16]:
```
# Pairplot
sns.pairplot(df[['lot area', 'Price', 'number of bedrooms', 'number of␣
 ↪bathrooms']])
plt.title('Pairplot')
plt.show()
```




[21]:
```
# Multivariate Analysis
# Parallel Coordinates Plot
fig = px.parallel_coordinates(df, dimensions=['lot area', 'Price', 'number of␣
 ↪bedrooms', 'number of bathrooms'], color="Price")
fig.show()

# 3D Scatter Plot
fig = px.scatter_3d(df, x='lot area', y='number of bedrooms', z='Price',␣
 ↪color='Price')
fig.show()

# Radial Plot (Polar plot)
fig = px.line_polar(df, r='Price', theta='lot area', line_close=True)
```

```
fig.update_traces(fill='toself')
fig.show()
```

[23]:
```
#Descriptive Statistics
descriptive_stats = df.describe()
print(descriptive_stats)
```

```
                 id          Date  number of bedrooms  number of bathrooms  \
count  1.462000e+04  14620.000000        14620.000000         14620.000000
mean   6.762821e+09  42604.538646            3.379343             2.129583
std    6.237575e+03     67.347991            0.938719             0.769934
min    6.762810e+09  42491.000000            1.000000             0.500000
25%    6.762815e+09  42546.000000            3.000000             1.750000
50%    6.762821e+09  42600.000000            3.000000             2.250000
75%    6.762826e+09  42662.000000            4.000000             2.500000
max    6.762832e+09  42734.000000           33.000000             8.000000

        living area      lot area  number of floors  waterfront present  \
count  14620.000000  1.462000e+04      14620.000000        14620.000000
mean    2098.262996  1.509328e+04          1.502360            0.007661
std      928.275721  3.791962e+04          0.540239            0.087193
min      370.000000  5.200000e+02          1.000000            0.000000
25%     1440.000000  5.010750e+03          1.000000            0.000000
50%     1930.000000  7.620000e+03          1.500000            0.000000
75%     2570.000000  1.080000e+04          2.000000            0.000000
max    13540.000000  1.074218e+06          3.500000            1.000000

       number of views  condition of the house  …    Built Year  \
count     14620.000000            14620.000000  …  14620.000000
mean          0.233105                3.430506  …   1970.926402
std           0.766259                0.664151  …     29.493625
min           0.000000                1.000000  …   1900.000000
25%           0.000000                3.000000  …   1951.000000
50%           0.000000                3.000000  …   1975.000000
75%           0.000000                4.000000  …   1997.000000
max           4.000000                5.000000  …   2015.000000

       Renovation Year    Postal Code     Lattitude     Longitude  \
count     14620.000000   14620.000000  14620.000000  14620.000000
mean         90.924008  122033.062244     52.792848   -114.404007
std         416.216661      19.082418      0.137522      0.141326
min           0.000000  122003.000000     52.385900   -114.709000
25%           0.000000  122017.000000     52.707600   -114.519000
50%           0.000000  122032.000000     52.806400   -114.421000
75%           0.000000  122048.000000     52.908900   -114.315000
max        2015.000000  122072.000000     53.007600   -113.505000
```

```
       living_area_renov  lot_area_renov  Number of schools nearby  \
count       14620.000000    14620.000000              14620.000000
mean         1996.702257    12753.500068                  2.012244
std           691.093366    26058.414467                  0.817284
min           460.000000      651.000000                  1.000000
25%          1490.000000     5097.750000                  1.000000
50%          1850.000000     7620.000000                  2.000000
75%          2380.000000    10125.000000                  3.000000
max          6110.000000   560617.000000                  3.000000

       Distance from the airport         Price
count               14620.000000  1.462000e+04
mean                   64.950958  5.389322e+05
std                     8.936008  3.675324e+05
min                    50.000000  7.800000e+04
25%                    57.000000  3.200000e+05
50%                    65.000000  4.500000e+05
75%                    73.000000  6.450000e+05
max                    80.000000  7.700000e+06

[8 rows x 23 columns]
```

[27]:
```
df.isnull().any()
```

[27]:
```
id                                      False
Date                                    False
number of bedrooms                      False
number of bathrooms                     False
living area                             False
lot area                                False
number of floors                        False
waterfront present                      False
number of views                         False
condition of the house                  False
grade of the house                      False
Area of the house(excluding basement)   False
Area of the basement                    False
Built Year                              False
Renovation Year                         False
Postal Code                             False
Lattitude                               False
Longitude                               False
living_area_renov                       False
lot_area_renov                          False
Number of schools nearby                False
Distance from the airport               False
Price                                   False
```

```
dtype: bool
```

[8]:
```
print(df.columns)
```

```
Index(['id', 'Date', 'number of bedrooms', 'number of bathrooms',
       'living area', 'lot area', 'number of floors', 'waterfront present',
       'number of views', 'condition of the house', 'grade of the house',
       'Area of the house(excluding basement)', 'Area of the basement',
       'Built Year', 'Renovation Year', 'Postal Code', 'Lattitude',
       'Longitude', 'living_area_renov', 'lot_area_renov',
       'Number of schools nearby', 'Distance from the airport', 'Price'],
      dtype='object')
```