INVESTIGATING THE MOVIE DATA BASE

The data set is from the movie data base which contains list of movies and its related parameters such as cast, director, production companies etc., from year 1960 to 2015.

We import proper libraries and read the .csv file to a data frame.

Data frame has 10866 rows and 21 columns.

List of columns are

```
Index(['id', 'imdb_id', 'popularity', 'budget', 'revenue', 'original_
title',
       'cast', 'homepage', 'director', 'tagline', 'keywords', 'overvi
ew',
       'runtime', 'genres', 'production_companies', 'release_date',
       'vote_count', 'vote_average', 'release_year', 'budget_adj',
       'revenue_adj'],
      dtype='object')
```

Columns in data frame has NaN rows, which can be cleaned as per need

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
id                      10866 non-null int64
imdb_id                 10856 non-null object
popularity              10866 non-null float64
budget                  10866 non-null int64
revenue                 10866 non-null int64
original_title          10866 non-null object
cast                    10790 non-null object
homepage                2936 non-null object
director                10822 non-null object
tagline                 8042 non-null object
keywords                9373 non-null object
overview                10862 non-null object
runtime                 10866 non-null int64
genres                  10843 non-null object
production_companies    9836 non-null object
release_date            10866 non-null object
vote_count              10866 non-null int64
vote_average            10866 non-null float64
release_year            10866 non-null int64
budget_adj              10866 non-null float64
revenue_adj             10866 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

It would be improper to clean data without knowing what we are looking for. So, we clean data along with answering a question.

Firstly, from the following data set, I would like to analyze,

WHICH GENRES ARE MOST POPULAR FROM YEAR TO YEAR?

To analyze and get the appropriate solution to the above posed question, we need to look at following columns

1) Genres
2) Popularity
3) Release_year

The column ‘popularity’ has no NaN, so it not needed to drop nulls, but the value counts of the popularity has long list of float values. Which will be hard to analyze. So, we round of the popularity column to nearest integer. Now, value counts of popularity counts looks as.

```
df['popularity']= round(df['popularity'])
```

```
df['popularity'].value_counts()
```

```
0.0     6739
1.0     3211
2.0      540
3.0      189
4.0       72
5.0       39
6.0       36
7.0       11
8.0       10
9.0        8
11.0       3
13.0       2
14.0       1
10.0       1
12.0       1
25.0       1
28.0       1
33.0       1
Name: popularity, dtype: int64
```

But there are outliers we need popularity on scale 10, so we drop the rows with popularity above 9.0.

```
df = df[df['popularity'] <= 9.0]
```

```
0.0    6739
1.0    3211
2.0     540
3.0     189
4.0      72
5.0      39
6.0      36
7.0      11
8.0      10
9.0       8
Name: popularity, dtype: int64
```

This is how the popularity looks after cleaning.

We then check for duplicated rows and drop them.

```
df[df.duplicated()]
```

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	...	overview	runtime	
2090	42194	tt0411951	1.0	30000000	967000	TEKKEN	Jon Foo|Kelly Overton|Cary-Hiroyuki Tagawa|Ian...	NaN	Dwight H. Little	Survival is no game	...	In the year of 2039, after World Wars destroy ...	92	Crime|Drama|Action|Thrille

1 rows × 21 columns

```
df = df.drop(2090)
```

```
df[df.duplicated()]
```

id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	...	overview	runtime	genres	production_companies	release_date

0 rows × 21 columns

With the following data frame, we have many genres, most of the genres have little movies associated with them, which are trivial for our analyzation, so we filter them out.

```
df = df.groupby("genres").filter(lambda x: len(x) >= 50)
```

we just used groupby to filter out genre elements whose contents are more than 50, i.e we dropped genres with movies less than 50.

```
df['genres'].value_counts()
```

```
Drama                                712
Comedy                               712
Documentary                          312
Drama|Romance                        289
Comedy|Drama                         280
Comedy|Romance                       268
Horror|Thriller                      259
Horror                               253
Comedy|Drama|Romance                 222
Drama|Thriller                       138
Comedy|Family                        102
Action|Thriller                      100
Thriller                              93
Drama|Comedy                          92
Animation|Family                      90
Crime|Drama|Thriller                  81
Crime|Drama                           74
Comedy|Horror                         72
Drama|Comedy|Romance                  64
Action                                63
Action|Comedy                         62
Drama|History                         58
Action|Crime|Drama|Thriller           54
Drama|Horror|Thriller                 53
Action|Crime|Thriller                 52
Horror|Science Fiction                52
Horror|Mystery|Thriller               51
Comedy|Crime                          50
Name: genres, dtype: int64
```

Release_year column does not need any cleaning, as it is clearly stated without any NaN values.

We groupby release_year and genres column and take the mean of the popularity column, we use mean over sum, because sum overlooks the importance of genres of little movies but with more popularity, so we take mean of popularity column.

```
res = df.groupby(['release_year','genres'])['popularity'].mean().reset_index()
```

We then take the list of all genres with maximum popularity in each year.

```
sol = list()
for year in res['release_year'].unique():
    sol.append(res.loc[(res[res['release_year']==year]['popularity'].max()==res['popularity']) & (res['release_year']==year)])
```

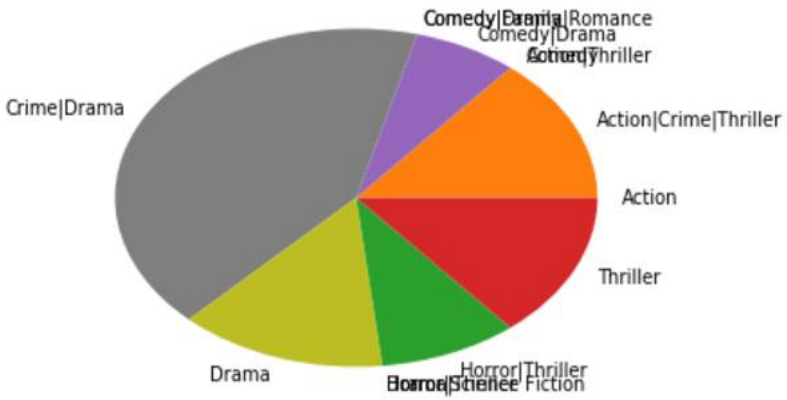
Now we define a function to get the most popular genres for a given year.

```
def popular_genre(year):
    for i in range(len(sol)):
        if all(sol[i]['release_year']==year):
            return sol[i]['genres'], plt.pie(res[res['release_year']==year]['popularity'],labels= res[res['release_year']==year][
```

So, we give a year to the function and it will give out all the genres according to its popularity in a pie chart.

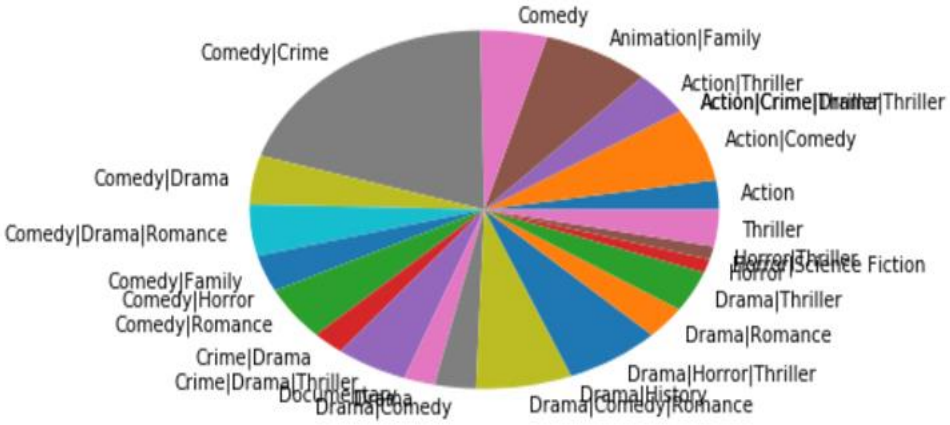
Example:

Year 1976: crime Drama is most popular



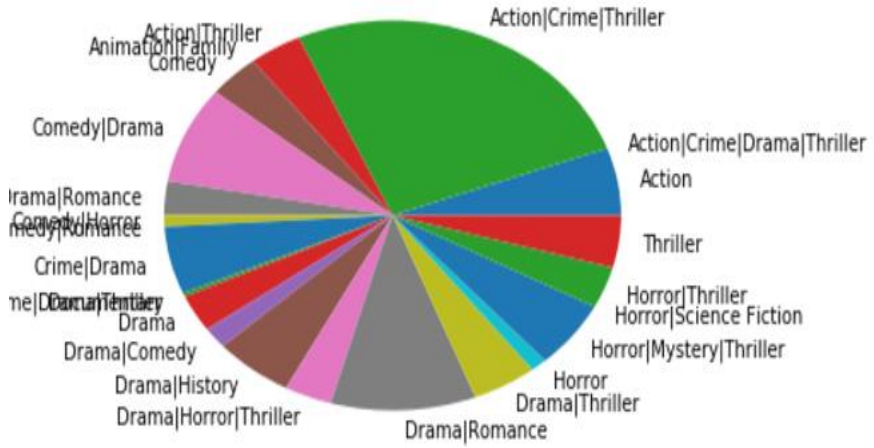
Inference: in year 1976 Crime drama has received more popularity, the same trend is observed over a decade. We can also observe that genres in old times are straight forward and will give the content of film straight away.

Year 2011: Crime Comedy is popular



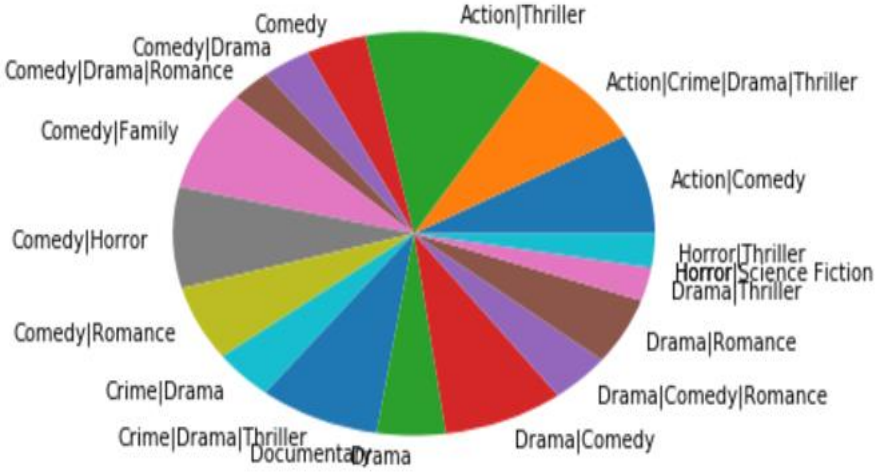
Inference: from past decade we can clearly observe that there is much more novelty in movies as more of Computer Graphics and animation come in. and super hero era starts and tend to produce more revenue and popularity

Year 2015 : Action Crime Thriller are popular



Inference: taste of viewers is tending to change more rapidly than ever, thanks to variety of genres that are avilable , in just 3 year Action thrillers became more popular than other genres.

Year 1995 : Action Thriller are popular

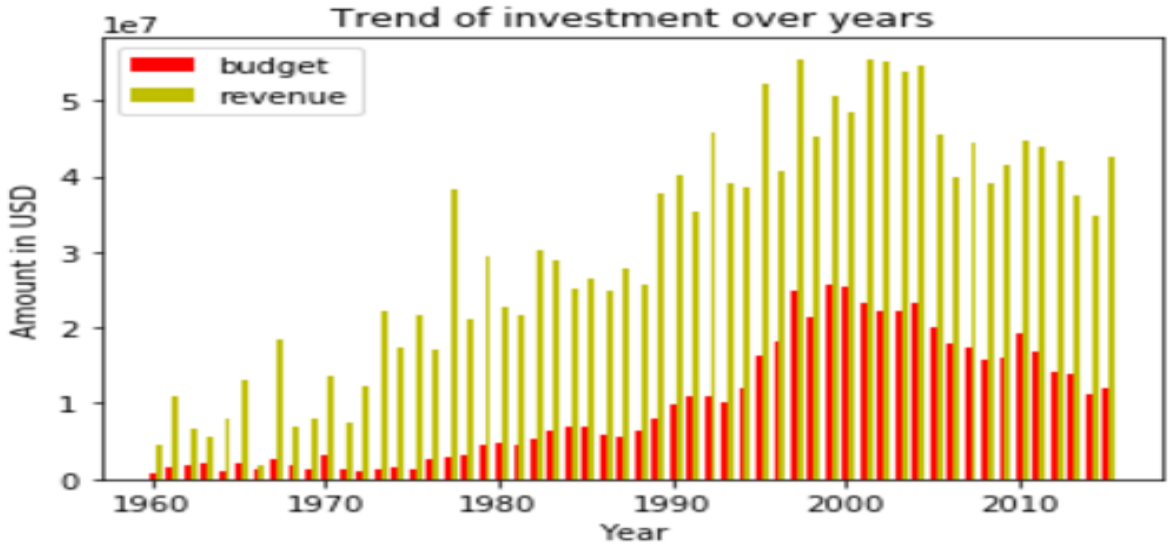


Inference: it is quite interesting to observe that in period 1990 – 1998, movies irrespective of genres has gained popularity, however Action Thrillers started to gain more popularity.

Plotting a yearlong trend of popular genres in a bar graph would create a mess, so to avoid it we created a function with input as year and then plotted the findings in a pie chart shown above.

TREND OF MEAN_REVENUE AND MEAN_BUDGET OVER THE YEAR?

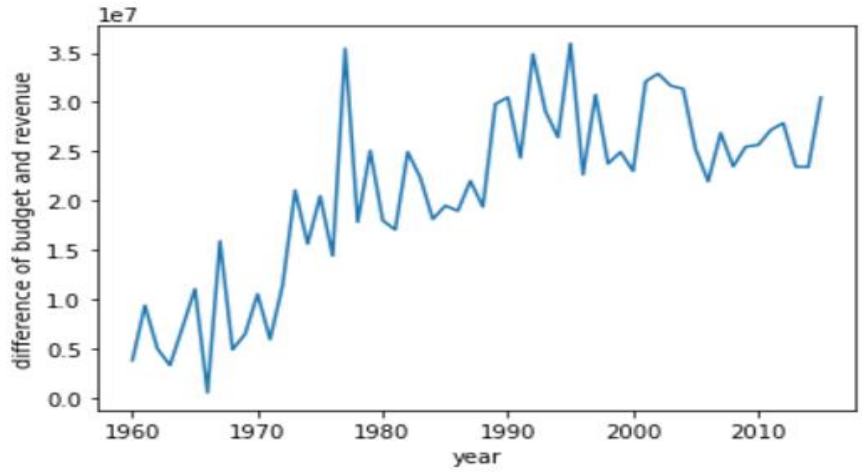
For this we groupby release year and take mean over budget and revenue columns and will plot the result.



Inference: We observe that there is steady increase in the investments and the revenues collected over the years. However, there is a major down fall in global recession period of 2007 -2008. Let's see how profitable the industry is over the years.

Create a profit column by substracting revenue and budget.

```
df['profit'] = np.subtract(df['revenue'],df['budget'])
```

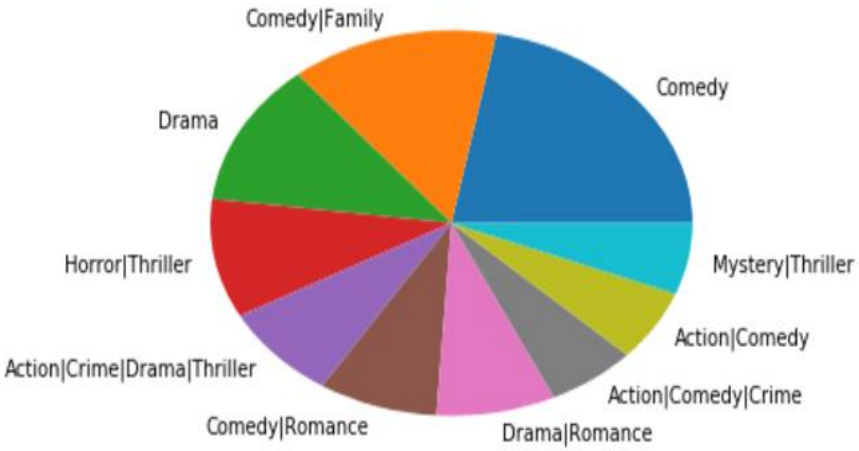


Inference: The profits have been increased by almost 3 times since 1960 to 2015. We can see the major down fall in global recession period of 2007 -2008, after that the profits are increasing at a steady rate

MOST NUMBER OF GENRES PRODUCED BY A COMPANY?

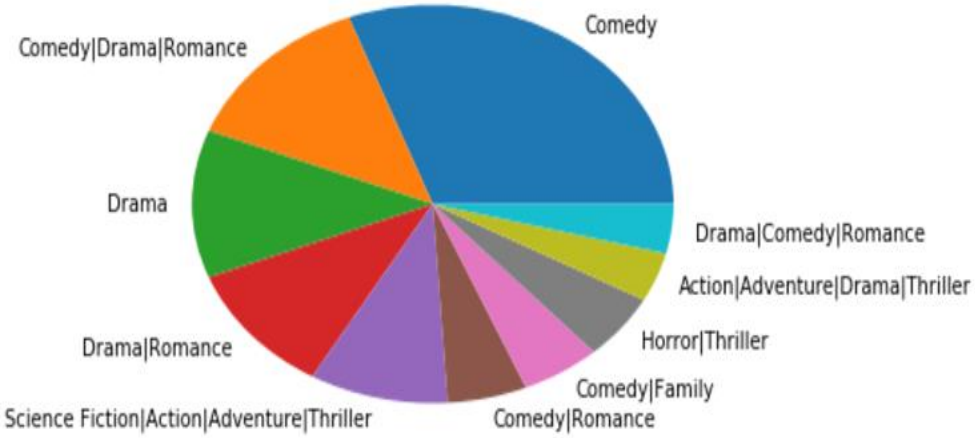
```
x=df[df['production_companies'] == 'Universal Pictures']['genres'].value_counts()[0:10]
plt.pie(x,labels=x.index.tolist());
```

Universal Studios: produced many comedy family genre movies



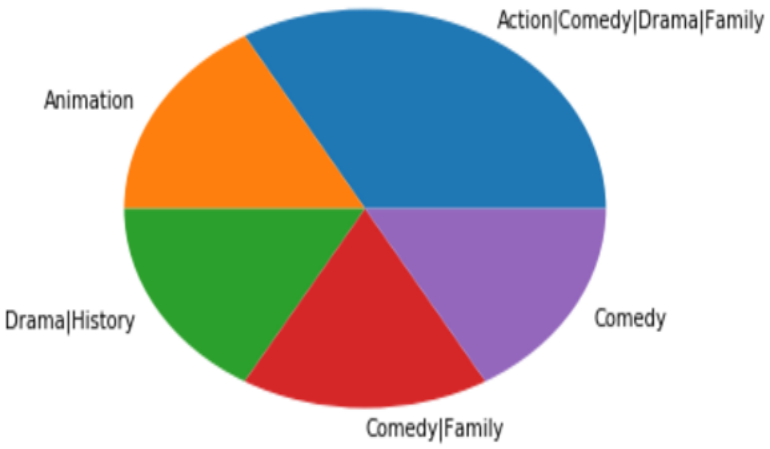
Inference: Universal studious produced most of genres which have comedy included in them, most of the comedy movies are made in earlier 1980's

Paramount Pictures: produced many Comedy Genre movies



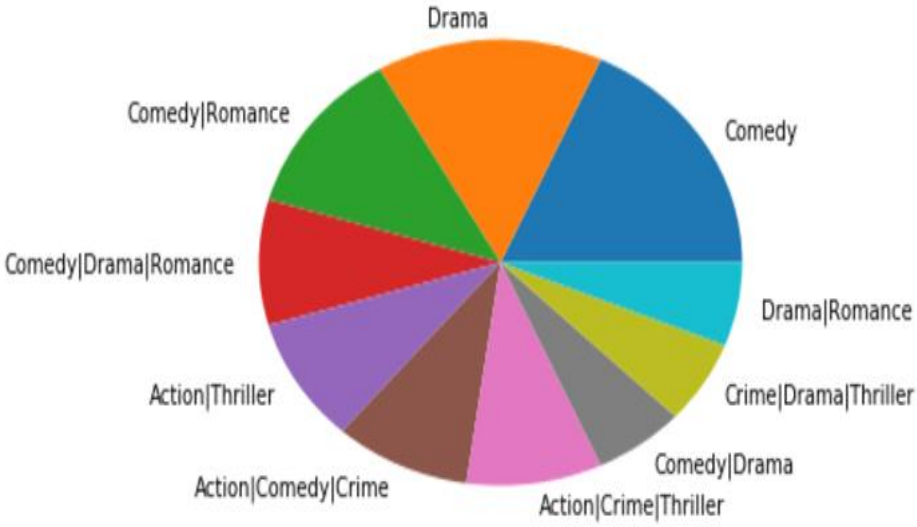
Inference: Paramount pictures has more comedy genre movies than any other production company.

Walt Disney: produced many Action Comedy Drama Family Genre movies



Inference: it is obvious that Walt Disney makes most of comedy and animated movies, which has huge impact in revenue generated in that particular genre.

Warner Bros.: produced many Drama Comedy Genre movies



Inference: Warner Bros. has more of Drama and comedy genre movies, company has gone with the flow of yearlong trends of genres that are producing highest revenues.

MOST REVENUE PRODUCING GENRES FOR A PRODUCTION COMPANY?

Let us do some cleaning, look for null data

```
df.isnull().sum()
```

```
id                        0
imdb_id                  10
popularity                0
budget                    0
revenue                   0
original_title            0
cast                     76
homepage               7930
director                 44
tagline                2824
keywords               1493
overview                  4
runtime                   0
genres                   23
production_companies   1030
release_date              0
vote_count                0
vote_average              0
release_year              0
budget_adj                0
revenue_adj               0
profit                    0
dtype: int64
```

If we use the dropna function, it would delete majority of rows as homepage has almost 7000 null values, so to avoid this we delete the columns with majority nulls and which are trivial for analyzing.

```
df =df.drop(['homepage','keywords','tagline'],axis =1)
```

```
df = df.dropna(how='any',axis=0)
```

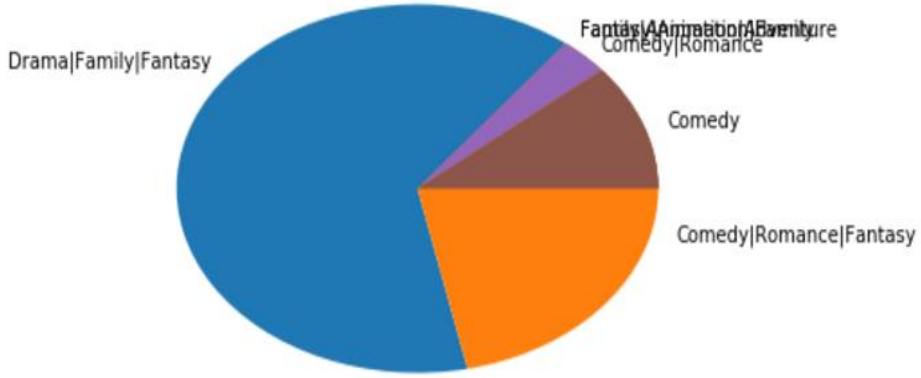
```
df.shape
```

```
(9771, 19)
```

Once the null values are deleted, we create a function for finding most revenue producing genres for the company and compare them with most genres produced by a company.

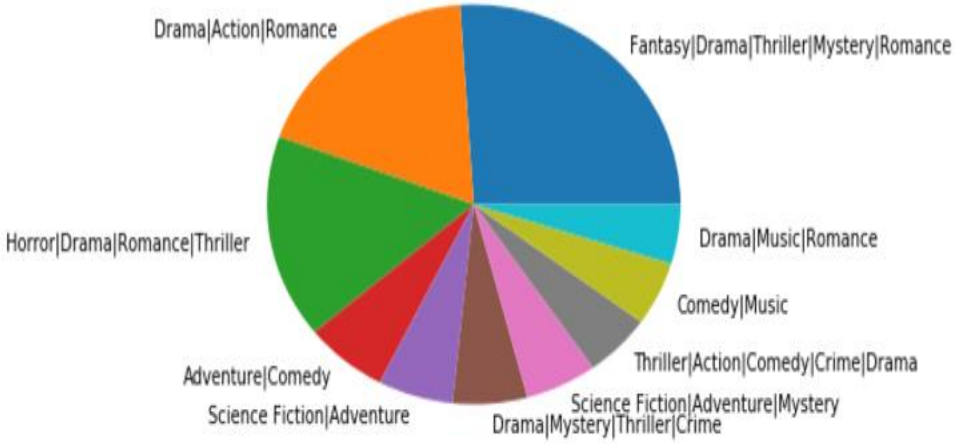
```
def revenue_genre(company):
    company = df[df['production_companies'] == company]
    com_gen= company.groupby('genres').mean().reset_index()
    com_gen.sort_values('profit',ascending=False,inplace=True)
    return plt.pie(com_gen['profit'][0:10],labels=com_gen['genres'][0:10]);
```

Universal Studios: obtained most profit producing Family Drama Genre movies



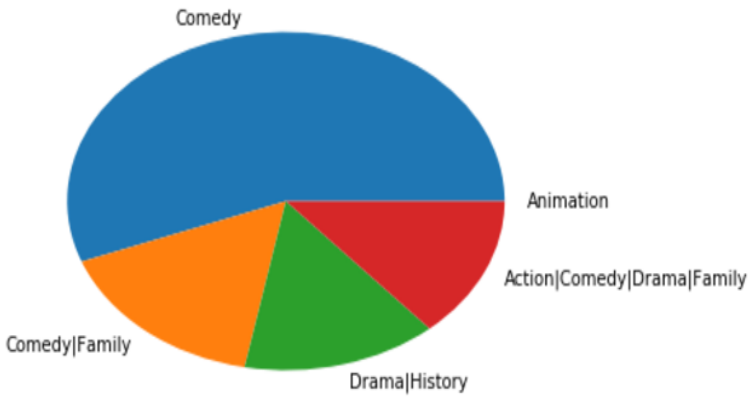
Inference: Drama has been some trust worthy revenue source genre movies for Universal studios, may be due to sequels and prequels that being produced in that genre. Even though comedy has been their most made genre, a good film in drama has more revenue produced than a bunch of comedy movies.

Paramount Pictures: obtained most profit producing Family Drama Mystery Genre movies



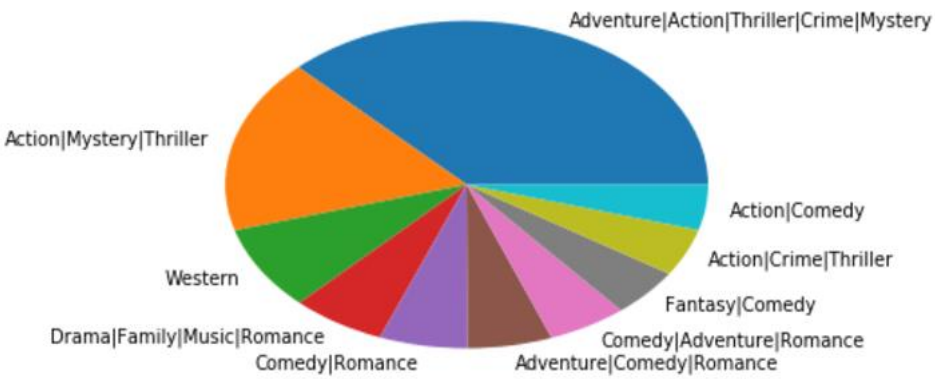
Inference: A detailed analysis is required in case of paramount pictures as there may be some missing parameters that limiting us to conclude about difference in most revenue generated sources and most commonly produced genres.

Walt Disney: obtained most profit producing comedy movies



Inference: observing the results from most produced genres and most revenue producing genres, Walt Disney has clearly followed the trend of producing what they are good at i.e. comedy and animated movies.

Warner Bros.: obtained most profit Family Drama Genre movies



Inference: even though there are lesser movies produced in Adventure action thriller crime mystery genre by Warner Bros. there are few iconic movies that stood apart generating more revenue than their routine comedy and drama movies.

Conclusion:

We observed over the years crime comedy has most revenue produced, we can conclude by saying that companies focusing on highest revenue producing genres have most popularity. With the given data set we can also find trends for a given month/season what genres are likely to produce more revenue. But the limited parameters have restricted us to get a region wise analysis, i.e., if at all the data has given sufficient amount information regarding region wise revenue and amount of budget spent on advertising we can analyze on how to generate more revenue for regions that have produced lower revenue.