-In this notebook we are going to analyze data based on anime series.

-We'll cover all the major operations related to it.

In [1]:

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

In [4]:

```
dataset=pd.read_excel(r"C:\Users\lenovo\Downloads\anime_data.xlsx")
```

An overview of excel data:

In [6]:

```
dataset.head()
```

Out[6]:

	title	description	mediaType	eps	duration	ongoing	sznOfRelease	years_running
0	Fullmetal Alchemist: Brotherhood	The foundation of alchemy is based on the law ...	TV	64	NaN	False	Spring	1
1	your name.	Mitsuha and Taki are two total strangers livin...	Movie	1	107.0	False	is_missing	0
2	A Silent Voice	After transferring into a new school, a deaf g...	Movie	1	130.0	False	is_missing	0
3	Haikyuu!! Karasuno High School vs Shiratorizaw...	Picking up where the second season ended, the ...	TV	10	NaN	False	Fall	0
4	Attack on Titan 3rd Season: Part II	The battle to retake Wall Maria begins now! Wi...	TV	10	NaN	False	Spring	0

5 rows × 44 columns

Number of rows and columns in the excel sheet:

In [13]:

```
dataset.shape
```

Out[13]:

```
(12101, 44)
```

In [8]:

```
dataset.describe()
```

Out[8]:

	eps	duration	years_running	studios_colab	contentWarn	watched	1
count	12101.000000	7465.000000	12101.000000	12101.000000	12101.000000	12101.000000	
mean	13.393356	24.230141	0.283200	0.051649	0.115362	2862.605694	
std	57.925097	31.468171	1.152234	0.221326	0.319472	7724.347024	
min	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	4.000000	0.000000	0.000000	0.000000	55.000000	
50%	2.000000	8.000000	0.000000	0.000000	0.000000	341.000000	
75%	12.000000	30.000000	0.000000	0.000000	0.000000	2026.000000	
max	2527.000000	163.000000	51.000000	1.000000	1.000000	161567.000000	7

8 rows × 38 columns

All the columns contained in the sheet:

In [9]:

```
dataset.columns
```

Out[9]:

```
Index(['title', 'description', 'mediaType', 'eps', 'duration', 'ongoing',
       'sznOfRelease', 'years_running', 'studio_primary', 'studios_colab',
       'contentWarn', 'watched', 'watching', 'wantWatch', 'dropped', 'ratin
g',
       'votes', 'tag_Based_on_a_Manga', 'tag_Comedy', 'tag_Action',
       'tag_Fantasy', 'tag_Sci_Fi', 'tag_Shounen', 'tag_Original_Work',
       'tag_Non_Human_Protagonists', 'tag_Drama', 'tag_Adventure',
       'tag_Family_Friendly', 'tag_Short_Episodes', 'tag_School_Life',
       'tag_Romance', 'tag_Shorts', 'tag_Slice_of_Life', 'tag_Seinen',
       'tag_Supernatural', 'tag_Magic', 'tag_Animal_Protagonists', 'tag_Ecch
i',
       'tag_Mecha', 'tag_Based_on_a_Light_Novel', 'tag_CG_Animation',
       'tag_Superpowers', 'tag_Others', 'tag_missing'],
      dtype='object')
```

In [35]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12101 entries, 0 to 12100
Data columns (total 44 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   title                        12100 non-null  object
 1   description                  7633 non-null   object
 2   mediaType                    12101 non-null  object
 3   eps                          12101 non-null  int64
 4   duration                     7465 non-null   float64
 5   ongoing                      12101 non-null  bool
 6   sznOfRelease                 12101 non-null  object
 7   years_running                12101 non-null  int64
 8   studio_primary               12101 non-null  object
 9   studios_colab                12101 non-null  int64
 10  contentWarn                  12101 non-null  int64
 11  watched                      12101 non-null  float64
 12  watching                     12101 non-null  int64
 13  wantWatch                    12101 non-null  int64
 14  dropped                      12101 non-null  int64
 15  rating                       12101 non-null  float64
 16  votes                        12101 non-null  int64
 17  tag_Based_on_a_Manga         12101 non-null  int64
 18  tag_Comedy                   12101 non-null  int64
 19  tag_Action                   12101 non-null  int64
 20  tag_Fantasy                  12101 non-null  int64
 21  tag_Sci_Fi                   12101 non-null  int64
 22  tag_Shounen                  12101 non-null  int64
 23  tag_Original_Work            12101 non-null  int64
 24  tag_Non_Human_Protagonists   12101 non-null  int64
 25  tag_Drama                    12101 non-null  int64
 26  tag_Adventure                12101 non-null  int64
 27  tag_Family_Friendly          12101 non-null  int64
 28  tag_Short_Episodes           12101 non-null  int64
 29  tag_School_Life              12101 non-null  int64
 30  tag_Romance                  12101 non-null  int64
 31  tag_Shorts                   12101 non-null  int64
 32  tag_Slice_of_Life            12101 non-null  int64
 33  tag_Seinen                   12101 non-null  int64
 34  tag_Supernatural             12101 non-null  int64
 35  tag_Magic                    12101 non-null  int64
 36  tag_Animal_Protagonists      12101 non-null  int64
 37  tag_Ecchi                    12101 non-null  int64
 38  tag_Mecha                    12101 non-null  int64
 39  tag_Based_on_a_Light_Novel   12101 non-null  int64
 40  tag_CG_Animation             12101 non-null  int64
 41  tag_Superpowers              12101 non-null  int64
 42  tag_Others                   12101 non-null  int64
 43  tag_missing                  12101 non-null  int64
dtypes: bool(1), float64(3), int64(35), object(5)
memory usage: 4.0+ MB
```

In [33]:

```
dataset.isna().sum()
```

Out[33]:

```
title                          1
description                 4468
mediaType                      0
eps                            0
duration                    4636
ongoing                        0
sznOfRelease                   0
years_running                  0
studio_primary                 0
studios_colab                  0
contentWarn                    0
watched                        0
watching                       0
wantWatch                      0
dropped                        0
rating                         0
votes                          0
tag_Based_on_a_Manga           0
tag_Comedy                     0
tag_Action                     0
tag_Fantasy                    0
tag_Sci_Fi                     0
tag_Shounen                    0
tag_Original_Work              0
tag_Non_Human_Protagonists     0
tag_Drama                      0
tag_Adventure                  0
tag_Family_Friendly            0
tag_Short_Episodes             0
tag_School_Life                0
tag_Romance                    0
tag_Shorts                     0
tag_Slice_of_Life              0
tag_Seinen                     0
tag_Supernatural               0
tag_Magic                      0
tag_Animal_Protagonists        0
tag_Ecchi                      0
tag_Mecha                      0
tag_Based_on_a_Light_Novel     0
tag_CG_Animation               0
tag_Superpowers                0
tag_Others                     0
tag_missing                    0
dtype: int64
```

In [36]:

```
dataset.describe().T
```

Out[36]:

	count	mean	std	min	25%	50%	7
eps	12101.0	13.393356	57.925097	1.000	1.000	2.000	12.0
duration	7465.0	24.230141	31.468171	1.000	4.000	8.000	30.0
years_running	12101.0	0.283200	1.152234	0.000	0.000	0.000	0.0
studios_colab	12101.0	0.051649	0.221326	0.000	0.000	0.000	0.0
contentWarn	12101.0	0.115362	0.319472	0.000	0.000	0.000	0.0
watched	12101.0	2862.605694	7724.347024	0.000	55.000	341.000	2026.0
watching	12101.0	256.334435	1380.840902	0.000	2.000	14.000	100.0
wantWatch	12101.0	1203.681431	2294.327380	0.000	49.000	296.000	1275.0
dropped	12101.0	151.568383	493.931710	0.000	3.000	12.000	65.0
rating	12101.0	2.949037	0.827385	0.844	2.304	2.965	3.6
votes	12101.0	2088.124700	5950.332228	10.000	34.000	219.000	1414.0
tag_Based_on_a_Manga	12101.0	0.290802	0.454151	0.000	0.000	0.000	1.0
tag_Comedy	12101.0	0.272870	0.445453	0.000	0.000	0.000	1.0
tag_Action	12101.0	0.231221	0.421631	0.000	0.000	0.000	0.0
tag_Fantasy	12101.0	0.181555	0.385493	0.000	0.000	0.000	0.0
tag_Sci_Fi	12101.0	0.166267	0.372336	0.000	0.000	0.000	0.0
tag_Shounen	12101.0	0.144864	0.351978	0.000	0.000	0.000	0.0
tag_Original_Work	12101.0	0.135195	0.341946	0.000	0.000	0.000	0.0
tag_Non_Human_Protagonists	12101.0	0.112470	0.315957	0.000	0.000	0.000	0.0
tag_Drama	12101.0	0.106107	0.307987	0.000	0.000	0.000	0.0
tag_Adventure	12101.0	0.103793	0.305005	0.000	0.000	0.000	0.0
tag_Family_Friendly	12101.0	0.097017	0.295993	0.000	0.000	0.000	0.0
tag_Short_Episodes	12101.0	0.096934	0.295880	0.000	0.000	0.000	0.0
tag_School_Life	12101.0	0.092306	0.289470	0.000	0.000	0.000	0.0
tag_Romance	12101.0	0.092141	0.289237	0.000	0.000	0.000	0.0
tag_Shorts	12101.0	0.089662	0.285709	0.000	0.000	0.000	0.0
tag_Slice_of_Life	12101.0	0.080820	0.272569	0.000	0.000	0.000	0.0
tag_Seinen	12101.0	0.077101	0.266763	0.000	0.000	0.000	0.0
tag_Supernatural	12101.0	0.070903	0.256674	0.000	0.000	0.000	0.0
tag_Magic	12101.0	0.064292	0.245283	0.000	0.000	0.000	0.0
tag_Animal_Protagonists	12101.0	0.060326	0.238099	0.000	0.000	0.000	0.0
tag_Ecchi	12101.0	0.057433	0.232678	0.000	0.000	0.000	0.0
tag_Mecha	12101.0	0.054541	0.227091	0.000	0.000	0.000	0.0
tag_Based_on_a_Light_Novel	12101.0	0.053384	0.224807	0.000	0.000	0.000	0.0
tag_CG_Animation	12101.0	0.050079	0.218116	0.000	0.000	0.000	0.0
tag_Superpowers	12101.0	0.044624	0.206486	0.000	0.000	0.000	0.0
tag_Others	12101.0	0.090654	0.287128	0.000	0.000	0.000	0.0

	count	mean	std	min	25%	50%	75
tag_missing	12101.0	0.025866	0.158741	0.000	0.000	0.000	0.0

Analysing how the number of episodes and their duration affecting the corresponding anime series.

In [14]:

```
dataset.eps.describe()
```

Out[14]:

```
count    12101.000000
mean        13.393356
std         57.925097
min          1.000000
25%          1.000000
50%          2.000000
75%         12.000000
max       2527.000000
Name: eps, dtype: float64
```

In [15]:

```
dataset[(dataset['eps']>24)&(dataset.duration.isna())].shape
```

Out[15]:

```
(1493, 44)
```

In [27]:

```
dataset_excluding_out=dataset[dataset['eps']<50]
```

In [28]:

```
dataset_excluding_out['eps_brackets'] = pd.cut(dataset_excluding_out['eps'],
                                               bins=[1, 10, 20, 30, 40, 50],
                                               labels=['cat1', 'cat2', 'cat3', 'cat4', 'ca
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_14684\3875442380.py:1: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  dataset_excluding_out['eps_brackets'] = pd.cut(dataset_excluding_out['ep
s'],
```

In [29]:

```
dataset_excluding_out.shape
```

Out[29]:

```
(11388, 45)
```

In [30]:

```
dataset_excluding_out.groupby(['eps_brackets']).duration.mean()
```

Out[30]:

```
eps_brackets
cat1    13.556684
cat2     7.419295
cat3     7.184783
cat4     8.549020
cat5     8.823529
Name: duration, dtype: float64
```

In [31]:

```
dataset_excluding_out[dataset_excluding_out['eps_brackets']=='cat1'].shape
```

Out[31]:

```
(1901, 45)
```

In [32]:

```
dataset_excluding_out.groupby('mediaType').agg({'duration':'mean','mediaType':'count'})
```

Out[32]:

mediaType	duration	mediaType
DVD Special	10.995798	802
Movie	57.869213	1928
Music Video	4.009412	1290
OVA	32.913809	1769
Other	7.219378	576
TV	7.130662	3308
TV Special	45.795181	504
Web	7.116523	1152
is_missing	17.555556	59

In [37]:

```
dataset.drop(columns=['title','description'],axis=1,inplace=True)
```

In [38]:

```
dataset.head()
```

Out[38]:

going	sznOfRelease	years_running	studio_primary	studios_colab	contentWarn	watched	...	tag
False	Spring	1	Bones	0	1	103707.0	...	
False	is_missing	0	Others	0	0	58831.0	...	
False	is_missing	0	Kyoto Animation	0	1	45892.0	...	
False	Fall	0	Production I.G	0	0	25134.0	...	
False	Spring	0	Others	0	1	21308.0	...	

In [39]:

```
dataset.rating.describe()
```

Out[39]:

```
count    12101.000000
mean         2.949037
std          0.827385
min          0.844000
25%          2.304000
50%          2.965000
75%          3.616000
max          4.702000
Name: rating, dtype: float64
```

In [40]:

```
dataset.dropna(inplace=True)
dataset.shape
```

Out[40]:

```
(7465, 42)
```

In [53]:

```
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'
    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()
```

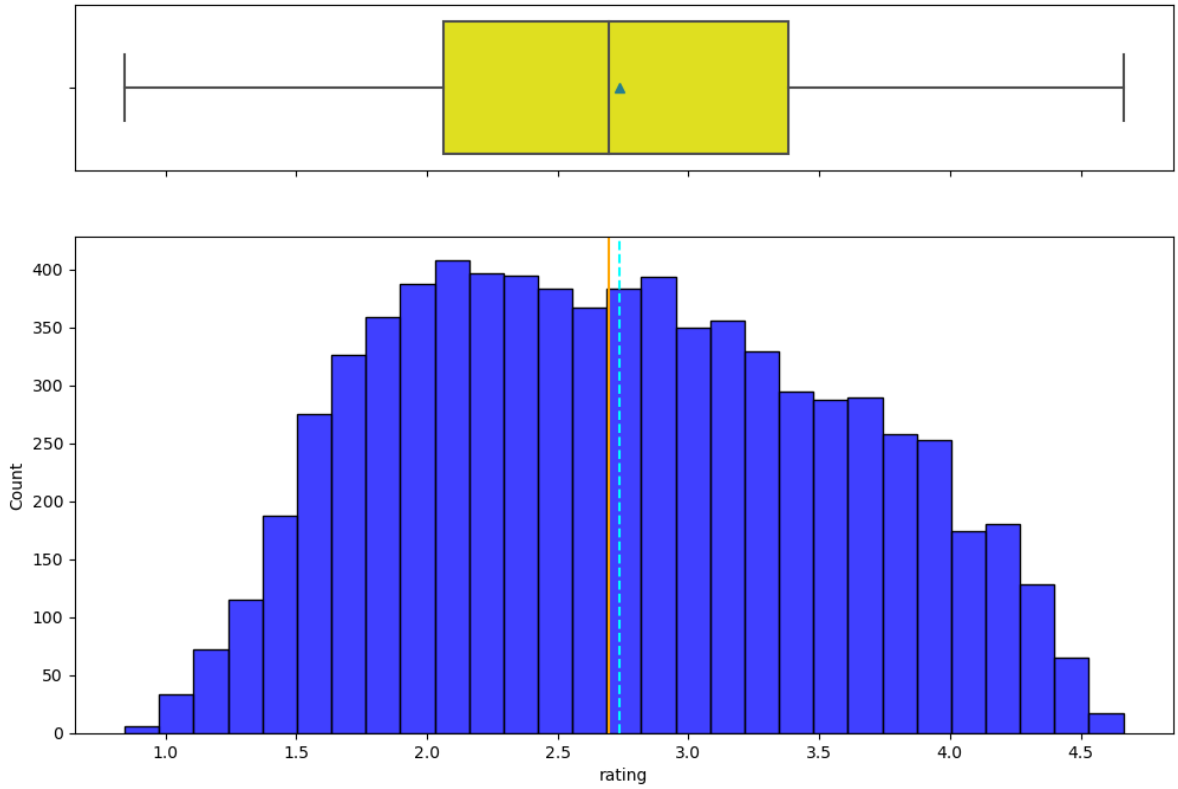
In [55]:

```
def discrete_univariate_analysis(data, feature, perc=False, n=None):
    total = len(data[feature])
    count = data[feature].nunique()

    if n is None:
        plt.figure(figsize=(count * 1.5, 5))
    else:
        plt.figure(figsize=(n + 1, 5))

    plt.xticks(rotation=45, fontsize=15)
    ax = sns.countplot(data=data, x=feature, palette="flare",
                       order=data[feature].value_counts().index[:n].sort_values(ascending=F

    for p in ax.patches:
        if perc:
            label = "{:.2f}%".format(100 * p.get_height() / total)
        else:
            label = int(p.get_height())

        x = p.get_x() + p.get_width() / 2
        y = p.get_height()

        ax.annotate(label, (x, y), ha="center", va="center",
                    size=12, xytext=(0, 5), textcoords="offset points")

    plt.show()
```

Boxplot and histograpgh for the datasheet related to rating

In [57]:

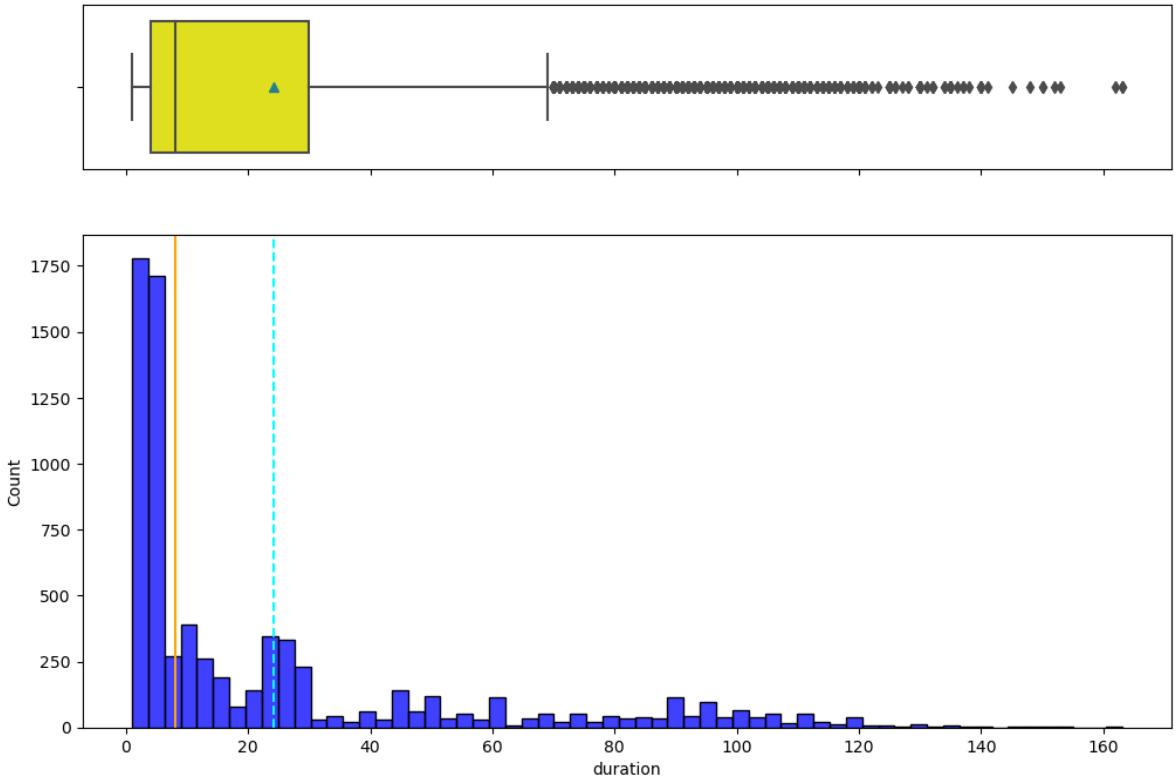
```
continuous_univariate_analysis(dataset,'rating')
```



Boxplot and histograpgh for the datasheet related to duration

In [58]:

```
continuous_univariate_analysis(dataset,'duration')
```



In [59]:

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

Out[59]:

```
3.5694732254047326
```

In [60]:

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

Out[60]:

```
3.729269121813031
```

In [61]:

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

Out[61]:

```
3.7585191256830606
```

In [64]:

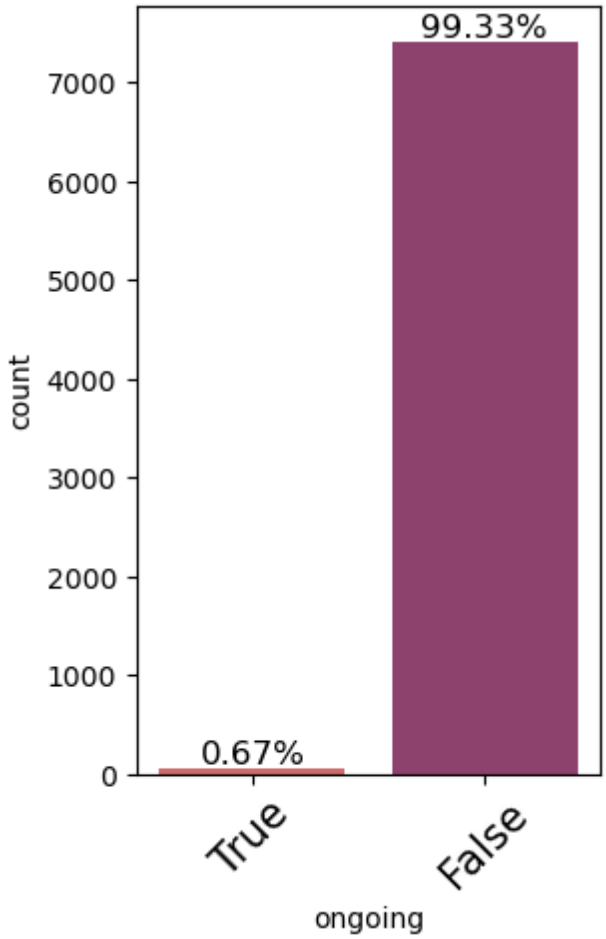
```
dataset[(dataset['duration']>=5)&(dataset['duration']<=30)]['rating'].mean()
```

Out[64]:

```
2.7890469755469756
```

In [65]:

```
discrete_univariate_analysis(dataset,"ongoing",perc=True)
```



In [66]:

```
dataset[dataset['ongoing']==True]['rating'].mean()
```

Out[66]:

```
3.1624600000000003
```

In [67]:

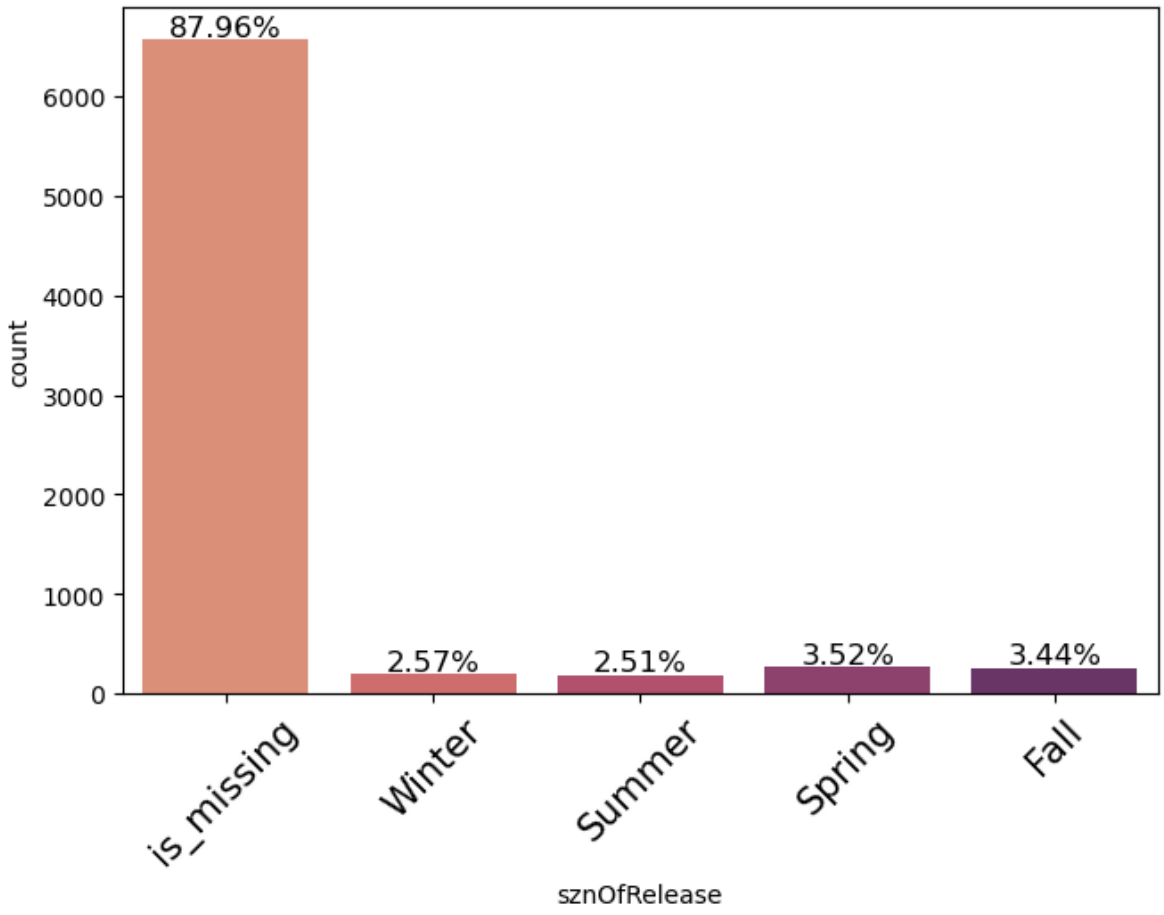
```
dataset[dataset['ongoing']==True]['duration'].mean()
```

Out[67]:

```
8.94
```

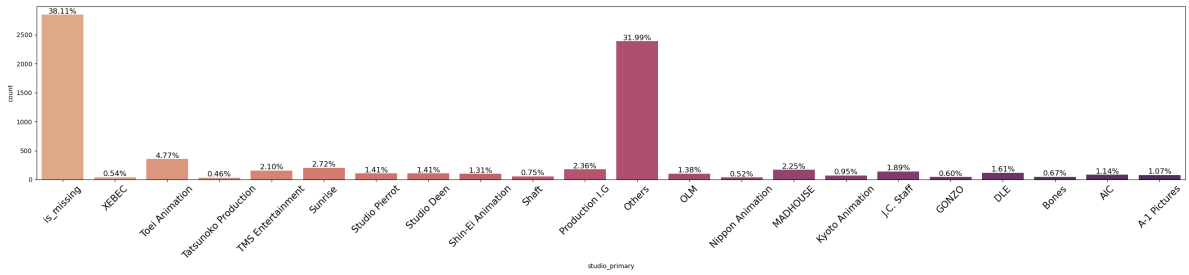
In [69]:

```
discrete_univariate_analysis(dataset,"sznOfRelease",perc=True)
```



In [70]:

```
discrete_univariate_analysis(dataset,"studio_primary",perc=True)
```



In [71]:

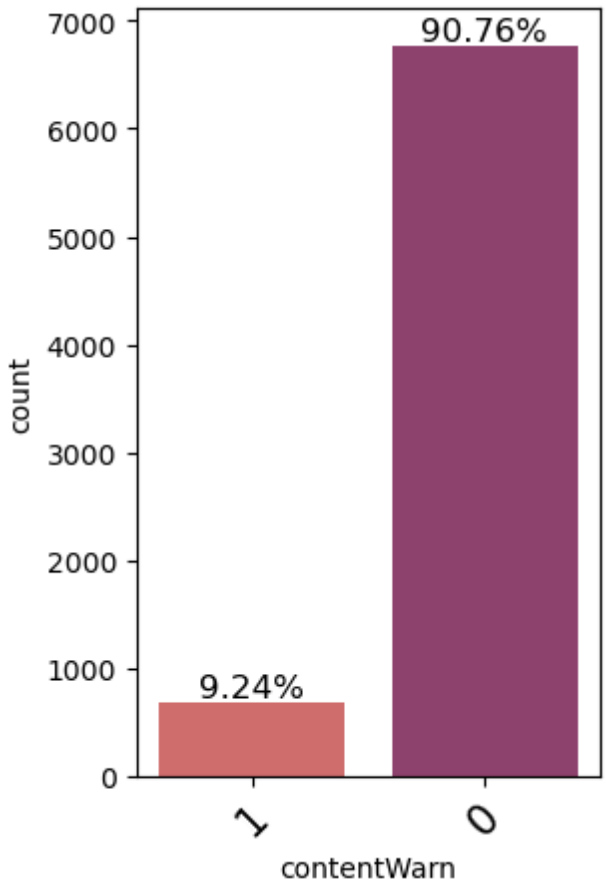
```
dataset[dataset['rating']>4]['studio_primary'].value_counts(normalize=True).mul(100).round(
```

Out[71]:

```
Others                  38.25
Production I.G           8.42
is_missing               7.02
TMS Entertainment        5.96
MADHOUSE                 5.96
Sunrise                  4.91
Kyoto Animation          4.04
Studio Deen              3.68
Bones                    3.68
A-1 Pictures             3.68
Toei Animation           3.51
Shaft                    3.33
J.C. Staff               3.16
Studio Pierrot           2.46
XEBEC                    0.35
Tatsunoko Production     0.35
Nippon Animation         0.35
OLM                      0.35
Shin-Ei Animation        0.35
GONZO                    0.18
Name: studio_primary, dtype: float64
```

In [72]:

```
discrete_univariate_analysis(dataset,"contentWarn",perc=True)
```



In [73]:

```
corr_cols=[item for item in dataset.columns if "tag" not in item]
```

In [74]:

```
corr_cols
```

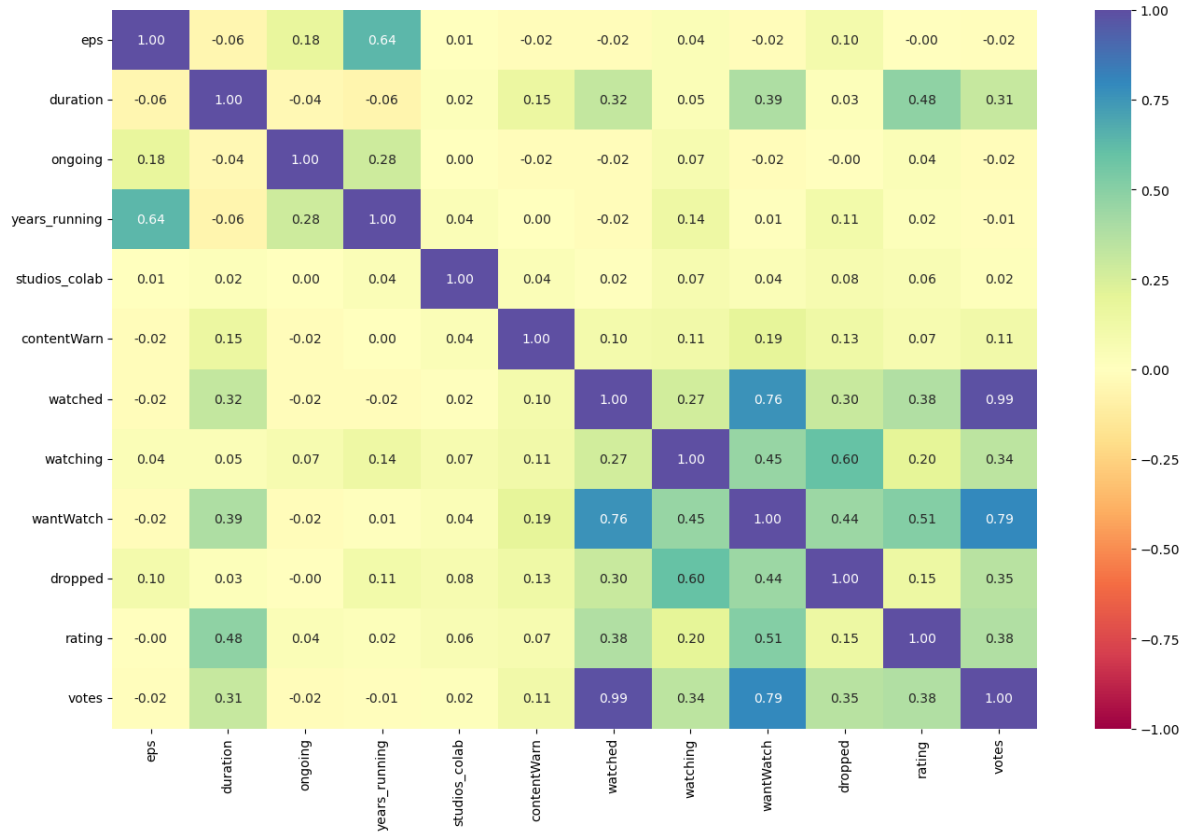
Out[74]:

```
['mediaType',
 'eps',
 'duration',
 'ongoing',
 'sznOfRelease',
 'years_running',
 'studio_primary',
 'studios_colab',
 'contentWarn',
 'watched',
 'watching',
 'wantWatch',
 'dropped',
 'rating',
 'votes']
```

In [77]:

```
plt.figure(figsize=(16,10))
sns.heatmap(dataset[corr_cols].corr(),annot=True,vmin=-1,vmax=1,fmt='.2f',cmap='Spectral')
plt.show()
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_14684\2954847673.py:2: FutureWar
ning: The default value of numeric_only in DataFrame.corr is deprecated. In a
future version, it will default to False. Select only valid columns or specif
y the value of numeric_only to silence this warning.
  sns.heatmap(dataset[corr_cols].corr(),annot=True,vmin=-1,vmax=1,fmt='.2f',c
map='Spectral')
```



In [78]:

```
dataset.drop(columns=['eps','watched'],inplace=True)
```

In [79]:

```
dataset.shape
```

Out[79]:

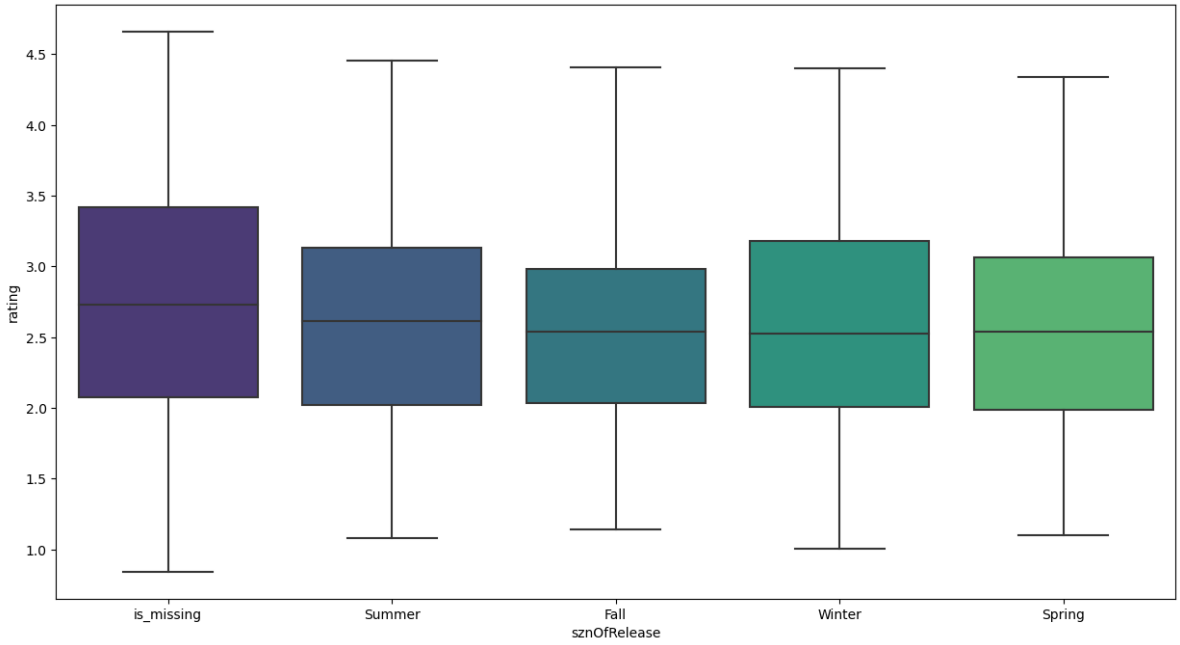
```
(7465, 40)
```

In [80]:

```
plt.figure(figsize=(15,8))
sns.boxplot(x='sznOfRelease',y='rating',data=dataset)
```

Out[80]:

```
<Axes: xlabel='sznOfRelease', ylabel='rating'>
```



Model building - Regression

In [81]:

```
x=dataset.drop(['rating'],axis=1)
y=dataset['rating']
```

In [82]:

```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7465 entries, 1 to 12100
Data columns (total 39 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   mediaType                    7465 non-null   object 
 1   duration                     7465 non-null   float64
 2   ongoing                      7465 non-null   bool   
 3   sznOfRelease                 7465 non-null   object 
 4   years_running                7465 non-null   int64  
 5   studio_primary               7465 non-null   object 
 6   studios_colab                7465 non-null   int64  
 7   contentWarn                  7465 non-null   int64  
 8   watching                     7465 non-null   int64  
 9   wantWatch                    7465 non-null   int64  
 10  dropped                      7465 non-null   int64  
 11  votes                        7465 non-null   int64  
 12  tag_Based_on_a_Manga         7465 non-null   int64  
 13  tag_Comedy                   7465 non-null   int64  
 14  tag_Action                   7465 non-null   int64  
 15  tag_Fantasy                  7465 non-null   int64  
 16  tag_Sci_Fi                   7465 non-null   int64  
 17  tag_Shounen                  7465 non-null   int64  
 18  tag_Original_Work            7465 non-null   int64  
 19  tag_Non_Human_Protagonists   7465 non-null   int64  
 20  tag_Drama                    7465 non-null   int64  
 21  tag_Adventure                7465 non-null   int64  
 22  tag_Family_Friendly          7465 non-null   int64  
 23  tag_Short_Episodes           7465 non-null   int64  
 24  tag_School_Life              7465 non-null   int64  
 25  tag_Romance                  7465 non-null   int64  
 26  tag_Shorts                   7465 non-null   int64  
 27  tag_Slice_of_Life            7465 non-null   int64  
 28  tag_Seinen                   7465 non-null   int64  
 29  tag_Supernatural             7465 non-null   int64  
 30  tag_Magic                    7465 non-null   int64  
 31  tag_Animal_Protagonists      7465 non-null   int64  
 32  tag_Ecchi                    7465 non-null   int64  
 33  tag_Mecha                    7465 non-null   int64  
 34  tag_Based_on_a_Light_Novel   7465 non-null   int64  
 35  tag_CG_Animation             7465 non-null   int64  
 36  tag_Superpowers              7465 non-null   int64  
 37  tag_Others                   7465 non-null   int64  
 38  tag_missing                  7465 non-null   int64  
dtypes: bool(1), float64(1), int64(34), object(3)
memory usage: 2.2+ MB
```

In [84]:

```
x=pd.get_dummies(x,columns=x.select_dtypes(include=['object','category']).columns.tolist(),
x.head()
```

Out[84]:

	duration	ongoing	years_running	studios_colab	contentWarn	watching	wantWatch	droppe
1	107.0	False	0	0	0	1453	21733	12
2	130.0	False	0	0	1	946	17148	13
8	111.0	False	0	0	0	280	6624	15
27	125.0	False	0	0	0	589	12388	16
31	117.0	False	0	0	0	538	15651	13

5 rows × 69 columns

In [85]:

```
x.drop(columns='ongoing',inplace=True)
```

In [86]:

```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7465 entries, 1 to 12100
Data columns (total 68 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   duration                              7465 non-null   float64
 1   years_running                         7465 non-null   int64
 2   studios_colab                         7465 non-null   int64
 3   contentWarn                           7465 non-null   int64
 4   watching                              7465 non-null   int64
 5   wantWatch                             7465 non-null   int64
 6   dropped                               7465 non-null   int64
 7   votes                                 7465 non-null   int64
 8   tag_Based_on_a_Manga                  7465 non-null   int64
 9   tag_Comedy                            7465 non-null   int64
 10  tag_Action                            7465 non-null   int64
 11  tag_Fantasy                           7465 non-null   int64
 12  tag_Sci_Fi                            7465 non-null   int64
 13  tag_Shounen                           7465 non-null   int64
 14  tag_Original_Work                     7465 non-null   int64
 15  tag_Non_Human_Protagonists            7465 non-null   int64
 16  tag_Drama                             7465 non-null   int64
 17  tag_Adventure                         7465 non-null   int64
 18  tag_Family_Friendly                   7465 non-null   int64
 19  tag_Short_Episodes                    7465 non-null   int64
 20  tag_School_Life                       7465 non-null   int64
 21  tag_Romance                           7465 non-null   int64
 22  tag_Shorts                            7465 non-null   int64
 23  tag_Slice_of_Life                     7465 non-null   int64
 24  tag_Seinen                            7465 non-null   int64
 25  tag_Supernatural                      7465 non-null   int64
 26  tag_Magic                             7465 non-null   int64
 27  tag_Animal_Protagonists               7465 non-null   int64
 28  tag_Ecchi                             7465 non-null   int64
 29  tag_Mecha                             7465 non-null   int64
 30  tag_Based_on_a_Light_Novel            7465 non-null   int64
 31  tag_CG_Animation                      7465 non-null   int64
 32  tag_Superpowers                       7465 non-null   int64
 33  tag_Others                            7465 non-null   int64
 34  tag_missing                           7465 non-null   int64
 35  mediaType_Movie                       7465 non-null   uint8
 36  mediaType_Music Video                 7465 non-null   uint8
 37  mediaType_OVA                         7465 non-null   uint8
 38  mediaType_Other                       7465 non-null   uint8
 39  mediaType_TV                          7465 non-null   uint8
 40  mediaType_TV Special                  7465 non-null   uint8
 41  mediaType_Web                         7465 non-null   uint8
 42  mediaType_is_missing                  7465 non-null   uint8
 43  sznOfRelease_Spring                   7465 non-null   uint8
 44  sznOfRelease_Summer                   7465 non-null   uint8
 45  sznOfRelease_Winter                   7465 non-null   uint8
 46  sznOfRelease_is_missing               7465 non-null   uint8
 47  studio_primary_AIC                    7465 non-null   uint8
 48  studio_primary_Bones                  7465 non-null   uint8
 49  studio_primary_DLE                    7465 non-null   uint8
 50  studio_primary_GONZO                  7465 non-null   uint8
 51  studio_primary_J.C. Staff             7465 non-null   uint8
 52  studio_primary_Kyoto Animation        7465 non-null   uint8
 53  studio_primary_MADHOUSE               7465 non-null   uint8
 54  studio_primary_Nippon Animation       7465 non-null   uint8
 55  studio_primary_OLM                    7465 non-null   uint8
```

```
 56  studio_primary_Others                7465 non-null   uint8
 57  studio_primary_Production I.G        7465 non-null   uint8
 58  studio_primary_Shaft                 7465 non-null   uint8
 59  studio_primary_Shin-Ei Animation     7465 non-null   uint8
 60  studio_primary_Studio Deen           7465 non-null   uint8
 61  studio_primary_Studio Pierrot        7465 non-null   uint8
 62  studio_primary_Sunrise               7465 non-null   uint8
 63  studio_primary_TMS Entertainment     7465 non-null   uint8
 64  studio_primary_Tatsunoko Production  7465 non-null   uint8
 65  studio_primary_Toei Animation        7465 non-null   uint8
 66  studio_primary_XEBEC                 7465 non-null   uint8
 67  studio_primary_is_missing            7465 non-null   uint8
dtypes: float64(1), int64(34), uint8(33)
memory usage: 2.3 MB
```

In [88]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
```

In [89]:

```
X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_size=0.2,random_state=1)
```

In [90]:

```
print("Number of samples for train",X_train.shape[0])
print("Number of samples for test",X_test.shape[0])
```

```
Number of samples for train 5972
Number of samples for test 1493
```

In [91]:

```
lin_model=LinearRegression()
lin_model.fit(X_train,Y_train)
```

Out[91]:

▾ LinearRegression

LinearRegression()

In [92]:

```
def Model_performance(model,predictor,target):
    pred=model.predict(predictor)
    r2=r2_score(target,pred)
    rmse=np.sqrt(mean_squared_error(target,pred))

    results=pd.DataFrame({
        "RMSE":rmse,
        "R2 Score":r2
    },index=[0]
    )
    return results
```

In [93]:

```
print("Training Data Performance")
lin_model_train=Model_performance(lin_model,X_train,Y_train)
lin_model_train
```

```
Training Data Performance
```

Out[93]:

	RMSE	R2 Score
0	0.580109	0.515527