Web-Series Analytics - by **ARMAN ALI**

Codes:

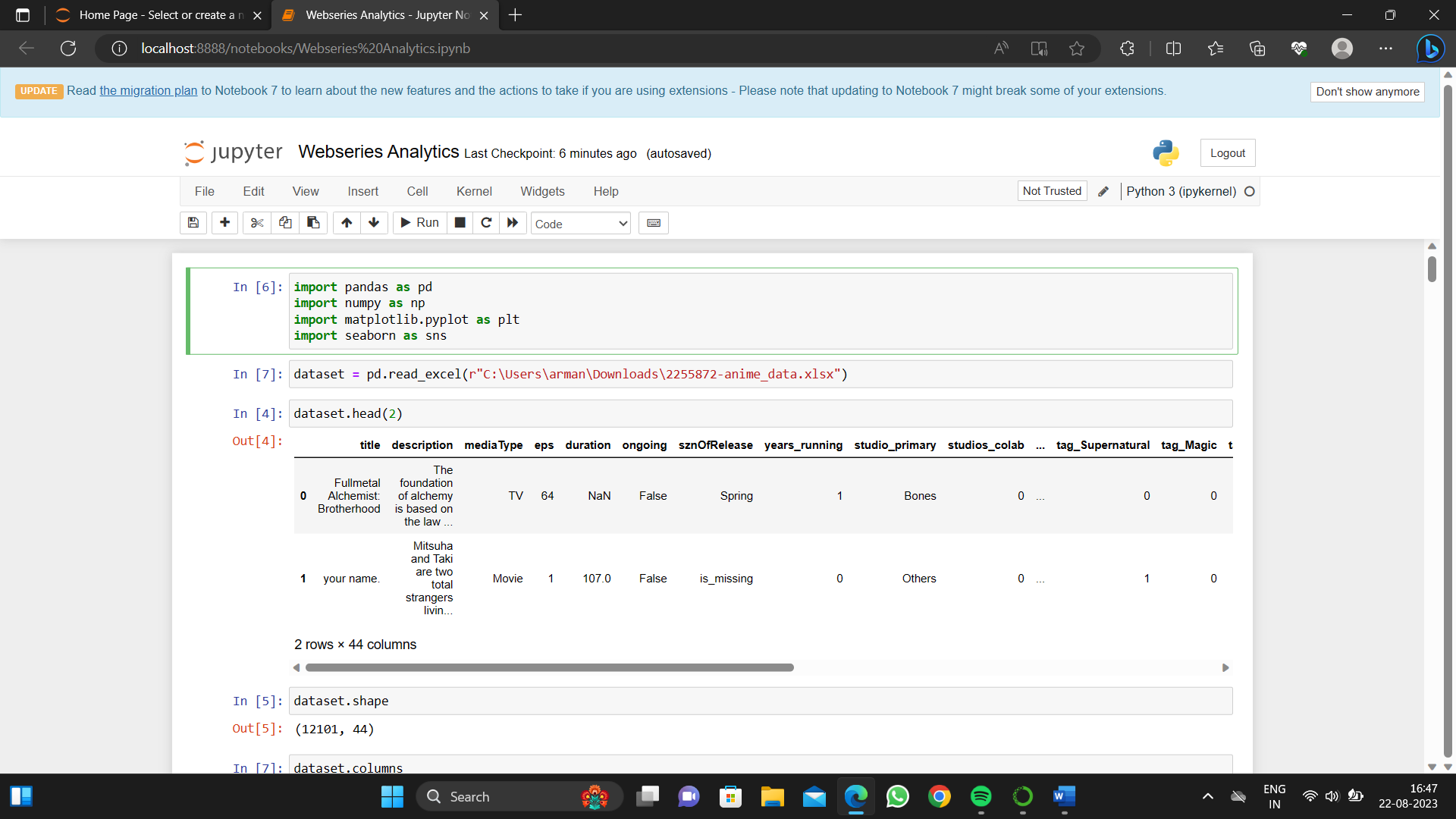
* import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

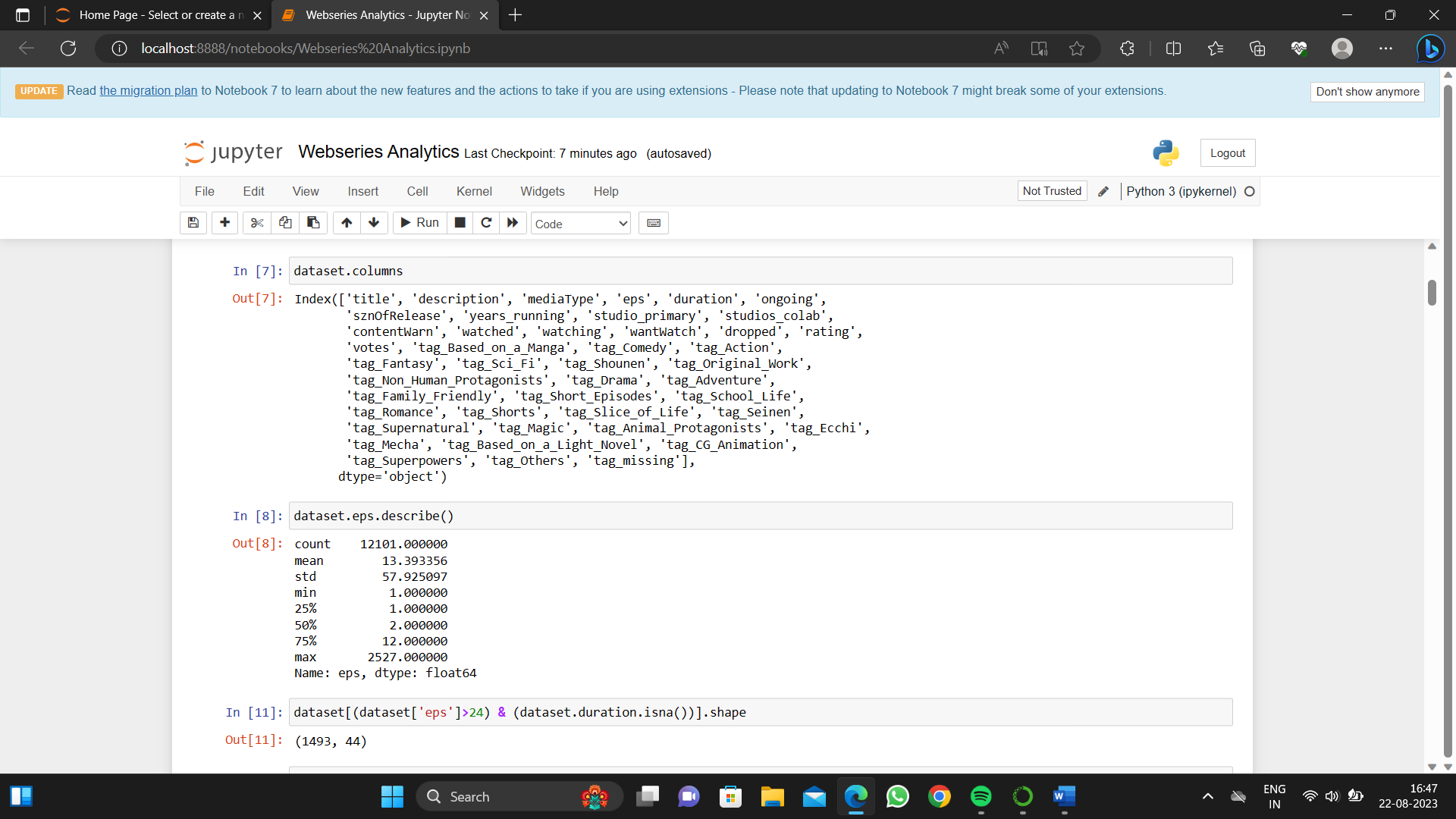
import seaborn as sns

* dataset = pd.read\_excel(r"C:\Users\arman\Downloads\2255872-anime\_data.xlsx")
* dataset.head(2)
* dataset.shape



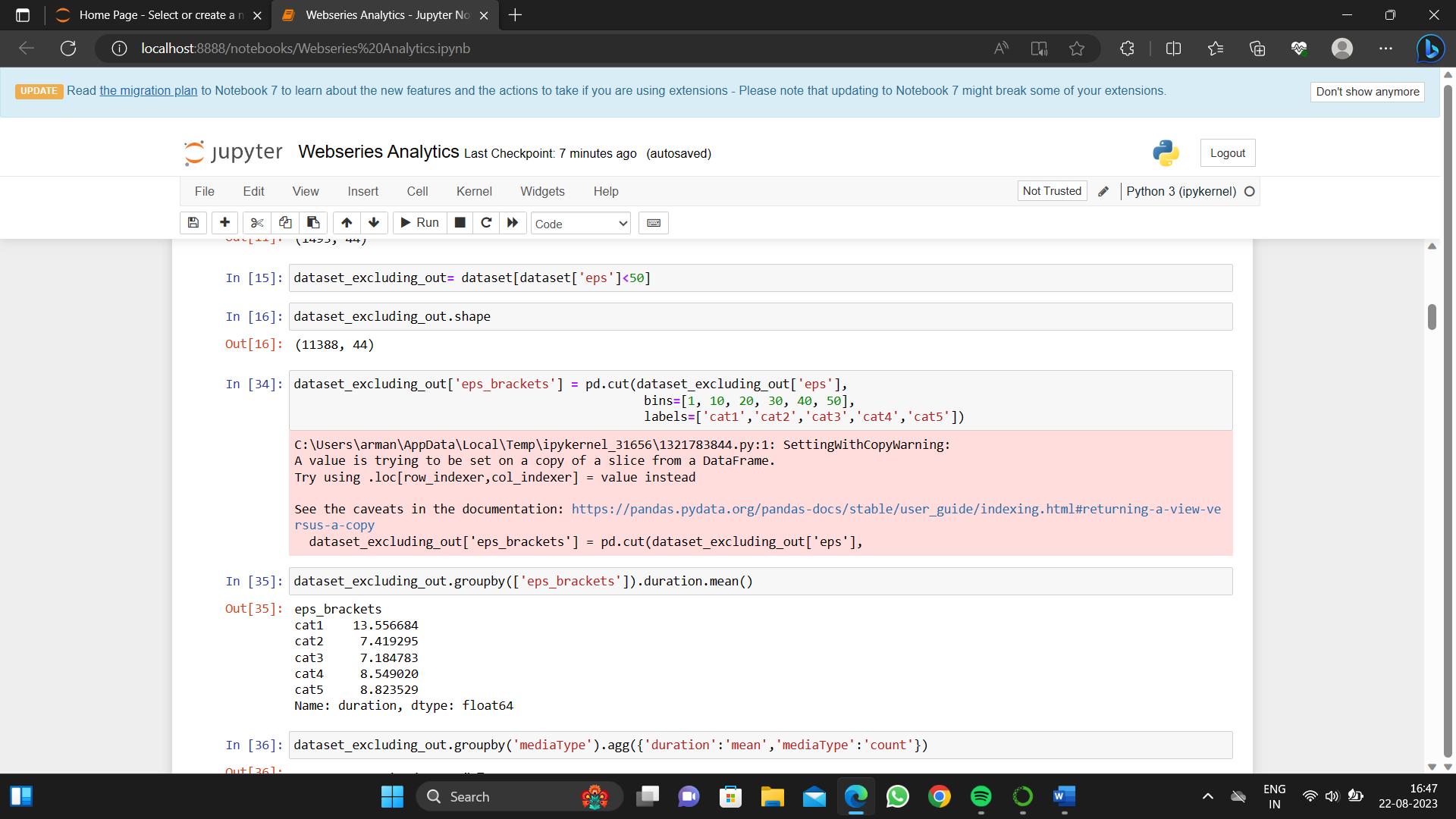
Codes:

* dataset.columns
* dataset.eps.describe()
* dataset[(dataset['eps']>24) & (dataset.duration.isna())].shape



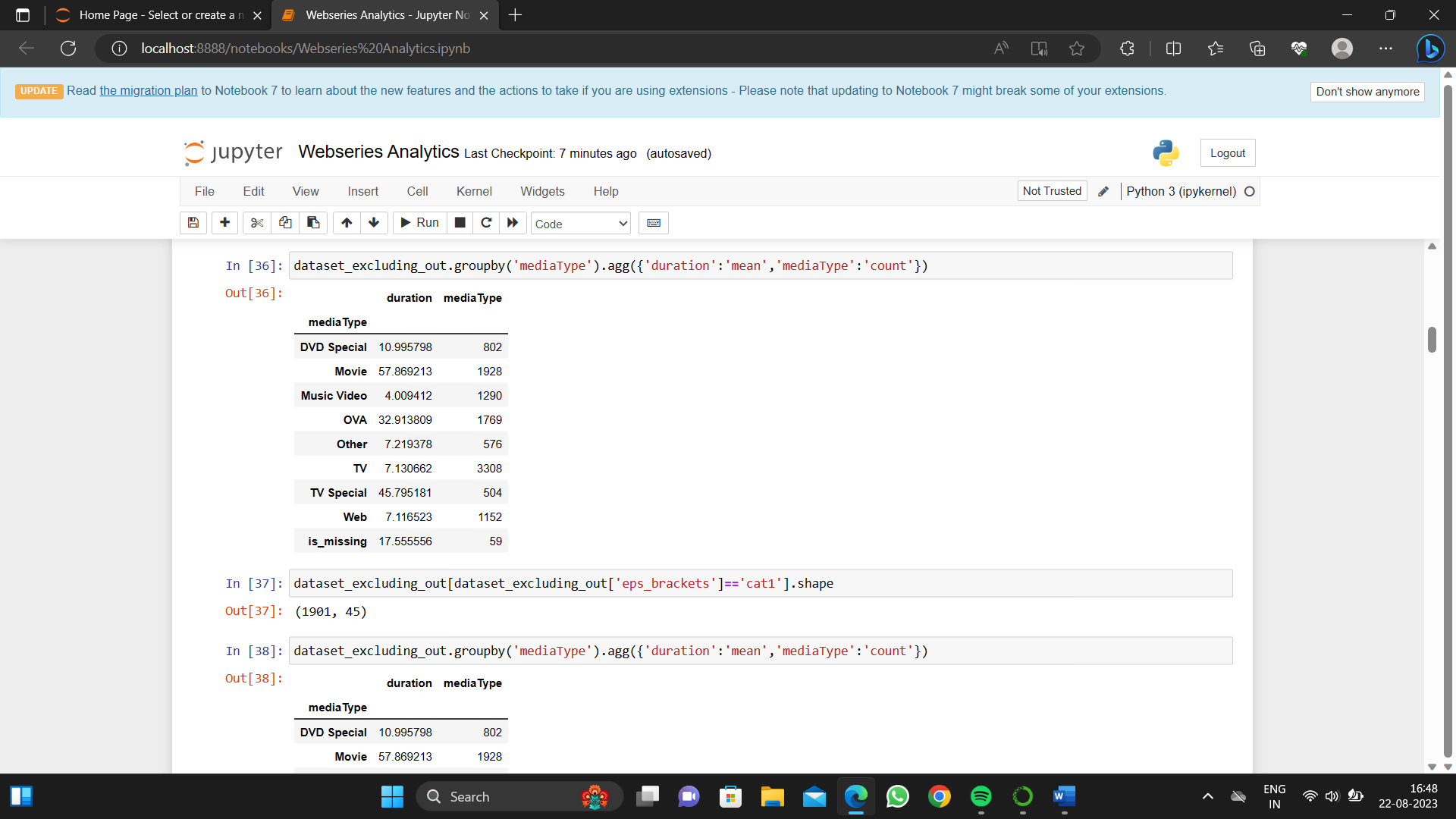
Codes:

