

```
In [48]: import pandas as pd
```

```
In [49]: movies=pd.read_csv('r'/Users/sasidharbhagavatula/Desktop/Movie-Rating.csv')
```

```
In [50]: movies
```

```
Out[50]:
```

| | Film | Genre | Rotten Tomatoes Ratings % | Audience Ratings % | Budget (million \$) | Year of release |
|-----|----------------------|-----------|---------------------------------|-----------------------|---------------------------|--------------------|
| 0 | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |
| 1 | 10,000 B.C. | Adventure | 9 | 44 | 105 | 2008 |
| 2 | 12 Rounds | Action | 30 | 52 | 20 | 2009 |
| 3 | 127 Hours | Adventure | 93 | 84 | 18 | 2010 |
| 4 | 17 Again | Comedy | 55 | 70 | 20 | 2009 |
| ... | ... | ... | ... | ... | ... | ... |
| 554 | Your Highness | Comedy | 26 | 36 | 50 | 2011 |
| 555 | Youth in Revolt | Comedy | 68 | 52 | 18 | 2009 |
| 556 | Zodiac | Thriller | 89 | 73 | 65 | 2007 |
| 557 | Zombieland | Action | 90 | 87 | 24 | 2009 |
| 558 | Zookeeper | Comedy | 14 | 42 | 80 | 2011 |

559 rows × 6 columns

```
In [51]: movies.head()
```

```
Out[51]:
```

| | Film | Genre | Rotten Tomatoes Ratings % | Audience Ratings % | Budget (million \$) | Year of release |
|---|----------------------|-----------|---------------------------------|-----------------------|---------------------------|--------------------|
| 0 | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |
| 1 | 10,000 B.C. | Adventure | 9 | 44 | 105 | 2008 |
| 2 | 12 Rounds | Action | 30 | 52 | 20 | 2009 |
| 3 | 127 Hours | Adventure | 93 | 84 | 18 | 2010 |
| 4 | 17 Again | Comedy | 55 | 70 | 20 | 2009 |

```
In [52]: movies.tail()
```

```
Out[52]:
```

| | Film | Genre | Rotten Tomatoes Ratings % | Audience Ratings % | Budget (million \$) | Year of release |
|-----|--------------------|----------|------------------------------|-----------------------|------------------------|--------------------|
| 554 | Your Highness | Comedy | 26 | 36 | 50 | 2011 |
| 555 | Youth in Revolt | Comedy | 68 | 52 | 18 | 2009 |
| 556 | Zodiac | Thriller | 89 | 73 | 65 | 2007 |
| 557 | Zombieland | Action | 90 | 87 | 24 | 2009 |
| 558 | Zookeeper | Comedy | 14 | 42 | 80 | 2011 |

```
In [53]: type(movies)
```

```
Out[53]: pandas.core.frame.DataFrame
```

```
In [54]: len(movies)
```

```
Out[54]: 559
```

```
In [55]: import numpy
print(numpy.__version__)
```

```
2.1.3
```

```
In [56]: import pandas
print(pandas.__version__)
```

```
2.2.3
```

```
In [57]: movies.columns
```

```
Out[57]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
               'Budget (million $)', 'Year of release'],
              dtype='object')
```

```
In [58]: movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Film                                559 non-null    object
1   Genre                              559 non-null    object
2   Rotten Tomatoes Ratings %          559 non-null    int64
3   Audience Ratings %                 559 non-null    int64
4   Budget (million $)                 559 non-null    int64
5   Year of release                     559 non-null    int64
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

```
In [59]: movies.shape
```

```
Out[59]: (559, 6)
```

```
In [60]: movies.head()
```

```
Out[60]:
```

| | Film | Genre | Rotten Tomatoes Ratings % | Audience Ratings % | Budget (million \$) | Year of release |
|---|----------------------|-----------|---------------------------|--------------------|---------------------|-----------------|
| 0 | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |
| 1 | 10,000 B.C. | Adventure | 9 | 44 | 105 | 2008 |
| 2 | 12 Rounds | Action | 30 | 52 | 20 | 2009 |
| 3 | 127 Hours | Adventure | 93 | 84 | 18 | 2010 |
| 4 | 17 Again | Comedy | 55 | 70 | 20 | 2009 |

```
In [61]: movies.tail()
```

```
Out[61]:
```

| | Film | Genre | Rotten Tomatoes Ratings % | Audience Ratings % | Budget (million \$) | Year of release |
|-----|-----------------|----------|---------------------------|--------------------|---------------------|-----------------|
| 554 | Your Highness | Comedy | 26 | 36 | 50 | 2011 |
| 555 | Youth in Revolt | Comedy | 68 | 52 | 18 | 2009 |
| 556 | Zodiac | Thriller | 89 | 73 | 65 | 2007 |
| 557 | Zombieland | Action | 90 | 87 | 24 | 2009 |
| 558 | Zookeeper | Comedy | 14 | 42 | 80 | 2011 |

```
In [62]: movies.columns
```

```
Out[62]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
              'Budget (million $)', 'Year of release'],
              dtype='object')
```

```
In [63]: movies.columns=['Film', 'Genre', 'CriticRating', 'AudienceRating', 'BudgetMil
```

```
In [64]: movies.head(1)
```

```
Out[64]:
```

| | Film | Genre | CriticRating | AudienceRating | BudgetMillions | Year |
|---|----------------------|--------|--------------|----------------|----------------|------|
| 0 | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |

```
In [65]: movies.shape
```

```
Out[65]: (559, 6)
```

```
In [66]: movies.describe()
```

```
Out[66]:
```

| | CriticRating | AudienceRating | BudgetMillions | Year |
|-------|--------------|----------------|----------------|-------------|
| count | 559.000000 | 559.000000 | 559.000000 | 559.000000 |
| mean | 47.309481 | 58.744186 | 50.236136 | 2009.152057 |
| std | 26.413091 | 16.826887 | 48.731817 | 1.362632 |
| min | 0.000000 | 0.000000 | 0.000000 | 2007.000000 |
| 25% | 25.000000 | 47.000000 | 20.000000 | 2008.000000 |
| 50% | 46.000000 | 58.000000 | 35.000000 | 2009.000000 |
| 75% | 70.000000 | 72.000000 | 65.000000 | 2010.000000 |
| max | 97.000000 | 96.000000 | 300.000000 | 2011.000000 |

```
In [67]: movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Film                  559 non-null   object
1   Genre                 559 non-null   object
2   CriticRating          559 non-null   int64
3   AudienceRating        559 non-null   int64
4   BudgetMillions        559 non-null   int64
5   Year                  559 non-null   int64
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

In [68]: `movies.describe().transpose()`

Out[68]:

| | count | mean | std | min | 25% | 50% | 75% | |
|-----------------------|-------|-------------|-----------|--------|--------|--------|--------|----|
| CriticRating | 559.0 | 47.309481 | 26.413091 | 0.0 | 25.0 | 46.0 | 70.0 | |
| AudienceRating | 559.0 | 58.744186 | 16.826887 | 0.0 | 47.0 | 58.0 | 72.0 | |
| BudgetMillions | 559.0 | 50.236136 | 48.731817 | 0.0 | 20.0 | 35.0 | 65.0 | 3 |
| Year | 559.0 | 2009.152057 | 1.362632 | 2007.0 | 2008.0 | 2009.0 | 2010.0 | 20 |

In [69]: `movies.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Film                  559 non-null   object
1   Genre                 559 non-null   object
2   CriticRating          559 non-null   int64
3   AudienceRating        559 non-null   int64
4   BudgetMillions        559 non-null   int64
5   Year                  559 non-null   int64
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

In [70]: `movies.Film=movies.Film.astype('category')`

In [71]: `movies.describe()`

Out[71]:

| | CriticRating | AudienceRating | BudgetMillions | Year |
|--------------|--------------|----------------|----------------|-------------|
| count | 559.000000 | 559.000000 | 559.000000 | 559.000000 |
| mean | 47.309481 | 58.744186 | 50.236136 | 2009.152057 |
| std | 26.413091 | 16.826887 | 48.731817 | 1.362632 |
| min | 0.000000 | 0.000000 | 0.000000 | 2007.000000 |
| 25% | 25.000000 | 47.000000 | 20.000000 | 2008.000000 |
| 50% | 46.000000 | 58.000000 | 35.000000 | 2009.000000 |
| 75% | 70.000000 | 72.000000 | 65.000000 | 2010.000000 |
| max | 97.000000 | 96.000000 | 300.000000 | 2011.000000 |

In [72]: `movies.Genre=movies.Genre.astype('category')`

In [73]: `movies.describe()`

Out[73]:

| | CriticRating | AudienceRating | BudgetMillions | Year |
|--------------|--------------|----------------|----------------|-------------|
| count | 559.000000 | 559.000000 | 559.000000 | 559.000000 |
| mean | 47.309481 | 58.744186 | 50.236136 | 2009.152057 |
| std | 26.413091 | 16.826887 | 48.731817 | 1.362632 |
| min | 0.000000 | 0.000000 | 0.000000 | 2007.000000 |
| 25% | 25.000000 | 47.000000 | 20.000000 | 2008.000000 |
| 50% | 46.000000 | 58.000000 | 35.000000 | 2009.000000 |
| 75% | 70.000000 | 72.000000 | 65.000000 | 2010.000000 |
| max | 97.000000 | 96.000000 | 300.000000 | 2011.000000 |

In [74]: `movies.Year=movies.Year.astype('category') # convert datatpye to categ`

In [75]: `movies.describe()`

Out[75]:

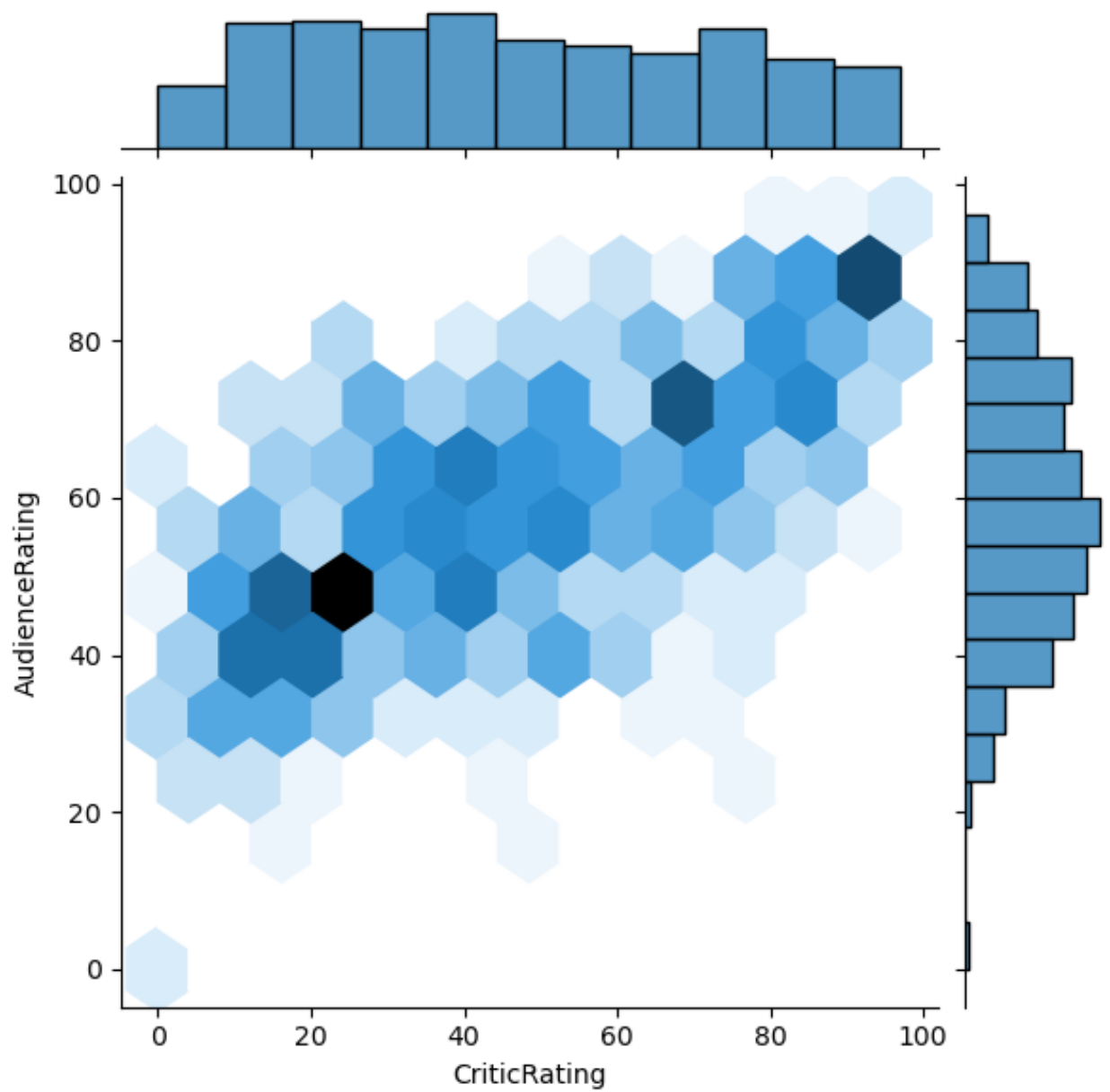
| | CriticRating | AudienceRating | BudgetMillions |
|--------------|--------------|----------------|----------------|
| count | 559.000000 | 559.000000 | 559.000000 |
| mean | 47.309481 | 58.744186 | 50.236136 |
| std | 26.413091 | 16.826887 | 48.731817 |
| min | 0.000000 | 0.000000 | 0.000000 |
| 25% | 25.000000 | 47.000000 | 20.000000 |
| 50% | 46.000000 | 58.000000 | 35.000000 |
| 75% | 70.000000 | 72.000000 | 65.000000 |
| max | 97.000000 | 96.000000 | 300.000000 |

PLOT

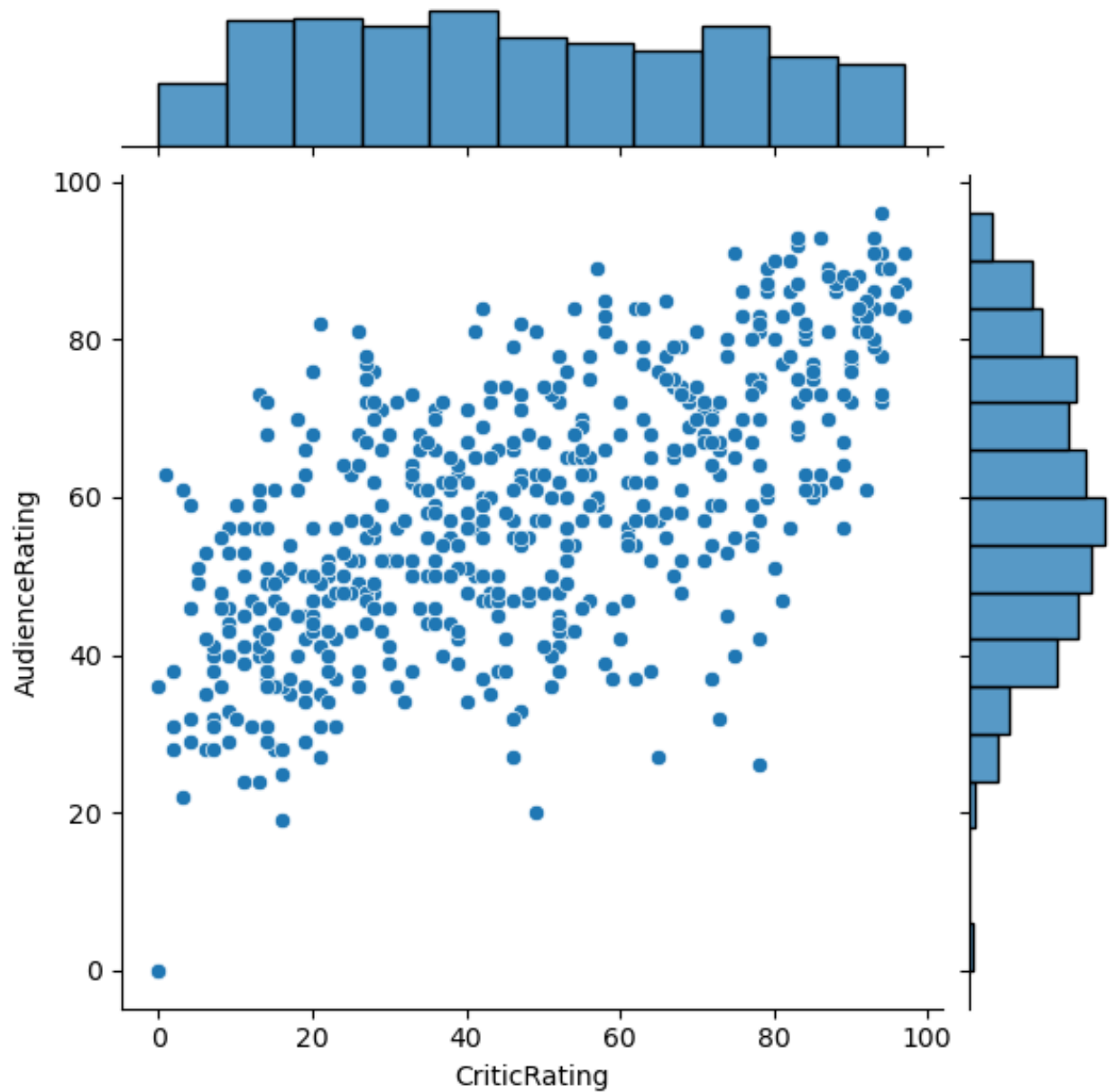
In [76]: `from matplotlib import pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')`

In [77]: `j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='hex`

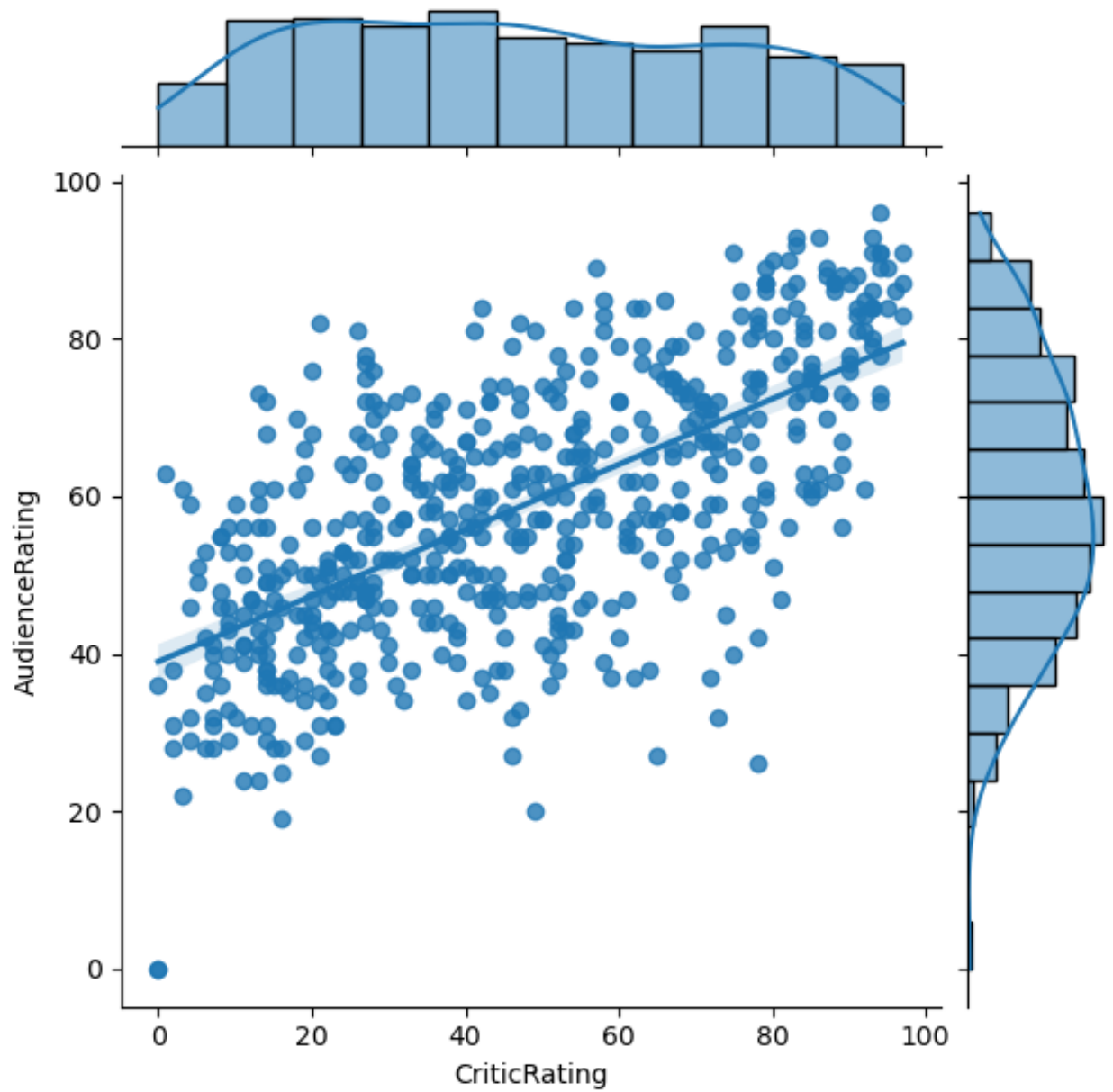


```
In [78]: j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='sca
```

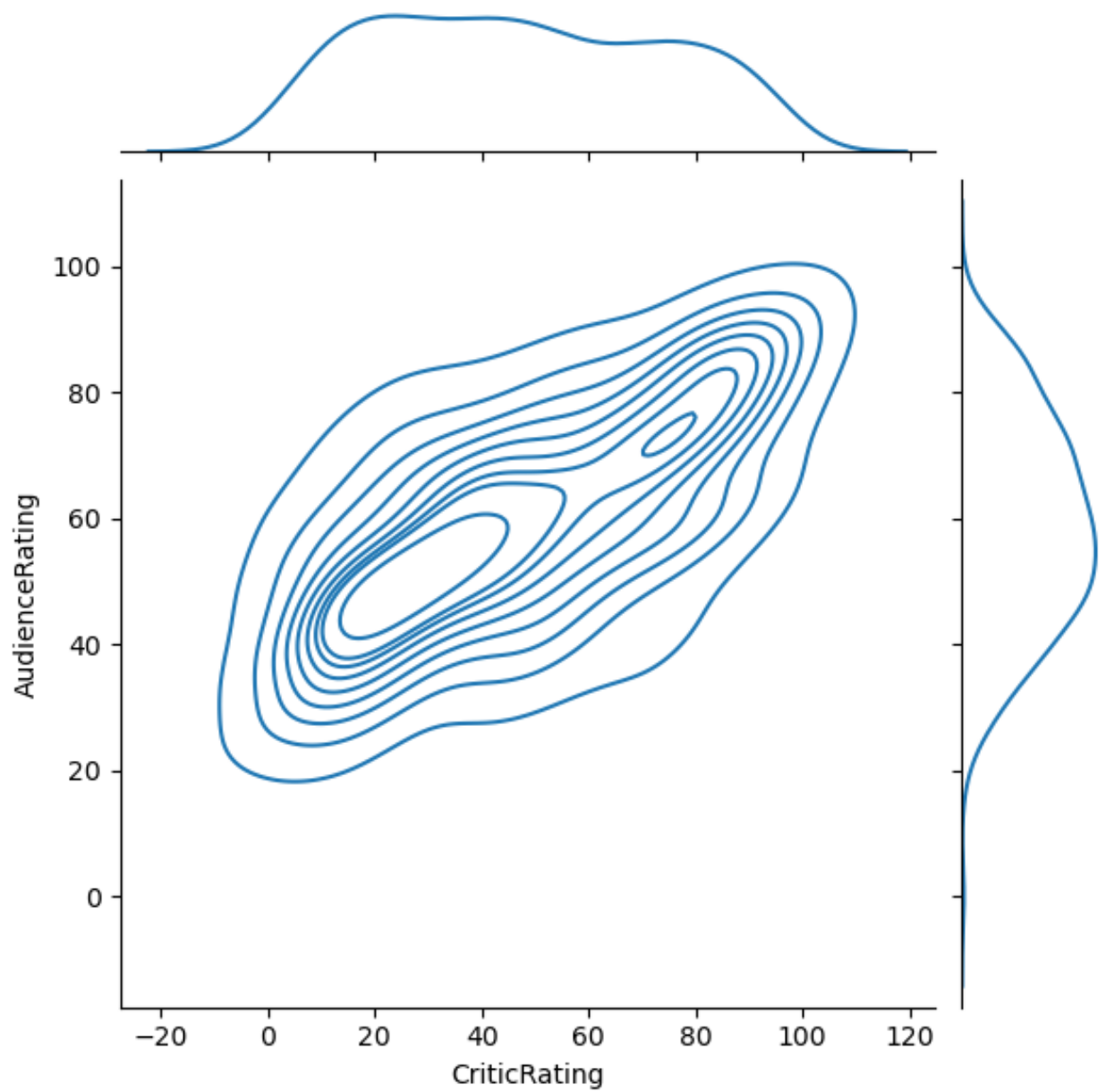


```
In [79]: #audience rating is more dominant than critic rating  
#most audience likely to watch audience rating than critic rating  
#joint plot
```

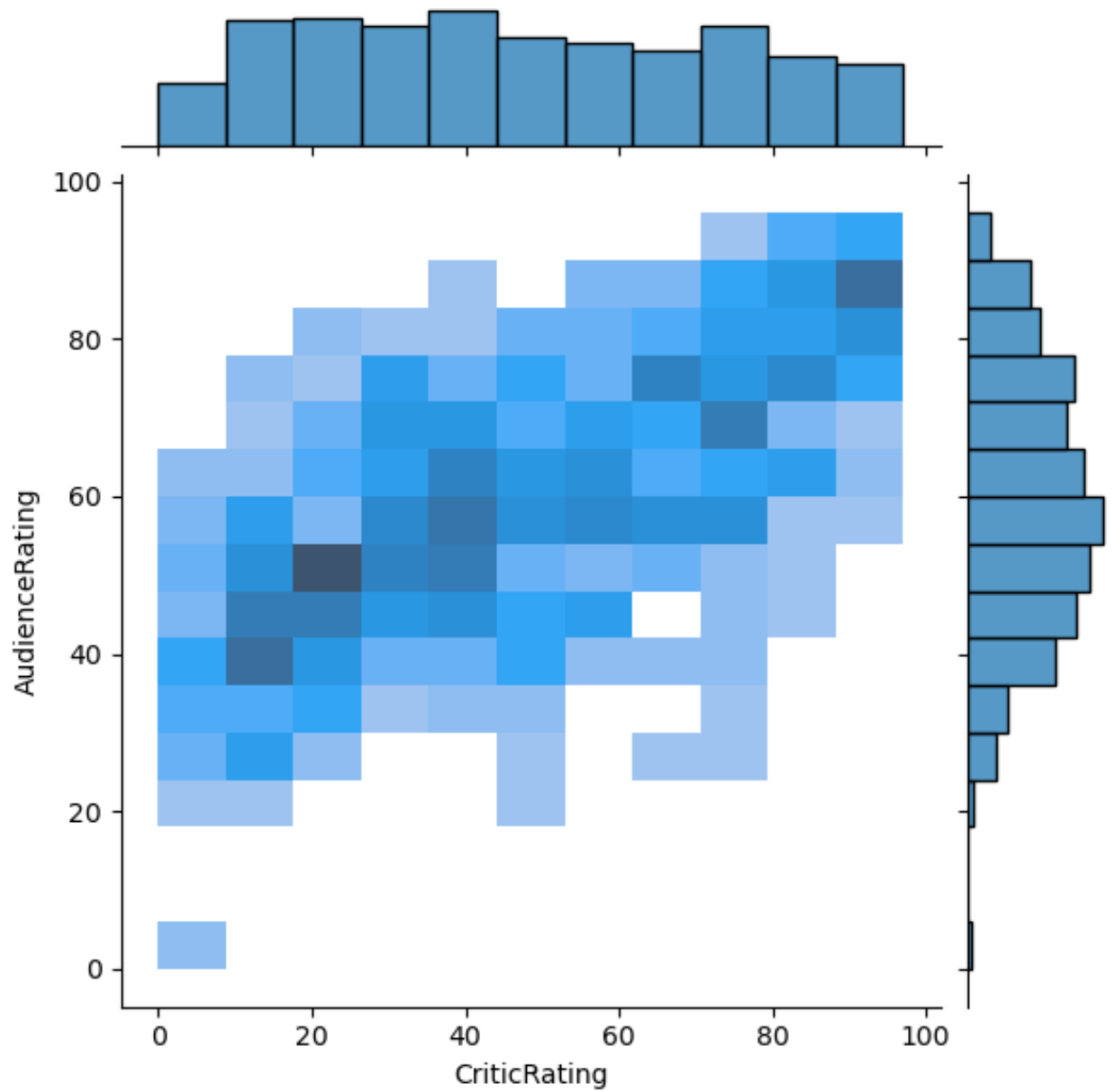
```
In [80]: j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='reg')
```

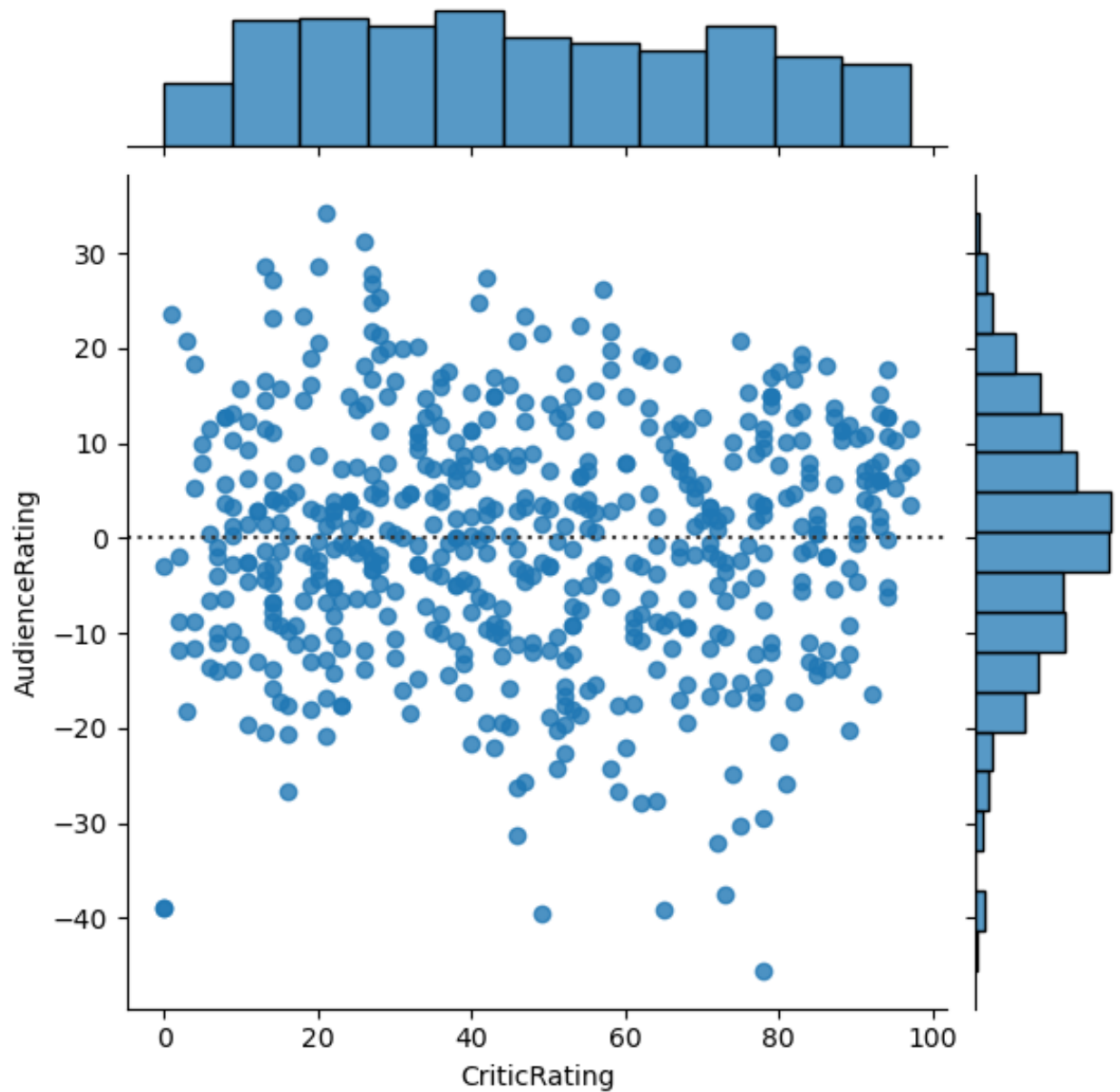
```
In [81]: j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='kde')
```



```
In [82]: j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='his
```



```
In [83]: j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='res
```

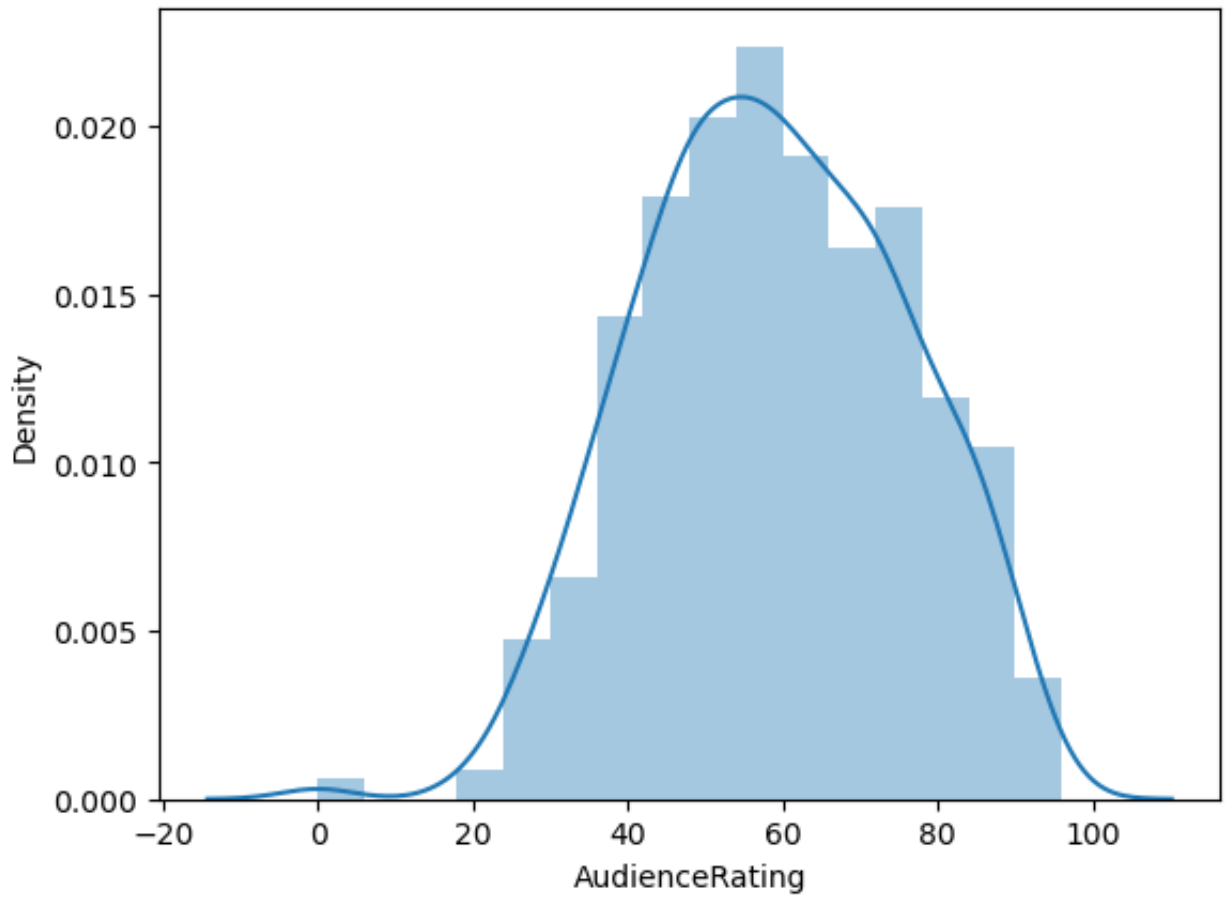


```
In [84]: #sns.set_style(style='dark grid')
```

```
In [85]: #Histograms
```

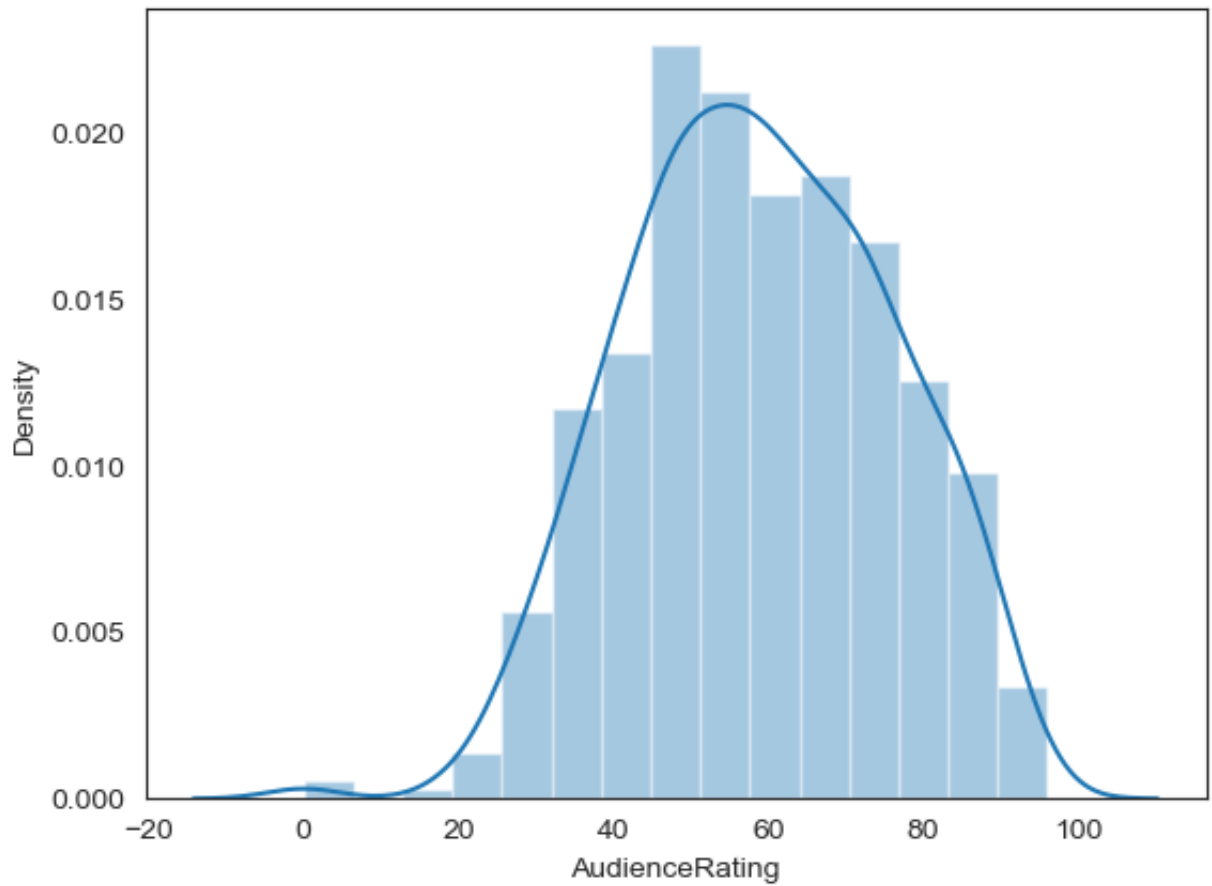
```
m1 = sns.distplot(movies.AudienceRating)
```

```
#y - axis generated by seaborn automatically that is the powerful
```

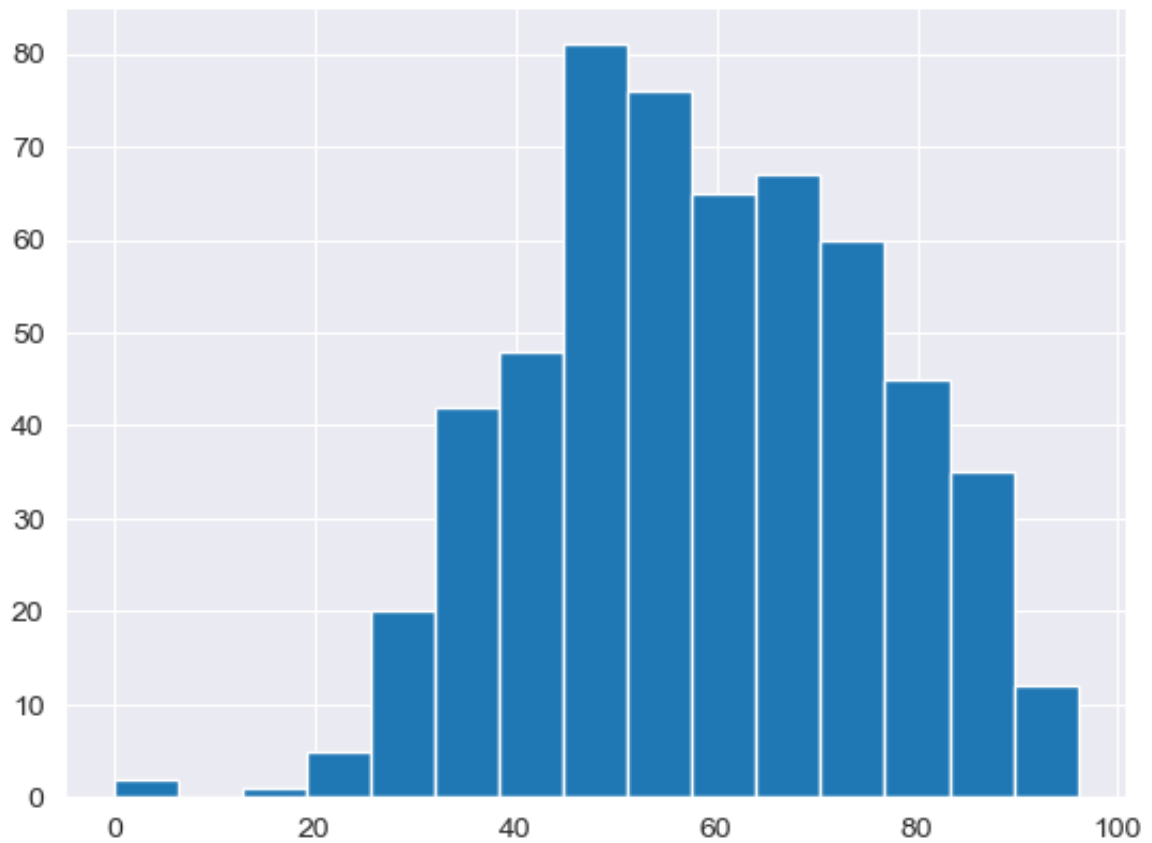


```
In [94]: sns.set_style('white')
```

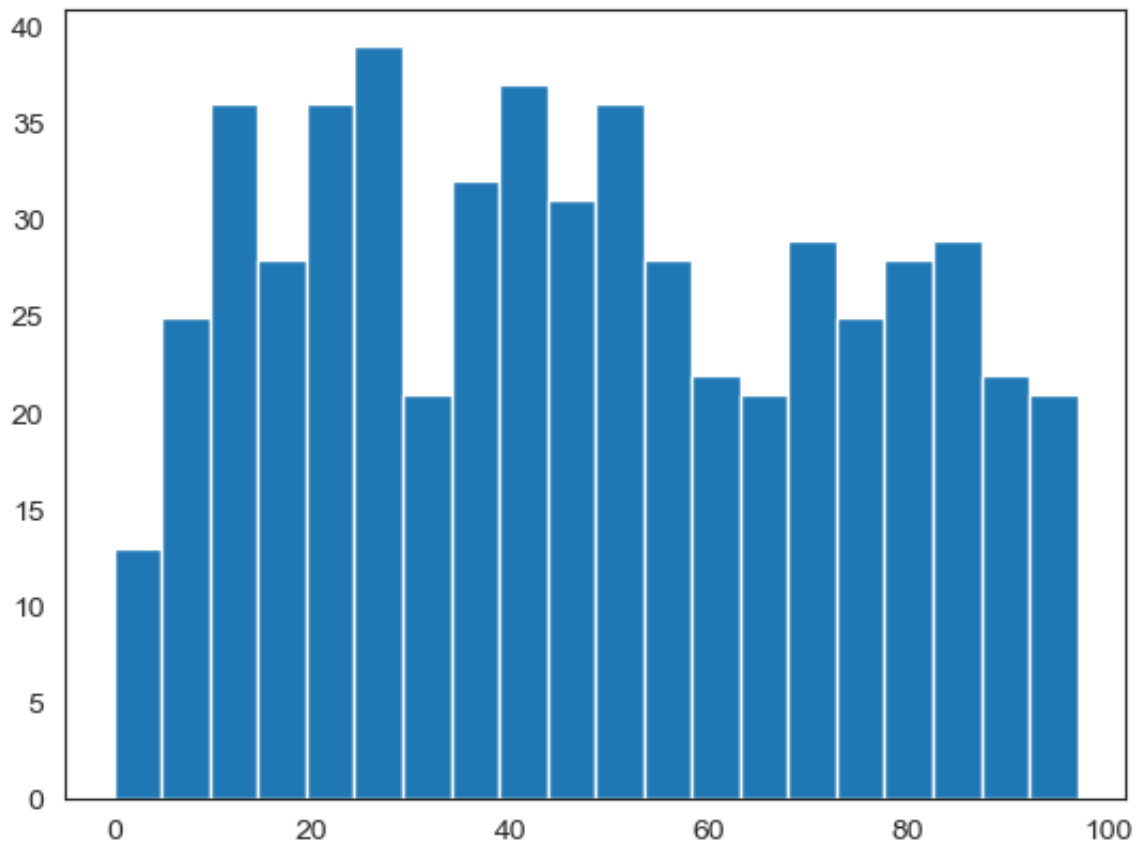
```
In [95]: m2 = sns.distplot(movies.AudienceRating, bins = 15)
```



```
In [89]: n1 = plt.hist(movies.AudienceRating, bins=15)
```

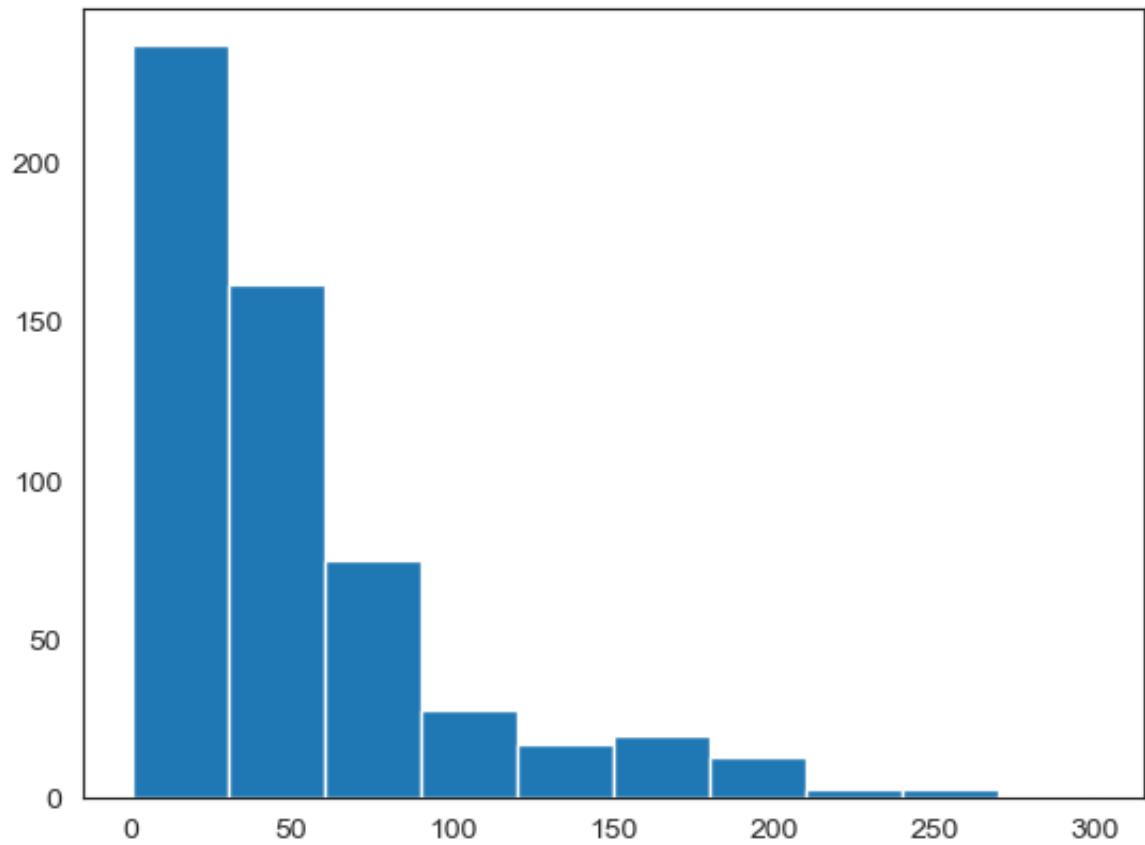


```
In [96]: n1 = plt.hist(movies.CriticRating, bins=20) #uniform distribution
```



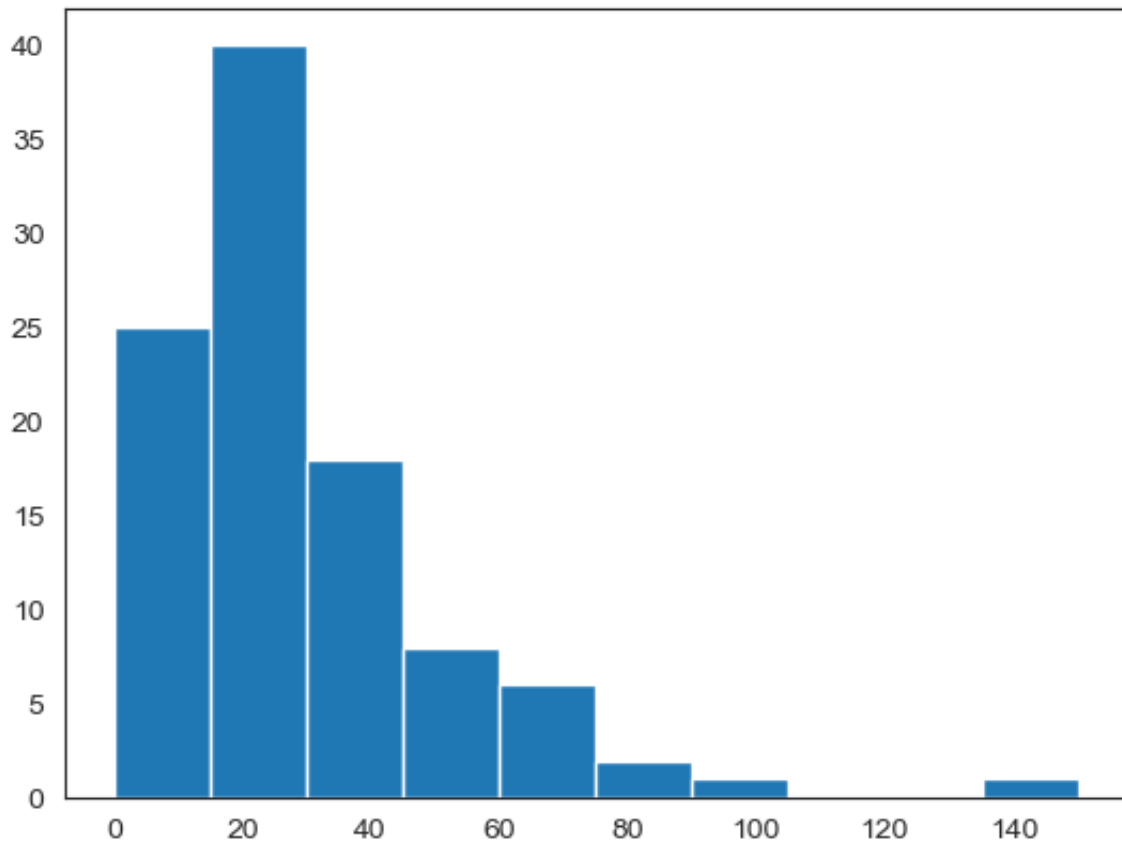
```
In [97]: #h1 = plt.hist(movies.BudgetMillions)

plt.hist(movies.BudgetMillions)
plt.show()
```



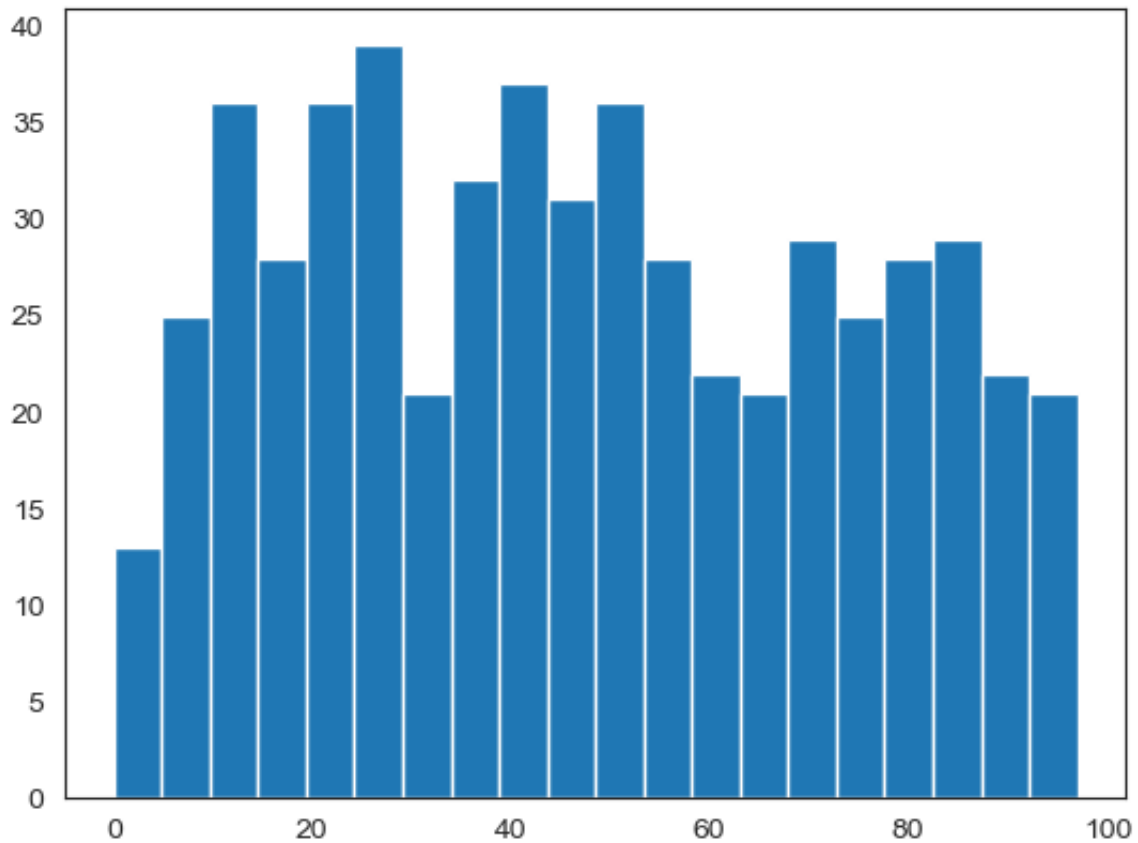
In []:

```
In [98]: plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
plt.show()
```

```
In [100...] plt.hist(movies.CriticRating,bins=20)
```

```
Out[100...] (array([13., 25., 36., 28., 36., 39., 21., 32., 37., 31., 36., 28., 22.,
        21., 29., 25., 28., 29., 22., 21.]),
        array([ 0. ,  4.85,  9.7 , 14.55, 19.4 , 24.25, 29.1 , 33.95, 38.8 ,
        43.65, 48.5 , 53.35, 58.2 , 63.05, 67.9 , 72.75, 77.6 , 82.45,
        87.3 , 92.15, 97.  ]),
        <BarContainer object of 20 artists>)
```



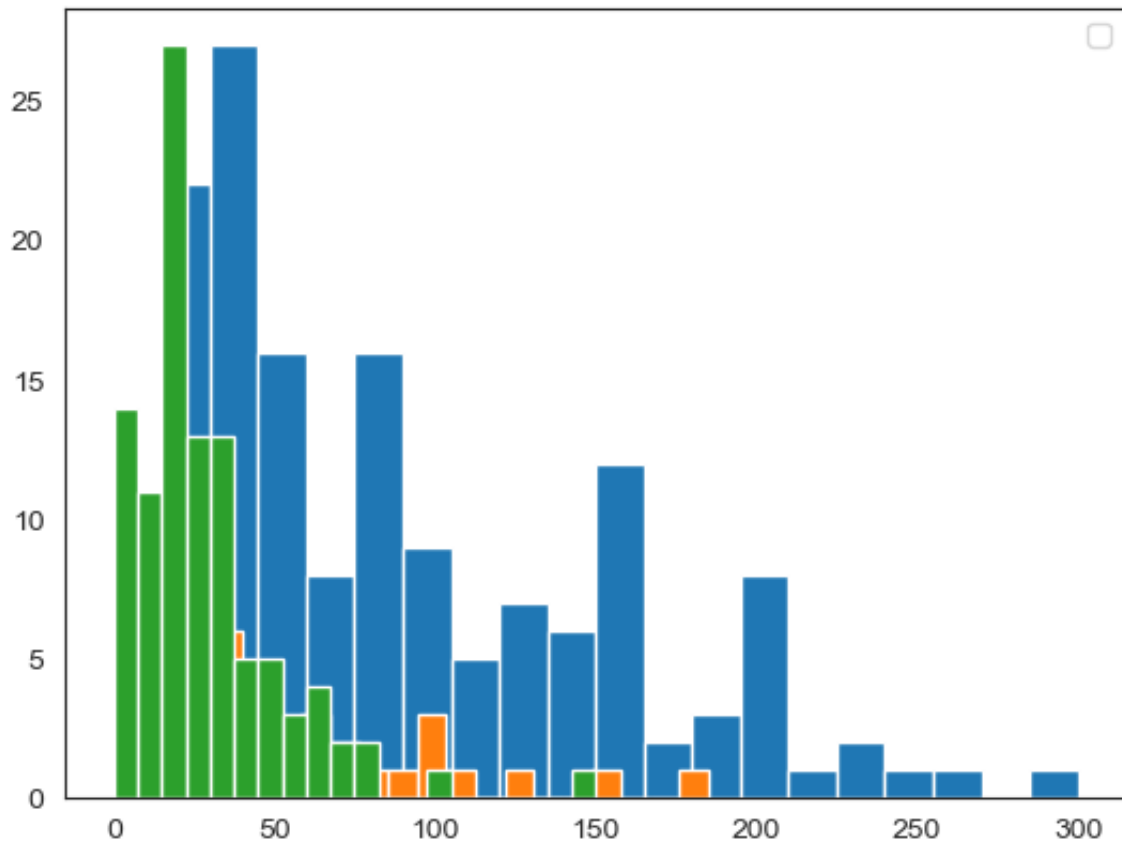
In [102... `movies.head()`

Out[102...

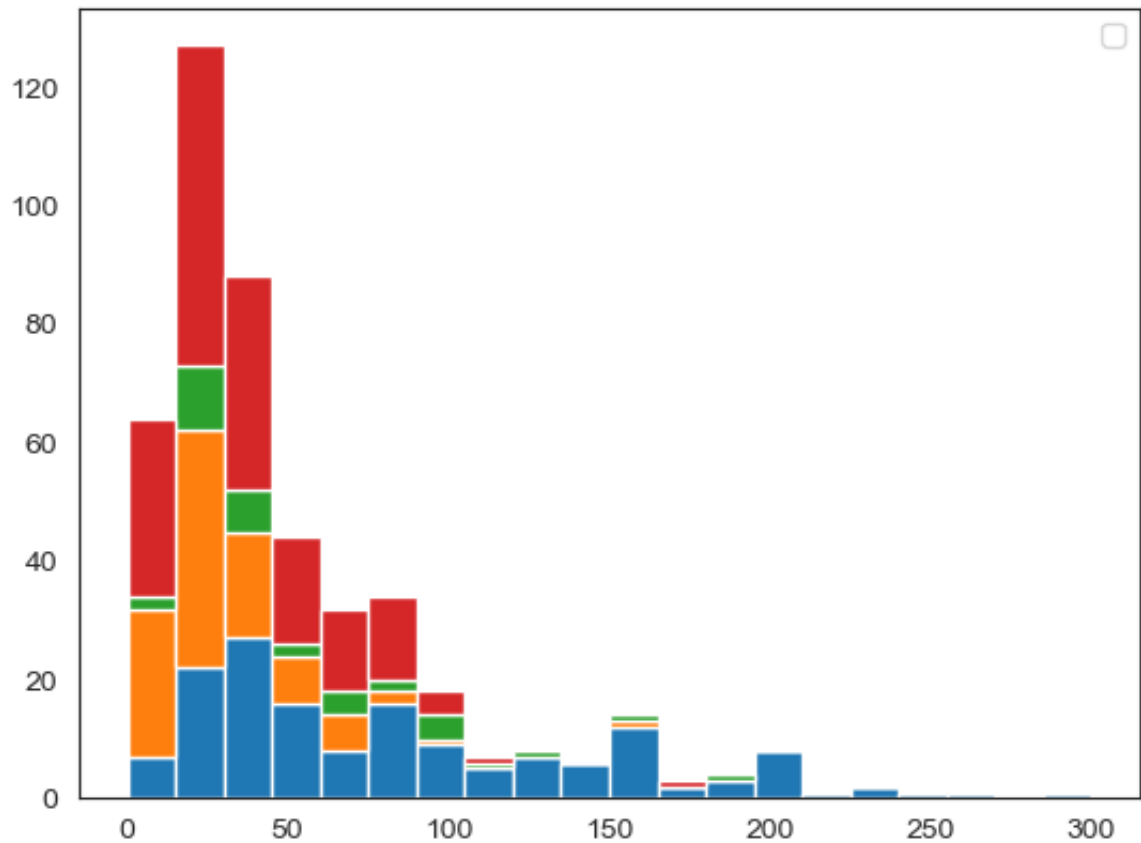
| | Film | Genre | CriticRating | AudienceRating | BudgetMillions | Year |
|---|----------------------|-----------|--------------|----------------|----------------|------|
| 0 | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |
| 1 | 10,000 B.C. | Adventure | 9 | 44 | 105 | 2008 |
| 2 | 12 Rounds | Action | 30 | 52 | 20 | 2009 |
| 3 | 127 Hours | Adventure | 93 | 84 | 18 | 2010 |
| 4 | 17 Again | Comedy | 55 | 70 | 20 | 2009 |

In [111... `plt.hist(movies[movies.Genre=='Action'].BudgetMillions,bins=20)`
`plt.hist(movies[movies.Genre=='Thriller'].BudgetMillions,bins=20)`
`plt.hist(movies[movies.Genre=='Drama'].BudgetMillions,bins=20)`
`plt.legend()`

Out[111... `<matplotlib.legend.Legend at 0x1467651d0>`



```
In [118... plt.hist([movies[movies.Genre == 'Action'].BudgetMillions,\n          movies[movies.Genre == 'Drama'].BudgetMillions, \n          movies[movies.Genre == 'Thriller'].BudgetMillions, \n          movies[movies.Genre == 'Comedy'].BudgetMillions],\n          bins = 20, stacked = True)\nplt.legend()\nplt.show()
```



```
In [119... for i in movies.Genre.cat.categories:
            print(i)
```

Action
Adventure
Comedy
Drama
Horror
Romance
Thriller

```
In [122... for m in movies.Film.cat.categories:
            print(m)
```

(500) Days of Summer
10,000 B.C.
12 Rounds
127 Hours
17 Again
2012
27 Dresses
30 Days of Night
30 Minutes or Less
50/50
88 Minutes
A Dangerous Method
A Nightmare on Elm Street
A Serious Man
A Very Harold and Kumar Christmas

Abduction
Across the Universe
Adventureland
Albert Nobbs
Alice in Wonderland
Alien vs. Predator -- Requiem
Aliens in the Attic
All About Steve
All Good Things
Amelia
American Gangster
An Education
Anonymous
Apollo 18
Appaloosa
Armored
Arthur
Australia
Avatar
Away We Go
Baby Mama
Babylon A.D.
Bad Teacher
Bandslam
Bangkok Dangerous
Battle: Los Angeles
Be Kind, Rewind
Beastly
Bedtime Stories
Beginners
Beowulf
Beverly Hills Chihuahua
Black Swan
Blades of Glory
Body of Lies
Bride Wars
Bridesmaids
Brooklyn's Finest
Brothers
Bruno
Buried
Burlesque
Burn After Reading
Captain America: The First Avenger
Case 39
Cats & Dogs: The Revenge of Kitty Galore
Cedar Rapids
Changeling
Cirque du Freak: The Vampire's Assistant
City Island
City of Ember
Clash of the Titans
Cloverfield

College Road Trip
Colombiana
Conan the Barbarian
Confessions of a Shopaholic
Contagion
Cop Out
Country Strong
Couples Retreat
Cowboys and Aliens
Crank: High Voltage
Crazy, Stupid, Love
Cyrus
Daddy Day Camp
Dance Flick
Date Night
Daybreakers
Dear John
Death Race
Death at a Funeral
Definitely, Maybe
Devil
Diary of a Wimpy Kid
Diary of a Wimpy Kid 2: Rodrick Rules
Did You Hear About the Morgans?
Dinner for Schmucks
Disaster Movie
District 9
Disturbia
Dolphin Tale
Don't Be Afraid Of The Dark
Doomsday
Doubt
Drag Me to Hell
Dragon Wars (aka D-War)
Dragonball Evolution
Dream House
Drillbit Taylor
Drive
Drive Angry
Due Date
Duplicity
Dylan Dog: Dead of Night
Eagle Eye
Easy A
Eat Pray Love
Enchanted
Epic Movie
Evan Almighty
Everybody's Fine
Everything Must Go
Extract
Extraordinary Measures
Extremely Loud and Incredibly Close

Fame
Fantastic Four: Rise of the Silver Surfer
Fast Five
Fast and Furious
Faster
Final Destination 5
Fired Up!
Fireproof
First Sunday
Fool's Gold
Footloose
For Colored Girls
Forgetting Sarah Marshall
Four Christmases
Friday the 13th
Friends With Benefits
Fright Night
From Paris with Love
Funny People
Furry Vengeance
G-Force
G.I. Joe: The Rise of Cobra
Gamer
Get Him to the Greek
Get Smart
Ghost Rider
Ghost Town
Ghosts of Girlfriends Past
Going the Distance
Good Luck Chuck
Gran Torino
Green Lantern
Green Zone
Greenberg
Grindhouse
Grown Ups
Gulliver's Travels
Hachiko: A Dog's Story
Hall Pass
Halloween
Halloween II
Hancock
Hanna
Hannah Montana: The Movie
Harold and Kumar Escape from Guantanamo Bay
Harry Potter and the Deathly Hallows Part 1
Harry Potter and the Half-Blood Prince
Harry Potter and the Order of the Phoenix
He's Just Not That Into You
Hellboy II: The Golden Army
Hesher
High School Musical 3: Senior Year
Hitman

Horrible Bosses
Hostel: Part II
Hot Tub Time Machine
Hotel for Dogs
Hugo
I Am Legend
I Am Number Four
I Love You Beth Cooper
I Love You Phillip Morris
I Love You, Man
I Now Pronounce You Chuck and Larry
I'm Still Here
Imagine That
Immortals
In Time
In the Land of Women
Inception
Indiana Jones and the Kingdom of the Crystal Skull
Inglourious Basterds
Inkheart
Insidious
Into the Wild
Iron Man
Iron Man 2
It's Complicated
J. Edgar
Jack and Jill
Jane Eyre
Jennifer's Body
Johnny English Reborn
Jonah Hex
Journey to the Center of the Earth
Jumper
Juno
Just Go With It
Just Wright
Kick-Ass
Killer Elite
Killers
Kit Kittredge: An American Girl
Knight and Day
Knocked Up
Knowing
Land of the Lost
Larry Crowne
Leap Year
Leatherheads
Legion
Letters to Juliet
License to Wed
Life as We Know It
Limitless
Lions for Lambs

Little Fockers
Live Free or Die Hard
Love & Other Drugs
Love Happens
MacGruber
Machete
Machine Gun Preacher
Mad Money
Made of Honor
Mamma Mia!
Margin Call
Marley and Me
Marmaduke
Max Payne
Meet Dave
Meet the Spartans
Michael Clayton
Midnight in Paris
Miracle at St. Anna
Mirrors
Miss Pettigrew Lives for a Day
Mission Impossible 4
Moneyball
Monte Carlo
Morning Glory
Mother and Child
Mr. Brooks
Mr. Popper's Penguins
Music and Lyrics
My Best Friend's Girl
My Life in Ruins
My Sister's Keeper
My Soul to Take
My Week with Marilyn
Nancy Drew
Nanny McPhee Returns
National Treasure: Book of Secrets
Never Back Down
Never Let Me Go
New Year's Eve
Next
Next Day Air
Nick and Norah's Infinite Playlist
Night at the Museum: Battle of the Smithsonian
Nights in Rodanthe
Nim's Island
Nine
Ninja Assassin
No Country for Old Men
No Reservations
No Strings Attached
Norbit
Not Easily Broken

Observe and Report
Obsessed
Ocean's 13
Old Dogs
One Day
One Missed Call
Orphan
Our Family Wedding
Our Idiot Brother
Over Her Dead Body
P.S. I Love You
Pandorum
Paranormal Activity
Paranormal Activity 2
Paranormal Activity 3
Pathfinder: Legend of the Ghost Warrior
Paul
Paul Blart: Mall Cop
Penelope
Percy Jackson & the Olympians: The Lightning Thief
Pineapple Express
Piranha 3-D
Pirate Radio (The Boat that Rocked, UK)
Pirates of the Caribbean: At World's End
Pirates of the Caribbean: On Stranger Tides
Please Give
Post Grad
Predators
Premonition
Priest
Prince of Persia: The Sands of Time
Prom Night
Punisher: War Zone
Push
Quantum of Solace
Quarantine
Race to Witch Mountain
Rachel Getting Married
Rambo
Real Steel
Red
Red Riding Hood
Red State
Remember Me
Repo Men
Resident Evil: Afterlife
Rise of the Planet of the Apes
Robin Hood
Role Models
Rush Hour 3
Salt
Sanctum
Saw IV

Saw VI
Scott Pilgrim vs. the World
Scream 4
Season Of The Witch
Semi-Pro
Seven Pounds
Sex Drive
Sex and the City
Sex and the City 2
Shame
Shark Night 3D
She's Out of My League
Sherlock Holmes
Sherlock Holmes 2
Shoot 'Em Up
Shooter
Shutter
Shutter Island
Skyline
Smart People
Solitary Man
Something Borrowed
Sorority Row
Soul Men
Soul Surfer
Source Code
Speed Racer
Spider-Man 3
Spy Kids: All The Time in the World
Star Trek
Stardust
Step Brothers
Step Up 2: The Streets
Step Up 3D
Stone
Stop-Loss
Street Fighter: The Legend of Chun-Li
Street Kings
Sucker Punch
Sunshine Cleaning
Super 8
Superbad
Superhero Movie
Surrogates
Swing Vote
TMNT
Take Me Home Tonight
Take Shelter
Taken
Takers
Taking Woodstock
Terminator Salvation
The A-Team

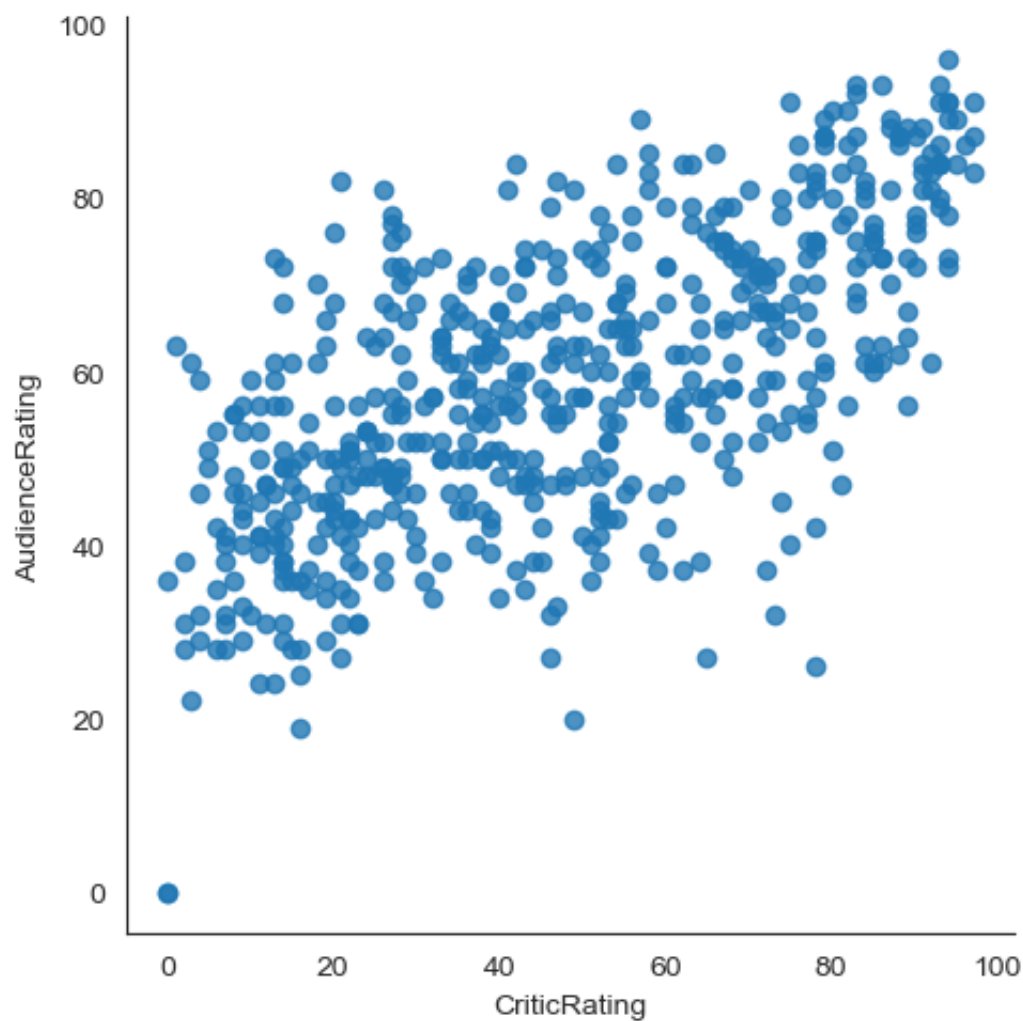
The Adjustment Bureau
The Adventures of Tintin
The Artist
The Back-up Plan
The Bank Job
The Beaver
The Big Year
The Book of Eli
The Bounty Hunter
The Bourne Ultimatum
The Box
The Boy in the Striped Pyjamas
The Brave One
The Bucket List
The Change Up
The Chronicles of Narnia: Prince Caspian
The Chronicles of Narnia: The Voyage of the Dawn Treader
The Company Men
The Conspirator
The Crazies
The Darjeeling Limited
The Dark Knight
The Day the Earth Stood Still
The Debt
The Descendants
The Dilemma
The Duchess
The Eagle
The Expendables
The Eye
The Fighter
The Final Destination
The Forbidden Kingdom
The Game Plan
The Girl With The Dragon Tattoo
The Golden Compass
The Green Hornet
The Hangover
The Hangover Part II
The Happening
The Haunting in Connecticut
The Haunting of Molly Hartley
The Heartbreak Kid
The Help
The Hills Have Eyes 2
The House Bunny
The Hurt Locker
The Ides of March
The Incredible Hulk
The Informant!
The International
The Invention of Lying
The Iron Lady

The Joneses
The Karate Kid
The Kids Are All Right
The Kingdom
The Kite Runner
The Last Airbender
The Last Exorcism
The Last Song
The Lincoln Lawyer
The Longshots
The Losers
The Love Guru
The Lovely Bones
The Mechanic
The Men Who Stare at Goats
The Mist
The Mummy: Tomb of the Dragon Emperor
The Muppets
The Nanny Diaries
The Other Boleyn Girl
The Other Guys
The Pink Panther 2
The Proposal
The Reaping
The Rite
The Road
The Roommate
The Ruins
The Rum Diary
The Secret Life of Bees
The Simpsons Movie
The Sitter
The Soloist
The Son Of No One
The Sorcerer's Apprentice
The Spirit
The Spy Next Door
The Strangers
The Switch
The Taking of Pelham 1 2 3
The Thing
The Three Musketeers
The Time Traveler's Wife
The Tourist
The Tree Of Life
The Twilight Saga: Eclipse
The Twilight Saga: New Moon
The Ugly Truth
The Unborn
The Uninvited
The Visitor
The Wolfman
The Women

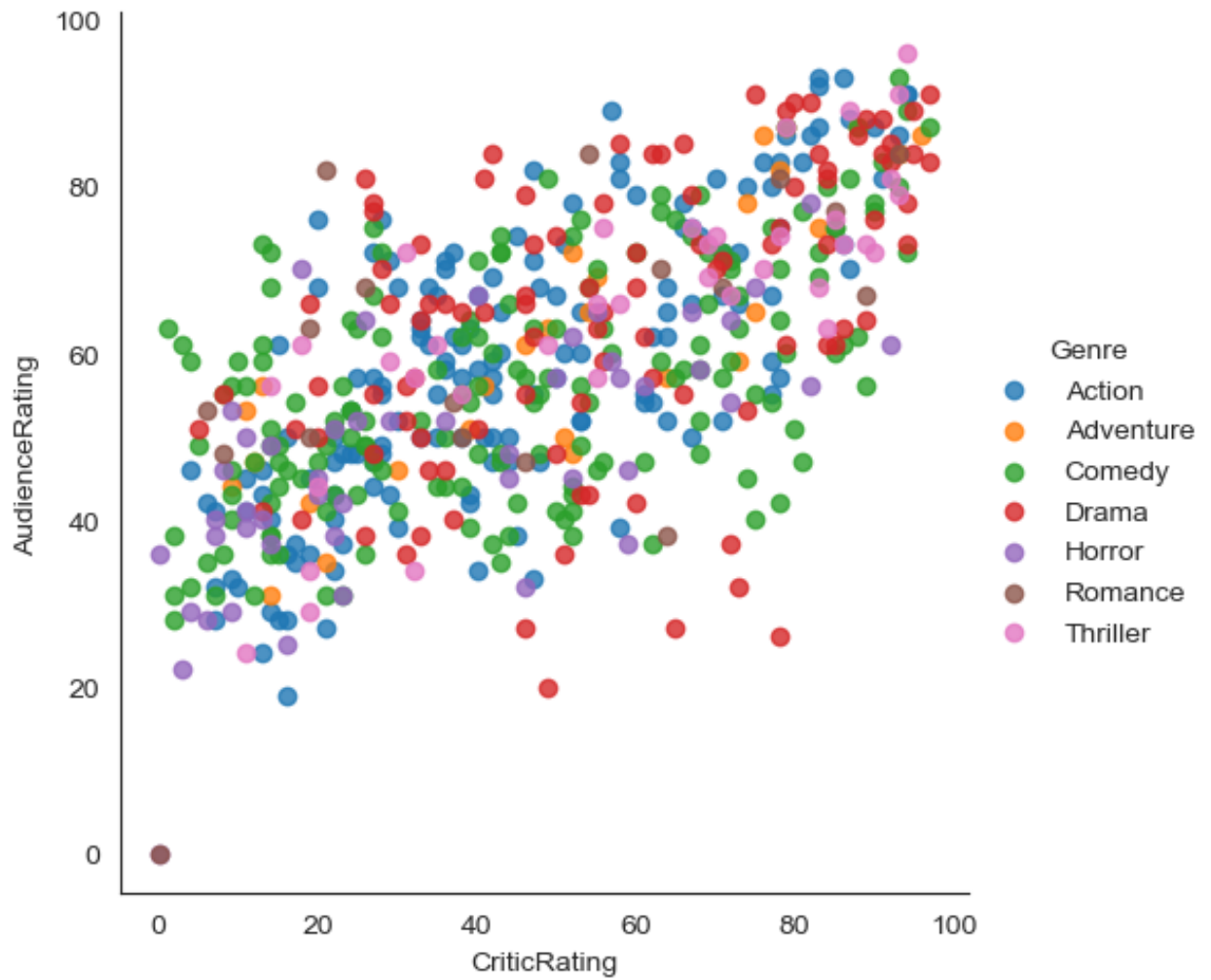
The X-Files: I Want to Believe
There Will Be Blood
Thor
Tinker Tailor Soldier Spy
Tooth Fairy
Tower Heist
Traitor
Transformers
Transformers: Dark of the Moon
Transformers: Revenge of the Fallen
Transporter 3
Trespass
Tron: Legacy
Tropic Thunder
True Grit
Twilight: Breaking Dawn
Tyler Perry's Meet the Browns
Tyler Perry's The Family That Preys
Tyler Perry's Why Did I get Married
Under the Same Moon
Underworld: Rise of the Lycans
Unknown
Unstoppable
Untraceable
Up in the Air
Vacancy
Valentine's Day
Valkyrie
Vampires Suck
Vanishing on 7th Street
Vantage Point
W.
Waiting For Forever
Waitress
Walk Hard: The Dewey Cox Story
Wall Street: Money Never Sleeps
Wanted
War Horse
Warrior
Watchmen
Water For Elephants
We Bought a Zoo
We Own the Night
Welcome Home, Roscoe Jenkins
What Happens in Vegas
What's Your Number
Whatever Works
When in Rome
Where the Wild Things Are
Whip It
Whiteout
Why Did I Get Married Too?
Wild Hogs

Winter's Bone
X-Men Origins: Wolverine
X-Men: First Class
Year One
Yes Man
Yogi Bear
You Again
You Don't Mess with the Zohan
You Will Meet a Tall Dark Stranger
Your Highness
Youth in Revolt
Zodiac
Zombieland
Zookeeper

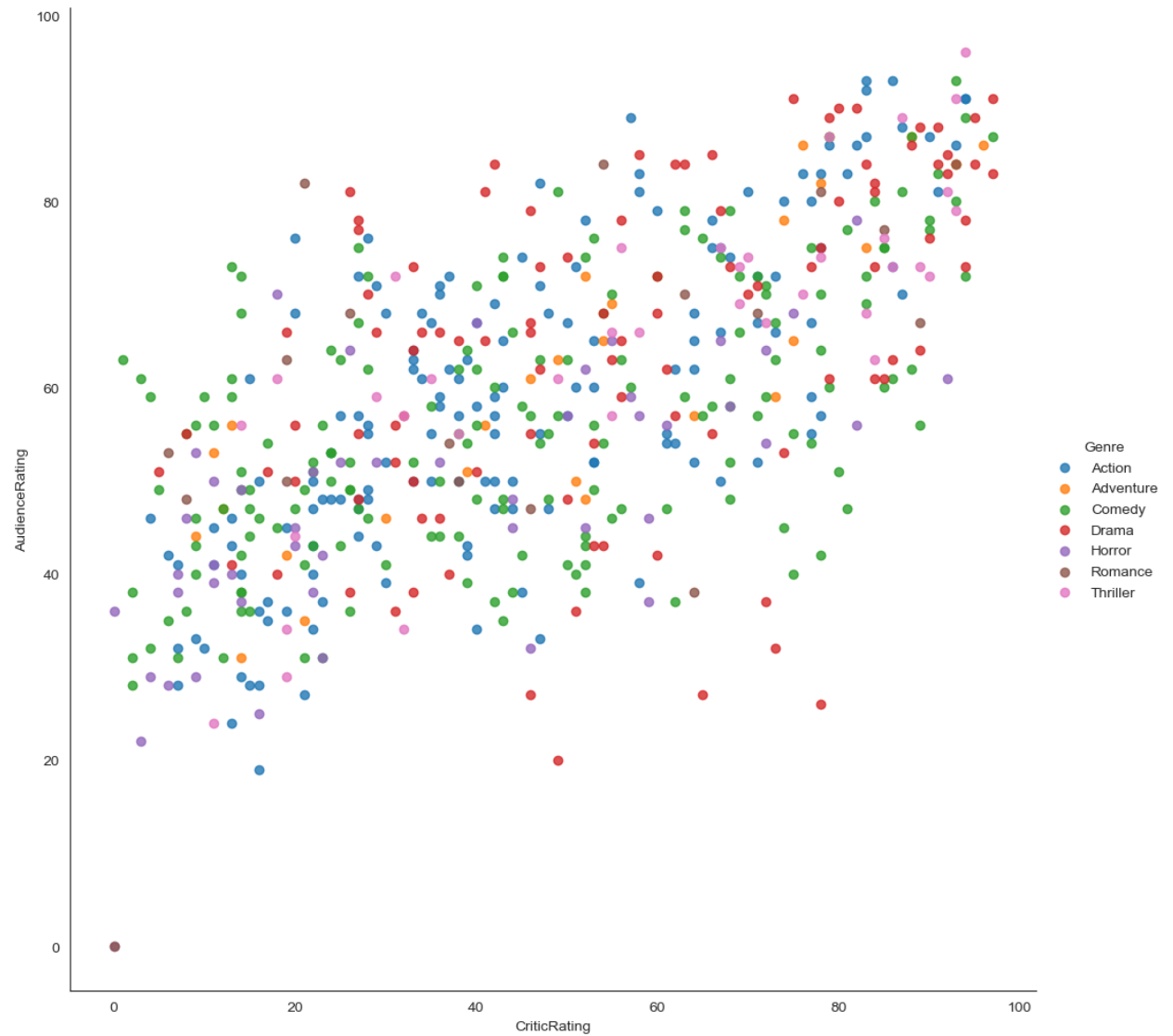
```
In [127... vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', fit_r
```



```
In [136... vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', fit_r
```



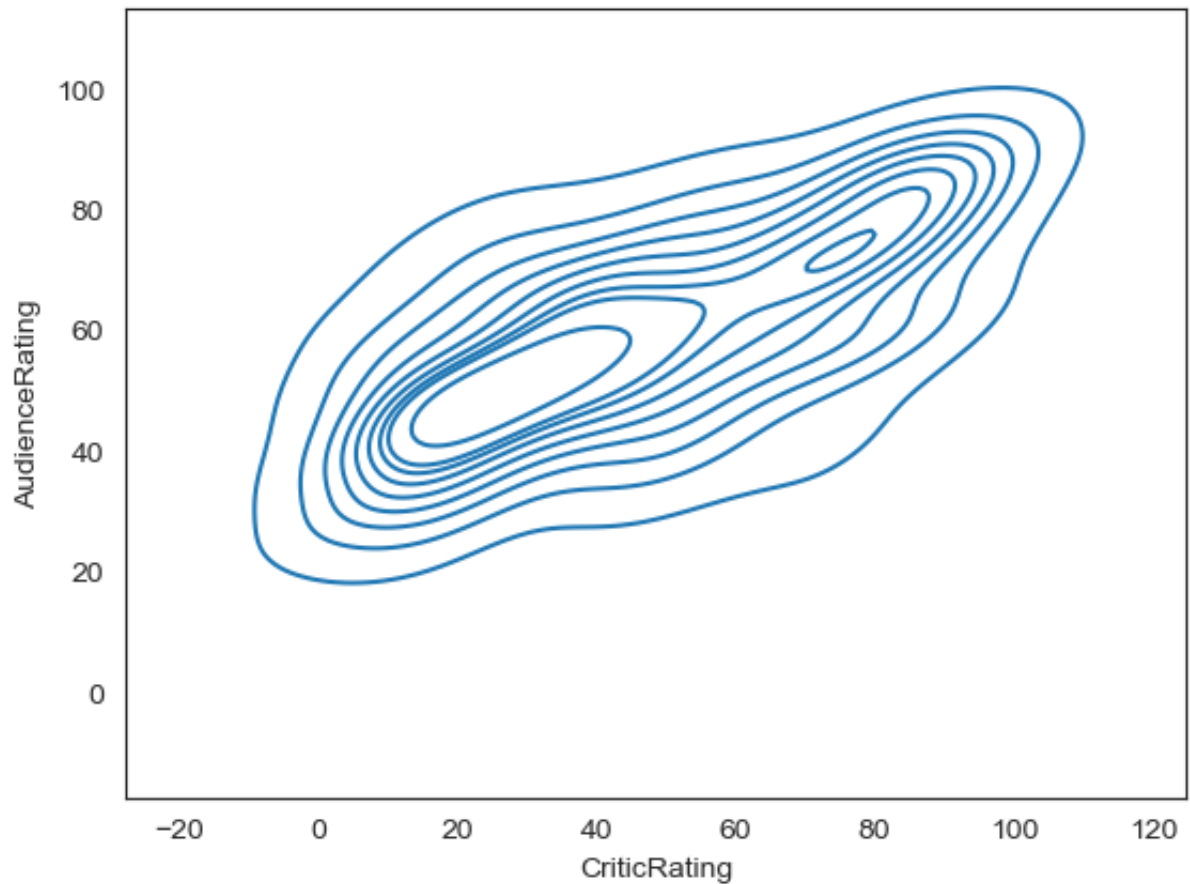
```
In [143... vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', fit_r
```

KERNEL DENSITY EXPLAINANTION PLOT KDE

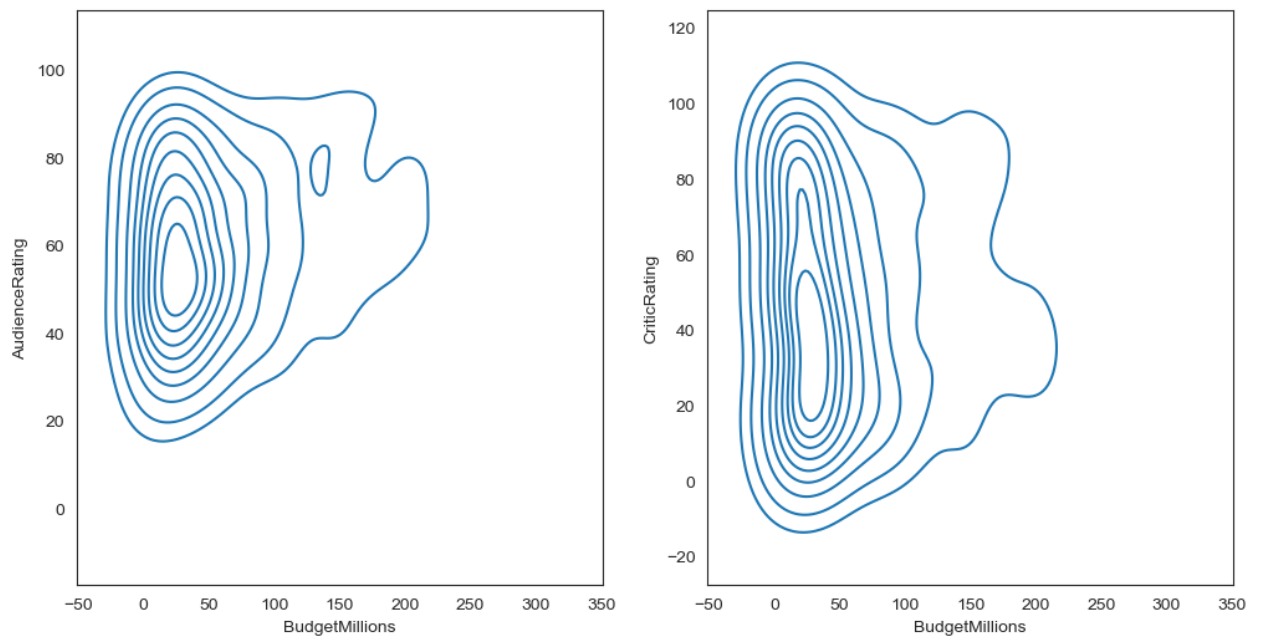
```
In [145...] sns.kdeplot(x=movies.CriticRating, y=movies.AudienceRating)
```

```
Out[145...] <Axes: xlabel='CriticRating', ylabel='AudienceRating'>
```

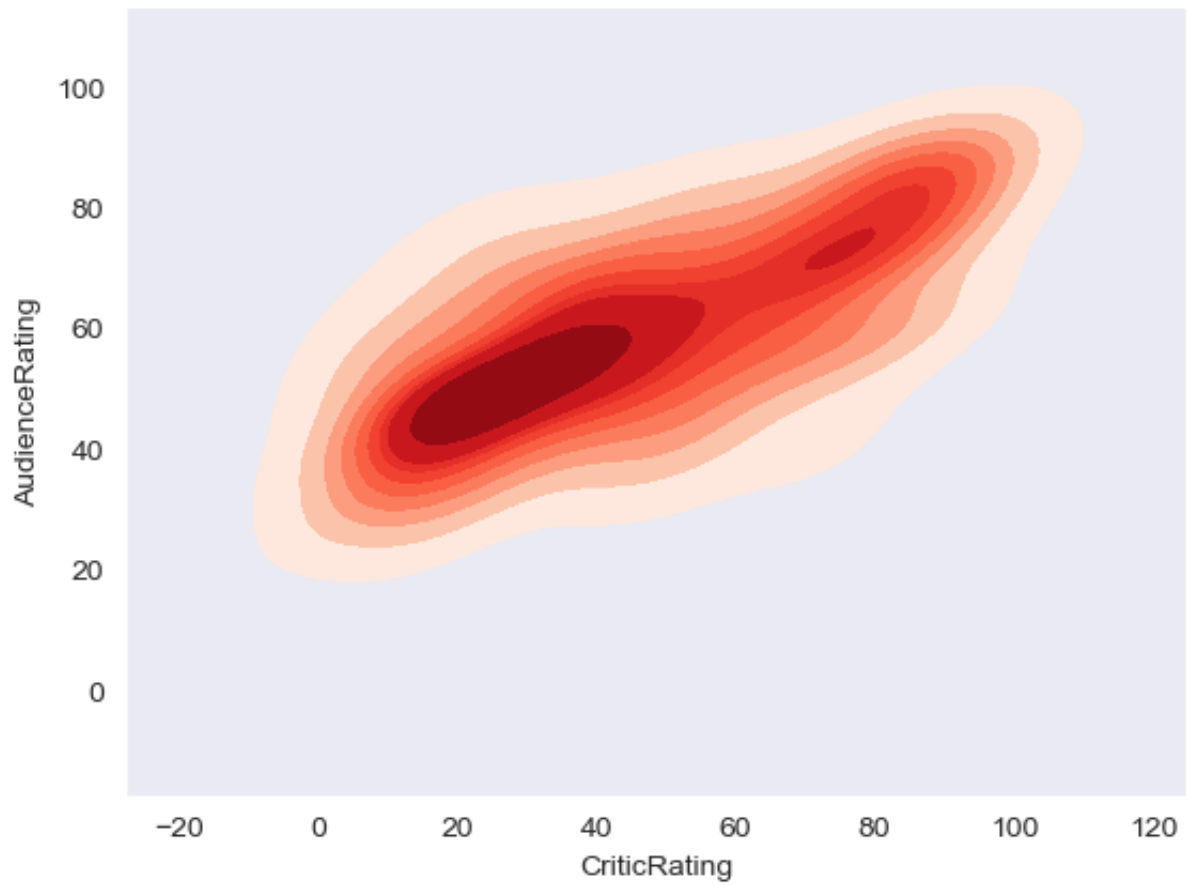


```
In [150... f, axes = plt.subplots(1,2, figsize =(12,6))

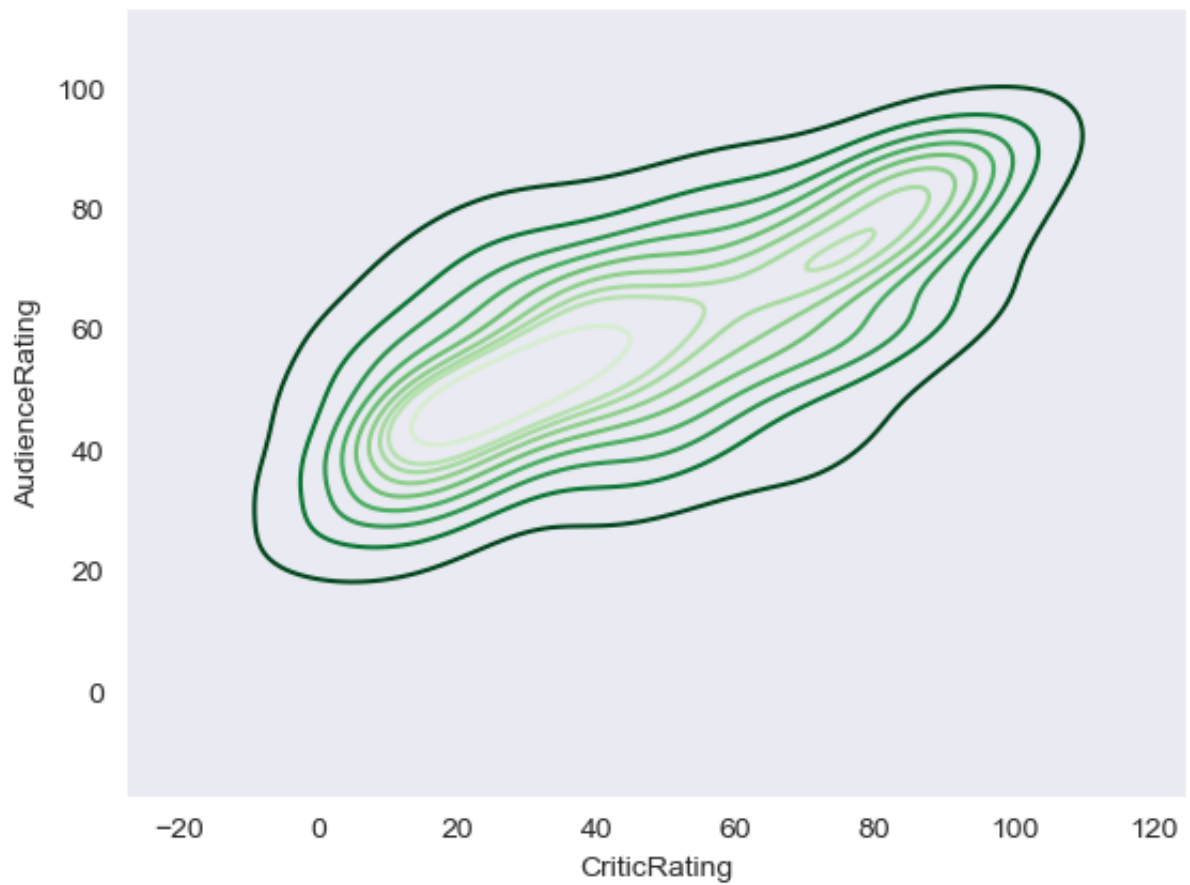
k1 = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,ax=axes[
k2 = sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating,ax=axes[1]
```



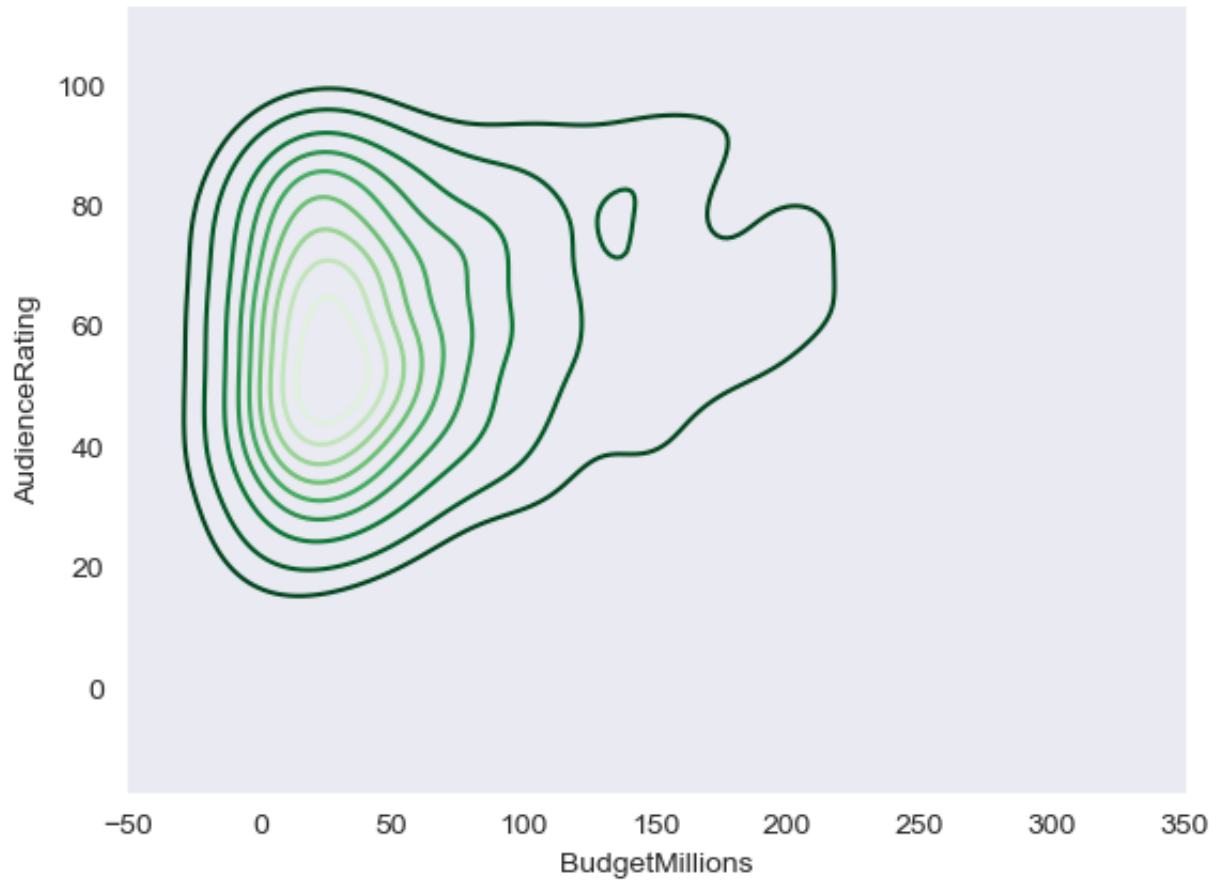
```
In [154... k1 = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade = Tr
sns.set_style("dark")
plt.show()
```



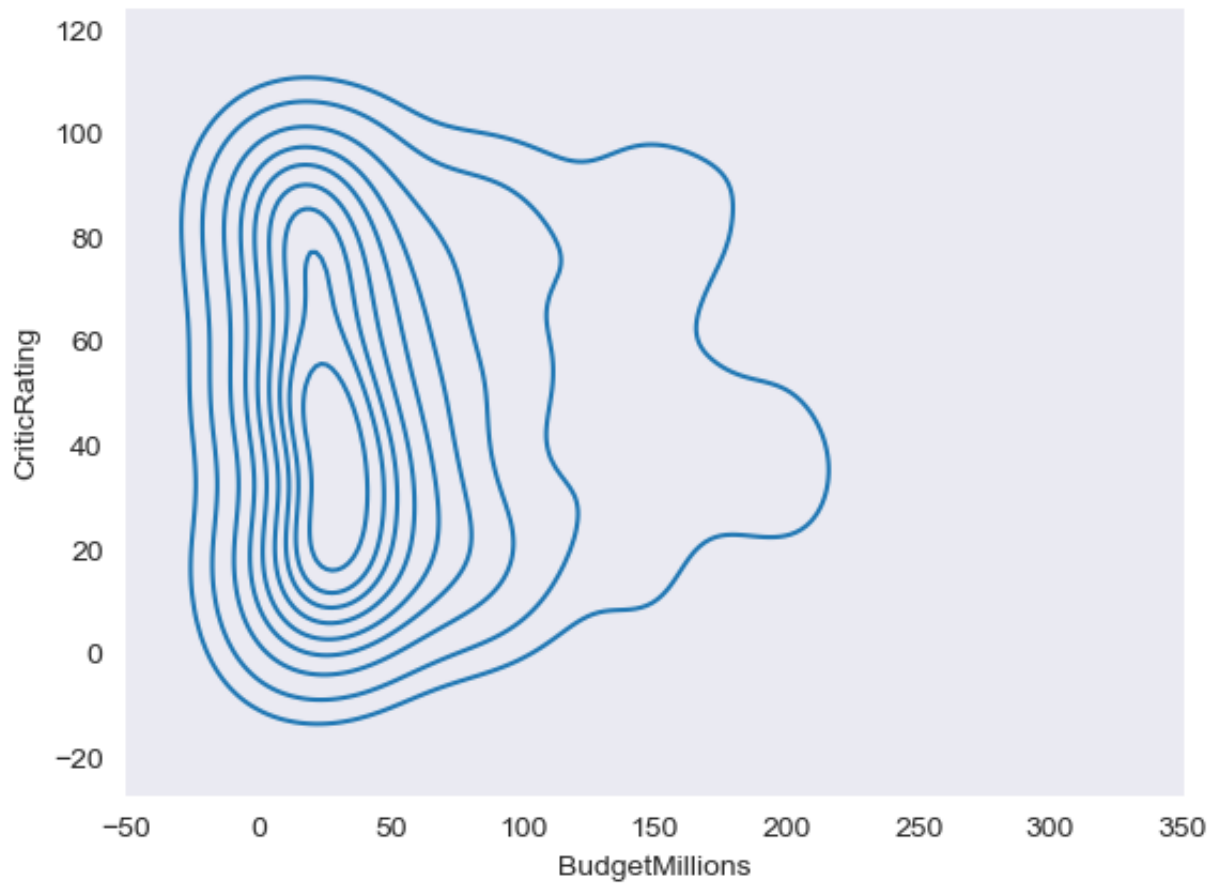
```
In [157... k1 = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade = Fa
```



```
In [158... k1 = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,shade =
```

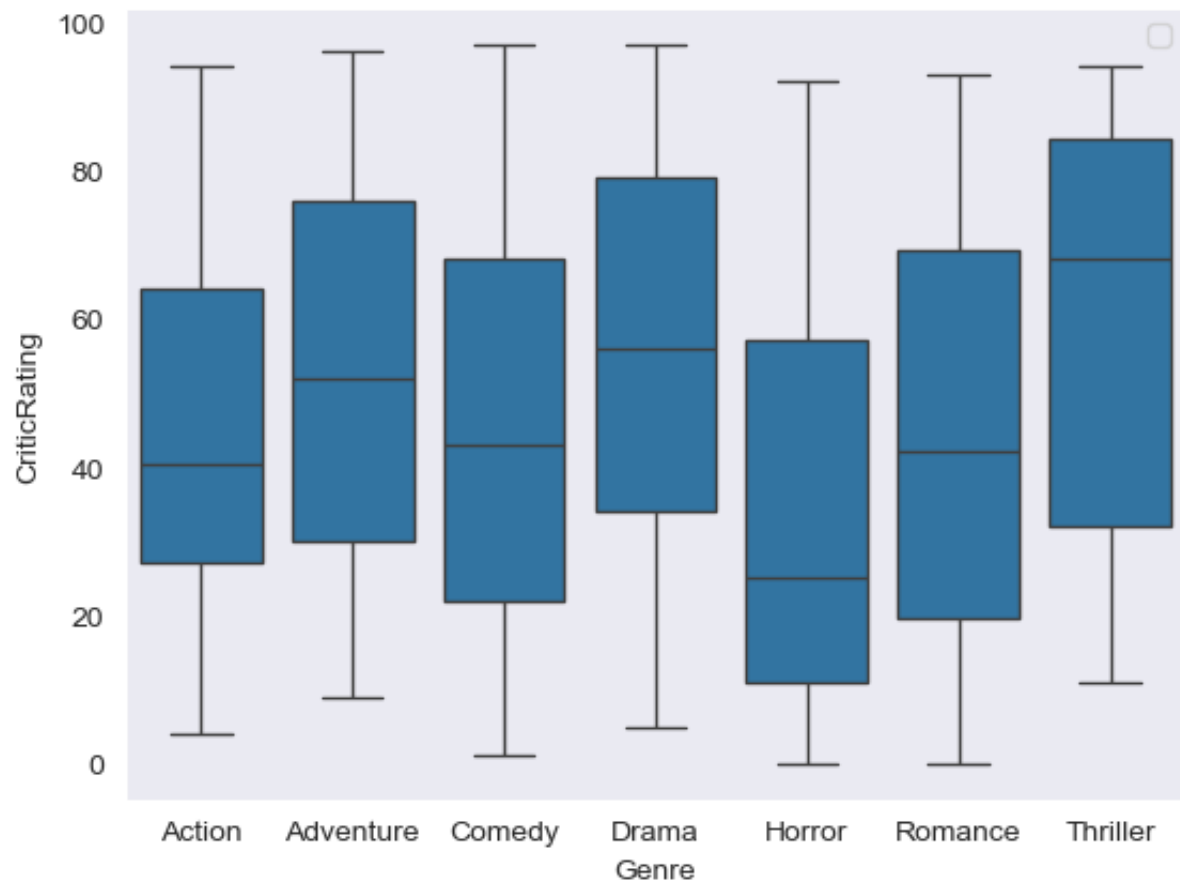


```
In [160... k=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating)
```

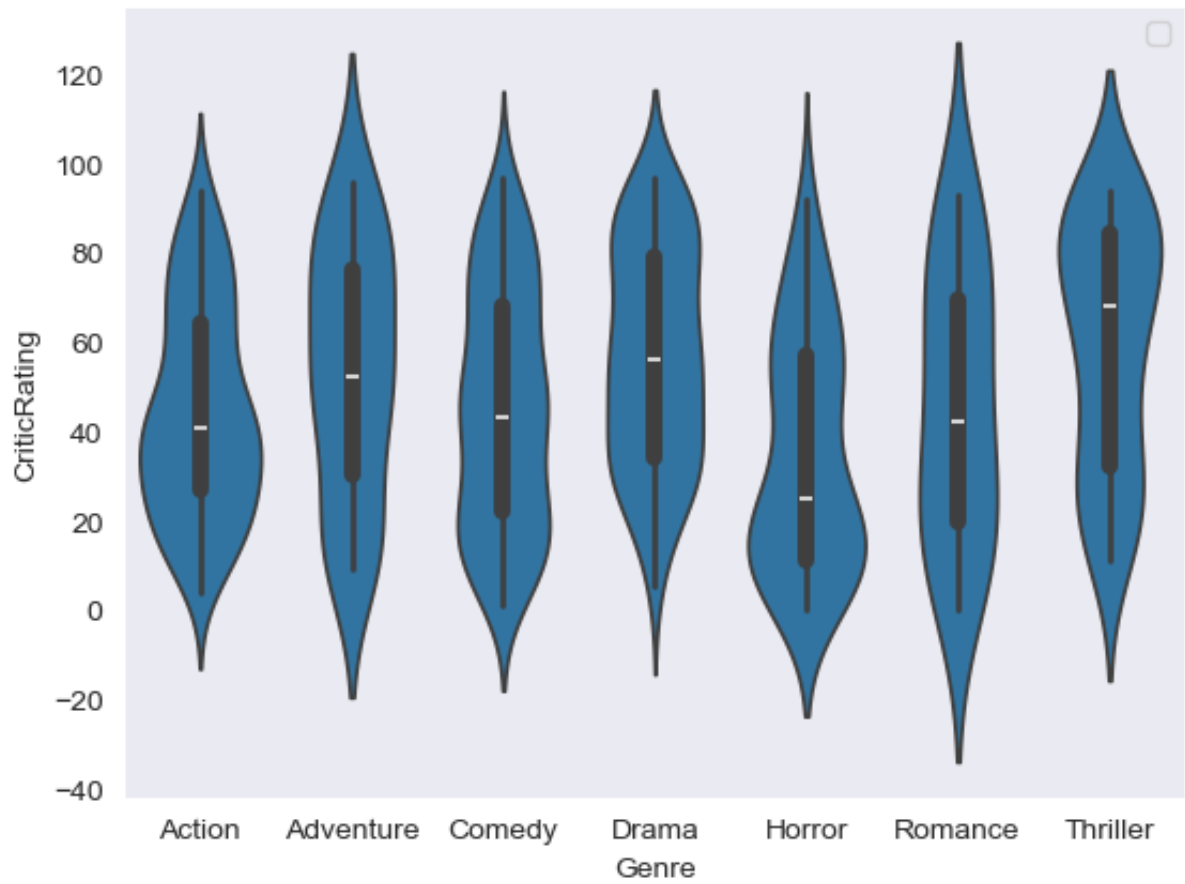


```
In [168... w = sns.boxplot(data=movies, x='Genre', y = 'CriticRating')  
plt.legend()
```

```
Out[168... <matplotlib.legend.Legend at 0x148084a50>
```



```
In [167... z = sns.violinplot(data=movies, x='Genre', y = 'CriticRating')
plt.legend()
plt.show()
```



In []: