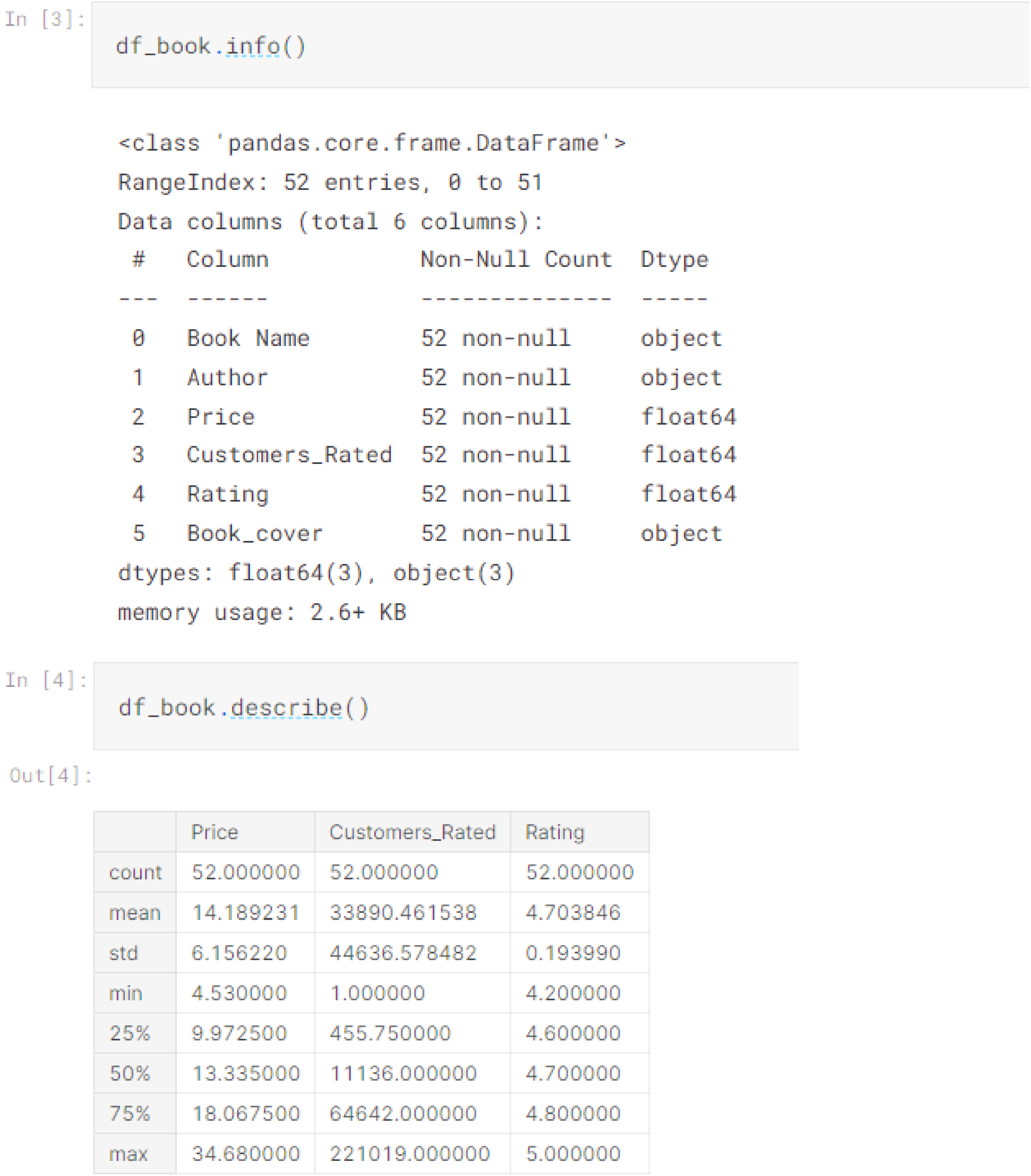
Barnes Noble, Ebay , Amazon books analysis



In [3]:

```
df_book.info()
```

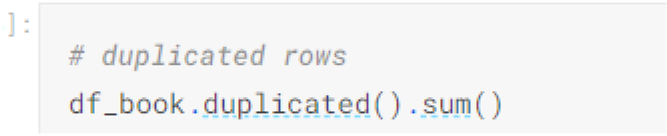
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Book Name         52 non-null     object 
 1   Author            52 non-null     object 
 2   Price             52 non-null     float64
 3   Customers_Rated   52 non-null     float64
 4   Rating            52 non-null     float64
 5   Book_cover        52 non-null     object 
dtypes: float64(3), object(3)
memory usage: 2.6+ KB
```

In [4]:

```
df_book.describe()
```

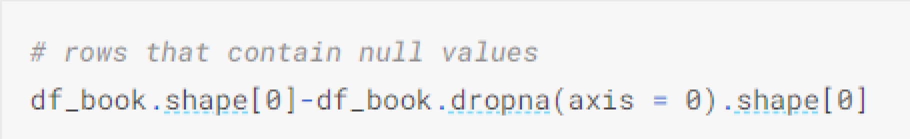
Out[4]:

	Price	Customers_Rated	Rating
count	52.000000	52.000000	52.000000
mean	14.189231	33890.461538	4.703846
std	6.156220	44636.578482	0.193990
min	4.530000	1.000000	4.200000
25%	9.972500	455.750000	4.600000
50%	13.335000	11136.000000	4.700000
75%	18.067500	64642.000000	4.800000
max	34.680000	221019.000000	5.000000



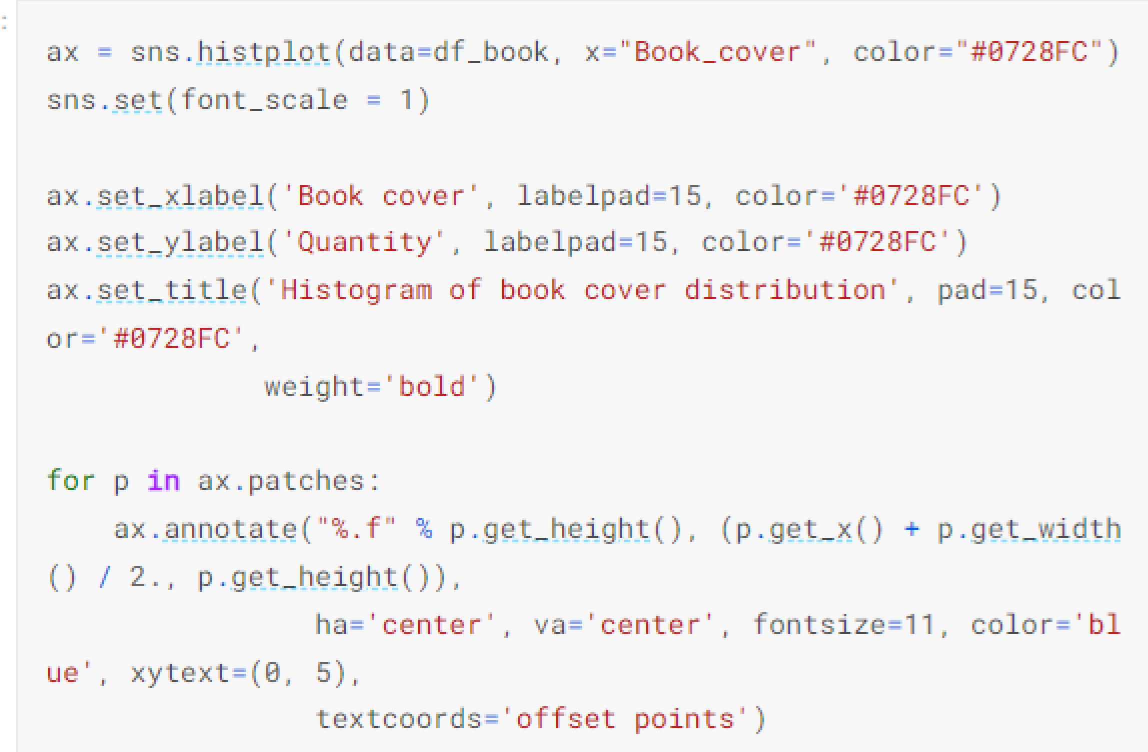
]:

