Import Required Libraries

In [ ]:
```
pip install pandoc
```

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Upload Dataset using pandas

In [4]:
```
dataset = pd.read_excel(r"C:\Users\bharg\DataScience TCR\2255872-anime_data
```

Understanding the dataset

In [7]:
```
dataset.head()
```

Out[7]:

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

5 rows × 44 columns

In [8]:
```
dataset.shape
```

Out[8]:
```
(12101, 44)
```

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', 'rati
ng',
       '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_Ec
chi',
       'tag_Mecha', 'tag_Based_on_a_Light_Novel', 'tag_CG_Animation',
       'tag_Superpowers', 'tag_Others', 'tag_missing'],
      dtype='object')
```

In [10]:
```
dataset.eps.describe()
```

Out[10]:
```
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 [11]:
```
dataset[(dataset['eps'] > 24) & (dataset.duration.isna())].shape
```

Out[11]:
```
(1493, 44)
```

In [12]:
```
dataset_excluding_out = dataset[dataset['eps'] < 50]
```

In [13]:
```
dataset_excluding_out['eps_brackets'] = pd.cut(dataset_excluding_out['eps']
                                    labels = ['cat1', 'cat2', 'cat3', 'cat4',
```

```
C:\Users\bharg\AppData\Local\Temp\ipykernel_4224\3829596462.py:1: SettingW
ithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  dataset_excluding_out['eps_brackets'] = pd.cut(dataset_excluding_out['ep
s'], bins = [1,10,20, 30, 40, 50],\
```

In [9]:
```
dataset_excluding_out.shape
```

Out[9]:
```
(11388, 45)
```

In [10]:
```
dataset_excluding_out.groupby(['eps_brackets']).duration.mean()
```

Out[10]:
```
eps_brackets
cat1    13.556684
cat2     7.419295
cat3     7.184783
cat4     8.549020
cat5     8.823529
Name: duration, dtype: float64
```

In [11]:
```
dataset_excluding_out.groupby(['eps_brackets']).title.count()
```

Out[11]:
```
eps_brackets
cat1    1901
cat2    2112
cat3    1038
cat4     220
cat5     169
Name: title, dtype: int64
```

In [29]:
```
dataset_excluding_out[dataset_excluding_out['eps_brackets'] == 'cat1'].shap
```

Out[29]:
```
(1901, 45)
```

In [15]:
```
dataset[(dataset['eps'] < 24) & (~dataset.duration.isna())].describe()
```

Out[15]:

	eps	duration	years_running	studios_colab	contentWarn	watched
count	7098.000000	7098.000000	7098.000000	7098.000000	7098.000000	7098.000000
mean	2.546210	25.080727	0.104959	0.034658	0.095661	1531.826289
std	3.611337	32.016127	0.556363	0.182924	0.294146	4699.844075
min	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	4.000000	0.000000	0.000000	0.000000	41.000000
50%	1.000000	9.000000	0.000000	0.000000	0.000000	170.000000
75%	1.000000	30.000000	0.000000	0.000000	0.000000	914.000000
max	23.000000	163.000000	20.000000	1.000000	1.000000	115949.000000

8 rows × 38 columns

In [32]:
```
dataset_excluding_out.groupby('mediaType').agg({'duration':'mean', 'mediaTy
```

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 [10]:
```
dataset.isna().sum()
```

Out[10]:
```
title                          0
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 [11]:
```
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                        12101 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 [12]:
```
dataset.describe().T
```

Out[12]:

	count	mean	std	min	25%	50%	
eps	12101.0	13.393356	57.925097	1.000	1.000	2.000	
duration	7465.0	24.230141	31.468171	1.000	4.000	8.000	
years_running	12101.0	0.283200	1.152234	0.000	0.000	0.000	
studios_colab	12101.0	0.051649	0.221326	0.000	0.000	0.000	
contentWarn	12101.0	0.115362	0.319472	0.000	0.000	0.000	
watched	12101.0	2862.605694	7724.347024	0.000	55.000	341.000	20
watching	12101.0	256.334435	1380.840902	0.000	2.000	14.000	1
wantWatch	12101.0	1203.681431	2294.327380	0.000	49.000	296.000	12
dropped	12101.0	151.568383	493.931710	0.000	3.000	12.000	
rating	12101.0	2.949037	0.827385	0.844	2.304	2.965	
votes	12101.0	2088.124700	5950.332228	10.000	34.000	219.000	14
tag_Based_on_a_Manga	12101.0	0.290802	0.454151	0.000	0.000	0.000	
tag_Comedy	12101.0	0.272870	0.445453	0.000	0.000	0.000	
tag_Action	12101.0	0.231221	0.421631	0.000	0.000	0.000	
tag_Fantasy	12101.0	0.181555	0.385493	0.000	0.000	0.000	
tag_Sci_Fi	12101.0	0.166267	0.372336	0.000	0.000	0.000	
tag_Shounen	12101.0	0.144864	0.351978	0.000	0.000	0.000	
tag_Original_Work	12101.0	0.135195	0.341946	0.000	0.000	0.000	
tag_Non_Human_Protagonists	12101.0	0.112470	0.315957	0.000	0.000	0.000	
tag_Drama	12101.0	0.106107	0.307987	0.000	0.000	0.000	
tag_Adventure	12101.0	0.103793	0.305005	0.000	0.000	0.000	
tag_Family_Friendly	12101.0	0.097017	0.295993	0.000	0.000	0.000	
tag_Short_Episodes	12101.0	0.096934	0.295880	0.000	0.000	0.000	
tag_School_Life	12101.0	0.092306	0.289470	0.000	0.000	0.000	
tag_Romance	12101.0	0.092141	0.289237	0.000	0.000	0.000	
tag_Shorts	12101.0	0.089662	0.285709	0.000	0.000	0.000	
tag_Slice_of_Life	12101.0	0.080820	0.272569	0.000	0.000	0.000	
tag_Seinen	12101.0	0.077101	0.266763	0.000	0.000	0.000	
tag_Supernatural	12101.0	0.070903	0.256674	0.000	0.000	0.000	
tag_Magic	12101.0	0.064292	0.245283	0.000	0.000	0.000	
tag_Animal_Protagonists	12101.0	0.060326	0.238099	0.000	0.000	0.000	
tag_Ecchi	12101.0	0.057433	0.232678	0.000	0.000	0.000	
tag_Mecha	12101.0	0.054541	0.227091	0.000	0.000	0.000	
tag_Based_on_a_Light_Novel	12101.0	0.053384	0.224807	0.000	0.000	0.000	
tag_CG_Animation	12101.0	0.050079	0.218116	0.000	0.000	0.000	
tag_Superpowers	12101.0	0.044624	0.206486	0.000	0.000	0.000	
tag_Others	12101.0	0.090654	0.287128	0.000	0.000	0.000	
tag_missing	12101.0	0.025866	0.158741	0.000	0.000	0.000	

