In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
data=pd.read_csv(r"D:\project\fandango_score_comparison.csv")
```

In [4]:
```
data
```

Out[4]:

	FILM	RottenTomatoes	RottenTomatoes_User	Metacritic	Metacritic_User	IMD
0	Avengers: Age of Ultron (2015)	74	8.6	66	7.1	7
1	Cinderella (2015)	85	8.0	67	7.5	7
2	Ant-Man (2015)	80	9.0	64	8.1	7
3	Do You Believe? (2015)	18	8.4	22	4.7	5
4	Hot Tub Time Machine 2 (2015)	14	2.8	29	3.4	5
...	...	...	...	...	...	
141	Mr. Holmes (2015)	87	7.8	67	7.9	7
142	'71 (2015)	97	8.2	83	7.5	7
143	Two Days, One Night (2014)	97	7.8	89	8.8	7
144	Gett: The Trial of Viviane Amsalem (2015)	100	8.1	90	7.3	7
145	Kumiko, The Treasure Hunter (2015)	87	6.3	68	6.4	6

146 rows × 22 columns

In [5]:

```
data.isnull().sum()
```

Out[5]:
```
FILM                          0
RottenTomatoes                0
RottenTomatoes_User           0
Metacritic                    0
Metacritic_User               0
IMDB                          0
Fandango_Stars                0
Fandango_Ratingvalue          0
RT_norm                       0
RT_user_norm                  0
Metacritic_norm               0
Metacritic_user_nom           0
IMDB_norm                     0
RT_norm_round                 0
RT_user_norm_round            0
Metacritic_norm_round         0
Metacritic_user_norm_round    0
IMDB_norm_round               0
Metacritic_user_vote_count    0
IMDB_user_vote_count          0
Fandango_votes                0
Fandango_Difference           0
dtype: int64
```

In [6]:
```
data.dtypes
```

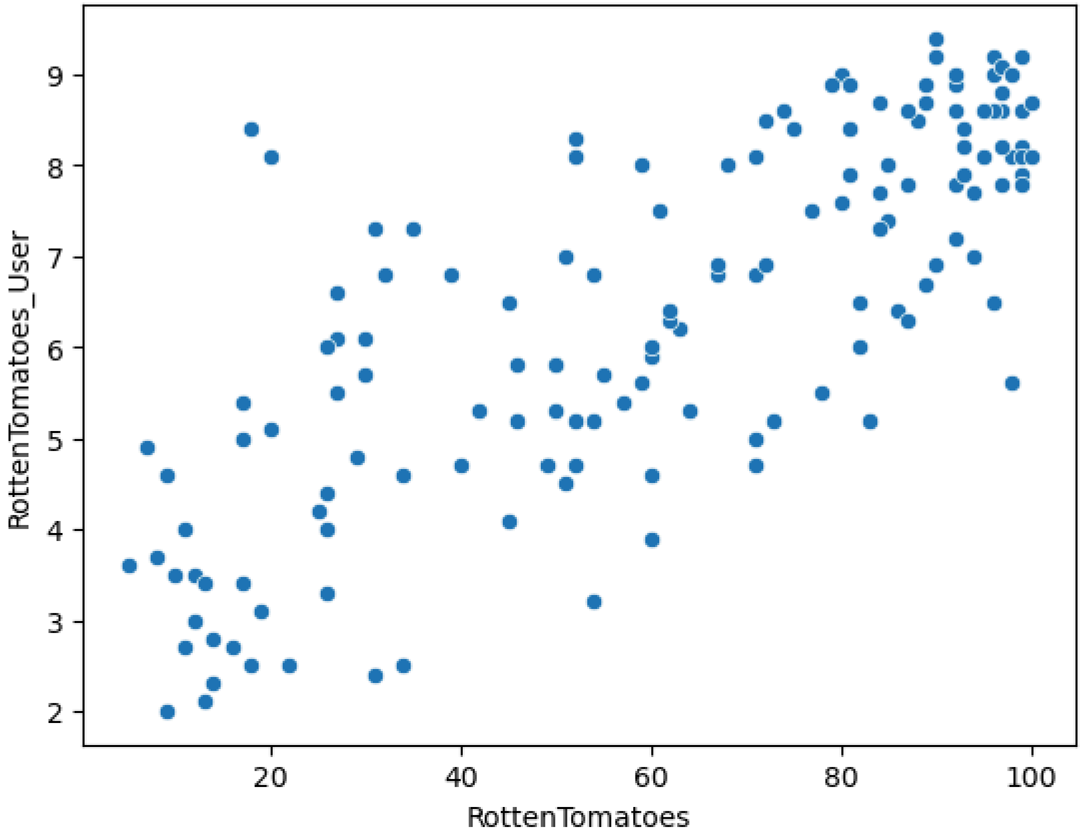
Out[6]:
```
FILM                           object
RottenTomatoes                  int64
RottenTomatoes_User           float64
Metacritic                      int64
Metacritic_User               float64
IMDB                          float64
Fandango_Stars                float64
Fandango_Ratingvalue          float64
RT_norm                       float64
RT_user_norm                  float64
Metacritic_norm               float64
Metacritic_user_nom           float64
IMDB_norm                     float64
RT_norm_round                 float64
RT_user_norm_round            float64
Metacritic_norm_round         float64
Metacritic_user_norm_round    float64
IMDB_norm_round               float64
Metacritic_user_vote_count      int64
IMDB_user_vote_count            int64
Fandango_votes                  int64
Fandango_Difference           float64
dtype: object
```

In [7]:
```
for i in data.columns:
    print(i,':','\n',data[i].unique())
```