```
# duplicated rows
df_book.duplicated().sum()
```



```
# rows that contain null values
df_book.shape[0]-df_book.dropna(axis = 0).shape[0]
```

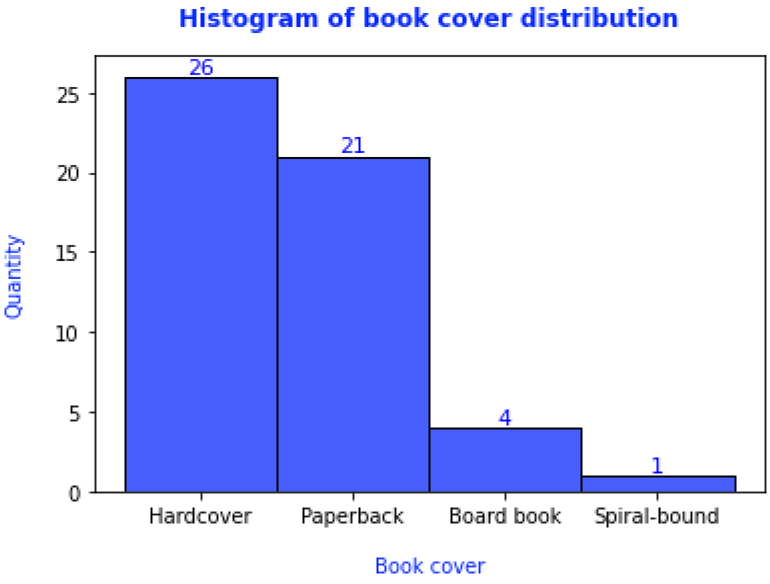
Book cover distribution



```
ax = sns.histplot(data=df_book, x="Book_cover", color="#0728FC")
sns.set(font_scale = 1)

ax.set_xlabel('Book cover', labelpad=15, color='#0728FC')
ax.set_ylabel('Quantity', labelpad=15, color='#0728FC')
ax.set_title('Histogram of book cover distribution', pad=15, color='#0728FC',
             weight='bold')

for p in ax.patches:
    ax.annotate("%.f" % p.get_height(), (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', fontsize=11, color='blue', xytext=(0, 5),
                textcoords='offset points')
```



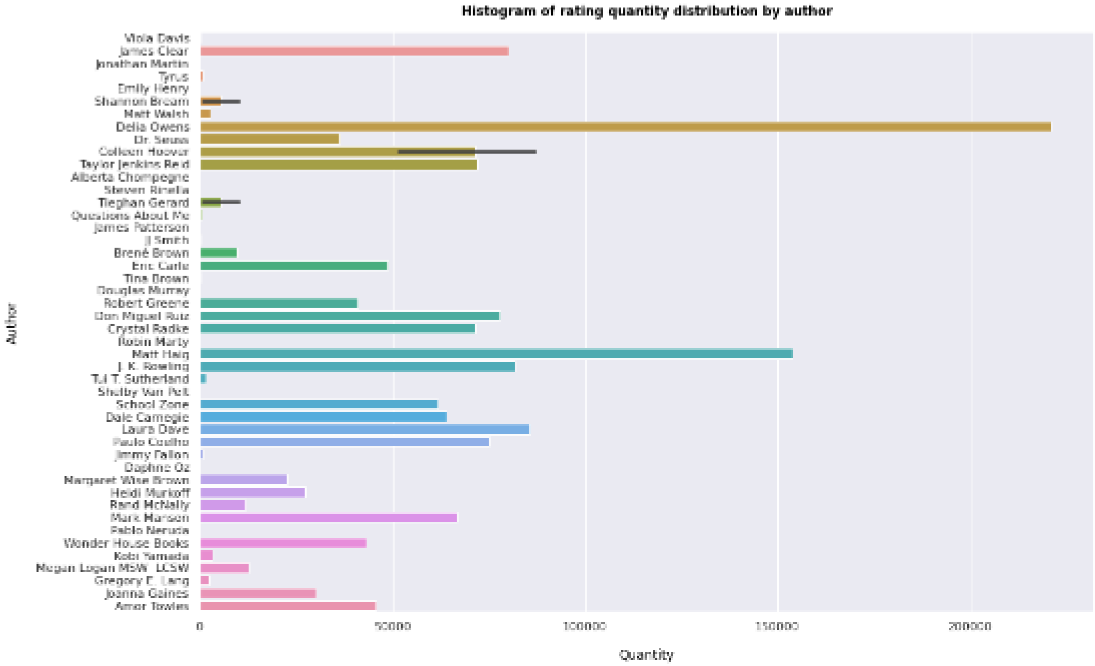
Number of ratings by Author

2]:

```
plt.figure(figsize=(15,10))
ax = sns.barplot(y="Author", x="Customers_Rated", data=df_book)

sns.set(font_scale = 1)

ax.set_xlabel('Quantity', labelpad=15, color='#000000')
ax.set_ylabel('Author', labelpad=15, color='#000000')
ax.set_title('Histogram of rating quantity distribution by autho
r', pad=15, color='#000000',
             weight='bold')
```

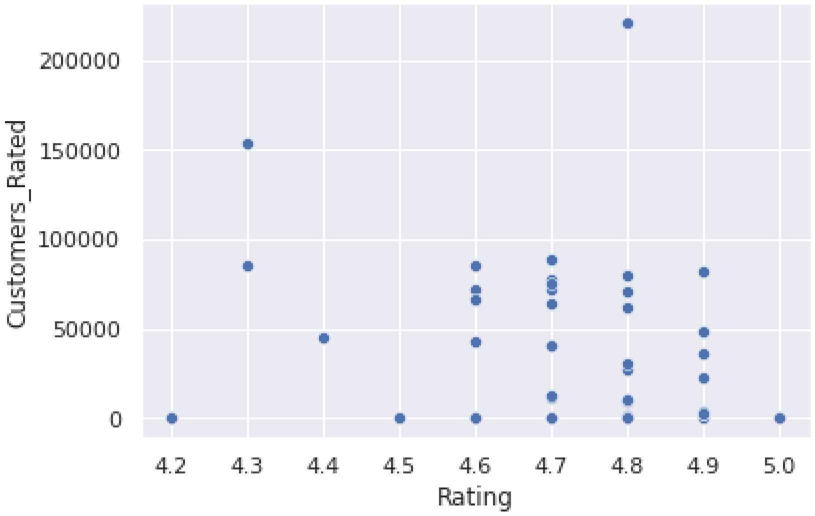


Histogram of rating quantity distribution by author

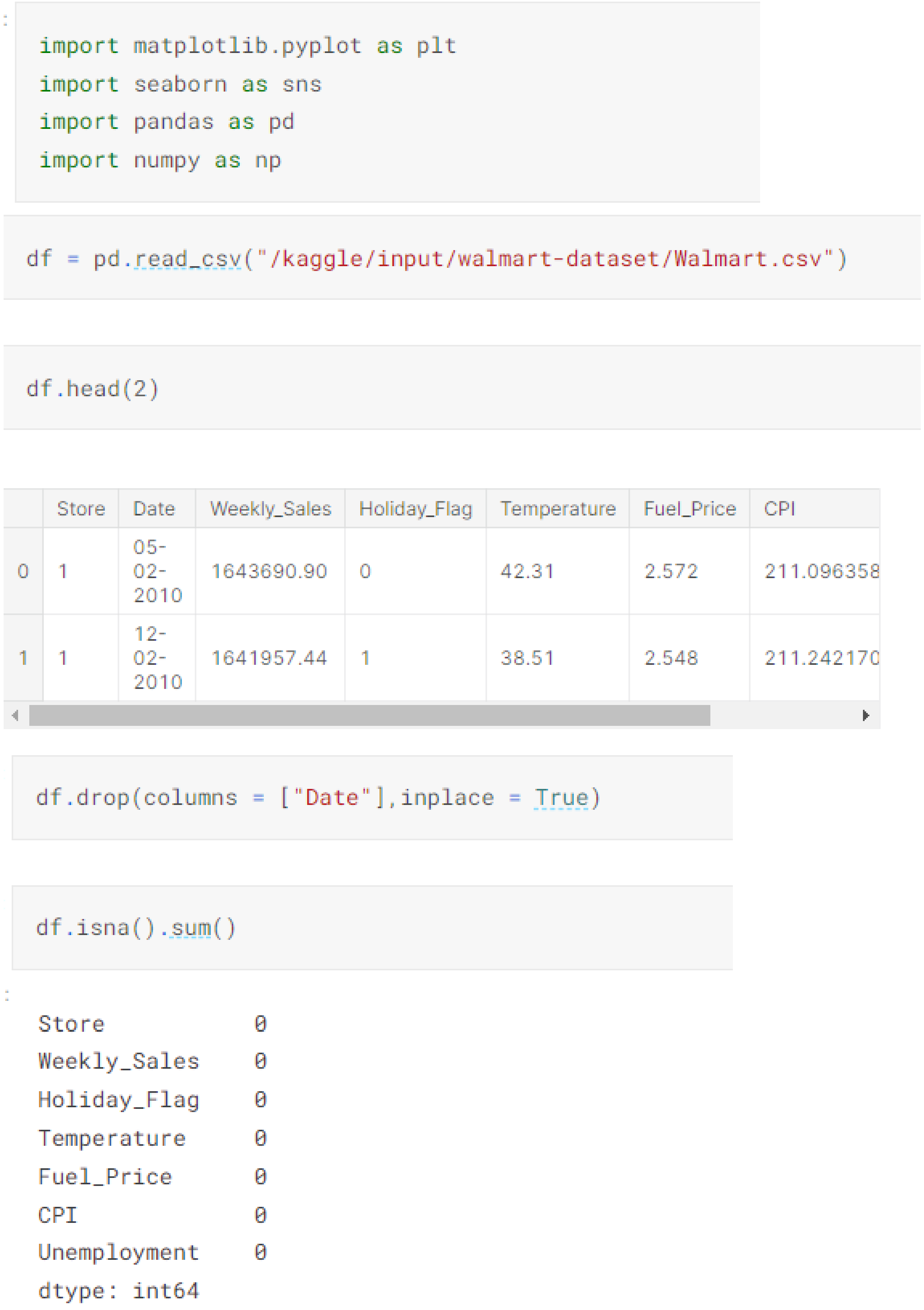
Rating and the quantity associated

```
sns.scatterplot(data=df_book, x="Rating", y="Customers_Rated")
```

```
<AxesSubplot:xlabel='Rating', ylabel='Customers_Rated'>
```



Walmart dataset analysis



```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
```

```
df = pd.read_csv("/kaggle/input/walmart-dataset/Walmart.csv")
```

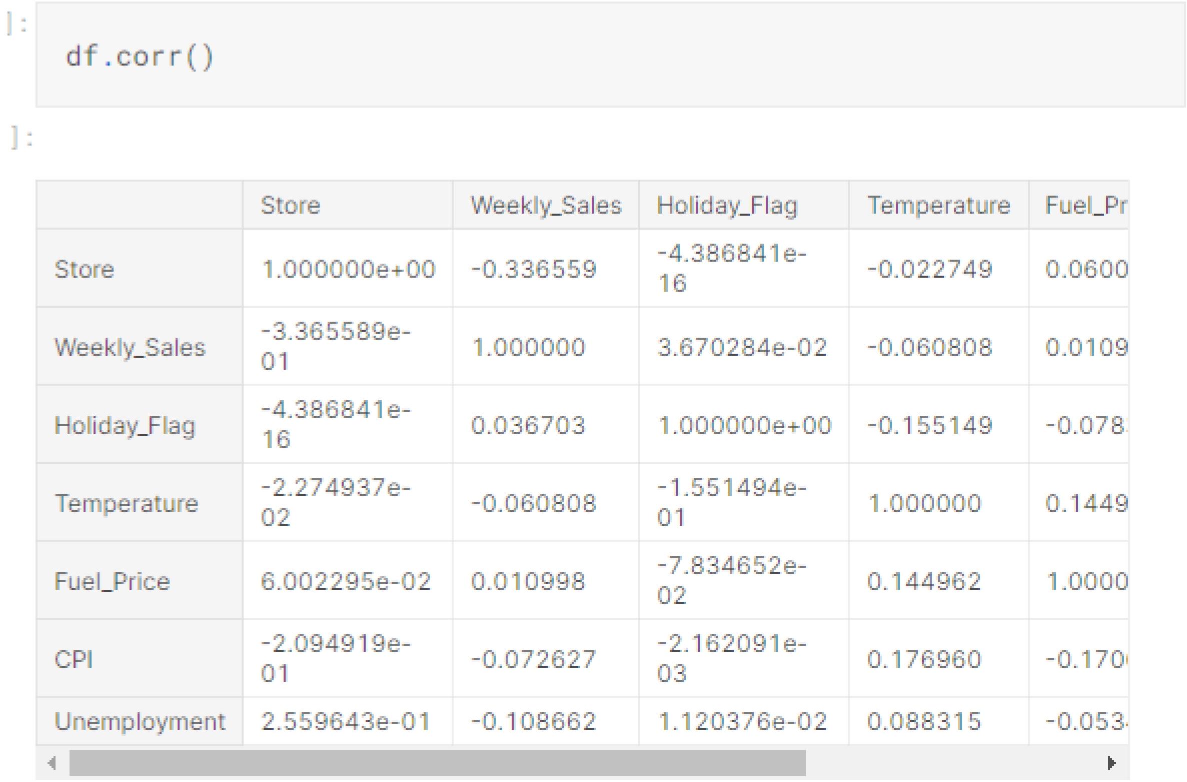
```
df.head(2)
```

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI
0	1	05-02-2010	1643690.90	0	42.31	2.572	211.096358
1	1	12-02-2010	1641957.44	1	38.51	2.548	211.242170

```
df.drop(columns = ["Date"],inplace = True)
```

```
df.isna().sum()
```

```
Store           0
Weekly_Sales    0
Holiday_Flag    0
Temperature     0
Fuel_Price      0
CPI             0
Unemployment    0
dtype: int64
```



]:

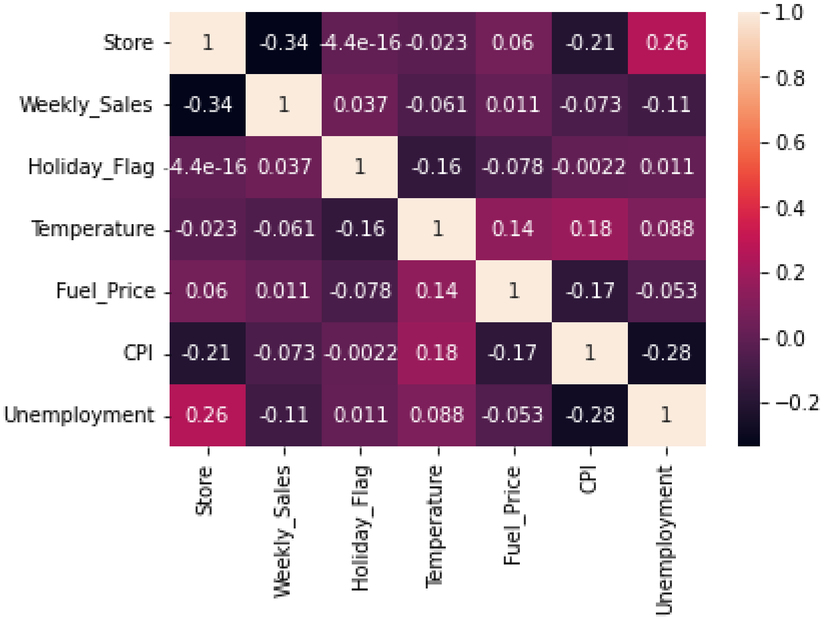
```
df.corr()
```

]:

	Store	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Pr
Store	1.000000e+00	-0.336559	-4.386841e-16	-0.022749	0.0600
Weekly_Sales	-3.365589e-01	1.000000	3.670284e-02	-0.060808	0.0109
Holiday_Flag	-4.386841e-16	0.036703	1.000000e+00	-0.155149	-0.078
Temperature	-2.274937e-02	-0.060808	-1.551494e-01	1.000000	0.1449
Fuel_Price	6.002295e-02	0.010998	-7.834652e-02	0.144962	1.0000
CPI	-2.094919e-01	-0.072627	-2.162091e-03	0.176960	-0.170
Unemployment	2.559643e-01	-0.108662	1.120376e-02	0.088315	-0.053

```
sns.heatmap(df.corr(), annot = True)
```

```
<AxesSubplot:>
```


Store
Weekly_Sales
Holiday_Flag
Temperature
Fuel_Price
CPI
Unemployment
1 -0.34 -4.4e-16 -0.023 0.06 -0.21 0.26
-4.4e-16 0.037 1 -0.16 -0.078 -0.0022 0.011