* dataset\_excluding\_out= dataset[dataset['eps']<50]
* dataset\_excluding\_out.shape
* dataset\_excluding\_out['eps\_brackets'] = pd.cut(dataset\_excluding\_out['eps'], bins=[1, 10, 20, 30, 40, 50], labels=['cat1','cat2','cat3','cat4','cat5'])
* dataset\_excluding\_out.groupby(['eps\_brackets']).duration.mean()



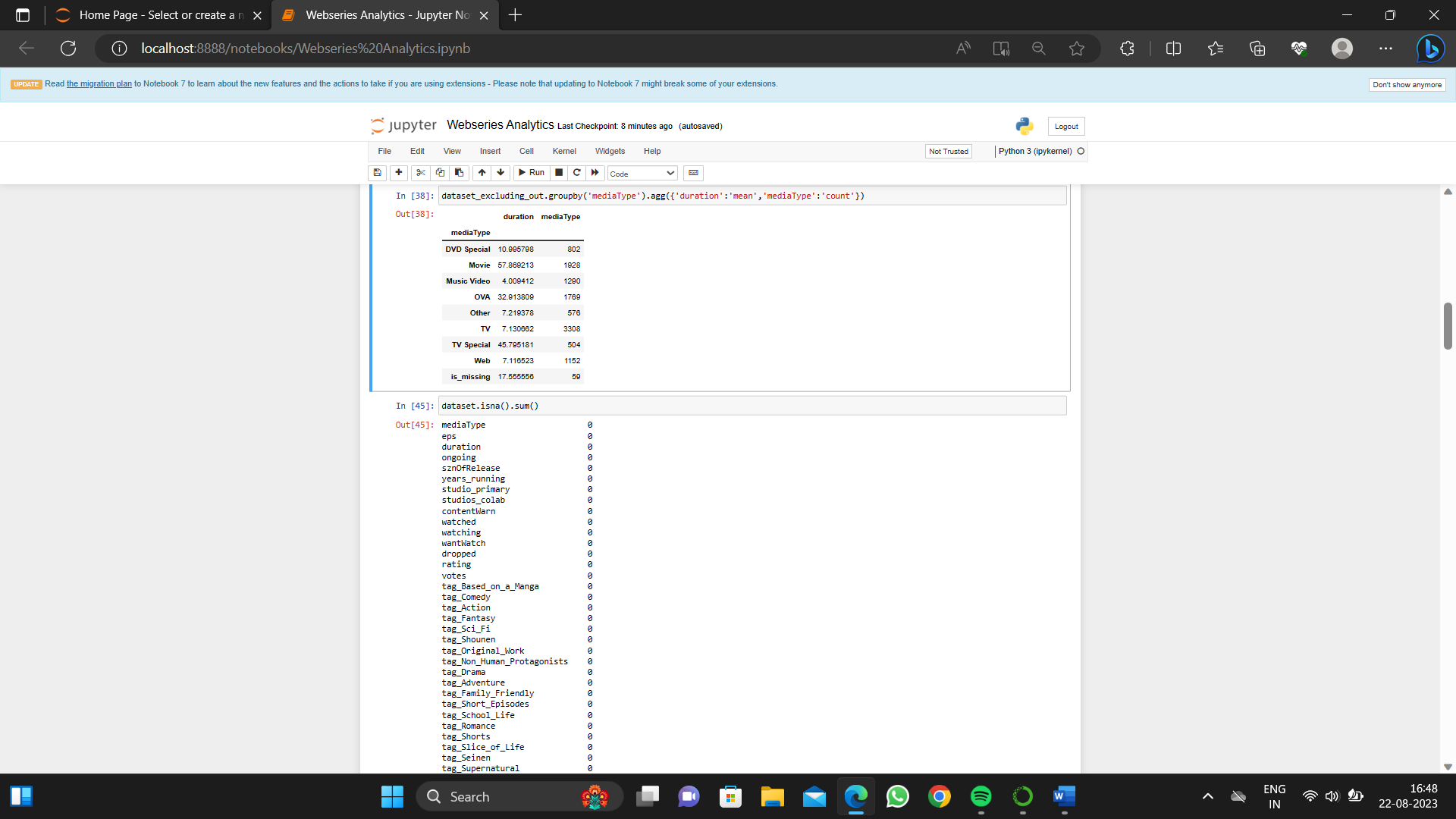
Codes:

* dataset\_excluding\_out.groupby('mediaType').agg({'duration':'mean','mediaType':'count'})
* dataset\_excluding\_out[dataset\_excluding\_out['eps\_brackets']=='cat1'].shape



Codes:

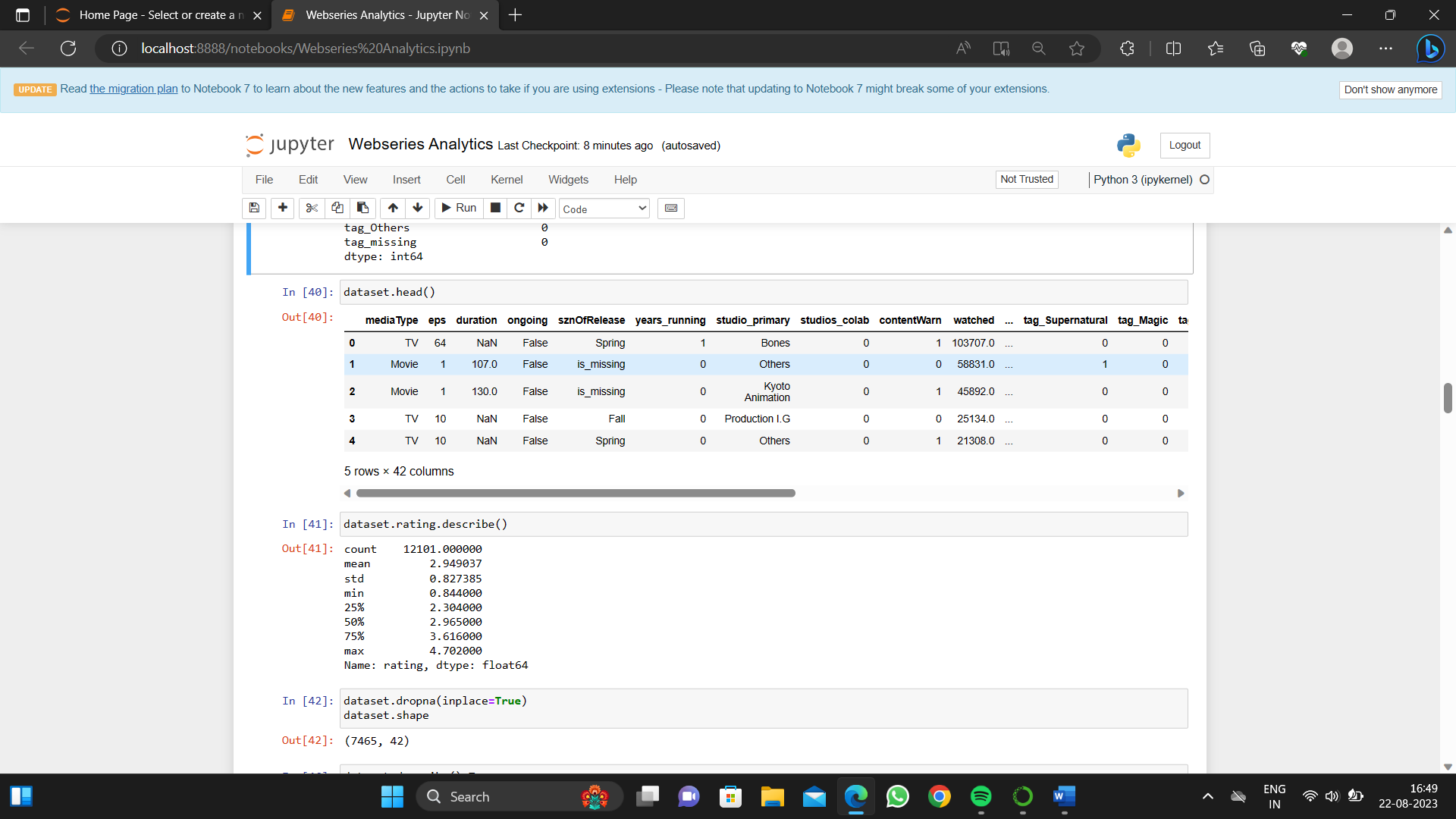
* dataset\_excluding\_out.groupby('mediaType').agg({'duration':'mean','mediaType':'count'})
* dataset.isna().sum()



Codes:

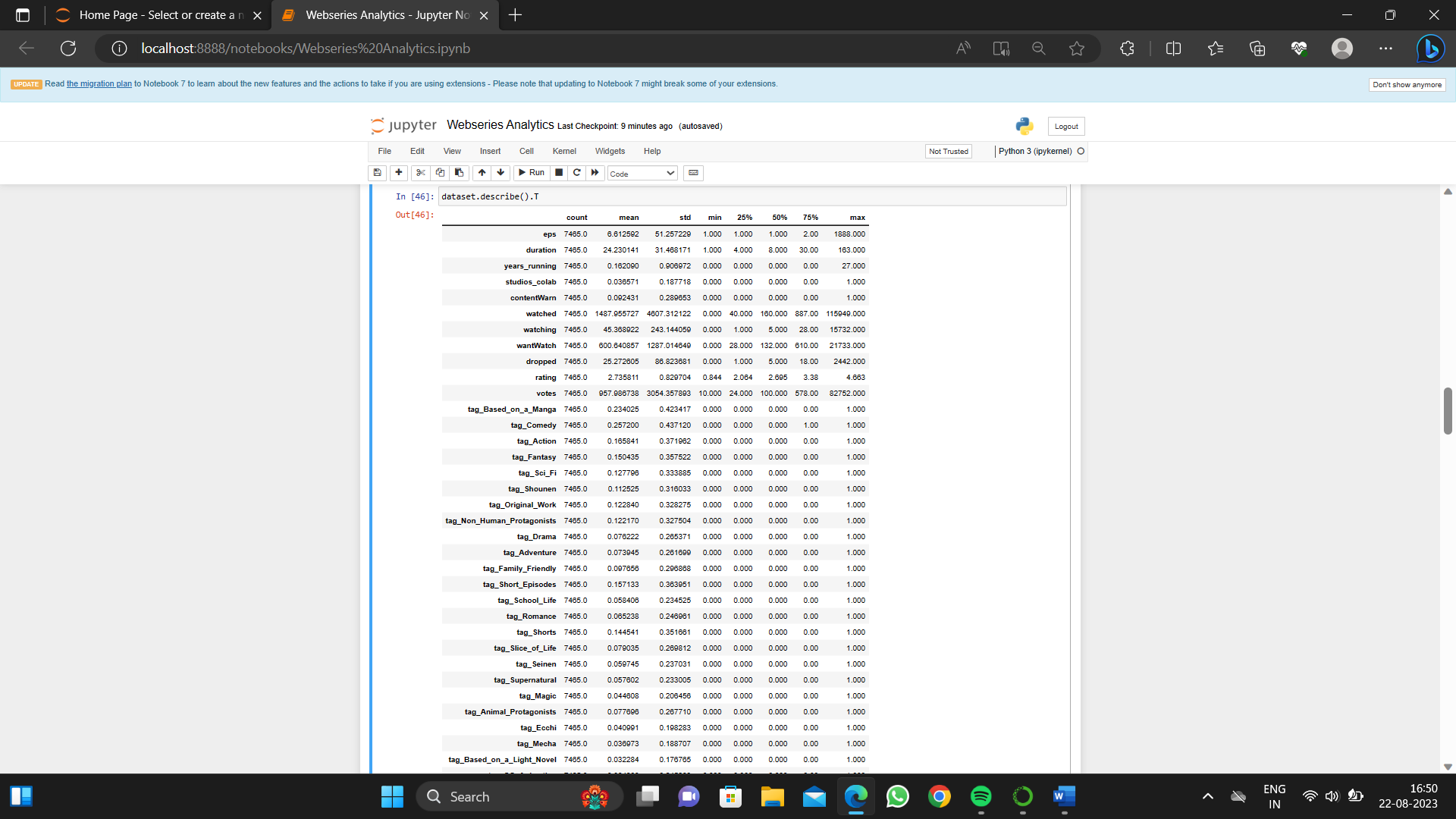
* dataset.head()
* dataset.rating.describe()
* dataset.dropna(inplace=True)

dataset.shape



Code:

* dataset.describe().T



Code:def continuous\_univariate\_analysis(data, feature, figsize=(12, 8), kde=False, bins=None):

f1, (ax\_box, ax\_hist) = plt.subplots(nrows=2, sharex=True, gridspec\_kw={'height\_ratios': (0.25, 0.75)}, figsize=figsize)

sns.set\_palette("viridis")

sns.boxplot(data=data, x=feature, ax=ax\_box, showmeans=True, color='yellow')

if bins:

sns.histplot(data=data, x=feature, ax=ax\_hist, kde=kde, color='crest', bins=bins)

else:

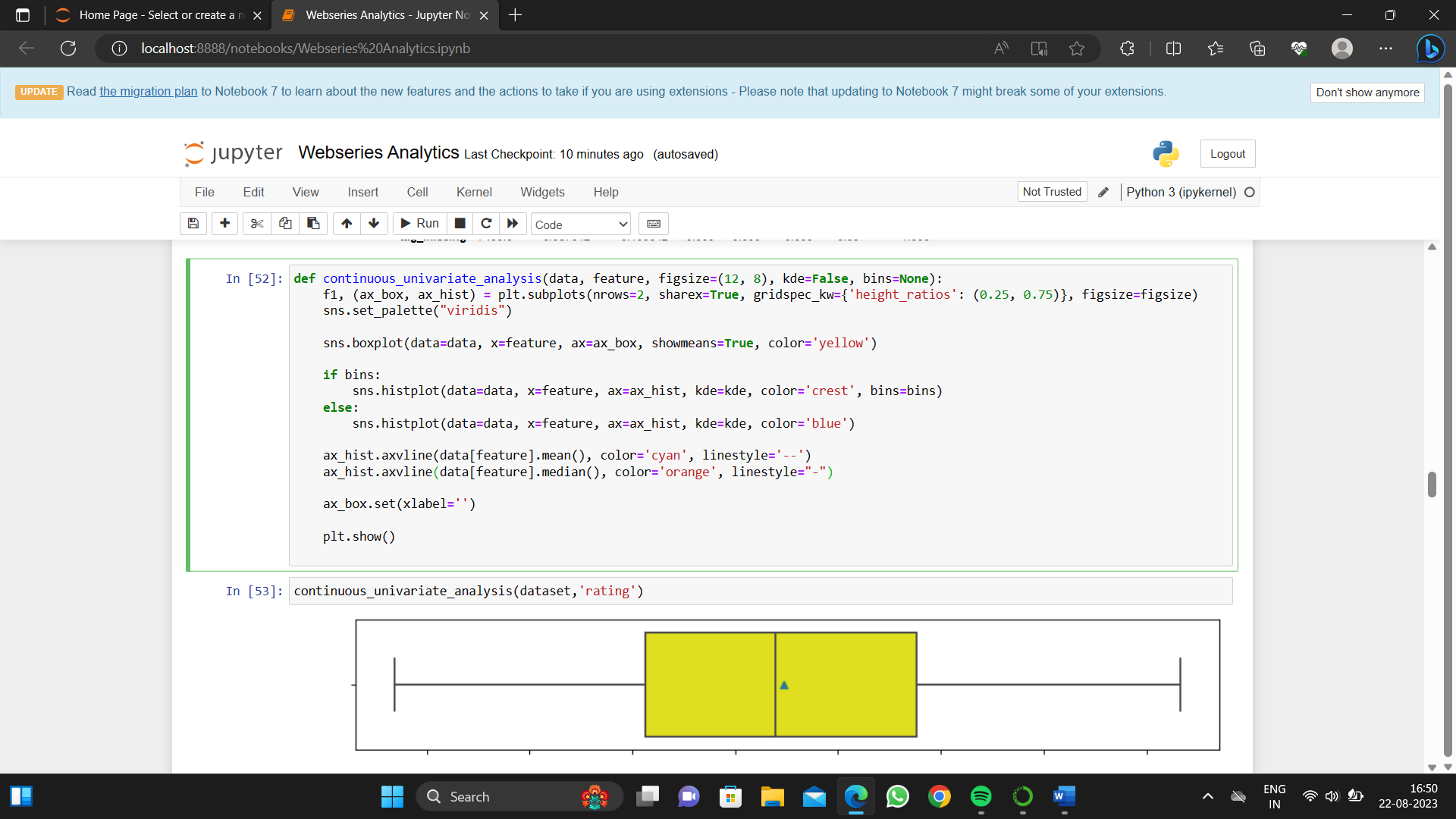
sns.histplot(data=data, x=feature, ax=ax\_hist, kde=kde, color='blue')

ax\_hist.axvline(data[feature].mean(), color='cyan', linestyle='--')

ax\_hist.axvline(data[feature].median(), color='orange', linestyle="-")

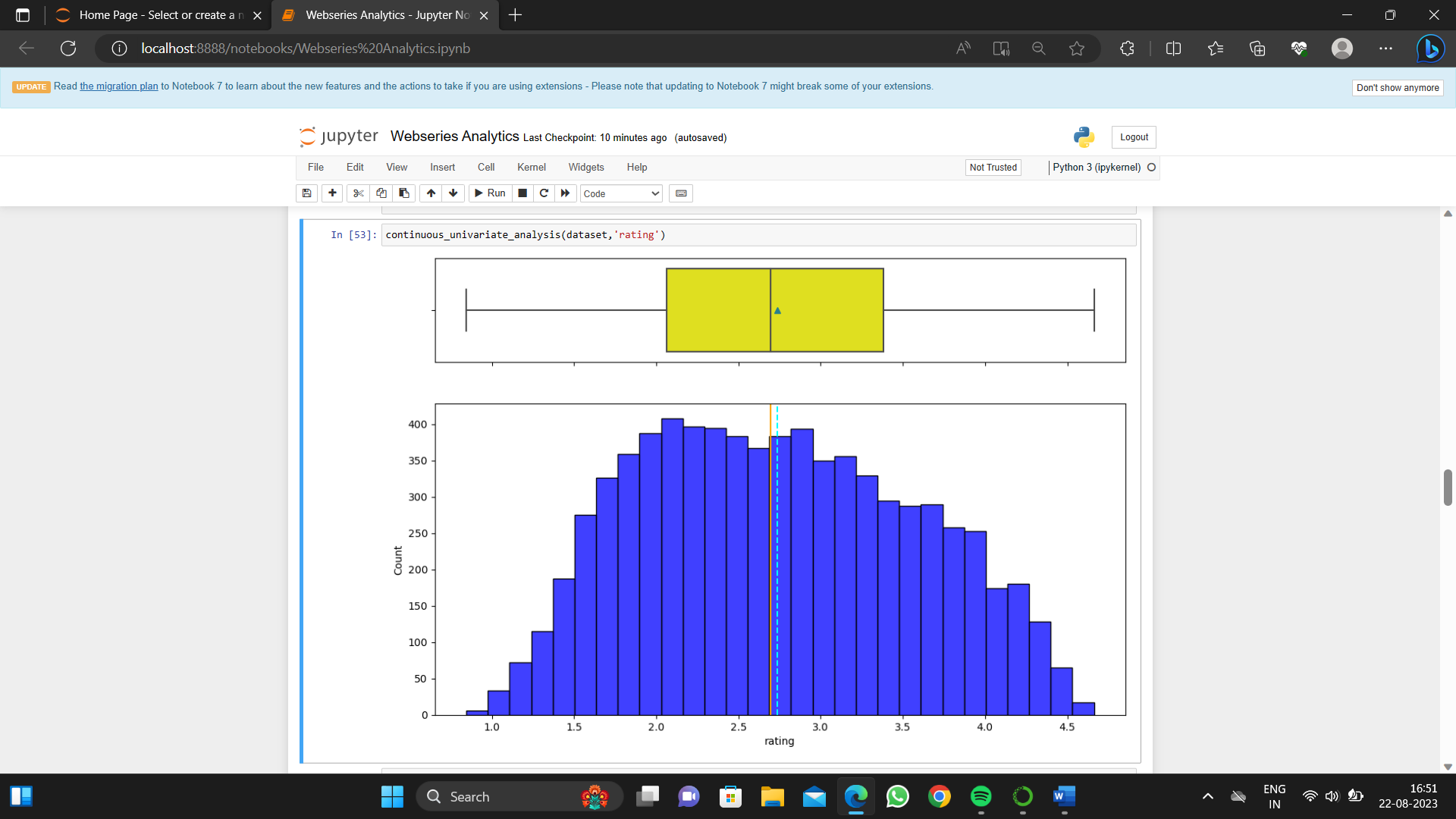
ax\_box.set(xlabel='')

plt.show()



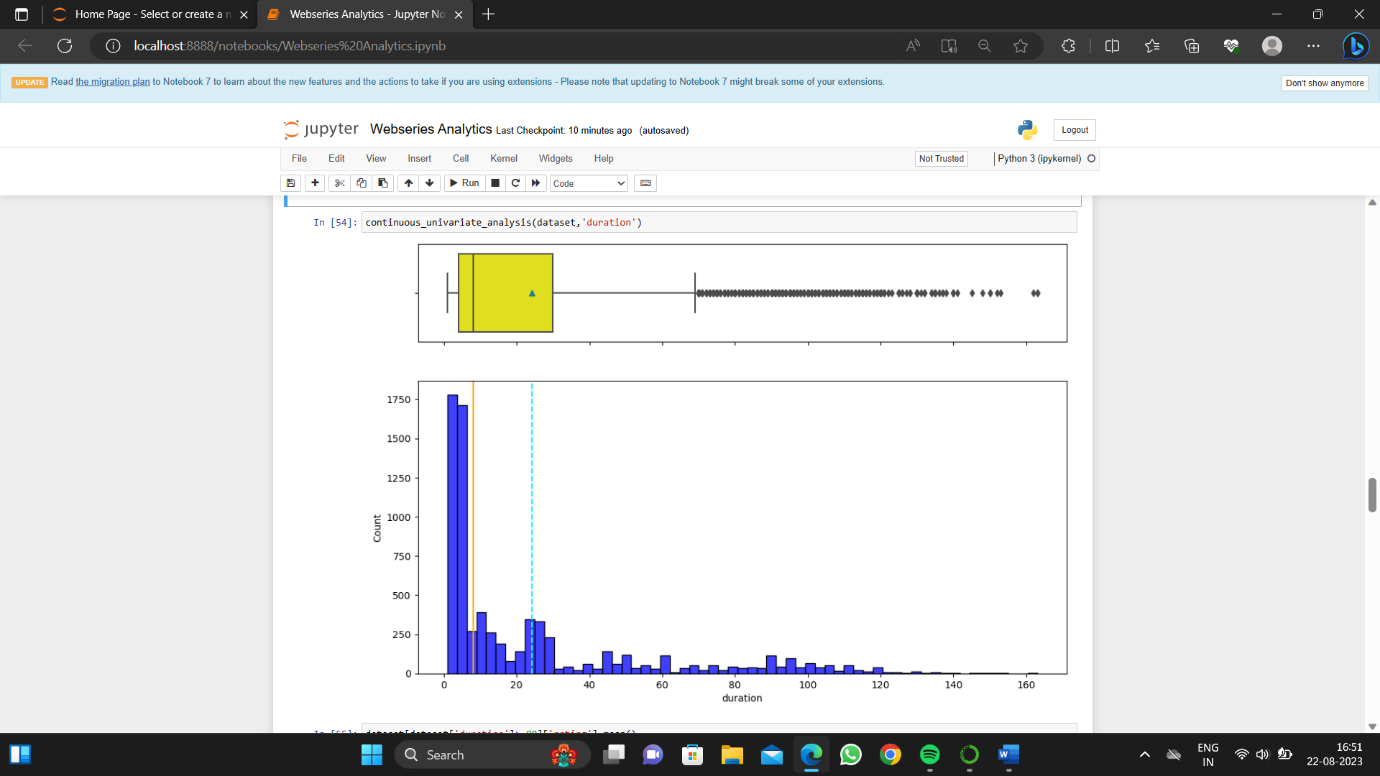
CODE:

* continuous\_univariate\_analysis(dataset,'rating')



CODE:

* continuous\_univariate\_analysis(dataset,'duration')



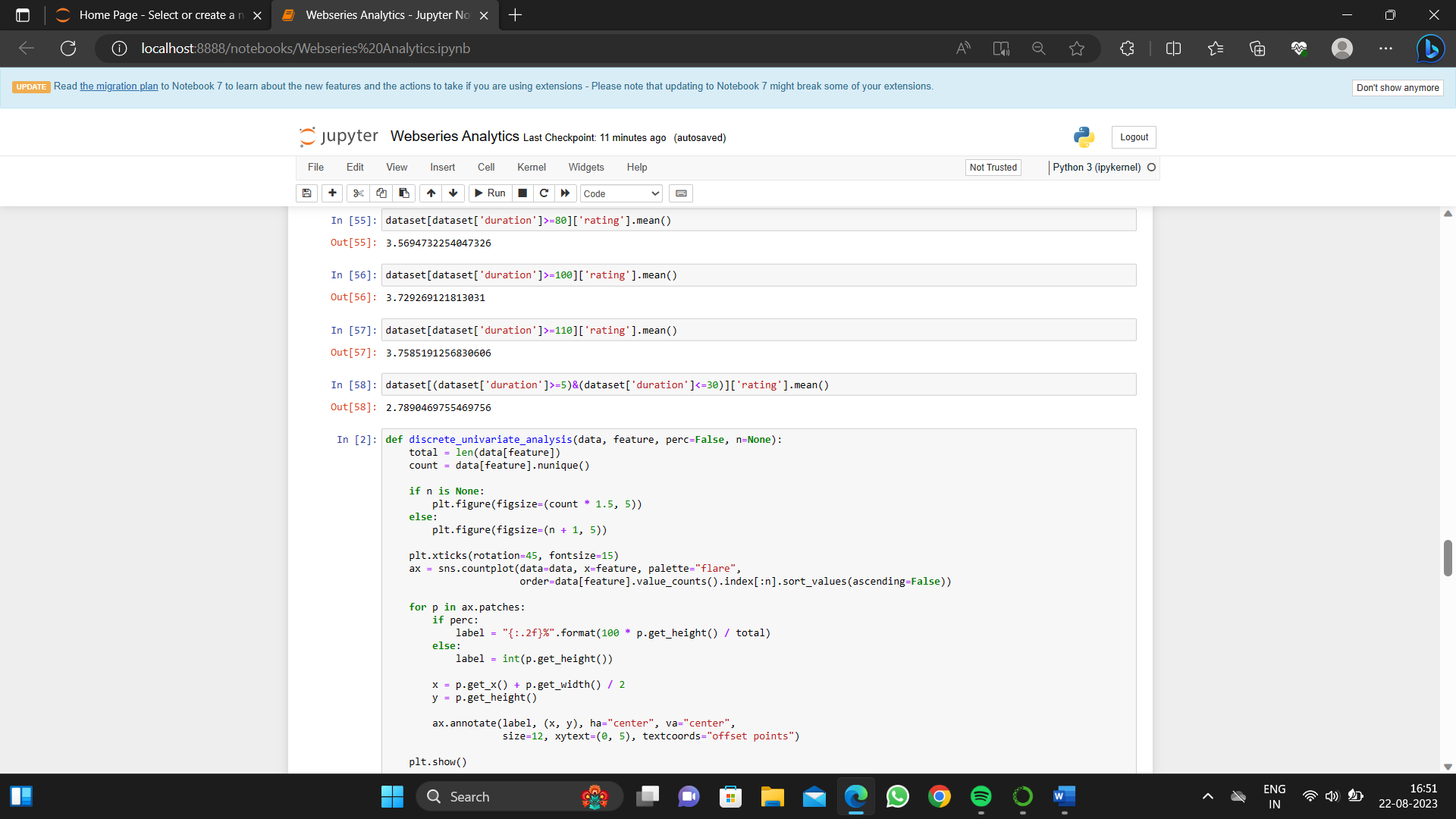
CODES:

\* dataset[dataset['duration']>=80]['rating'].mean()

\* dataset[dataset['duration']>=100]['rating'].mean()

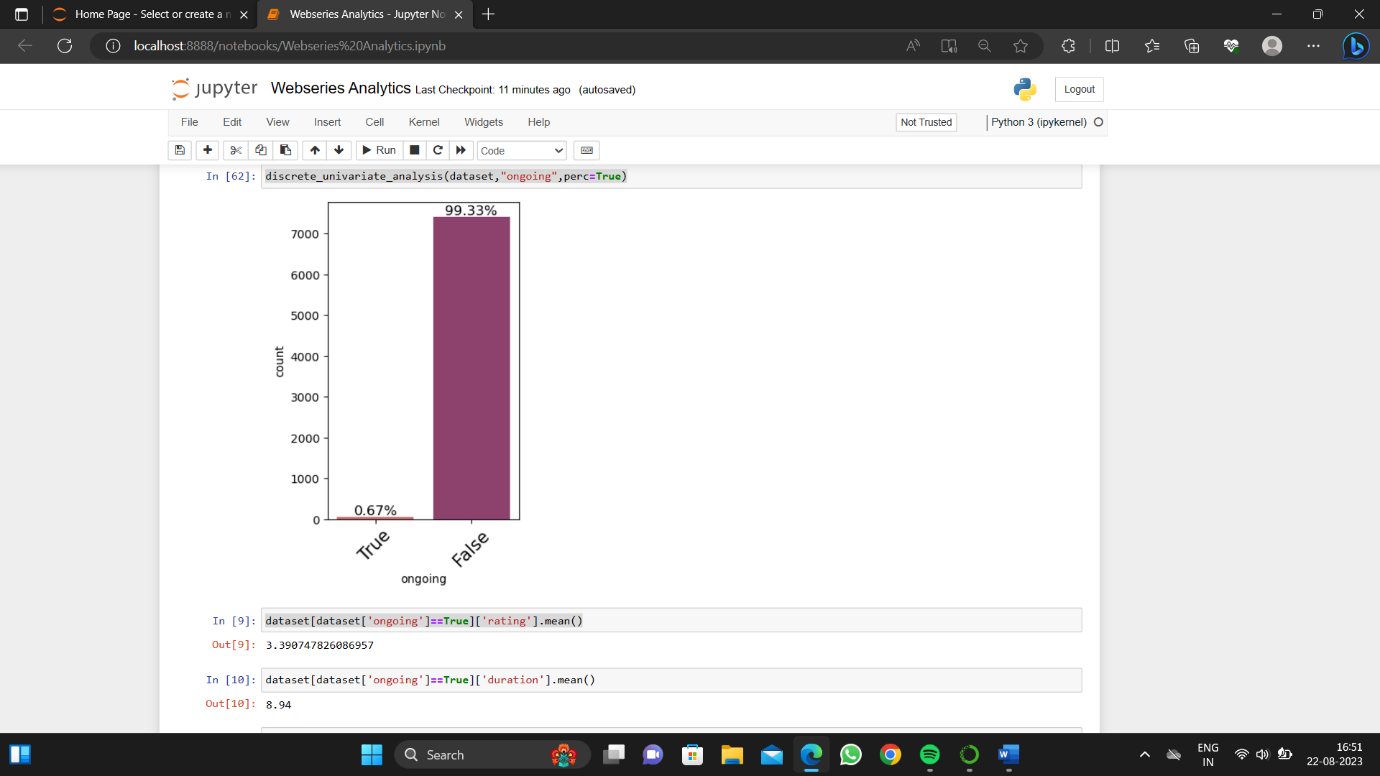
\* dataset[dataset['duration']>=110]['rating'].mean()

\* dataset[(dataset['duration']>=5)&(dataset['duration']<=30)]['rating'].mean()



CODE:

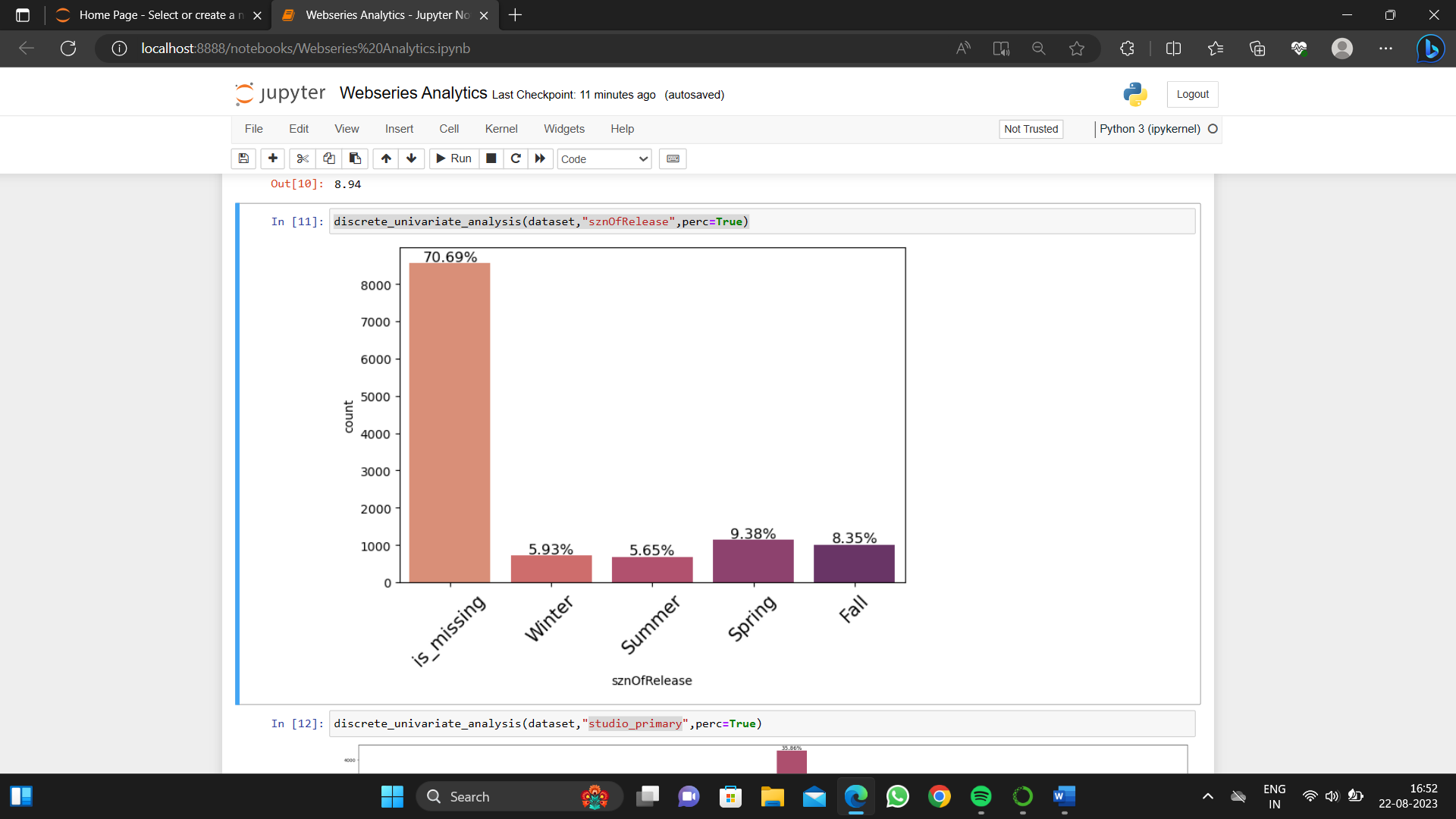
* discrete\_univariate\_analysis(dataset,"ongoing",perc=True)
* dataset[dataset['ongoing']==True]['rating'].mean()
* dataset[dataset['ongoing']==True]['duration'].mean()



CODE:

* discrete\_univariate\_analysis(dataset,"ongoing",

perc=True)



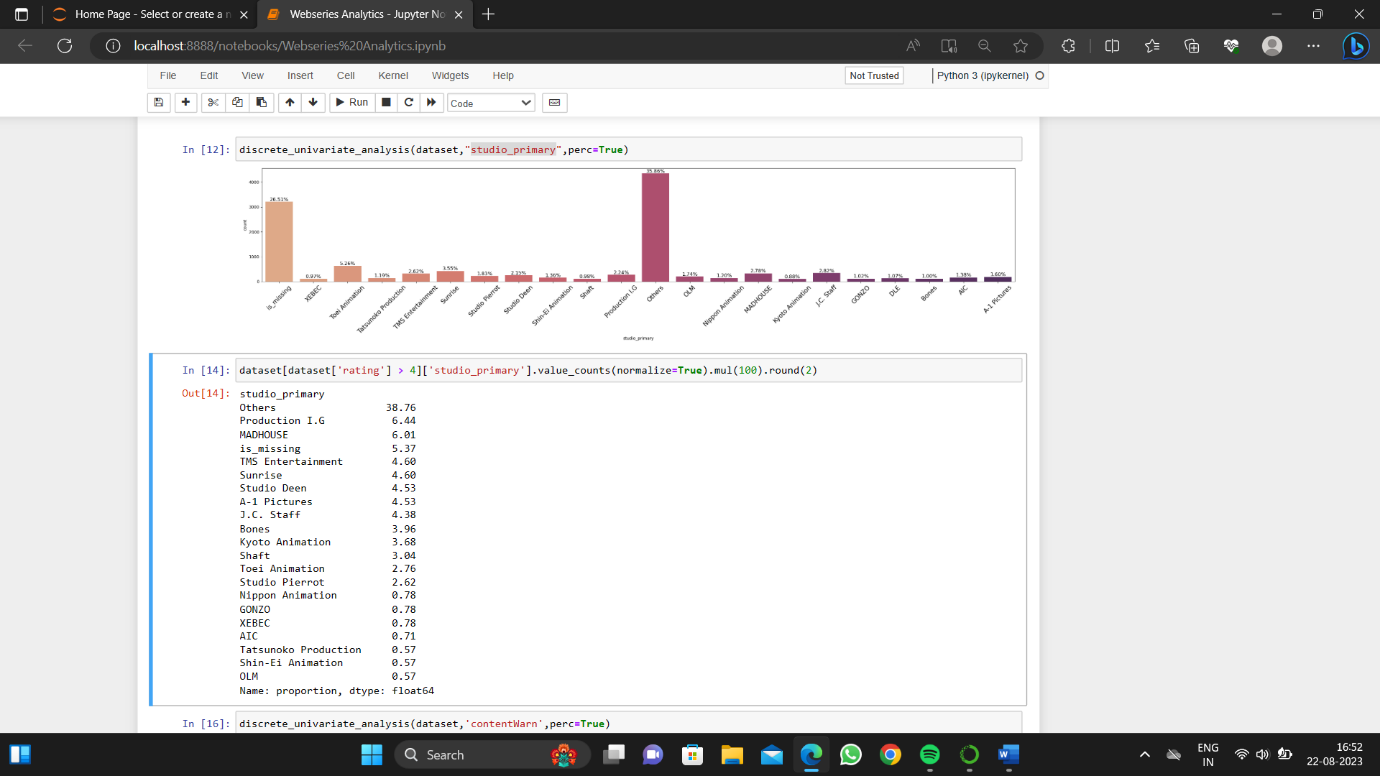
CODE:

* discrete\_univariate\_analysis(dataset,"studio

\_primary",perc=True)

* dataset[dataset['rating'] > 4]

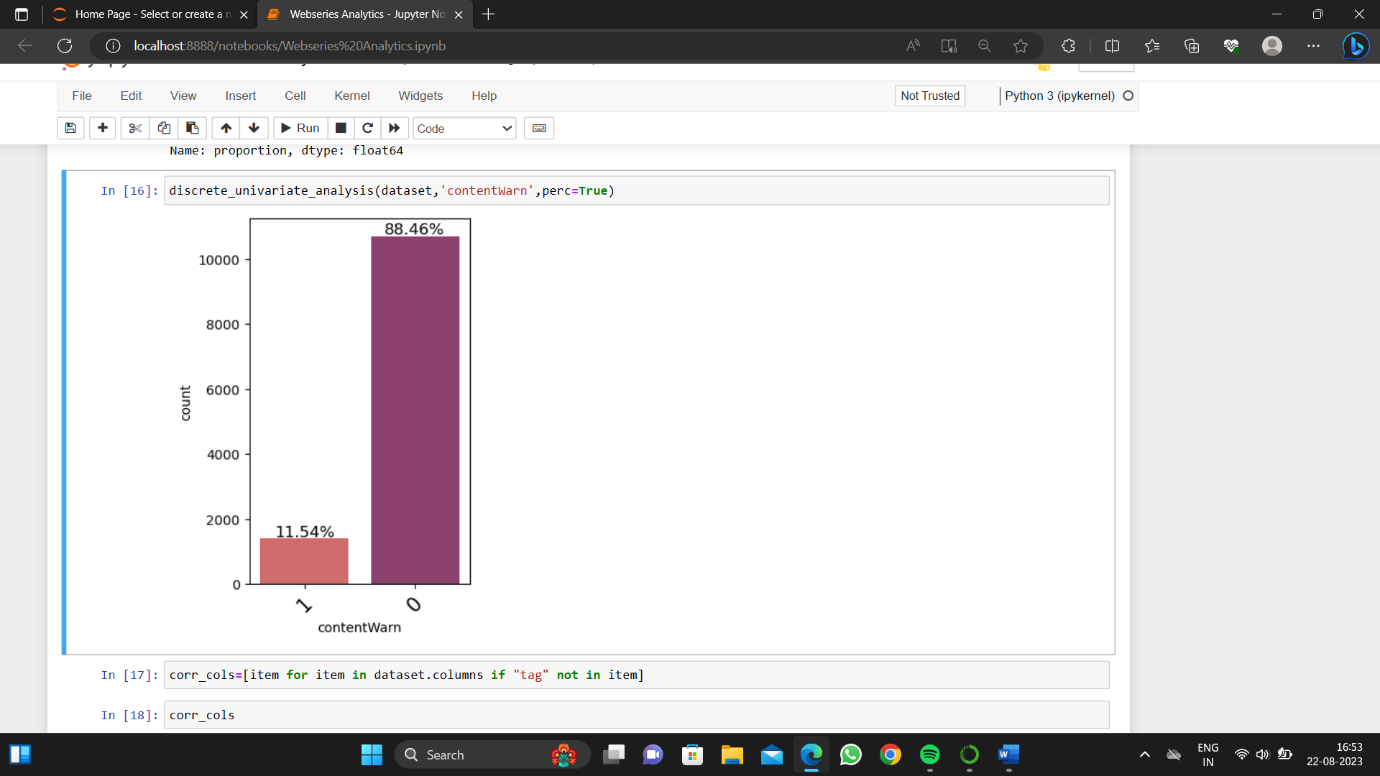
['studio\_primary'].value\_counts(normalize=True).mul(100).round(2)



CODE:

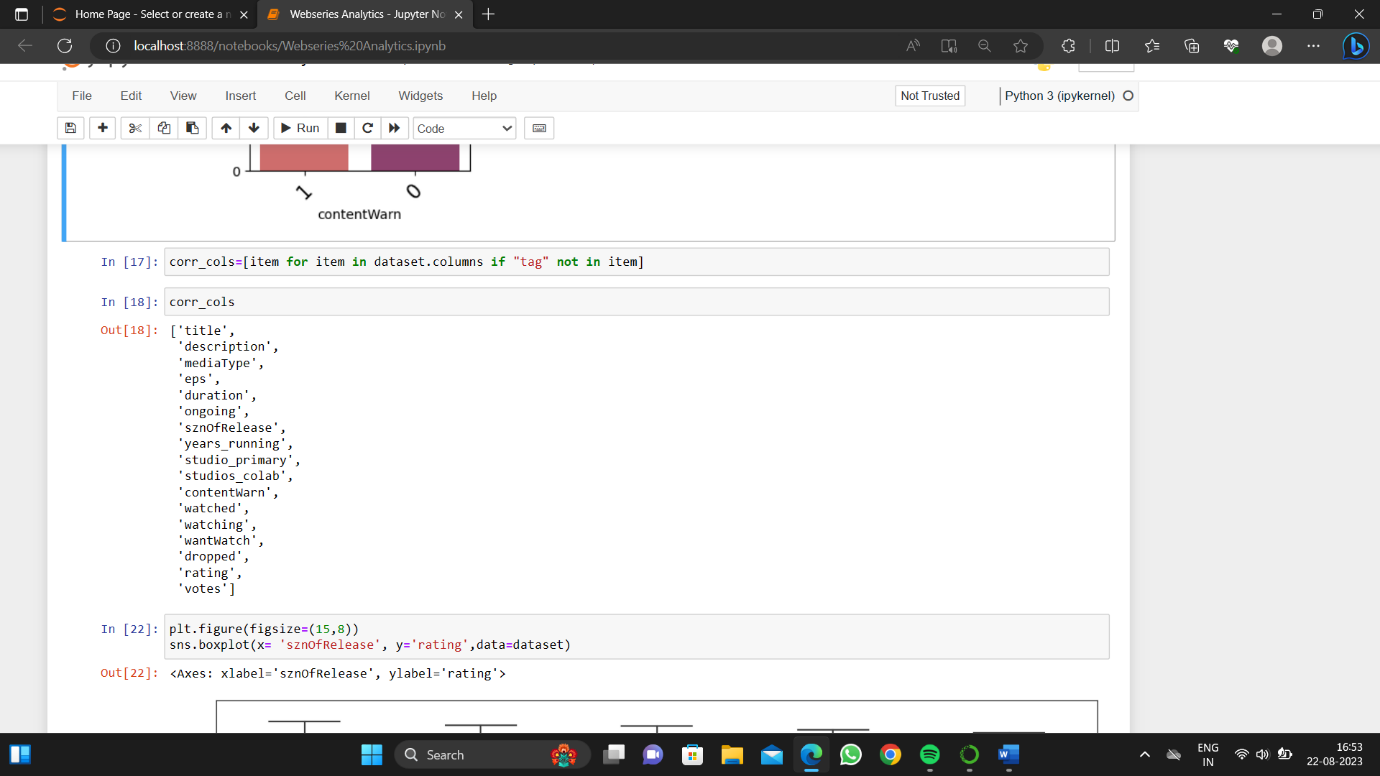
* discrete\_univariate\_analysis(dataset,

'contentWarn',perc=True)



CODE:

* corr\_cols=[item for item in dataset.columns if "tag" not in item]
* corr\_cols



CODE:

* plt.figure(figsize=(15,8))

sns.boxplot(x= 'sznOfRelease', y='rating',data=dataset)