In [5]:
```
dataset.drop(columns = ['title', 'description'], axis=1, inplace=True)
```

In [6]:
```
dataset.head()
```

Out[6]:

	mediaType	eps	duration	ongoing	sznOfRelease	years_running	studio_primary	studios
0	TV	64	NaN	False	Spring	1	Bones	
1	Movie	1	107.0	False	is_missing	0	Others	
2	Movie	1	130.0	False	is_missing	0	Kyoto Animation	
3	TV	10	NaN	False	Fall	0	Production I.G	
4	TV	10	NaN	False	Spring	0	Others	

5 rows × 42 columns

In [15]:
```
dataset.rating.describe()
```

Out[15]:
```
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 [7]:
```
dataset.dropna(inplace=True)
dataset.shape
```

Out[7]:
```
(7465, 42)
```

In [17]:
```
12000-7465
```

Out[17]:
```
4535
```

Creating a Fuction so Each and every variable can be displayed in graphs using univartiate analysis

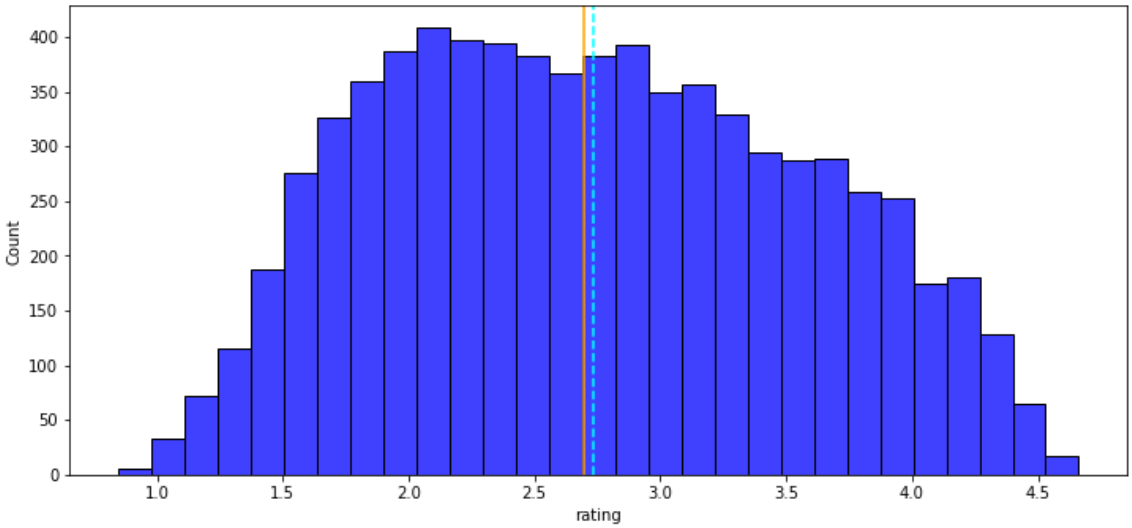
In [18]:
```
def continuos_univariate_analysis(data,
                                  feature,
                                  figsize=(12, 8),
                                  kde=False,
                                  bins=None):
    f1, (ax_box,
         ax_hist) = plt.subplots(nrows=2,
                                 sharex=True,
                                 gridspec_kw={'height_ratios': (0.25, 0.75)
                                 figsize=figsize)
    sns.color_palette("viridis", as_cmap=True)
    sns.boxplot(data=data,
                x=feature,
                ax=ax_box,
                showmeans=True,
                color='yellow')
    sns.histplot(data=data,
                 x=feature,
                 ax=ax_hist,
                 showmeans=True,
                 color='crest',
                 bins=bins,
                 kde=kde) if 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="-")
```

In [19]:
```
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))
    else:
        plt.figure(figsize=(n + 1, 5))
    plt.xticks(rotation=90, fontsize=15)
    ax = sns.countplot(
        data=data,
        x=feature,
        palette="flare",
        order=data[feature].value_counts().index[:n].sort_values(
            ascending=False))
    for p in ax.patches:
        if perc == True:
            label = "{:.1f}%".format(100 * p.get_height() / total)
        else:
            label = 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()
```

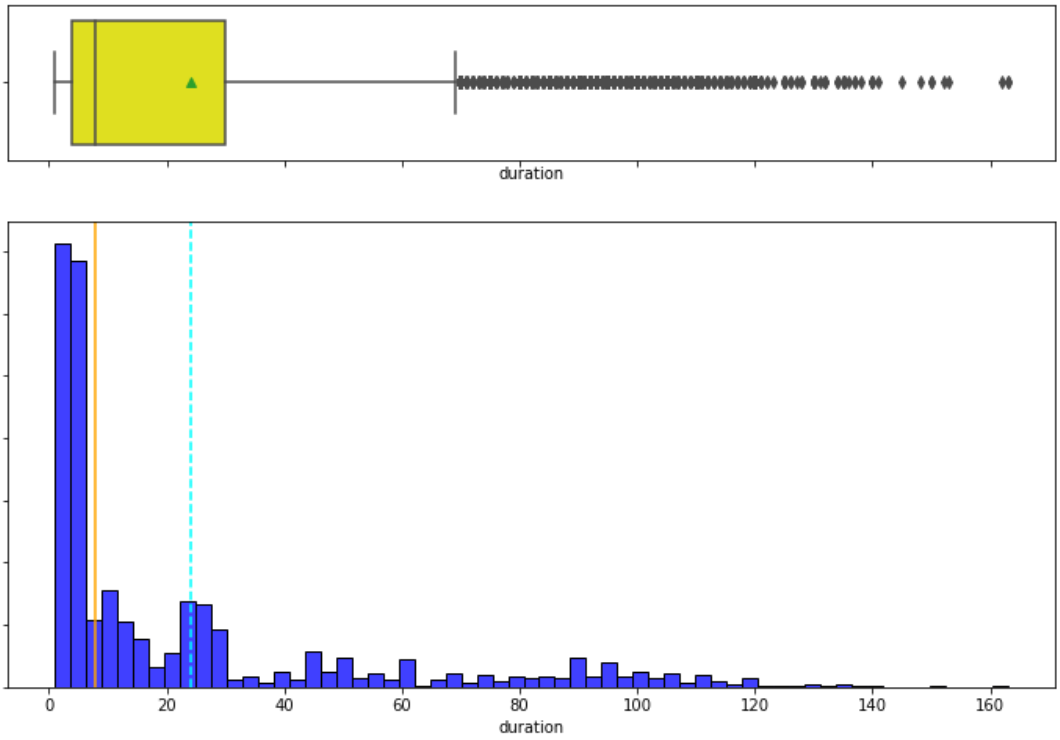
In [20]:
```
dataset.columns
```

Out[20]:
```
Index(['mediaType', 'eps', 'duration', 'ongoing', 'sznOfRelease',
       'years_running', 'studio_primary', 'studios_colab', 'contentWarn',
       'watched', 'watching', 'wantWatch', 'dropped', 'rating', '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_Ec
chi',
       'tag_Mecha', 'tag_Based_on_a_Light_Novel', 'tag_CG_Animation',
       'tag_Superpowers', 'tag_Others', 'tag_missing'],
      dtype='object')
```

In [21]:
```
continuos_univariate_analysis(dataset, 'rating')
```



In [22]:
```
continuos_univariate_analysis(dataset, 'duration')
```



In [23]:
```
dataset[dataset['duration'] >=80]['rating'].mean()
```

Out[23]: 3.569473225404726

In [24]:
```
dataset[dataset['duration'] >=100]['rating'].mean()
```

Out[24]: 3.729269121813027

In [25]:
```
dataset[dataset['duration'] >=110]['rating'].mean()
```

Out[25]: 3.7585191256830592

In [26]:
```
dataset[(dataset['duration'] >=5) & (dataset['duration']<=30)]['rating'].me
```

Out[26]: 2.789046975546977

In [6]:
```
dataset['Duration_category'] = pd.cut(dataset['duration'], bins = [0,20,40,
                                      labels = ['0-20', '20-40', '40-60', '60-80
```

In [15]:
```
exp = dataset.groupby(['mediaType', 'Duration_category']).agg({'rating' : [
```

In [18]:
```
exp.columns = exp.columns.droplevel(0)
```

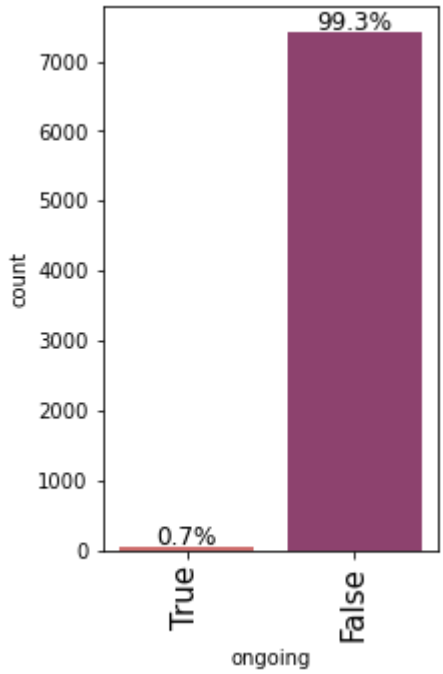
In [20]:
```
exp.columns = ['Mediatype', 'duration', 'rating']
```

In [22]:
```
exp.pivot_table(columns='duration', values='rating', index='Mediatype')
```

Out[22]:

duration	0-20	20-40	40-60	60-80	>80
Mediatype					
DVD Special	2.914576	3.445481	3.207250	NaN	2.838000
Movie	2.091484	2.958930	3.301082	3.146733	3.619978
Music Video	2.374712	2.292750	2.689000	NaN	NaN
OVA	2.716476	3.133294	2.751825	3.113306	3.151345
Other	2.394337	3.322878	2.991667	2.722000	3.558750
TV	2.428329	3.507727	3.452000	NaN	2.587000
TV Special	2.636727	3.229709	3.338108	3.143355	3.667397
Web	2.353887	3.331457	2.521500	2.979250	3.518500
is_missing	2.327500	3.277500	3.448667	2.718500	NaN

In [27]:
```
discrete_univariate_analysis(dataset, "ongoing", perc=True)
```



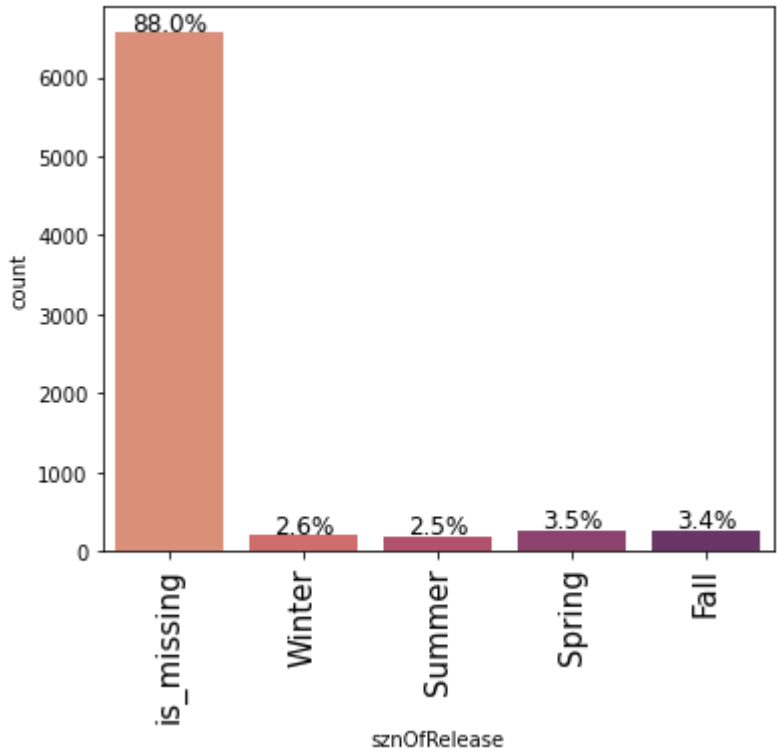
In [28]:
```
dataset[dataset['ongoing'] == True]['rating'].mean()
```

Out[28]: 3.1624600000000003

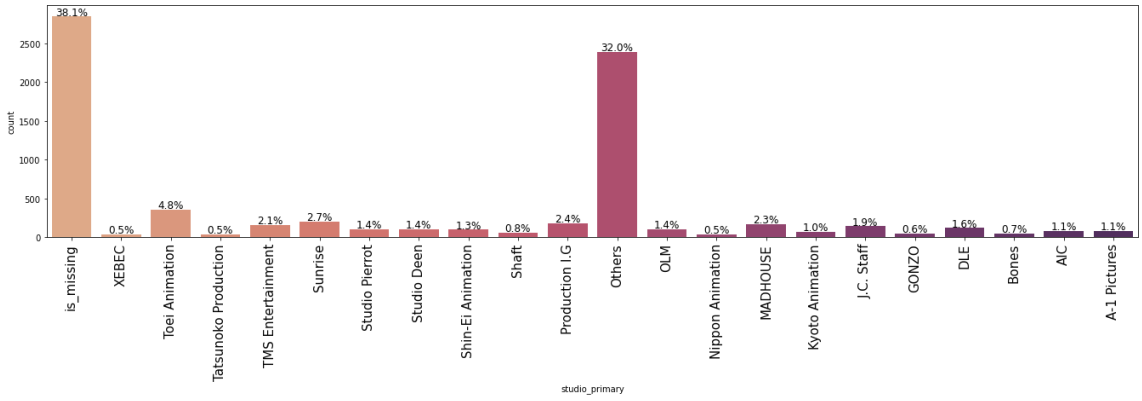
In [29]:
```
dataset[dataset['ongoing'] == True]['duration'].mean()
```

Out[29]: 8.94

In [30]:
```
discrete_univariate_analysis(dataset, "sznOfRelease", perc=True)
```



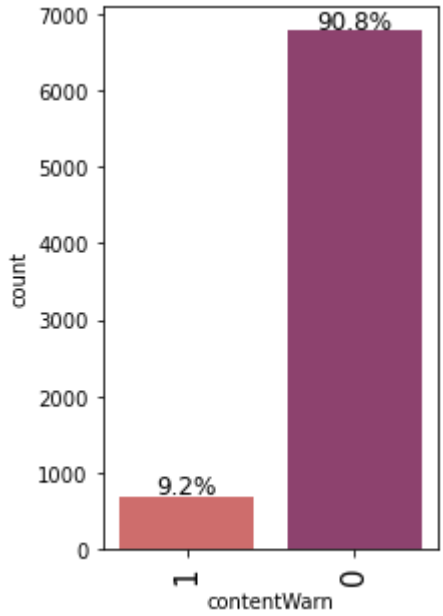
In [31]:
```
discrete_univariate_analysis(dataset, "studio_primary", perc=True)
```



In [34]:
```
dataset[dataset['rating'] > 4]['studio_primary'].value_counts(normalize=Tru
```

Out[34]:
```
Others                 38.25
Production I.G          8.42
is_missing              7.02
MADHOUSE                5.96
TMS Entertainment       5.96
Sunrise                 4.91
Kyoto Animation         4.04
Studio Deen             3.68
A-1 Pictures            3.68
Bones                   3.68
Toei Animation          3.51
Shaft                   3.33
J.C. Staff              3.16
Studio Pierrot          2.46
Shin-Ei Animation       0.35
Nippon Animation        0.35
Tatsunoko Production    0.35
XEBEC                   0.35
OLM                     0.35
GONZO                   0.18
Name: studio_primary, dtype: float64
```

In [35]:
```
discrete_univariate_analysis(dataset, 'contentWarn', perc=True)
```

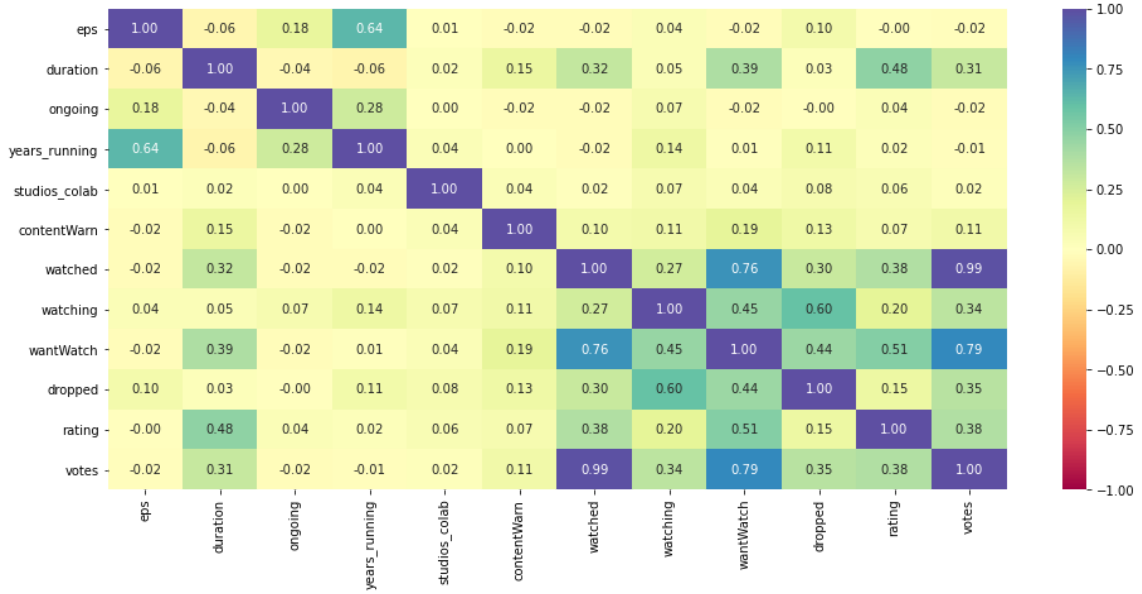


In [39]:
```
corr_cols = [item for item in dataset.columns if "tag" not in item]
```

In [40]:
```
corr_cols
```

Out[40]:
```
['mediaType',
 'eps',
 'duration',
 'ongoing',
 'sznOfRelease',
 'years_running',
 'studio_primary',
 'studios_colab',
 'contentWarn',
 'watched',
 'watching',
 'wantWatch',
 'dropped',
 'rating',
 'votes']
```

In [43]:
```
plt.figure(figsize=(16,7))
sns.heatmap(dataset[corr_cols].corr(), annot=True, vmin = -1, vmax= 1, fmt=
plt.show()
```



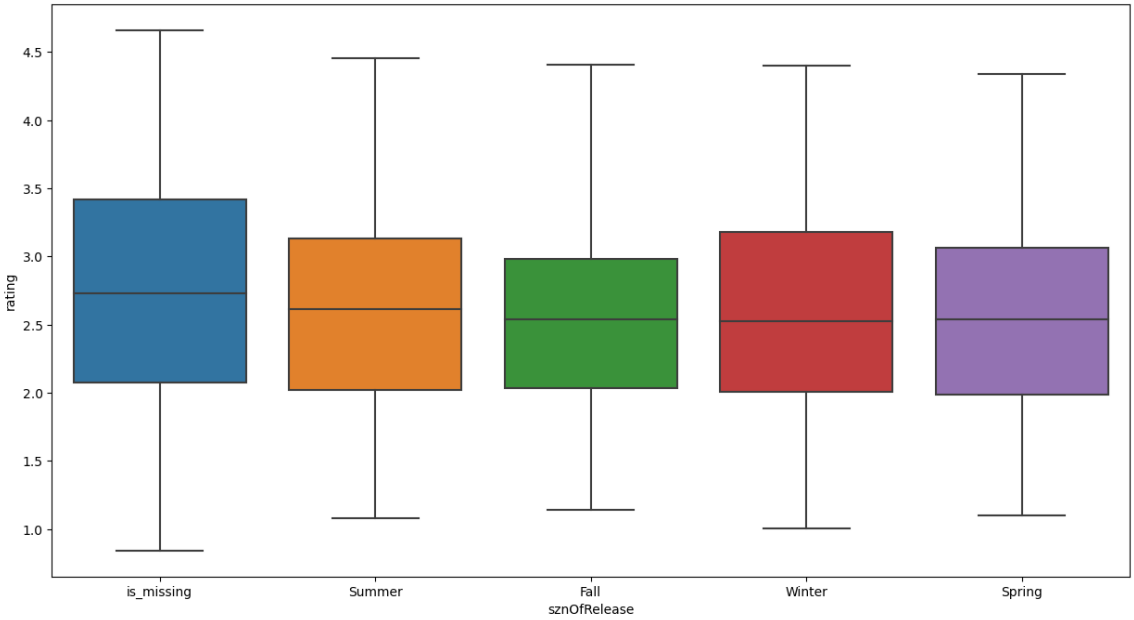
In [8]:
```
dataset.drop(columns= ['eps', 'watched'], inplace=True)
```

In [9]:
```
dataset.shape
```

Out[9]:
```
(7465, 40)
```

In [10]:
```
plt.figure(figsize=(15,8))
sns.boxplot(x = 'sznOfRelease', y='rating', data=dataset)
```

Out[10]: <AxesSubplot:xlabel='sznOfRelease', ylabel='rating'>



Regression Model

In [12]:
```
x = dataset.drop(['rating'], axis=1)
y = dataset['rating']
```

In [13]:
```
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 [14]:
```
x = pd.get_dummies(x, columns=x.select_dtypes(include=['object', 'category'
x.head()
```

Out[14]:

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

5 rows × 69 columns

In [15]:
```
x.drop(columns='ongoing', inplace=True)
```

In [54]:
```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7465 entries, 1 to 12100
Data columns (total 69 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   watched                               7465 non-null   float64
 5   watching                              7465 non-null   int64  
 6   wantWatch                             7465 non-null   int64  
 7   dropped                               7465 non-null   int64  
 8   votes                                 7465 non-null   int64  
 9   tag_Based_on_a_Manga                  7465 non-null   int64  
 10  tag_Comedy                            7465 non-null   int64  
 11  tag_Action                            7465 non-null   int64  
 12  tag_Fantasy                           7465 non-null   int64  
 13  tag_Sci_Fi                            7465 non-null   int64  
 14  tag_Shounen                           7465 non-null   int64  
 15  tag_Original_Work                     7465 non-null   int64  
 16  tag_Non_Human_Protagonists            7465 non-null   int64  
 17  tag_Drama                             7465 non-null   int64  
 18  tag_Adventure                         7465 non-null   int64  
 19  tag_Family_Friendly                   7465 non-null   int64  
 20  tag_Short_Episodes                    7465 non-null   int64  
 21  tag_School_Life                       7465 non-null   int64  
 22  tag_Romance                           7465 non-null   int64  
 23  tag_Shorts                            7465 non-null   int64  
 24  tag_Slice_of_Life                     7465 non-null   int64  
 25  tag_Seinen                            7465 non-null   int64  
 26  tag_Supernatural                      7465 non-null   int64  
 27  tag_Magic                             7465 non-null   int64  
 28  tag_Animal_Protagonists               7465 non-null   int64  
 29  tag_Ecchi                             7465 non-null   int64  
 30  tag_Mecha                             7465 non-null   int64  
 31  tag_Based_on_a_Light_Novel            7465 non-null   int64  
 32  tag_CG_Animation                      7465 non-null   int64  
 33  tag_Superpowers                       7465 non-null   int64  
 34  tag_Others                            7465 non-null   int64  
 35  tag_missing                           7465 non-null   int64  
 36  mediaType_Movie                       7465 non-null   uint8  
 37  mediaType_Music Video                 7465 non-null   uint8  
 38  mediaType_OVA                         7465 non-null   uint8  
 39  mediaType_Other                       7465 non-null   uint8  
 40  mediaType_TV                          7465 non-null   uint8  
 41  mediaType_TV Special                  7465 non-null   uint8  
 42  mediaType_Web                         7465 non-null   uint8  
 43  mediaType_is_missing                  7465 non-null   uint8  
 44  sznOfRelease_Spring                   7465 non-null   uint8  
 45  sznOfRelease_Summer                   7465 non-null   uint8  
 46  sznOfRelease_Winter                   7465 non-null   uint8  
 47  sznOfRelease_is_missing               7465 non-null   uint8  
 48  studio_primary_AIC                    7465 non-null   uint8  
 49  studio_primary_Bones                  7465 non-null   uint8  
 50  studio_primary_DLE                    7465 non-null   uint8  
 51  studio_primary_GONZO                  7465 non-null   uint8  
 52  studio_primary_J.C. Staff             7465 non-null   uint8  
 53  studio_primary_Kyoto Animation        7465 non-null   uint8  
 54  studio_primary_MADHOUSE               7465 non-null   uint8  
 55  studio_primary_Nippon Animation       7465 non-null   uint8  
```

```
 56  studio_primary_OLM                    7465 non-null   uint8
 57  studio_primary_Others                 7465 non-null   uint8
 58  studio_primary_Production I.G         7465 non-null   uint8
 59  studio_primary_Shaft                  7465 non-null   uint8
 60  studio_primary_Shin-Ei Animation      7465 non-null   uint8
 61  studio_primary_Studio Deen            7465 non-null   uint8
 62  studio_primary_Studio Pierrot         7465 non-null   uint8
 63  studio_primary_Sunrise                7465 non-null   uint8
 64  studio_primary_TMS Entertainment      7465 non-null   uint8
 65  studio_primary_Tatsunoko Production   7465 non-null   uint8
 66  studio_primary_Toei Animation         7465 non-null   uint8
 67  studio_primary_XEBEC                  7465 non-null   uint8
 68  studio_primary_is_missing             7465 non-null   uint8
dtypes: float64(2), int64(34), uint8(33)
memory usage: 2.6 MB
```

In [16]:
```
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_err
```

In [18]:
```
X_train, X_test, Y_train, Y_test = train_test_split(x,y, test_size=0.2, ran
```

In [19]:
```
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 [20]:
```
lin_model = LinearRegression()
lin_model.fit(X_train, Y_train)
```

Out[20]:
```
LinearRegression()
```

In [21]:
```
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 [22]:
```
print("Training Data Performance")
lin_model_train = Model_performance(lin_model, X_train, Y_train)
lin_model_train
```

```
Training Data Performance
```

Out[22]:

	RMSE	R2 Score
0	0.580109	0.515527

In [83]:
```
print("Test Data Performance")
lin_model_test = Model_performance(lin_model, X_test, Y_test)
lin_model_test
```

```
Test Data Performance
```

Out[83]:

	RMSE	R2 Score
0	0.564058	0.519397

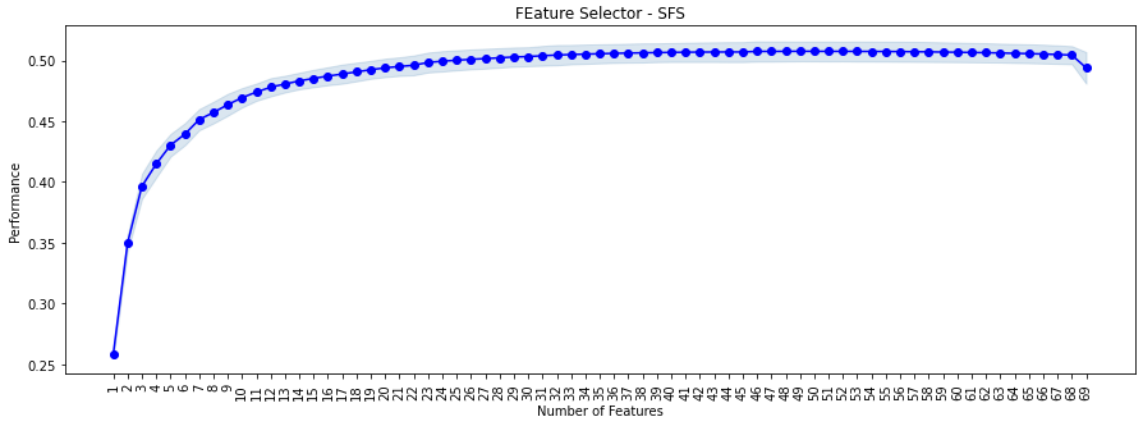
In [64]:
```
x.columns
```

Out[64]:
```
Index(['duration', 'years_running', 'studios_colab', 'contentWarn', 'watch
ed',
       'watching', 'wantWatch', 'dropped', 'votes', 'tag_Based_on_a_Mang
a',
       'tag_Comedy', 'tag_Action', 'tag_Fantasy', 'tag_Sci_Fi', 'tag_Shoun
en',
       '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_Lif
e',
       'tag_Seinen', 'tag_Supernatural', 'tag_Magic',
       'tag_Animal_Protagonists', 'tag_Ecchi', 'tag_Mecha',
       'tag_Based_on_a_Light_Novel', 'tag_CG_Animation', 'tag_Superpower
s',
       'tag_Others', 'tag_missing', 'mediaType_Movie', 'mediaType_Music Vi
deo',
       'mediaType_OVA', 'mediaType_Other', 'mediaType_TV',
       'mediaType_TV Special', 'mediaType_Web', 'mediaType_is_missing',
       'sznOfRelease_Spring', 'sznOfRelease_Summer', 'sznOfRelease_Winte
r',
       'sznOfRelease_is_missing', 'studio_primary_AIC', 'studio_primary_Bo
nes',
       'studio_primary_DLE', 'studio_primary_GONZO',
       'studio_primary_J.C. Staff', 'studio_primary_Kyoto Animation',
       'studio_primary_MADHOUSE', 'studio_primary_Nippon Animation',
       'studio_primary_OLM', 'studio_primary_Others',
       'studio_primary_Production I.G', 'studio_primary_Shaft',
       'studio_primary_Shin-Ei Animation', 'studio_primary_Studio Deen',
       'studio_primary_Studio Pierrot', 'studio_primary_Sunrise',
       'studio_primary_TMS Entertainment',
       'studio_primary_Tatsunoko Production', 'studio_primary_Toei Animati
on',
       'studio_primary_XEBEC', 'studio_primary_is_missing'],
      dtype='object')
```

Feature Selection

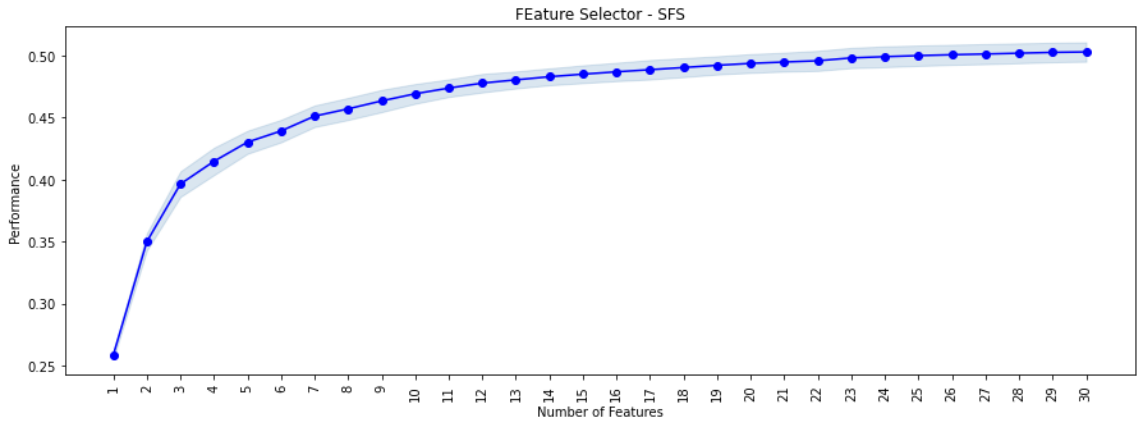
In [69]:
```
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
reg = LinearRegression()
sfs = SFS(reg, k_features = X_train.shape[1],
          forward = True, floating=False, scoring='r2', n_jobs= -1, cv=5)
sfs = sfs.fit(X_train, Y_train)
```

In [71]:
```
from mlxtend.plotting import plot_sequential_feature_selection as plot_sfs
fig1 = plot_sfs(sfs.get_metric_dict(), kind='std_err', figsize=(15,5))
plt.title("FEature Selector - SFS")
plt.xticks(rotation=90)
plt.show()
```



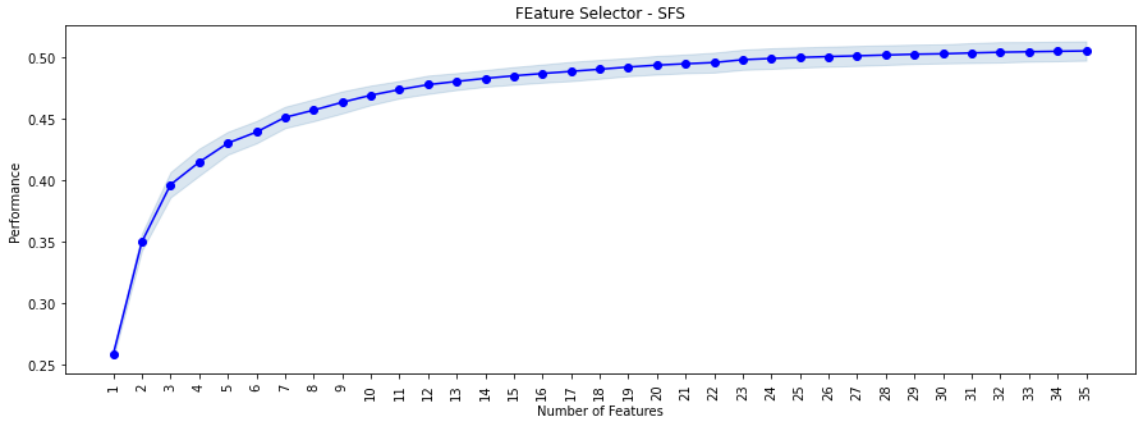
In [72]:
```
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
reg = LinearRegression()
sfs = SFS(reg, k_features = 30,
          forward = True, floating=False, scoring='r2', n_jobs= -1, cv=5)
sfs = sfs.fit(X_train, Y_train)
```

In [73]:
```
from mlxtend.plotting import plot_sequential_feature_selection as plot_sfs
fig1 = plot_sfs(sfs.get_metric_dict(), kind='std_err', figsize=(15,5))
plt.title("FEature Selector - SFS")
plt.xticks(rotation=90)
plt.show()
```



In [74]:
```
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
reg = LinearRegression()
sfs = SFS(reg, k_features = 35,
          forward = True, floating=False, scoring='r2', n_jobs= -1, cv=5)
sfs = sfs.fit(X_train, Y_train)
```

In [75]:
```
from mlxtend.plotting import plot_sequential_feature_selection as plot_sfs
fig1 = plot_sfs(sfs.get_metric_dict(), kind='std_err', figsize=(15,5))
plt.title("FEature Selector - SFS")
plt.xticks(rotation=90)
plt.show()
```



In [76]:
```
feature_index = list(sfs.k_feature_idx_)
print(feature_index)
```

```
[0, 1, 3, 4, 6, 8, 9, 11, 12, 14, 17, 19, 20, 21, 23, 24, 25, 26, 29, 31,
36, 38, 39, 40, 42, 49, 50, 57, 58, 59, 61, 63, 64, 66, 68]
```

In [78]:
```
X_train.columns[feature_index]
```

Out[78]:
```
Index(['duration', 'years_running', 'contentWarn', 'watched', 'wantWatch',
       'votes', 'tag_Based_on_a_Manga', 'tag_Action', 'tag_Fantasy',
       'tag_Shounen', 'tag_Drama', 'tag_Family_Friendly', 'tag_Short_Episo
des',
       'tag_School_Life', 'tag_Shorts', 'tag_Slice_of_Life', 'tag_Seinen',
       'tag_Supernatural', 'tag_Ecchi', 'tag_Based_on_a_Light_Novel',
       'mediaType_Movie', 'mediaType_OVA', 'mediaType_Other', 'mediaType_T
V',
       'mediaType_Web', 'studio_primary_Bones', 'studio_primary_DLE',
       'studio_primary_Others', 'studio_primary_Production I.G',
       'studio_primary_Shaft', 'studio_primary_Studio Deen',
       'studio_primary_Sunrise', 'studio_primary_TMS Entertainment',
       'studio_primary_Toei Animation', 'studio_primary_is_missing'],
      dtype='object')
```

In [80]:
```
X_train_final = X_train[X_train.columns[feature_index]]
X_test_final = X_test[X_test.columns[feature_index]]
```

In [82]:
```
lin_model_v2 = LinearRegression()
lin_model_v2.fit(X_train_final, Y_train)
```

Out[82]:
```
LinearRegression()
```

In [84]:
```
print("Training Data Performance")
lin_model_train = Model_performance(lin_model, X_train, Y_train)
lin_model_train
```

```
Training Data Performance
```

Out[84]:

	RMSE	R2 Score
0	0.578312	0.518524

In [85]:
```
print("Training Data Performance")
lin_model_train = Model_performance(lin_model, X_test, Y_test)
lin_model_train
```

```
Training Data Performance
```

Out[85]:

	RMSE	R2 Score
0	0.564058	0.519397

In [88]:
```
X_train.columns[feature_index]
```

Out[88]:
```
Index(['duration', 'years_running', 'contentWarn', 'watched', 'wantWatch',
       'votes', 'tag_Based_on_a_Manga', 'tag_Action', 'tag_Fantasy',
       'tag_Shounen', 'tag_Drama', 'tag_Family_Friendly', 'tag_Short_Episo
des',
       'tag_School_Life', 'tag_Shorts', 'tag_Slice_of_Life', 'tag_Seinen',
       'tag_Supernatural', 'tag_Ecchi', 'tag_Based_on_a_Light_Novel',
       'mediaType_Movie', 'mediaType_OVA', 'mediaType_Other', 'mediaType_T
V',
       'mediaType_Web', 'studio_primary_Bones', 'studio_primary_DLE',
       'studio_primary_Others', 'studio_primary_Production I.G',
       'studio_primary_Shaft', 'studio_primary_Studio Deen',
       'studio_primary_Sunrise', 'studio_primary_TMS Entertainment',
       'studio_primary_Toei Animation', 'studio_primary_is_missing'],
      dtype='object')
```

In [87]:
```
X_train.head(2)
```

Out[87]:

	duration	years_running	studios_colab	contentWarn	watched	watching	wantWatch
8843	1.0	0	0	0	23.0	0	17
1599	24.0	0	0	0	1278.0	24	1117

2 rows × 69 columns