```
FILM :
 ['Avengers: Age of Ultron (2015)' 'Cinderella (2015)' 'Ant-Man (2015)'
 'Do You Believe? (2015)' 'Hot Tub Time Machine 2 (2015)'
 'The Water Diviner (2015)' 'Irrational Man (2015)' 'Top Five (2014)'
 'Shaun the Sheep Movie (2015)' 'Love & Mercy (2015)'
 'Far From The Madding Crowd (2015)' 'Black Sea (2015)' 'Leviathan (2014)'
 'Unbroken (2014)' 'The Imitation Game (2014)' 'Taken 3 (2015)'
 'Ted 2 (2015)' 'Southpaw (2015)'
 'Night at the Museum: Secret of the Tomb (2014)' 'Pixels (2015)'
 'McFarland, USA (2015)' 'Insidious: Chapter 3 (2015)'
 'The Man From U.N.C.L.E. (2015)' 'Run All Night (2015)'
 'Trainwreck (2015)' 'Selma (2014)' 'Ex Machina (2015)'
 'Still Alice (2015)' 'Wild Tales (2014)' 'The End of the Tour (2015)'
 'Red Army (2015)' 'When Marnie Was There (2015)'
 'The Hunting Ground (2015)' 'The Boy Next Door (2015)' 'Aloha (2015)'
 'The Loft (2015)' '5 Flights Up (2015)' 'Welcome to Me (2015)'
 'Saint Laurent (2015)' 'Maps to the Stars (2015)'
 "I'll See You In My Dreams (2015)" 'Timbuktu (2015)' 'About Elly (2015)'
 'The Diary of a Teenage Girl (2015)'
 'Kingsman: The Secret Service (2015)' 'Tomorrowland (2015)'
 'The Divergent Series: Insurgent (2015)' 'Annie (2014)'
 'Fantastic Four (2015)' 'Terminator Genisys (2015)'
 'Pitch Perfect 2 (2015)' 'Entourage (2015)' 'The Age of Adaline (2015)'
 'Hot Pursuit (2015)' 'The DUFF (2015)' 'Black or White (2015)'
 'Project Almanac (2015)' 'Ricki and the Flash (2015)'
 'Seventh Son (2015)' 'Mortdecai (2015)' 'Unfinished Business (2015)'
 'American Ultra (2015)' 'True Story (2015)' 'Child 44 (2015)'
 'Dark Places (2015)' 'Birdman (2014)' 'The Gift (2015)'
 'Unfriended (2015)' 'Monkey Kingdom (2015)' 'Mr. Turner (2014)'
 'Seymour: An Introduction (2015)' 'The Wrecking Crew (2015)'
 'American Sniper (2015)' 'Furious 7 (2015)'
 'The Hobbit: The Battle of the Five Armies (2014)' 'San Andreas (2015)'
 'Straight Outta Compton (2015)' 'Vacation (2015)' 'Chappie (2015)'
 'Poltergeist (2015)' 'Paper Towns (2015)' 'Big Eyes (2014)'
 'Blackhat (2015)' 'Self/less (2015)' 'Sinister 2 (2015)'
 'Little Boy (2015)' 'Me and Earl and The Dying Girl (2015)'
 'Maggie (2015)' 'Mad Max: Fury Road (2015)' 'Spy (2015)'
 'The SpongeBob Movie: Sponge Out of Water (2015)' 'Paddington (2015)'
 'Dope (2015)' 'What We Do in the Shadows (2015)' 'The Overnight (2015)'
 'The Salt of the Earth (2015)' 'Song of the Sea (2014)'
 'Fifty Shades of Grey (2015)' 'Get Hard (2015)' 'Focus (2015)'
 'Jupiter Ascending (2015)' 'The Gallows (2015)'
 'The Second Best Exotic Marigold Hotel (2015)' 'Strange Magic (2015)'
 'The Gunman (2015)' 'Hitman: Agent 47 (2015)' 'Cake (2015)'
 'The Vatican Tapes (2015)' 'A Little Chaos (2015)'
 'The 100-Year-Old Man Who Climbed Out the Window and Disappeared (2015)'
 'Escobar: Paradise Lost (2015)' 'Into the Woods (2014)'
 'It Follows (2015)' 'Inherent Vice (2014)' 'A Most Violent Year (2014)'
 "While We're Young (2015)" 'Clouds of Sils Maria (2015)'
 'Testament of Youth (2015)' 'Infinitely Polar Bear (2015)'
 'Phoenix (2015)' 'The Wolfpack (2015)'
 'The Stanford Prison Experiment (2015)' 'Tangerine (2015)'
 'Magic Mike XXL (2015)' 'Home (2015)' 'The Wedding Ringer (2015)'
 'Woman in Gold (2015)' 'The Last Five Years (2015)'
 'Mission: Impossible â€“ Rogue Nation (2015)' 'Amy (2015)'
 'Jurassic World (2015)' 'Minions (2015)' 'Max (2015)'
 'Paul Blart: Mall Cop 2 (2015)' 'The Longest Ride (2015)'
 'The Lazarus Effect (2015)' 'The Woman In Black 2 Angel of Death (2015)'
 'Danny Collins (2015)' 'Spare Parts (2015)' 'Serena (2015)'
 'Inside Out (2015)' 'Mr. Holmes (2015)' "'71 (2015)"
 'Two Days, One Night (2014)' 'Gett: The Trial of Viviane Amsalem (2015)'
 'Kumiko, The Treasure Hunter (2015)']
RottenTomatoes :
 [ 74  85  80  18  14  63  42  86  99  89  84  82  51  90   9  46  59  50
  17  79  68  60  92  88  96  10  19  11  52  71  94  97  95  75  30  27
  26  67  32  54   8  39  34  64  12  45  93  98 100  72  81  61  31  55
  20  13  78  87  25  29  57  16  62   7  49  40  73  83  35   5  22  77]
RottenTomatoes_User :
 [8.6 8.  9.  8.4 2.8 6.2 5.3 6.4 8.2 8.7 7.7 6.  7.9 7.  9.2 4.6 5.8 5.4
 8.9 5.6 5.9 7.4 8.5 7.2 3.5 3.1 4.  4.7 4.5 7.8 8.1 6.1 2.  6.8 3.7 3.
 2.7 5.2 4.1 4.4 3.3 3.9 7.5 9.4 5.5 5.7 2.4 6.9 2.5 5.1 3.4 3.2 8.8 6.5
 4.2 4.8 6.3 5.  4.9 2.1 6.7 7.6 7.3 6.6 9.1 3.6 2.3 8.3]
Metacritic :
 [66 67 64 22 29 50 53 81 80 71 62 92 59 73 26 48 57 47 27 60 52 55 75 89
 78 72 77 84 82 30 40 24 91 87 58 42 33 38 63 51 31 56 45 54 32 41 39 88
 94 83 43 34 74 65 86 46 25 28 49 37 69 79 76 68 35 85 13 36 90]
Metacritic_User :
 [7.1 7.5 8.1 4.7 3.4 6.8 7.6 8.8 8.5 6.6 7.2 6.5 8.2 4.6 5.8 5.3 6.9 7.9
 7.3 6.  7.8 7.4 6.4 5.5 4.  2.4 9.6 6.3 5.4 4.8 2.5 5.7 5.2 3.7 7.  3.9
 3.2 3.8 8.  8.3 6.2 8.4 5.  5.9 8.7 8.6 4.5 6.1 4.3 3.3 6.7 4.9 4.4 8.9]
IMDB :
 [7.8 7.1 5.4 5.1 7.2 6.9 6.5 7.4 6.4 7.7 8.1 6.1 6.6 6.3 5.6 7.5 7.6 6.7
 8.2 7.9 4.6 5.5 5.9 7.  5.2 4.  7.3 4.9 6.2 8.4 5.  8.3 4.2 4.4 5.7 5.8
 6.  6.8 8.  4.3 8.6]
Fandango_Stars :
 [5.  3.5 4.5 4.  3. ]
Fandango_Ratingvalue :
 [4.5 3.  4.  3.5 4.1 4.6 3.6 3.1 4.2 3.7 2.7 3.2 4.8 4.3 3.8 2.8 3.3 3.9
 3.4 2.9 4.4]
RT_norm :
 [3.7  4.25 4.   0.9  0.7  3.15 2.1  4.3  4.95 4.45 4.2  4.1  2.55 4.5
 0.45 2.3  2.95 2.5  0.85 3.95 3.4  3.   4.6  4.4  4.8  0.5  0.95 0.55
 2.6  3.55 4.7  4.85 4.75 3.75 1.5  1.35 1.3  3.35 1.6  2.7  0.4  1.95
 1.7  3.2  0.6  2.25 4.65 4.9  5.   3.6  4.05 3.05 1.55 2.75 1.   0.65
 3.9  4.35 1.25 1.45 2.85 0.8  3.1  0.35 2.45 2.   3.65 4.15 1.75 0.25
 1.1  3.85]
RT_user_norm :
 [4.3  4.   4.5  4.2  1.4  3.1  2.65 3.2  4.1  4.35 3.85 3.   3.95 3.5
 4.6  2.3  2.9  2.7  4.45 2.8  2.95 3.7  4.25 3.6  1.75 1.55 2.   2.35
 2.25 3.9  4.05 3.05 1.   3.4  1.85 1.5  1.35 2.6  2.05 2.2  1.65 1.95
 3.75 4.7  2.75 2.85 1.2  3.45 1.25 2.55 1.7  1.6  4.4  3.25 2.1  2.4
 3.15 2.5  2.45 1.05 3.35 3.8  3.65 3.3  4.55 1.8  1.15 4.15]
Metacritic_norm :
 [3.3  3.35 3.2  1.1  1.45 2.5  2.65 4.05 4.   3.55 3.1  4.6  2.95 3.65
 1.3  2.4  2.85 2.35 1.35 3.   2.6  2.75 3.75 4.45 3.9  3.6  3.85 4.2
 4.1  1.5  2.   1.2  4.55 4.35 2.9  2.1  1.65 1.9  3.15 2.55 1.55 2.8
 2.25 2.7  1.6  2.05 1.95 4.4  4.7  4.15 2.15 1.7  3.7  3.25 4.3  2.3
 1.25 1.4  2.45 1.85 3.45 3.95 3.8  3.4  1.75 4.25 0.65 1.8  4.5 ]
Metacritic_user_nom :
 [3.55 3.75 4.05 2.35 1.7  3.4  3.8  4.4  4.25 3.3  3.6  3.25 4.1  2.3
 2.9  2.65 3.45 3.95 3.65 3.   3.9  3.7  3.2  2.75 2.   1.2  4.8  3.15
 2.7  2.4  1.25 2.85 2.6  1.85 3.5  1.95 1.6  1.9  4.   4.15 3.1  4.2
 2.5  2.95 4.35 4.3  2.25 3.05 2.15 1.65 3.35 2.45 2.2  4.45]
IMDB_norm :
 [3.9  3.55 2.7  2.55 3.6  3.45 3.25 3.7  3.2  3.85 4.05 3.05 3.3  3.15
 2.8  3.75 3.8  3.35 4.1  3.95 2.3  2.75 2.95 3.5  2.6  2.   3.65 2.45
 3.1  4.2  2.5  4.15 2.1  2.2  2.85 2.9  3.   3.4  4.   2.15 4.3 ]
RT_norm_round :
 [3.5 4.5 4.  1.  0.5 3.  2.  5.  2.5 1.5]
RT_user_norm_round :
 [4.5 4.  1.5 3.  2.5 3.5 2.  1. ]
Metacritic_norm_round :
 [3.5 3.  1.  1.5 2.5 4.  4.5 2.  0.5]
Metacritic_user_norm_round :
 [3.5 4.  2.5 1.5 4.5 3.  2.  1.  5. ]
IMDB_norm_round :
 [4.  3.5 2.5 3.  2.  4.5]
Metacritic_user_vote_count :
 [1330  249  627   31   88   34   17  124   62   54   35   37  145  218
   566  240  197  128  103  246   59  115  144  141  169  316  672  153
   107   19   11   29    6   75   67   80   33    8   46   14   23   18
  1054  262  206  108  421  779  192   96  100   78   69   24   95  126
    39   42   73 1171  121  130   15   98    4  850  764  903  199   90
    72  637  142   51  127   77   38   41 2375  318  196  149   43   13
   778  167  503   49   44    5    7  307  551  286  133   65   36   21
    52  177   20  362   60 1281  204  211   55  807  123]
IMDB_user_vote_count :
 [271107  65709 103660   3136  19560  39373   2680  16876  12227   5367
   12129  16547  22521  77518 334164 104235  49102  23561  50291  19521
   13769  25134  22104  50438  27380  45344 154499  57123  50285   1320
    2275   4160   1196  19658  12255  21319   2174   8301   2196  22440
    1151   6246  20659   1107 272204  42937  89618  19222  39838  85585
   56333  21914  45510  17061  33594   4857  40057   1769  41177  31878
   14346   3017  16069  19220   9856 303505  10891  22348    883  13296
     243    732 251856 207211 289464  45723  15982   8179 125088  21372
   14156  39152  27328   5626   3200   5927   5269  18986 292023  66636
   26046  38593   4911  39561    867   6605  14067 179506  50022 101264
  105412   5511  12940   3658  16663   4260  19627    952   4778  17237
    7819  81679  64656  44711  32166  17647  11392   5495   1062   3687
    1488    950    696  11937  41158  37292  17957   4110  82579   5630
  241807  55895   5444  15004  25214  17691  14873  11206  47377  12165
   96252   7367  24116  24345   1955   5289]
Fandango_votes :
 [14846 12640 12055  1793  1021   397   252  3223   896   864   804   218
     64  9443  8055  6757  6437  5597  5445  3886  3364  3276  2686  2066
   8381  7025  3458  1258   235   121    54    46    42  2800  2284   811
     79    56    43    35   281    74    38 15205  8077  7123  6835  6288
   6272  4577  4279  3325  2618  2273  1862  1834  1462  1213  1196   821
    638   540   308    55  4194  2680  2507   701   290    41 34085 33538
  15337  9749  8096  3815  3642  2704  1750  1501  1430  1415   973   624
     95 10509  9418  4493  4045  2195   259   110    83    66 34846  5933
   4933  4122  1896  1870  1117   996   917   482   210    63    48 13055
   2097  1078   675   449   162   127   124    70    51    36  9363  7705
   6506  2435    99  8357   729 34390 14998  3412  3054  2603  1651  1333
    531   450    50 15749  1348   192   118    59]
Fandango_Difference :
 [0.5 0.4 0.3 0.2 0.1 0. ]
```

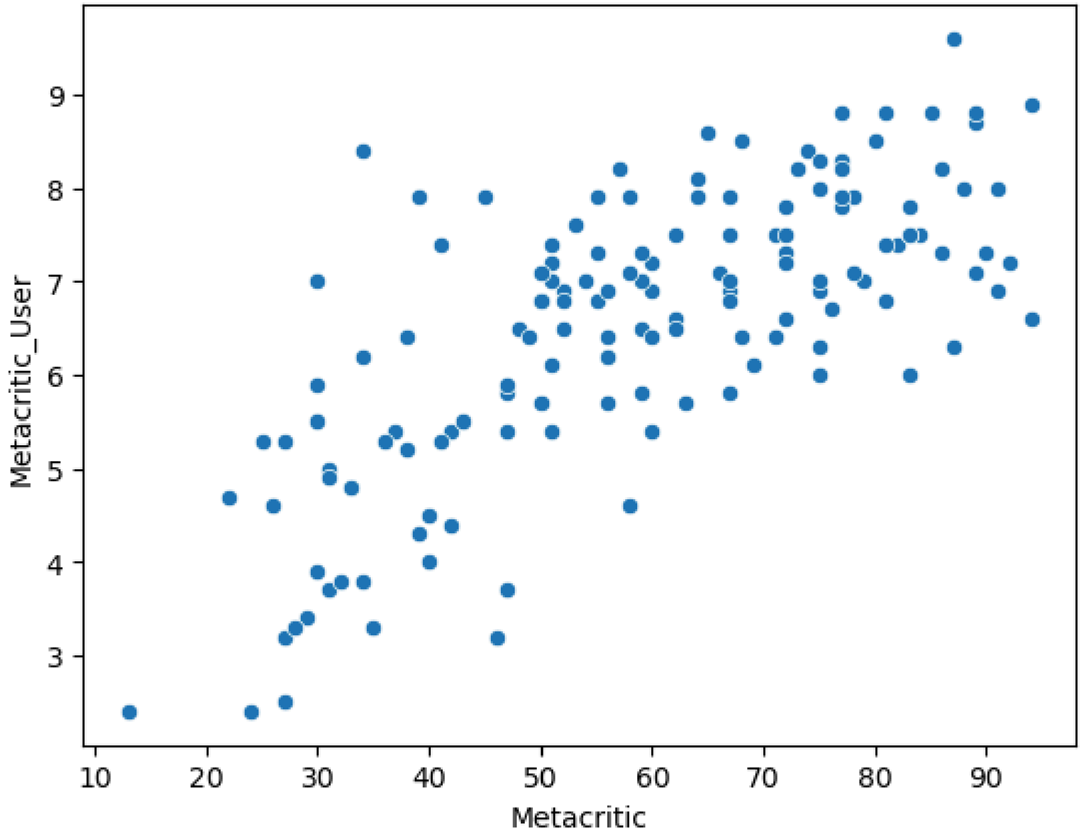
In [8]:
```
sns.scatterplot(x=data.RottenTomatoes,y=data.RottenTomatoes_User)
```

Out[8]:
```
<Axes: xlabel='RottenTomatoes', ylabel='RottenTomatoes_User'>
```



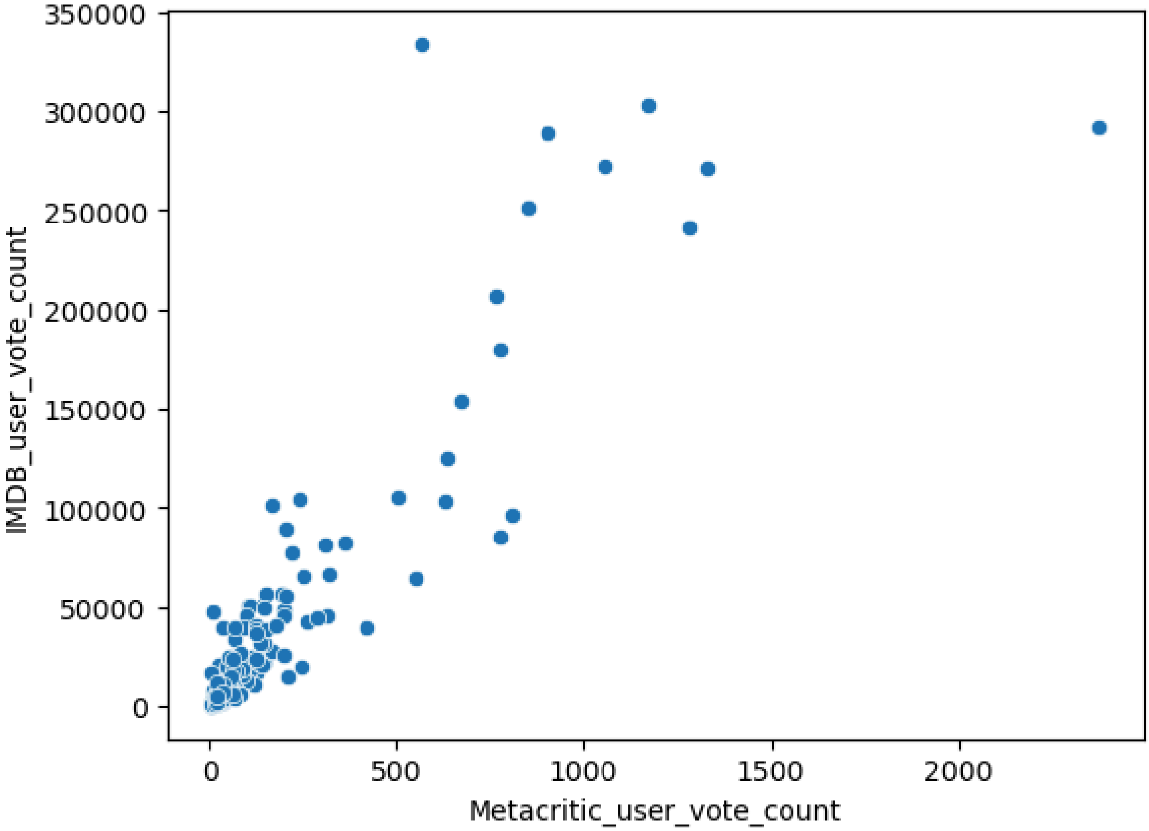
In [9]:
```
sns.scatterplot(x=data.Metacritic,y=data.Metacritic_User)
```

Out[9]:
```
<Axes: xlabel='Metacritic', ylabel='Metacritic_User'>
```



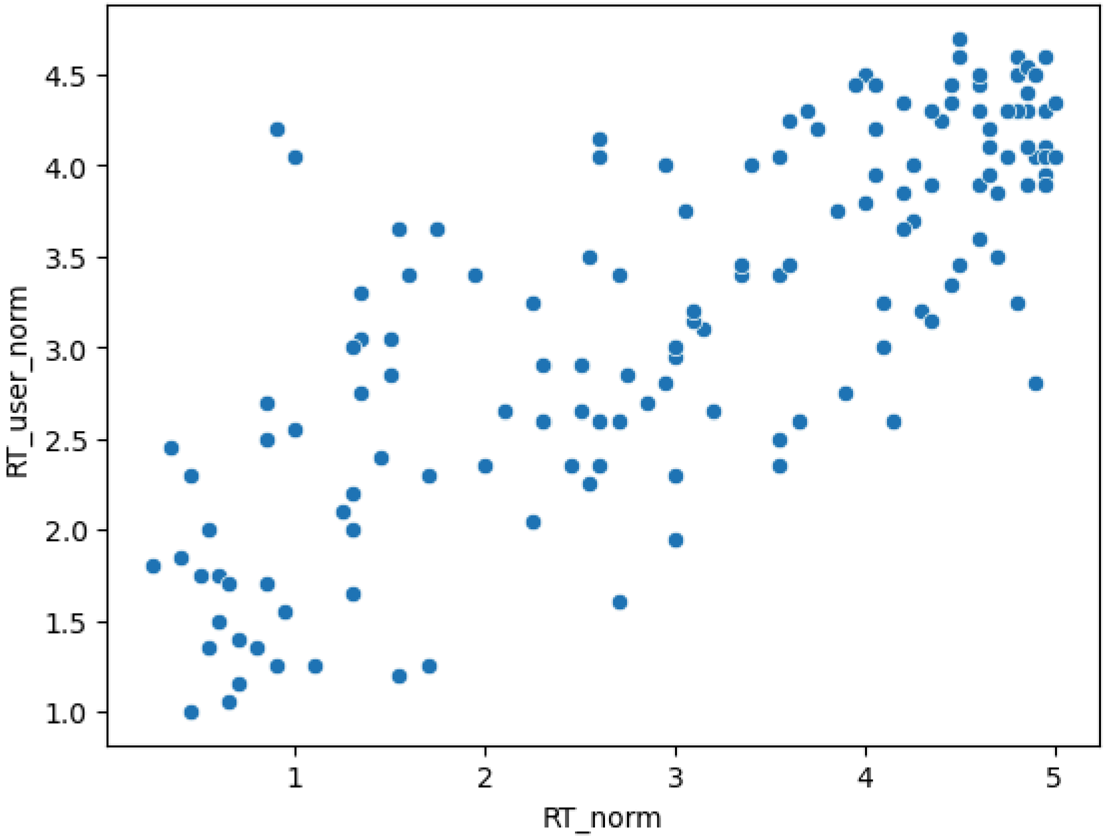
In [10]:
```
#relationship between vote counts on MetaCritic versus vote counts on IMDB
sns.scatterplot(x=data.Metacritic_user_vote_count,y=data.IMDB_user_vote_count)
```

```
Out[10]: <Axes: xlabel='Metacritic_user_vote_count', ylabel='IMDB_user_vote_count'>
```



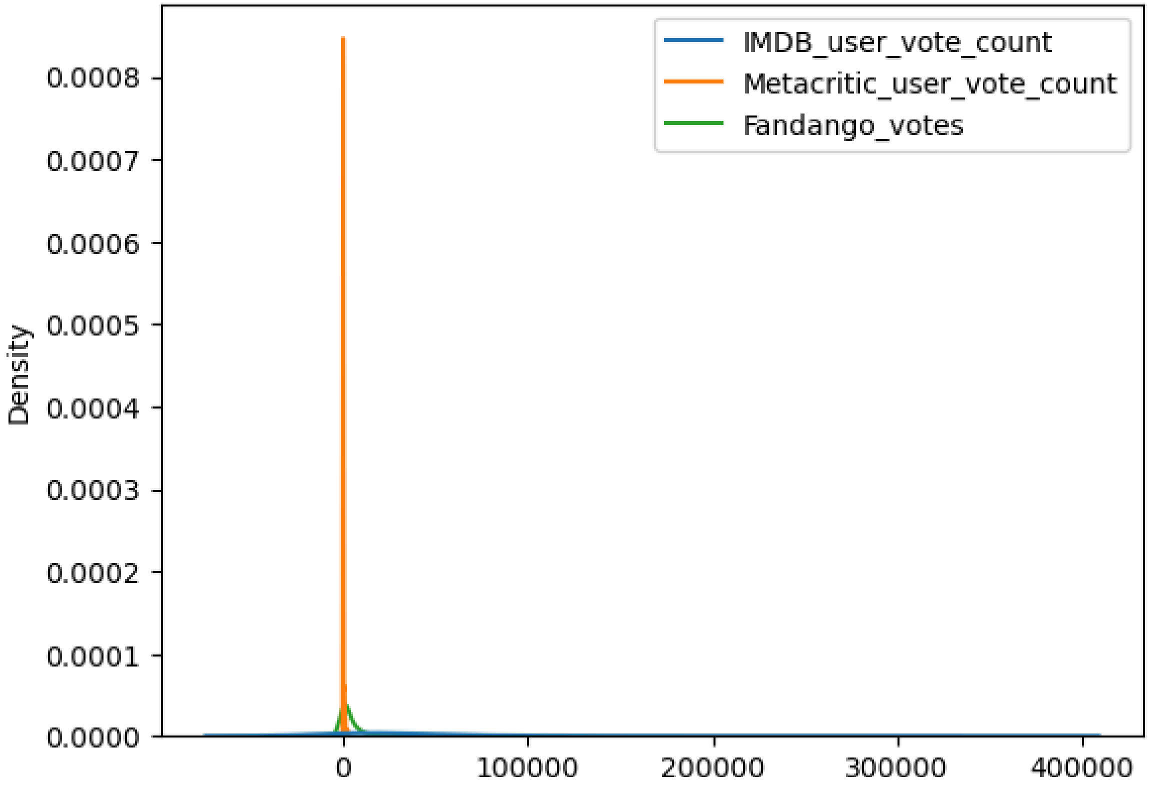
```
In [11]: sns.scatterplot(x=data.RT_norm,y=data.RT_user_norm)
```

```
Out[11]: <Axes: xlabel='RT_norm', ylabel='RT_user_norm'>
```



```
In [12]: df=data[['IMDB_user_vote_count','Metacritic_user_vote_count','Fandango_votes']]
In [13]: sns.kdeplot(data=df)
```

```
Out[13]: <Axes: ylabel='Density'>
```



```
In [14]: #norm_scores DataFrame that only contains the normalizes ratings include both ST
```

```
In [15]: norm_score=data[['Fandango_Stars','Fandango_Ratingvalue','RT_norm_round','RT_use
                          'IMDB_norm','Metacritic_user_norm_round','IMDB_norm_round']]
```

```
In [16]: norm_score
```

Out[16]:

	Fandango_Stars	Fandango_Ratingvalue	RT_norm_round	RT_user_norm_round	Met
0	5.0	4.5	3.5	4.5	
1	5.0	4.5	4.5	4.0	
2	5.0	4.5	4.0	4.5	
3	5.0	4.5	1.0	4.0	
4	3.5	3.0	0.5	1.5	
...	...	...	...	...	
141	4.0	4.0	4.5	4.0	
142	3.5	3.5	5.0	4.0	
143	3.5	3.5	5.0	4.0	
144	3.5	3.5	5.0	4.0	
145	3.5	3.5	4.5	3.0	

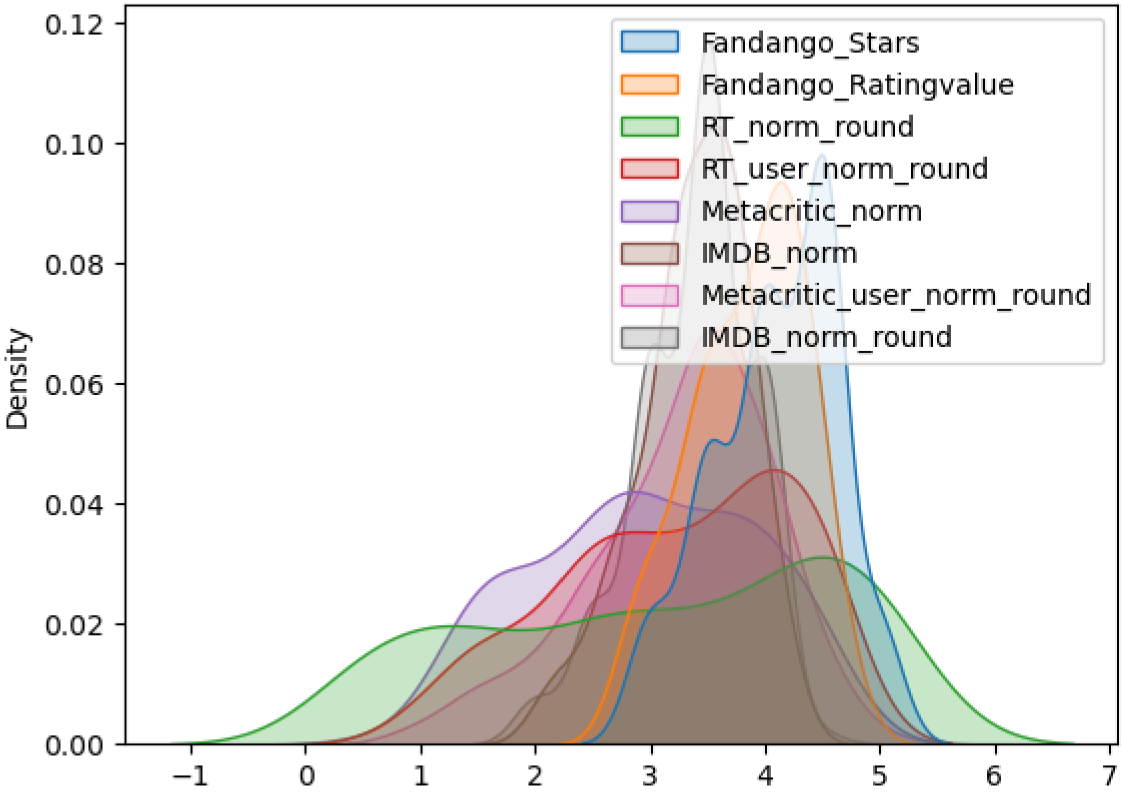
146 rows × 8 columns

In [17]:

```
sns.kdeplot(data=norm_score,fill=True)
```

Out[17]:

```
<Axes: ylabel='Density'>
```



In [18]:

```
data
```

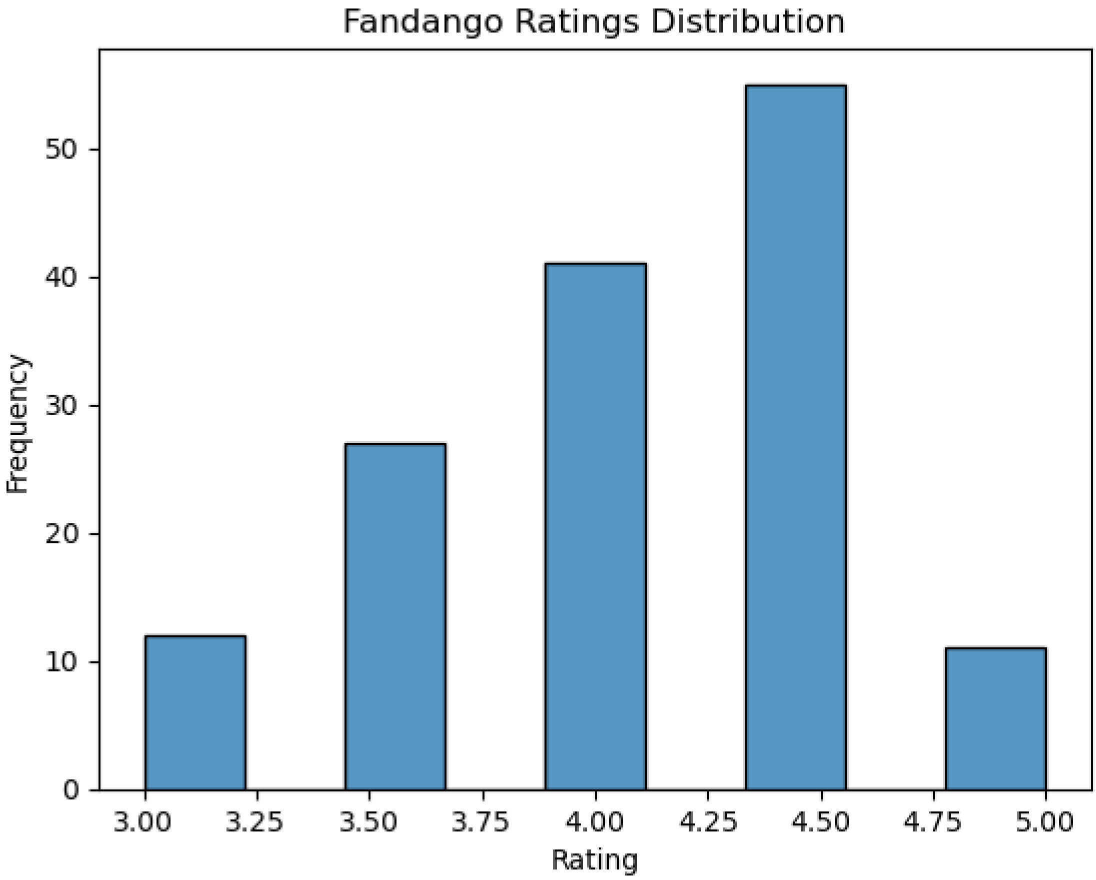
Out[18]:

	FILM	RottenTomatoes	RottenTomatoes_User	Metacritic	Metacritic_User	IMD
0	Avengers: Age of Ultron (2015)	74	8.6	66	7.1	7
1	Cinderella (2015)	85	8.0	67	7.5	7
2	Ant-Man (2015)	80	9.0	64	8.1	7
3	Do You Believe? (2015)	18	8.4	22	4.7	5
4	Hot Tub Time Machine 2 (2015)	14	2.8	29	3.4	5
...	...	...	...	...	...	
141	Mr. Holmes (2015)	87	7.8	67	7.9	7
142	'71 (2015)	97	8.2	83	7.5	7
143	Two Days, One Night (2014)	97	7.8	89	8.8	7
144	Gett: The Trial of Viviane Amsalem (2015)	100	8.1	90	7.3	7
145	Kumiko, The Treasure Hunter (2015)	87	6.3	68	6.4	6

146 rows × 22 columns

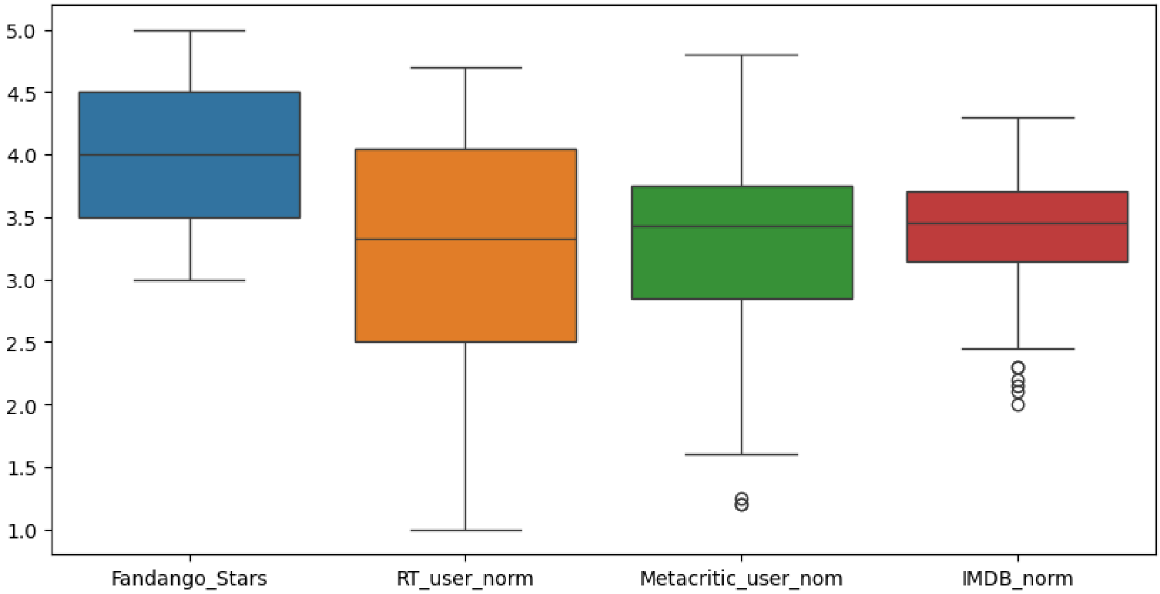
In [19]:

```
sns.histplot(data['Fandango_Stars'])
plt.title('Fandango Ratings Distribution')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```



In [20]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=data[['Fandango_Stars','RT_user_norm','Metacritic_user_nom','IM
```

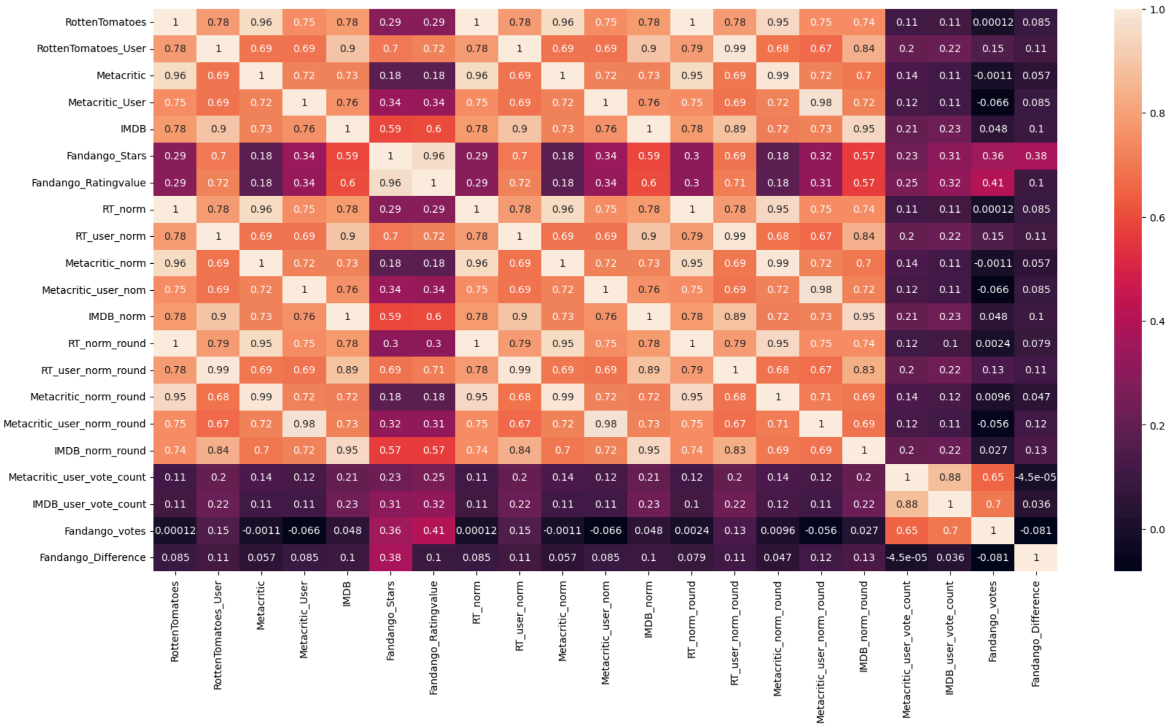
Out[20]: `<Axes: >`



In [21]:
```
data1=data.drop(['FILM'],axis=1)
```

In [22]:
```
d=data1.corr()
plt.figure(figsize=(20,10))
sns.heatmap(d,annot=True)
```

Out[22]: `<Axes: >`



In [23]:
```
# We'll predict Fandango rating using other ratings
ip=data1.drop(['Fandango_Stars'],axis=1)
op=data1['Fandango_Stars']
```

In [24]:
```
#train test split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(ip,op,test_size=0.45,random_state
```

In [25]:
```
#standard scalar transform
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
```

In [26]:
```
from sklearn.linear_model import LinearRegression
alg=LinearRegression()
alg.fit(x_train,y_train)
```

Out[26]:
```
▾ LinearRegression
LinearRegression()
```

In [27]:
```
yp=alg.predict(x_test)
```

In [28]:
```
df=pd.DataFrame({'y':list(y_test),'prediction':yp})
```

In [29]:
```
#accuracy
from sklearn.metrics import mean_squared_error,r2_score
mse=mean_squared_error(y_test,yp)
r2=r2_score(y_test,yp)
```

In [30]:
```
print(mse)
print(r2)
```

```
0.006185390001541189
0.9774010829551576
```

In [ ]:

In [ ]: