Project 3

Supratik Chanda and Joseph Ditton

Analysis of popular TV series

The goal of this analysis is to show interesting facts and trends across 4 popular tv series, namely: Game of Thrones, Dare Devil, The Flash, and Arrow. Using the information gathered we make a prediction about either the continued popularity of each show or its inevitable decline. We then turn our attention to all of the movie genres hoping to discover which genre produced the most popular movies. Finally we look at movie publishers to see which publisher has the higest user reviews based on their movies.

Dataset

The dataset is gathered from the Rotten Tomatoes website. We are specifically looking for audience ratings on the various pages we are looking at. This includes ratings for tv series, movie genres and movie studios. The webpages are downloaded and scraped for relevant information. The scraping is performed by using the Beautiful Soup library. Once the data has been inserted into a dataframe we use several simple analysis techniques to so interesting facts about the data.

The webpages that are downloaded need a lot of processing to turn them into usable data. First the pages are downloaded for each season of the 4 tv shows. Next, we look at various divs and anchor tags to grab the audience rating of each of the seasons and store it in number format.

We perform a similar process for the other analysis' that we do.

Analysis Technique

Comparison of various TV shows

First we compare trends in popularity across multiple tv shows. Using a bar charts we campare the review scores for every season of each show. We can clearly see that Game of Thrones is the most well liked out of all the shows and also has the most consistent reviews while the other shows tend to have more fluctuation across seasons. Arrow had the worst season out of all of them but seems to have recovered. Based on the popularity trends it is likely that we will see a new season for each of the shows. One thing that is worth pointing out is that the graph might accidentally imply that some shows have stopped running while others continue. However that is not the case, while some show have been running longer (they have more seasons) all of the shows are still currently running.

Genre popularity

We also look at the popularity of each genre. Rotten Tomatos groups some genres together (it groups action adventure, kids and family, and animation together for some reason) so it is hard to get an accurate look at each individual genre but we can get pretty close. The way we determine the popularity of each genre is by looking at the audience score for each movie in that genre. The scores are agregated and we calculate the median score and the standard deviation and throw that into a box ane whisker plot.

We can see that the genre: Drama, Documentary ,mystery and suspense is the highest rated. Comedy is also another genre that the audience has liked but not as much as Drama or Documentary. Horror doesn't go well with the common audience. Science fiction is also a really popular genre.

Publisher ratings

We perorm a similar analysis based on the publisher. We download the data about the publisher from the Rotten Tomatos site and scrape the page for all the movies an their audience scores. Again, we use a box and whisker plot to show where every publisher stands. We also look at the data from 2012 to 2018.

We can see some interesting things from the plots:

- In 2014, Only Waner Bros. Pictures has been able to get good reviews.
- In 2014 and 2015, the audience ratins for low for the movies of Warner and Walt Disney Pictures.
- In 2015 Walt Disney pictures had a really good year in terms of review scores.
- In 2017, only Warner again has gotten good reviews.
- In 2018 ,Universal Pictures was the lowest reviewed while Walt Disney was highly praised by the audiences. We can that overall Walt Disney pictures have been widely appreciated by a large chunk of audiences. Universal pictures has also got pretty good appreciation by the audience in the year of 2017 and 2018! May be it's because of the Jurassic World! In the last 3 years , Warner Bros is also getting a lot of audience appreciation . Let's hope we see moe DC . But What we found out that Marvel Studios is not catching up . I don't know why but they have not been able to get a large variety audience poll.

We have found out that overall Walt Disney pictures have been widely appreciated by a large chunk of audience. We expected Universal to be higher because of Jurrasic World, which is one of the highest grossing movies of all time, but their other movies must be getting lower scores. In the last 3 years , Warner Bros is also getting a lot of audience appreciation . Let's hope we see more DC movies. But What we found out that Marvel Studios is not catching up. We don't know why but they have not been able to get a large variety audience poll.

As a part of this we performed a T test between the 2017 ratings and the 2018 ratings. We come into a conclusion that the p_value for both of the two datasets is significantly low ,much lower than 0.05. p_value of 0.05 says thet there is a 5 % chance that the data is random and there is a real difference. T value relates the size of the differences. Now the p_value that we are getting is approximately 2% which is way lower than 5% . Henceforth we can infer conclusively that there is 2% chance that the data is random.

Results

Figure 1 shows the popularity trend of the selected TV shows across all of their seasons.

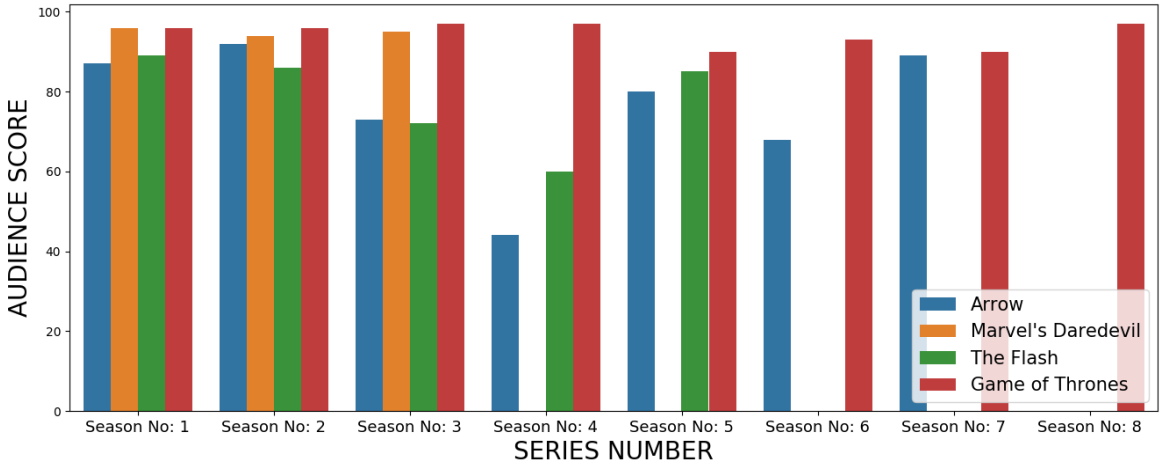


Figure 2 shows popularity of each genre that was calculated by looking at the audience reviews for the movies in the genre.

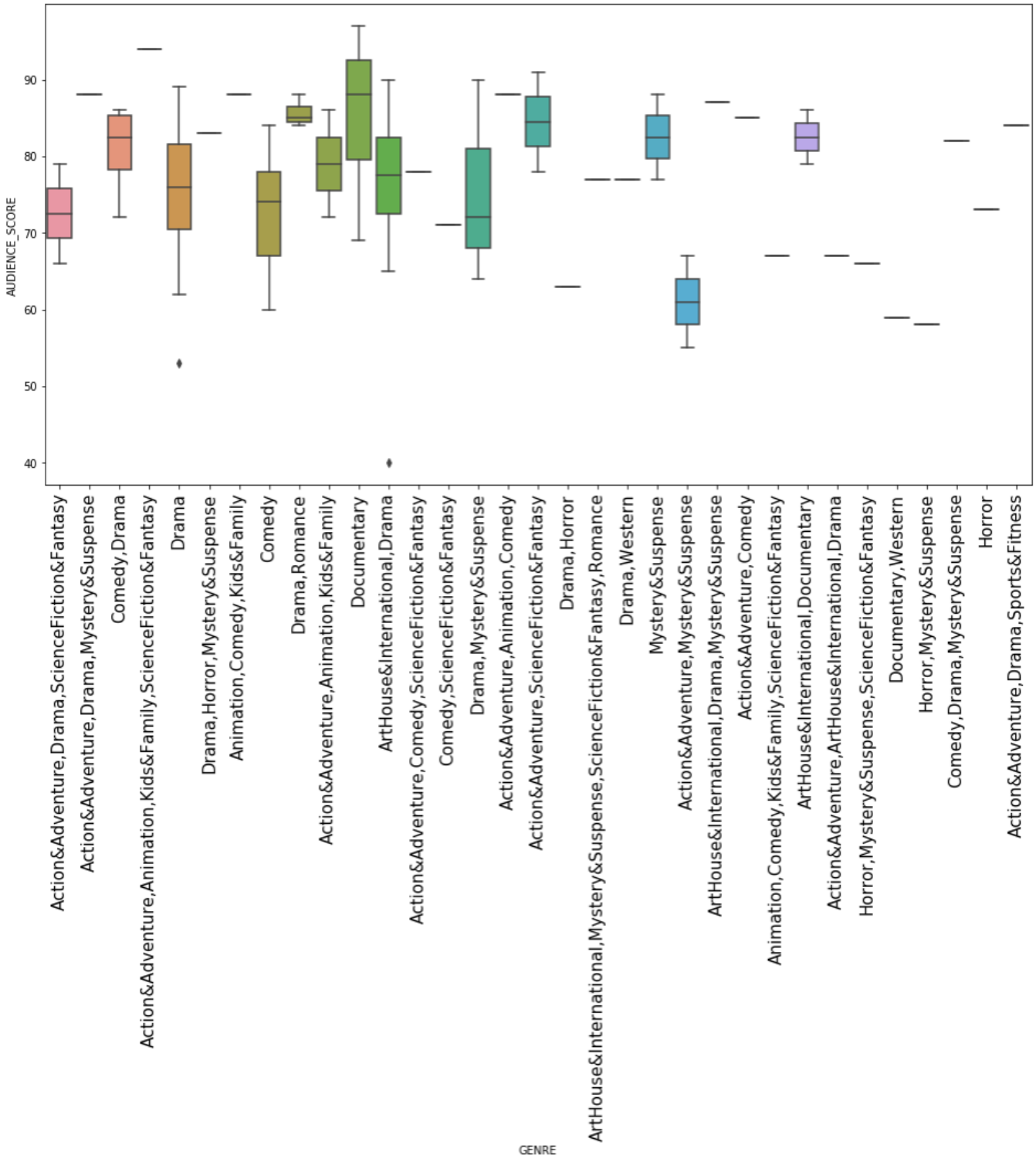
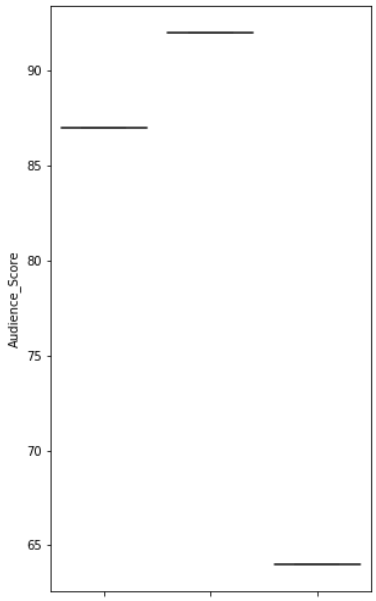
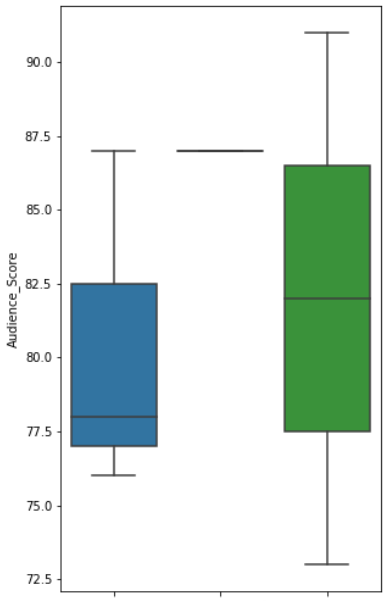
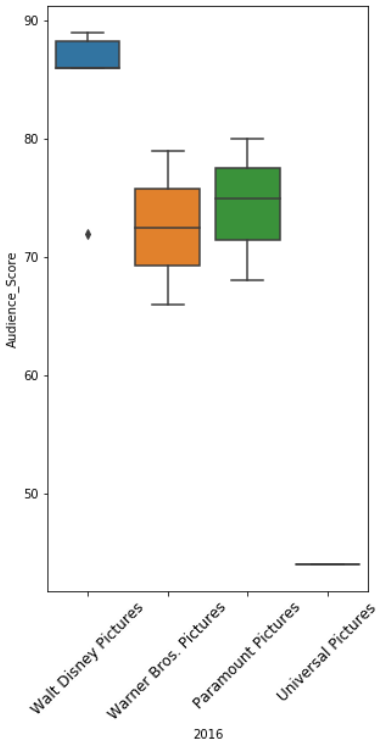
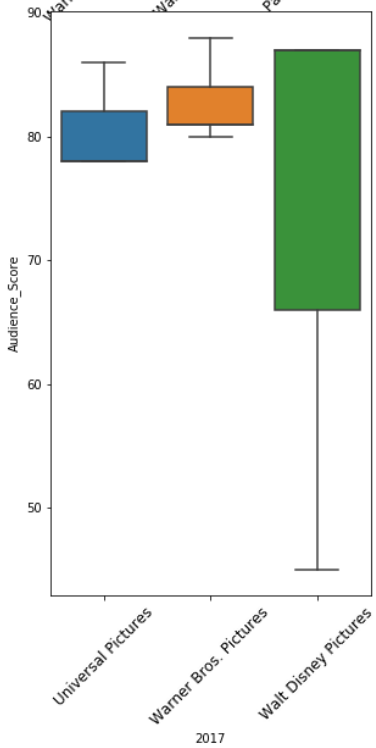


Figure 3 show the popularity of each movie studio based on the movies they produced. It shows this for each year in the range 2014 to 2018.


Audience_Score
90
85
80
75
70
65
Warner Bros. Pictures
Walt Disney Pictures
Paramount Pictures


Audience_Score
90.0
87.5
85.0
82.5
80.0
77.5
75.0
72.5
Walt Disney Pictures
Paramount Pictures
Universal Pictures


Audience_Score
90
80
70
60
50
Walt Disney Pictures
Warner Bros. Pictures
Paramount Pictures
Universal Pictures
2016


Audience_Score
90
80
70
60
50
Universal Pictures
Warner Bros. Pictures
Walt Disney Pictures
2017

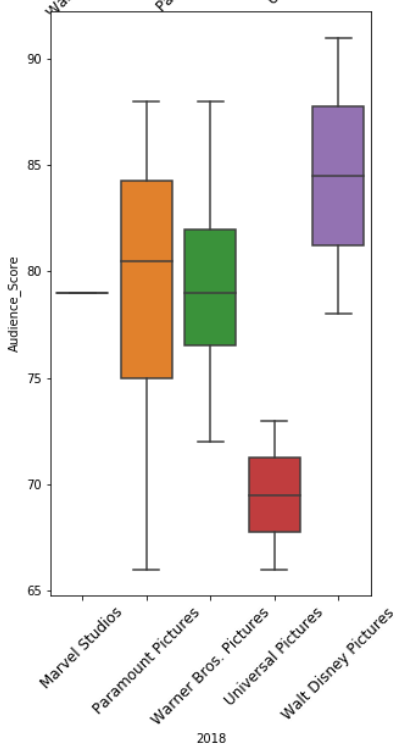
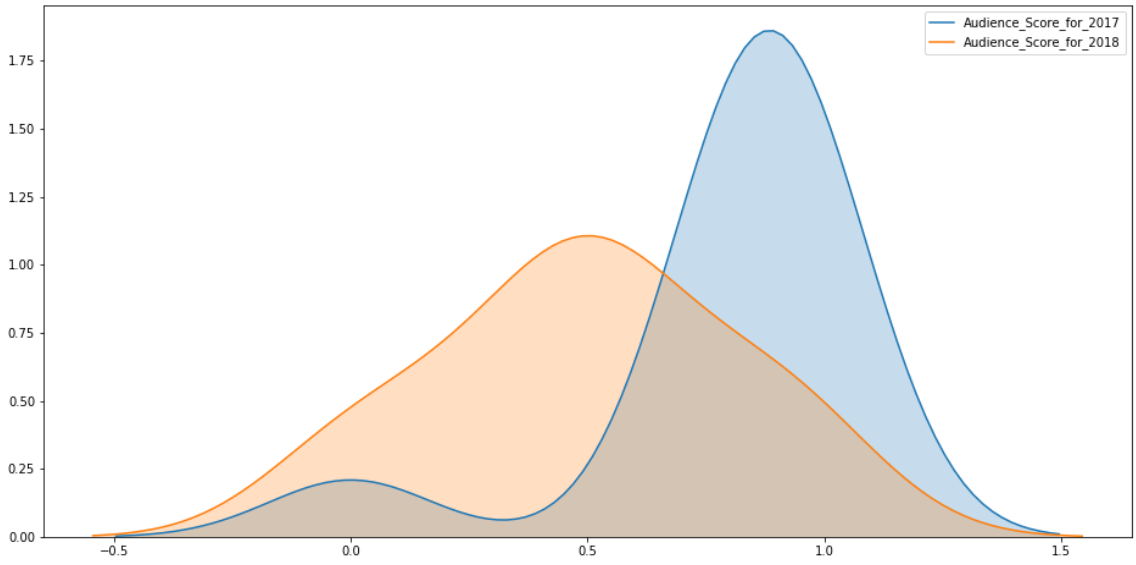

Audience_Score
90
85
80
75
70
65
Marvel Studios
Paramount Pictures
Warner Bros. Pictures
Universal Pictures
Walt Disney Pictures
2018

Figure 4 shows the results of performing the T test to determine if the change in popularity was random or not between 2017 and 2018



Analysis 1

In [1]:

```
import requests
import re
from bs4 import BeautifulSoup as BS

def comicseries(c,ind):
    fl = str('flash')
    marvelHeroes = ['arrow','daredevil',fl,'game_of_thrones']
    rotTomtoes = requests.get('https://www.rottentomatoes.com/tv/' + str
(marvelHeroes[ind]) +'/s0'+ str(c))
    Text = BS(rotTomtoes.text,'html5lib')
    c+=1
    return Text
```

In [2]:

```
import json,re,pandas as pd
from IPython.display import display
finalDT= pd.DataFrame()
end=False
count=1
filmNo=1
ind=0
audienceScore=pd.Series()
tvshowName = pd.Series()
DTFrame=pd.DataFrame()
tvSeriesNo =pd.Series()
while end == False:
    if str(comicseries(count,ind)).find('Sorry, please try again later.'
) != -1:
        ind+=1
        count=1
        if ind >3:
            break
        continue
    if ind ==3 and count==8:
        show = comicseries(count,ind).find('a',attrs={'id':'tvPosterLin
k'}).text.replace('\n','').strip()
        tvshowName = tvshowName.append(pd.Series(show))
        tvSeriesNo = tvSeriesNo.append(pd.Series('Season No: '+ str(coun
t)))
        k = comicseries(count,ind).find('div',attrs={'class':'meter-valu
e superPageFontColor'}).find('span',attrs={'class':'superPageFontColor'
}).text
        audienceScore = audienceScore.append(pd.Series(k))
        break
    else:
        show = comicseries(count,ind).find('a',attrs={'id':'tvPosterLin
k'}).text.replace('\n','').strip()
        tvshowName = tvshowName.append(pd.Series(show))
        tvSeriesNo = tvSeriesNo.append(pd.Series('Season No: '+ str(coun
t)))
        k = comicseries(count,ind).find('div',attrs={'class':'meter-valu
e'}).find('span',attrs={'class':'superPageFontColor'}).text
        audienceScore = audienceScore.append(pd.Series(k))
    count+=1
DTFrame =pd.concat([tvSeriesNo,tvshowName,audienceScore],axis=1)
DTFrame.reset_index(drop=True,inplace=True)
DTFrame.columns=['Series Number','TV_Series_Name','AudienceScore']
DTFrame['AudienceScore']= DTFrame['AudienceScore'].replace('%','',regex=
True)
DTFrame['AudienceScore'] = DTFrame['AudienceScore'].astype('int')
```

In [1]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(16,6))
sns.barplot(x=DTFrame['Series Number'],y=DTFrame['AudienceScore'],hue=DT
Frame['TV_Series_Name'])
plt.legend(loc='lower right',fontsize=15)
plt.xlabel('SERIES NUMBER',fontsize=20)
plt.ylabel('AUDIENCE SCORE',fontsize=20)
plt.xticks(fontsize=13)
plt.savefig('Movie_Review_1.png')
plt.show()
```

```
---------------------------------------------------------------------
----
NameError                                 Traceback (most recent call l
ast)
<ipython-input-1-dad5d35fb573> in <module>
      2 import seaborn as sns
      3 plt.figure(figsize=(16,6))
----> 4 sns.barplot(x=DTFrame['Series Number'],y=DTFrame['AudienceScor
e'],hue=DTFrame['TV_Series_Name'])
      5 plt.legend(loc='lower right',fontsize=15)
      6 plt.xlabel('SERIES NUMBER',fontsize=20)

NameError: name 'DTFrame' is not defined
```

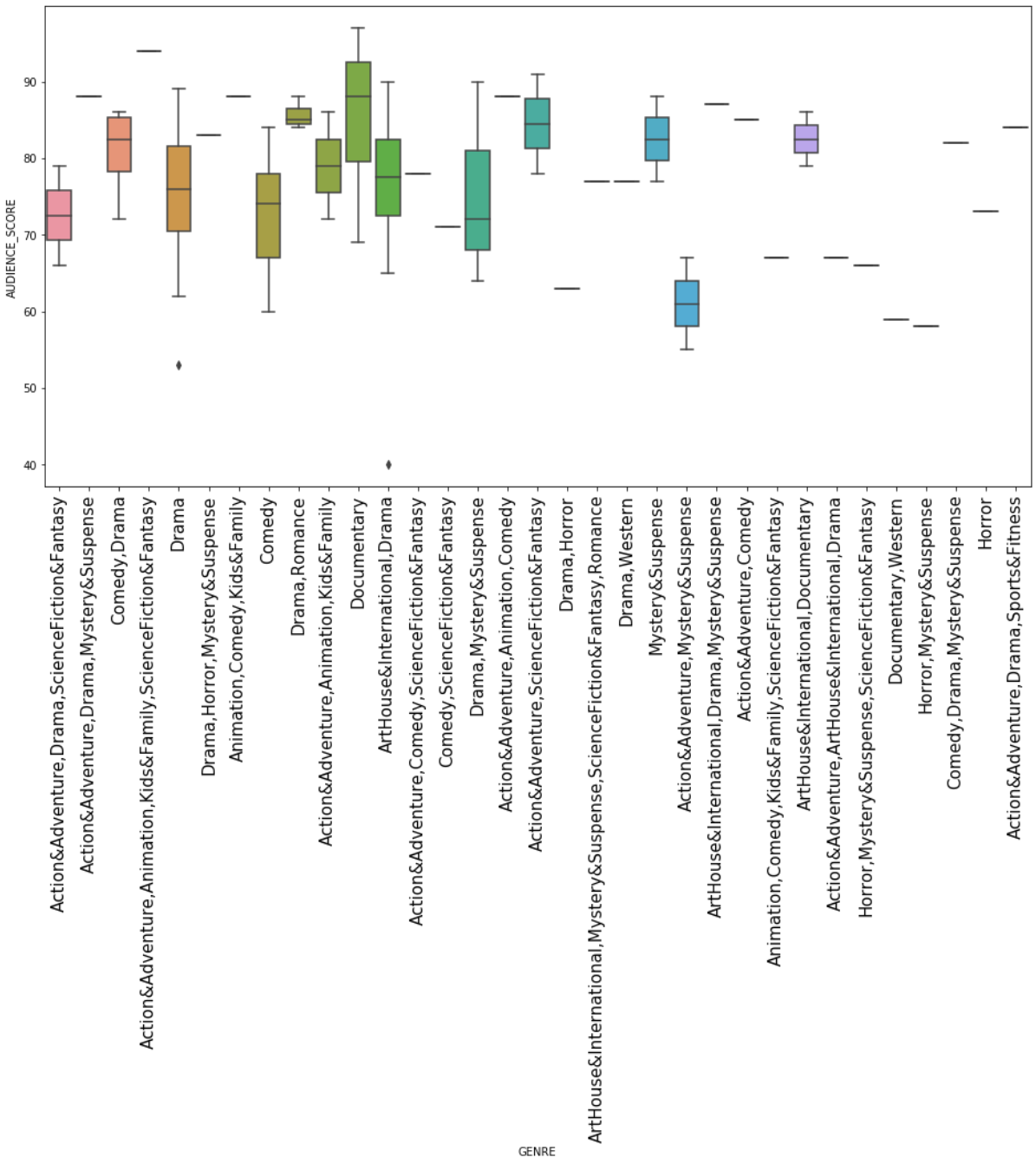
Analysis 2

In [4]:

```
import requests
import re,pandas as pd
from bs4 import BeautifulSoup as soup
from urllib import request
from collections import Counter
from IPython.display import display
def fn1():
    movieList = requests.get('https://www.rottentomatoes.com/top/bestofr
t/?year=2018')
    text = soup(movieList.text,'html5lib')
    topMovies = text.find_all('a',attrs={'class':'unstyled articleLink',
'href':True})
    k = [('https://www.rottentomatoes.com' + each['href']) for each in t
opMovies if re.search('/m',str(each)) if re.search('^/m',each['href']) ]
    genreDTFrame=pd.DataFrame()
    genreList=[]
    AudienceScoreList=[]
    audi=[]
    for each in k:
        i=request.urlopen(each)
        result = soup(i,'html5lib')
        nameTag = result.find_all('div',attrs={'class':'meta-value'})
        audiScr = result.find('div',attrs={'class':'meter-value'}).find(
'span',attrs={'class':'superPageFontColor'}).text
        AudienceScoreList.append(audiScr.replace('%',''))
        [genreList.append(each.text.replace(' ','').replace('\n','')) fo
r each in nameTag if str(each).find('genres')>-1]
    #print(genreList,type(genreList))
    return (AudienceScoreList,genreList)
```

In [5]:

```
def fn2():
    import matplotlib.pyplot as plt
    import seaborn as sns
    from IPython.display import display
    audiscore,genre= fn1()
    audiPDSeries=pd.Series(audiscore)
    genrePDSeries = pd.Series(genre)
    DTFrame = pd.concat([genrePDSeries,audiPDSeries],axis=1)
    DTFrame.columns=['GENRE','AUDIENCE_SCORE']
    DTFrame.AUDIENCE_SCORE = DTFrame.AUDIENCE_SCORE.astype('int')
    #display(DTFrame)
    plt.figure(figsize=(16,8))
    sns.boxplot(x=DTFrame['GENRE'],y=DTFrame['AUDIENCE_SCORE'])
    plt.xticks(rotation=90,fontsize=15)
    plt.savefig('Movie_Review_2.png')
    plt.show()
fn2()
```



Analysis 3

In [6]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from bs4 import BeautifulSoup as sp
import requests
from IPython.display import display
```

In [7]:

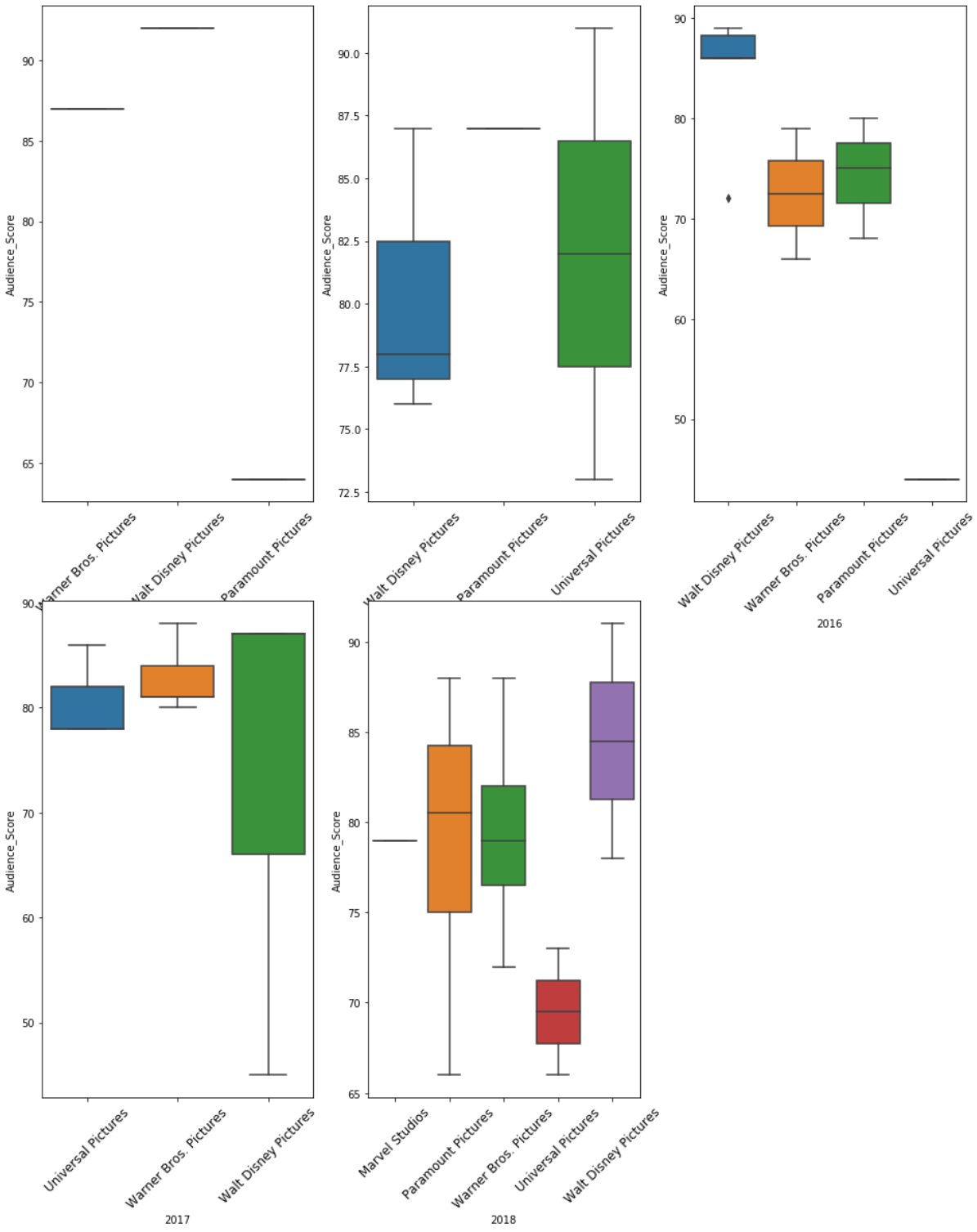
```
def PlotSeries(year):
    rot = 'https://www.rottentomatoes.com'
    scrapedData= requests.get('https://www.rottentomatoes.com/top/bestof
rt/?year='+year)
    #print(year)
    filmNames = sp(scrapedData.text,'html5lib')
    k = filmNames.find_all('a',attrs={'class':'unstyled articleLink'})
    nameList=[]
    totalDTFrame=pd.DataFrame()
    for each in k:
        if each['href'].startswith('/m'):
            nameList.append(rot+each['href'])
    StudioComparison=['Marvel Studios','Paramount Pictures','Warner Bro
s. Pictures','Universal Pictures','Walt Disney Pictures']
    AudienceScoreList=[]
    from urllib import request
    for each in nameList:
        Flag=False
        try:
            urlText = request.urlopen(each)
            soupText = sp(urlText,'html5lib')
        except:
            pass


        audiScr = soupText.find('div',attrs={'class':'meter-value'}).fin
d('span',attrs={'class':'superPageFontColor'}).text
        DTFrame=pd.DataFrame()
        try:
            k = soupText.find('a',attrs={'target':'movie-studio'}).text.
strip()
            j= soupText.find('h1',attrs={'class':'title hidden-xs'}).tex
t.strip()
            #print(j)
            if k in StudioComparison:
               # print('kk1')
                tempMovieStudio=pd.DataFrame([k])
                tempAudiScore = pd.DataFrame([audiScr])
                DTFrame=pd.concat([tempMovieStudio,tempAudiScore],axis=1
)
                #display(DTFrame)
                Flag=True
        except:
            j=soupText.find('div',attrs={'class':'meta-value'}).text.rep
lace(' ','')
            if j in StudioComparison:
                #print('kk2')
                tempMovieStudio=pd.DataFrame([k])
                tempAudiScore = pd.DataFrame([audiScr])
                DTFrame=pd.concat([tempMovieStudio,tempAudiScore],axis=1
)
                Flag=True
                #display(DTFrame)
        if Flag:
            totalDTFrame = pd.concat([totalDTFrame,DTFrame],axis=0)
    #print(totalDTFrame)
```

```
    return totalDTFrame
    #display(totalDTFrame)
```

In [10]:

```
yearList=['0','2014','2015','2016','2017','2018']
fig=plt.figure(figsize=(16,30))
fig.subplots_adjust(hspace=0.2,wspace=0.2)
for i in range(1,len(yearList)):
    DTFrame = PlotSeries(yearList[i])
    #print(DTFrame)
    DTFrame.columns=['Studio_Names','Audience_Score']
    DTFrame.Audience_Score = DTFrame.Audience_Score.replace('%','',regex
=True).astype('int')
    ax=fig.add_subplot(3,3,i)
    ax=sns.boxplot(x=DTFrame['Studio_Names'],y=DTFrame['Audience_Score'
])
    plt.xlabel(yearList[i])
    plt.xticks(fontsize=12,rotation=45)
    plt.savefig('Movie_Review_3.png')
    #plt.show()
#,
```


Audience_Score
Warner Bros. Pictures
Walt Disney Pictures
Paramount Pictures
Walt Disney Pictures
Paramount Pictures
Universal Pictures
Walt Disney Pictures
Warner Bros. Pictures
Paramount Pictures
Universal Pictures
2016
Universal Pictures
Warner Bros. Pictures
Walt Disney Pictures
2017
Marvel Studios
Paramount Pictures
Warner Bros. Pictures
Universal Pictures
Walt Disney Pictures
2018

Analysis 4

In [11]:
```
DTFrame2018 = PlotSeries('2018')
from scipy.stats import ttest_ind
#print(DTFrame)
DTFrame2018.columns=['Studio_Names','Audience_Score']
DTFrame2018.Audience_Score = DTFrame2018.Audience_Score.replace('%','',r
egex=True).astype('int')
display(DTFrame2018)
```

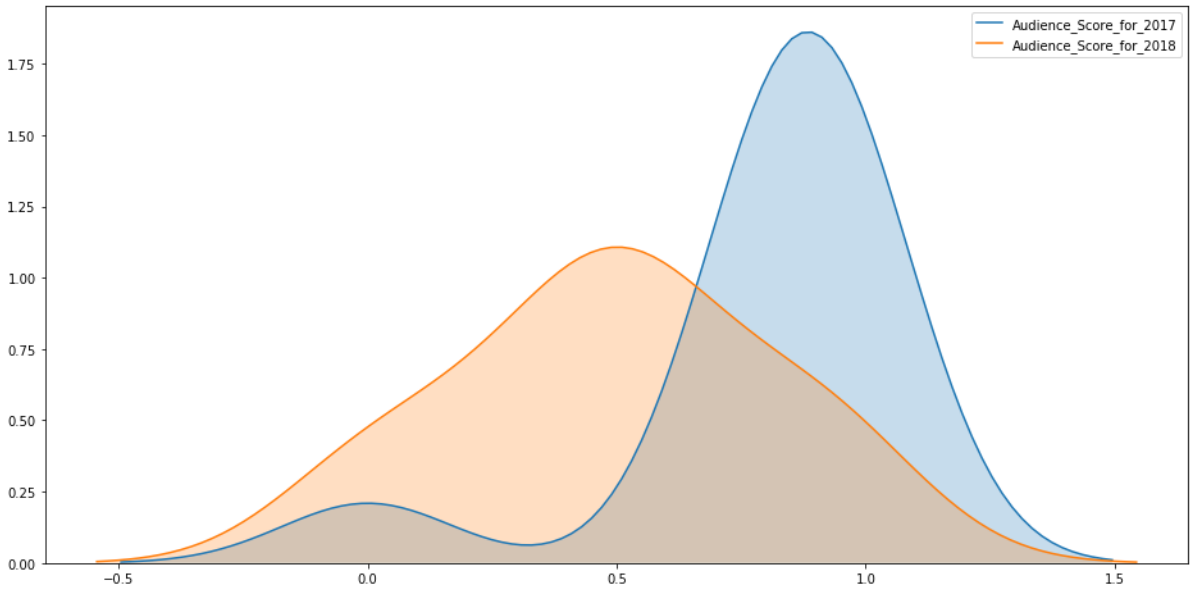
	Studio_Names	Audience_Score
0	Marvel Studios	79
0	Paramount Pictures	88
0	Warner Bros. Pictures	80
0	Paramount Pictures	83
0	Warner Bros. Pictures	88
0	Universal Pictures	66
0	Walt Disney Pictures	78
0	Warner Bros. Pictures	78
0	Walt Disney Pictures	91
0	Paramount Pictures	78
0	Paramount Pictures	66
0	Warner Bros. Pictures	72
0	Universal Pictures	73

In [12]:
```
DTFrame2017 = PlotSeries('2017')
from scipy.stats import ttest_ind
#print(DTFrame)
DTFrame2017.columns=['Studio_Names','Audience_Score']
DTFrame2017.Audience_Score = DTFrame2017.Audience_Score.replace('%','',r
egex=True).astype('int')
display(DTFrame2017)
```

	Studio_Names	Audience_Score
0	Universal Pictures	86
0	Warner Bros. Pictures	88
0	Warner Bros. Pictures	81
0	Walt Disney Pictures	45
0	Walt Disney Pictures	87
0	Warner Bros. Pictures	81
0	Warner Bros. Pictures	80
0	Walt Disney Pictures	87
0	Warner Bros. Pictures	84
0	Universal Pictures	78
0	Universal Pictures	78

In [16]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
DTFrame2017['Audience_Score'] = scaler.fit_transform(np.array(DTFrame201
7['Audience_Score']).astype('float').reshape(-1,1))
DTFrame2018['Audience_Score'] = scaler.fit_transform(np.array(DTFrame201
8['Audience_Score']).astype('float').reshape(-1,1))
print(ttest_ind(DTFrame2017['Audience_Score'],DTFrame2018['Audience_Scor
e']))
plt.figure(figsize=(16,8))
sns.kdeplot(DTFrame2017['Audience_Score'],label='Audience_Score_for_201
7',shade=True)
sns.kdeplot(DTFrame2018['Audience_Score'],label='Audience_Score_for_201
8',shade=True)
plt.legend()
plt.savefig('t_test.png')
plt.show()
```

```
Ttest_indResult(statistic=2.484270789853766, pvalue=0.021078072962328408)
```



So, from the above analysis , we come into a conclusion that the p_value for both of the two datasets is significantly low ,much lower than 0.05. p_value of 0.05 says thet there is a 5 % chance that the data is random and there is a real difference. T value relates the size of the differences. Now the p_value that we are getting is approximately 2% which is way lower than 5% . Henceforth we can infer conclusively that there is 2% chance that the data is random.

In [ ]: