DATA ANALYSIS REPORT

IMDb data from 2006

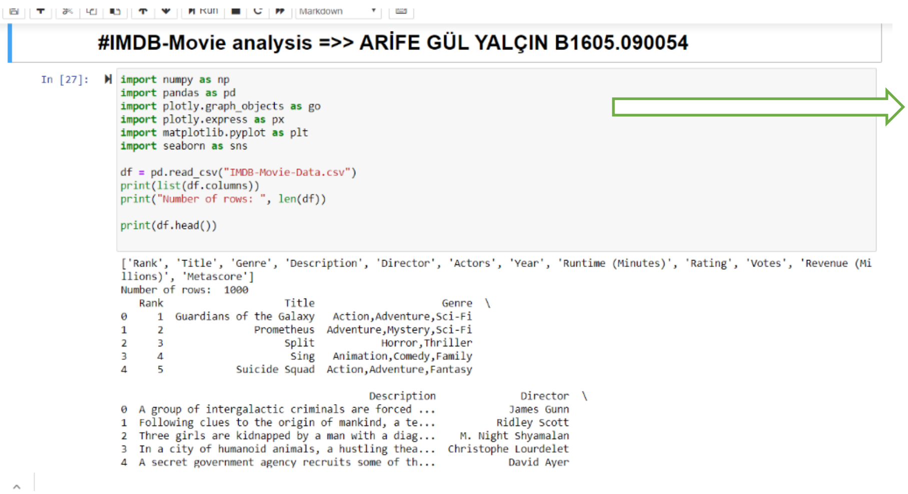
Arife Gül Yalçın

B1605.090054

A data set of 1000 popular movies on IMDb in the last years

First of all, I want to explain my aim in choosing this subject and dataset;

I personally like watching movies and doing research about them. Questions such as who directed the movie ,which movie is the most popular (user ranking) or what are movies genre, are very important. IMDb is a very advanced platform. By researching the movie you want to watch, you can reach very accurate results. On the IMDb platform, movies are rated by good people and appropriate ratings are revealed, which you will like almost. In addition, users can contribute to the scoring by voting.



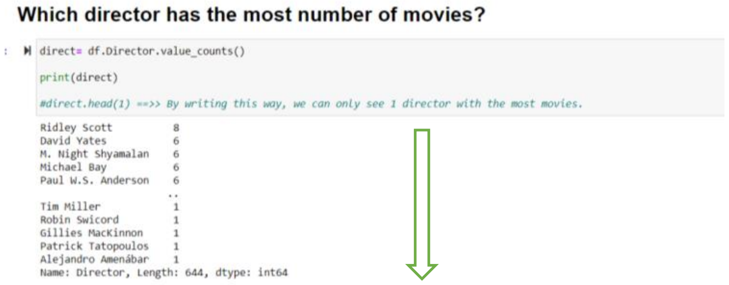
#IMDB-Movie analysis =>> ARİFE GÜL YALÇIN B1605.090054

```
In [27]:  import numpy as np
          import pandas as pd
          import plotly.graph_objects as go
          import plotly.express as px
          import matplotlib.pyplot as plt
          import seaborn as sns

          df = pd.read_csv("IMDB-Movie-Data.csv")
          print(list(df.columns))
          print("Number of rows: ", len(df))

          print(df.head())
```

```
['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year', 'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Mi
llions)', 'Metascore']
Number of rows:  1000
   Rank                    Title                     Genre  \
0     1  Guardians of the Galaxy   Action,Adventure,Sci-Fi
1     2               Prometheus  Adventure,Mystery,Sci-Fi
2     3                    Split           Horror,Thriller
3     4                     Sing   Animation,Comedy,Family
4     5            Suicide Squad  Action,Adventure,Fantasy

                                         Description              Director  \
0  A group of intergalactic criminals are forced ...            James Gunn
1  Following clues to the origin of mankind, a te...          Ridley Scott
2  Three girls are kidnapped by a man with a diag...    M. Night Shyamalan
3  In a city of humanoid animals, a hustling thea...  Christophe Lourdelet
4  A secret government agency recruits some of th...            David Ayer
```

In this part, I imported pandas to read our csv file. I also continue importing some things that will work with my future questions here.

⇨ In this section, we checked which columns we have in our csv file named 'IMDB-Movie-Data' and we read our file.

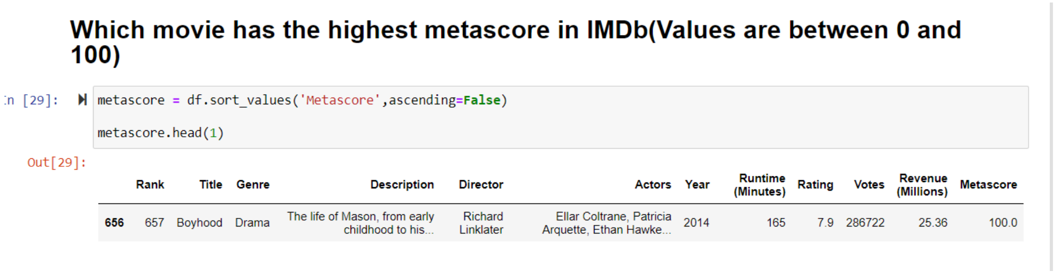


Which director has the most number of movies?

```
direct= df.Director.value_counts()

print(direct)

#direct.head(1) ==>> By writing this way, we can only see 1 director with the most movies.
```

```
Ridley Scott          8
David Yates           6
M. Night Shyamalan    6
Michael Bay           6
Paul W.S. Anderson    6
                     ..
Tim Miller            1
Robin Swicord         1
Gillies MacKinnon     1
Patrick Tatopoulos    1
Alejandro Amenábar    1
Name: Director, Length: 644, dtype: int64
```

I wrote this way to see more than one director in this section. But as I mentioned; By writing this way, we can only see 1 director with the most movies.



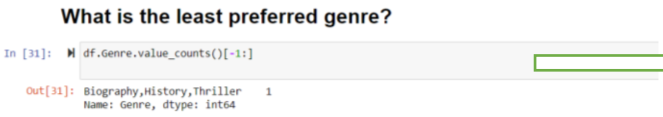
Which movie has the highest metascore in IMDb(Values are between 0 and 100)

```
In [29]:  metascore = df.sort_values('Metascore',ascending=False)

          metascore.head(1)
```

Out[29]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
656	657	Boyhood	Drama	The life of Mason, from early childhood to his...	Richard Linklater	Ellar Coltrane, Patricia Arquette, Ethan Hawke...	2014	165	7.9	286722	25.36	100.0

⇨ Here, Metascor is a value between 0 and 100, and by taking 100 Ms, the Boyhood movie in the Drama genre has the highest Ms.

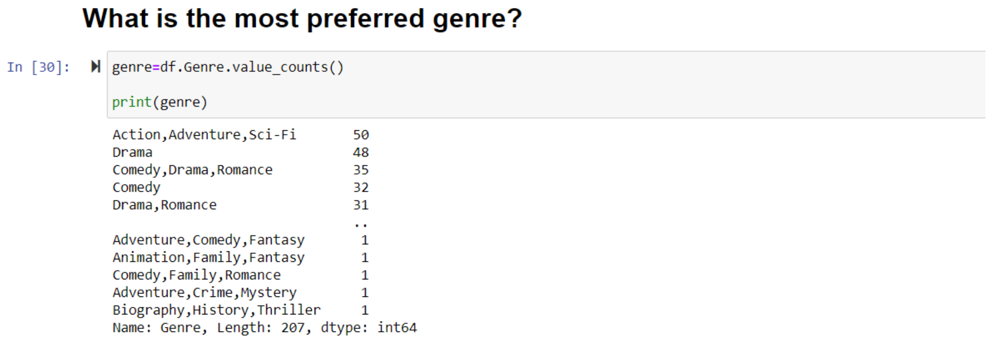


What is the least preferred genre?

```
In [31]:  df.Genre.value_counts()[-1:]

Out[31]:  Biography,History,Thriller    1
          Name: Genre, dtype: int64
```

[-1:] ➔ shows the least

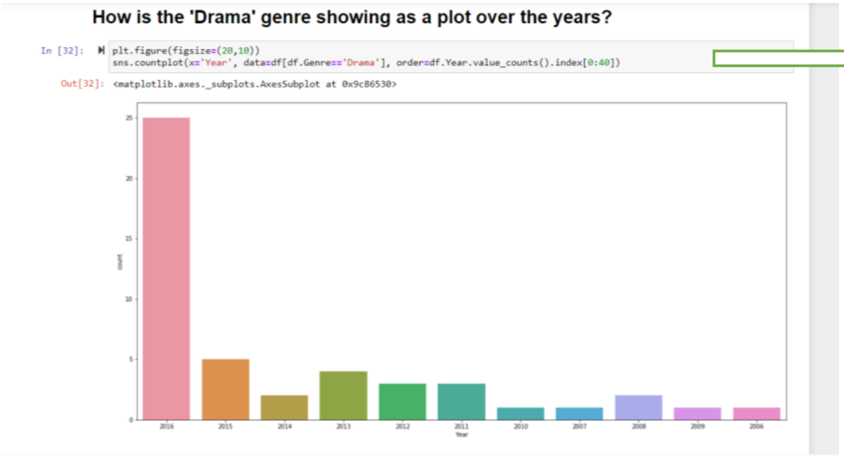
⇨ When we check out the least preferred movie genre, we have 1 movies in Action,Fantasy,Thriller.



What is the most preferred genre?

```
In [30]:  genre=df.Genre.value_counts()

          print(genre)

          Action,Adventure,Sci-Fi       50
          Drama                         48
          Comedy,Drama,Romance          35
          Comedy                        32
          Drama,Romance                 31
                                        ..
          Adventure,Comedy,Fantasy       1
          Animation,Family,Fantasy       1
          Comedy,Family,Romance          1
          Adventure,Crime,Mystery        1
          Biography,History,Thriller     1
          Name: Genre, Length: 207, dtype: int64
```

⇨ When we check out the most preferred movie genre, we have 50 movies in Action, Adventure, Sci-Fi. By writing this way, we can only see 1 genre .(genre.head(1))

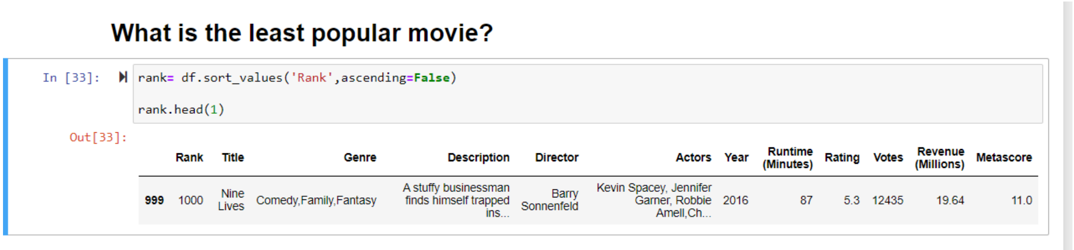


How is the 'Drama' genre showing as a plot over the years?

```
In [32]:  plt.figure(figsize=(20,10))
          sns.countplot(x='Year', data=df[df.Genre=='Drama'], order=df.Year.value_counts().index[0:40])

Out[32]:  <matplotlib.axes._subplots.AxesSubplot at 0x9c86530>
```

I use sns.countplot

⇨ In this section, we checked the plot representation of the 'Drama' type from our datasets between 2006-2016.



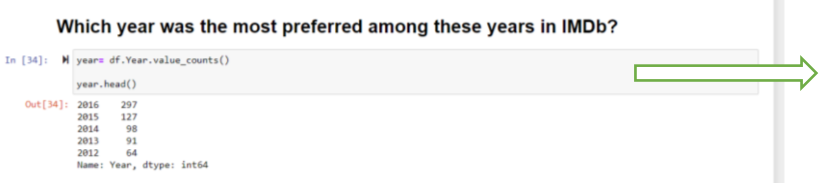
What is the least popular movie?

```
In [33]:  rank= df.sort_values('Rank',ascending=False)

          rank.head(1)
```

Out[33]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
999	1000	Nine Lives	Comedy,Family,Fantasy	A stuffy businessman finds himself trapped ins...	Barry Sonnenfeld	Kevin Spacey, Jennifer Garner, Robbie Amell,Ch...	2016	87	5.3	12435	19.64	11.0

⇨ The 'Nine Lives' movie at the bottom of the list is our least watched movie between these years.

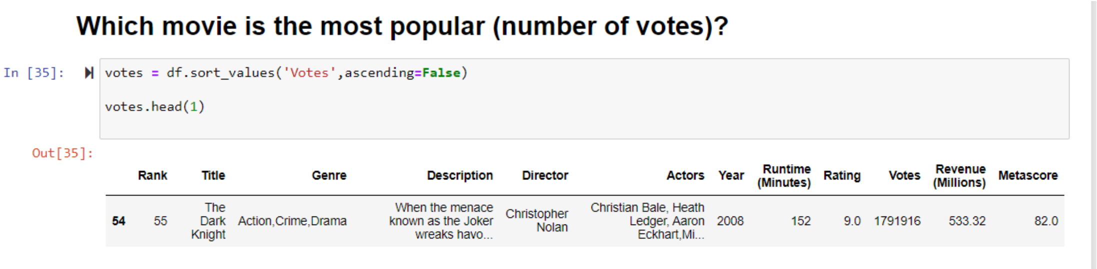


Which year was the most preferred among these years in IMDb?

```
In [34]:  year= df.Year.value_counts()

          year.head()

Out[34]:  2016    297
          2015    127
          2014     98
          2013     91
          2012     64
          Name: Year, dtype: int64
```

We can see the number of movies in all years

⇨ The most preferred year among these years in IMDb is 2016. This year there are 297 films.



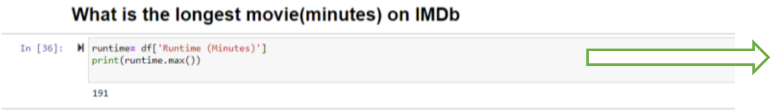
Which movie is the most popular (number of votes)?

```
In [35]:  votes = df.sort_values('Votes',ascending=False)

          votes.head(1)
```

Out[35]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
54	55	The Dark Knight	Action,Crime,Drama	When the menace known as the Joker wreaks havo...	Christopher Nolan	Christian Bale, Heath Ledger, Aaron Eckhart,Mi...	2008	152	9.0	1791916	533.32	82.0

⇨ In this part, the movie "The Dark Knight" in the genre of Action, Crime, Drama with the votes of 1791916 is the most popular movie.

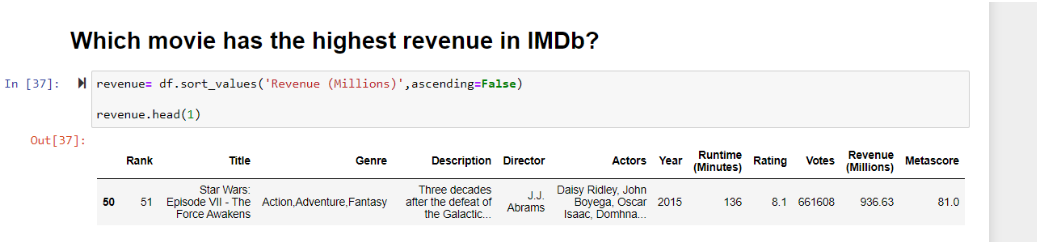


What is the longest movie(minutes) on IMDb

```
In [36]:  runtime= df['Runtime (Minutes)']
          print(runtime.max())

          191
```

I used **max ()** to see this value

⇨ Checked that the duration of the longest movie was 191 minutes



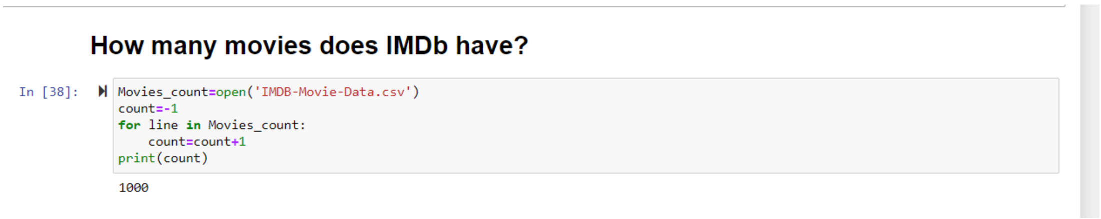
Which movie has the highest revenue in IMDb?

```
In [37]:  revenue= df.sort_values('Revenue (Millions)',ascending=False)

          revenue.head(1)
```

Out[37]:

	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
50	51	Star Wars: Episode VII - The Force Awakens	Action,Adventure,Fantasy	Three decades after the defeat of the Galactic...	J.J. Abrams	Daisy Ridley, John Boyega, Oscar Isaac, Domhna...	2015	136	8.1	661608	936.63	81.0

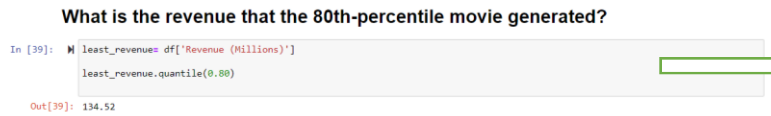
⇨ 'Star Wars: Episode VII - The Force Awakens' movie, Action, Adventure, Fantasy, has earned 936.63 million revenues.



How many movies does IMDb have?

```
In [38]:  Movies_count=open('IMDB-Movie-Data.csv')
          count=-1
          for line in Movies_count:
              count=count+1
          print(count)

          1000
```

⇨ I found the total movie count by using the for loop and increasing it by 1 each time

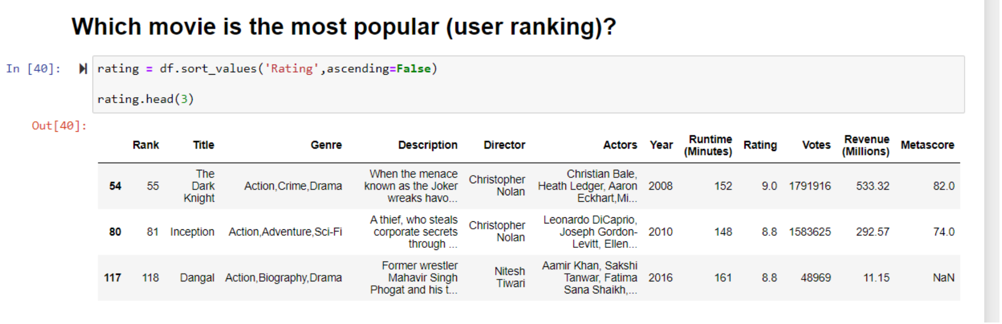


What is the revenue that the 80th-percentile movie generated?

```
In [39]:  least_revenue= df['Revenue (Millions)']

          least_revenue.quantile(0.80)

Out[39]:  134.52
```

I used **quantile()**

⇨ This is how I checked the as 'quantile (0.80)' revenue from the 80th percentile, and revenue=134.52



Which movie is the most popular (user ranking)?

```
In [40]:  rating = df.sort_values('Rating',ascending=False)

          rating.head(3)

Out[40]:
```

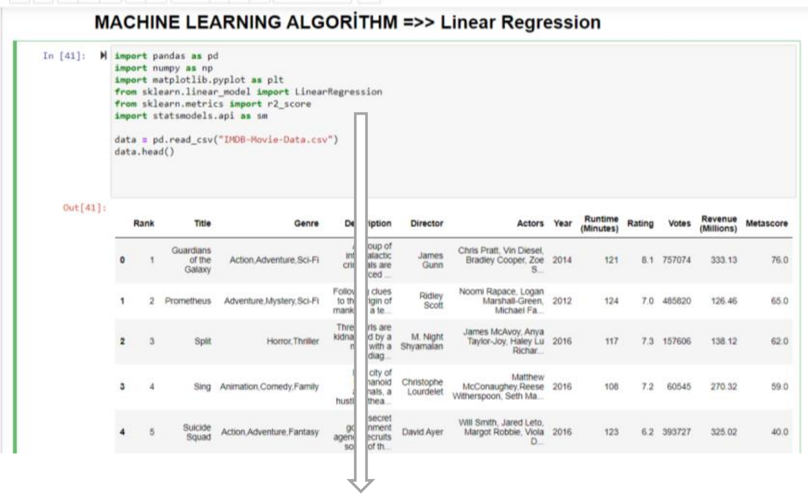
	Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
54	55	The Dark Knight	Action,Crime,Drama	When the menace known as the Joker wreaks havo...	Christopher Nolan	Christian Bale, Heath Ledger, Aaron Eckhart,Mi...	2008	152	9.0	1791916	533.32	82.0
80	81	Inception	Action,Adventure,Sci-Fi	A thief, who steals corporate secrets through ...	Christopher Nolan	Leonardo DiCaprio, Joseph Gordon-Levitt, Ellen...	2010	148	8.8	1583625	292.57	74.0
117	118	Dangal	Action,Biography,Drama	Former wrestler Mahavir Singh Phogat and his t...	Nitesh Tiwari	Aamir Khan, Sakshi Tanwar, Fatima Sana Shaikh,...	2016	161	8.8	48969	11.15	NaN

⇨ The Dark Knight movie with Rating 9 is the most popular movie(user ranking)

MACHINE LEARNING

- **Linear Regression**
- Logistic Regression
- Decision Tree
- SVM
- Naive Bayes
- **kNN**
- K-Means
- Random Forest
- Dimensionality Reduction Algorithms
- Gradient Boosting algorithms
  - GBM
  - XGBoost
  - LightGBM
  - CatBoost

I tried to apply Linear Regression and kNN on my dataset



MACHINE LEARNING ALGORİTHM =>> Linear Regression

In [41]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
import statsmodels.api as sm

data = pd.read_csv("IMDB-Movie-Data.csv")
data.head()
```

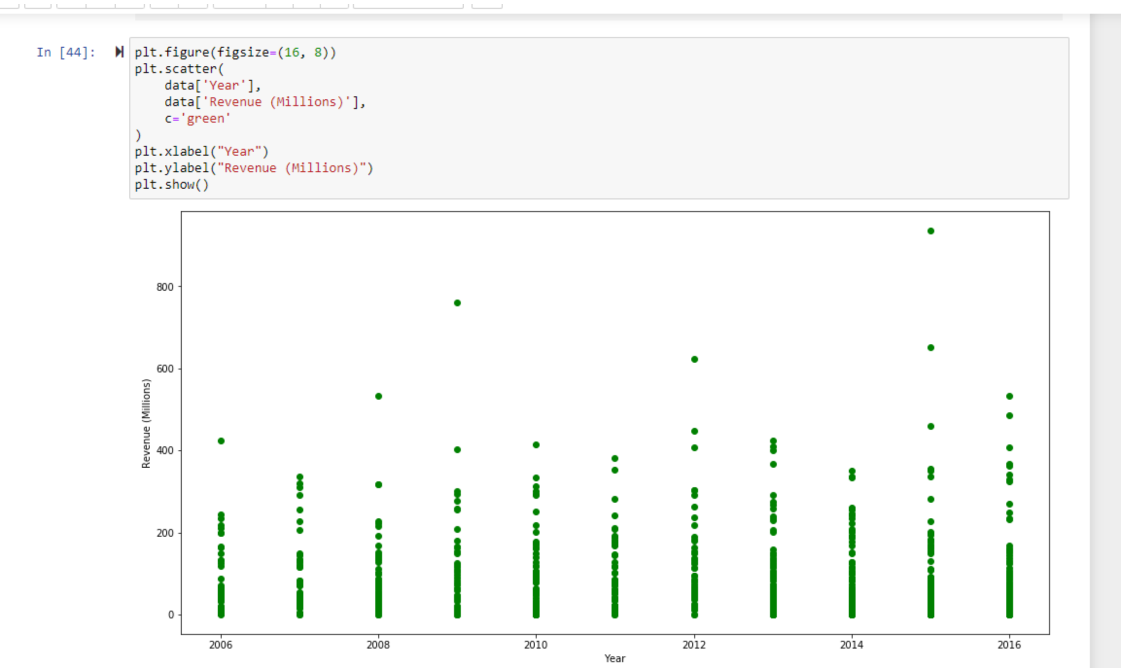
Out[41]:

	Rank	Title	Genre	De iption	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A oup of int alactic cri als are ced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13	76.0
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Follov clues to th igin of mank a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46	65.0
2	3	Split	Horror,Thriller	Thre rls are kidna d by a n with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.12	62.0
3	4	Sing	Animation,Comedy,Family	city of hanoid hals, a hustl thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.32	59.0
4	5	Suicide Squad	Action,Adventure,Fantasy	secret gc nment agen ecruits so of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02	40.0

First of all;

import LinearRegression ➔ from sklearn.linear_model

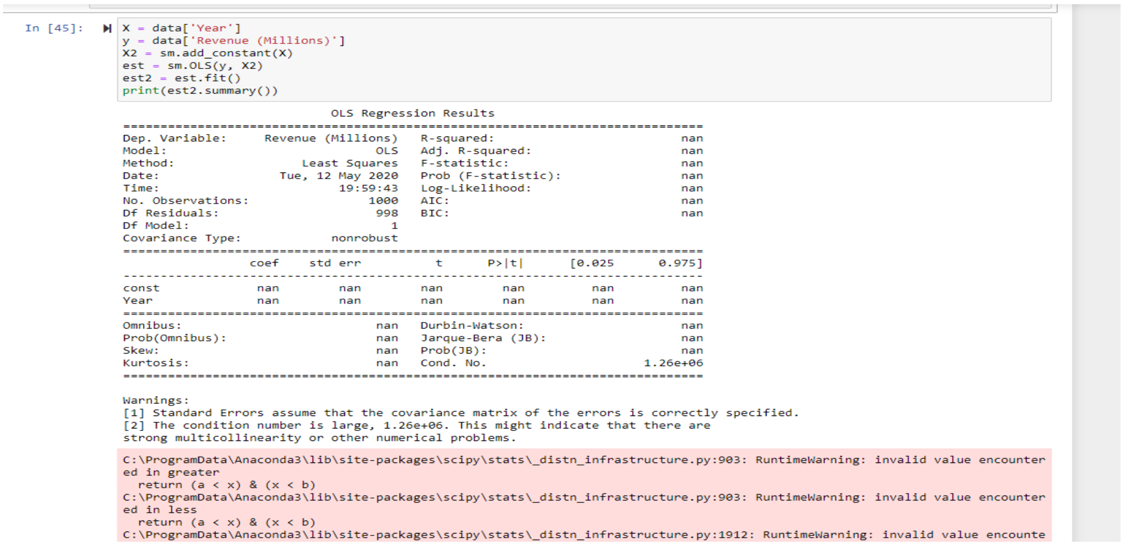
import r2_score ➔ from sklearn.metrics



In [44]:
```
plt.figure(figsize=(16, 8))
plt.scatter(
    data['Year'],
    data['Revenue (Millions)'],
    c='green'
)
plt.xlabel("Year")
plt.ylabel("Revenue (Millions)")
plt.show()
```

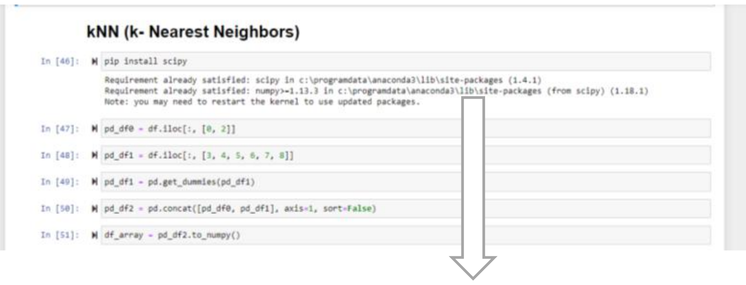
⇨ Run this cell of code and you should see this graph:

As you can see, there is a clear relationship between the Year and Revenue (Millions).



```
In [45]:  X = data['Year']
          y = data['Revenue (Millions)']
          X2 = sm.add_constant(X)
          est = sm.OLS(y, X2)
          est2 = est.fit()
          print(est2.summary())

                            OLS Regression Results
==============================================================================
Dep. Variable:     Revenue (Millions)   R-squared:                       nan
Model:                            OLS   Adj. R-squared:                  nan
Method:                 Least Squares   F-statistic:                     nan
Date:                Tue, 12 May 2020   Prob (F-statistic):              nan
Time:                        19:59:43   Log-Likelihood:                  nan
No. Observations:                1000   AIC:                             nan
Df Residuals:                     998   BIC:                             nan
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const             nan        nan        nan        nan         nan         nan
Year              nan        nan        nan        nan         nan         nan
==============================================================================
Omnibus:                          nan   Durbin-Watson:                   nan
Prob(Omnibus):                    nan   Jarque-Bera (JB):                nan
Skew:                             nan   Prob(JB):                        nan
Kurtosis:                         nan   Cond. No.                   1.26e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.26e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\_distn_infrastructure.py:903: RuntimeWarning: invalid value encounter
ed in greater
  return (a < x) & (x < b)
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\_distn_infrastructure.py:903: RuntimeWarning: invalid value encounter
ed in less
  return (a < x) & (x < b)
C:\ProgramData\Anaconda3\lib\site-packages\scipy\stats\_distn_infrastructure.py:1912: RuntimeWarning: invalid value encounte
```

⇨ Looking at both coefficients, we have a p-value that is very low (although it is probably not exactly 0). This means that there is a strong correlation between these coefficients and the target.

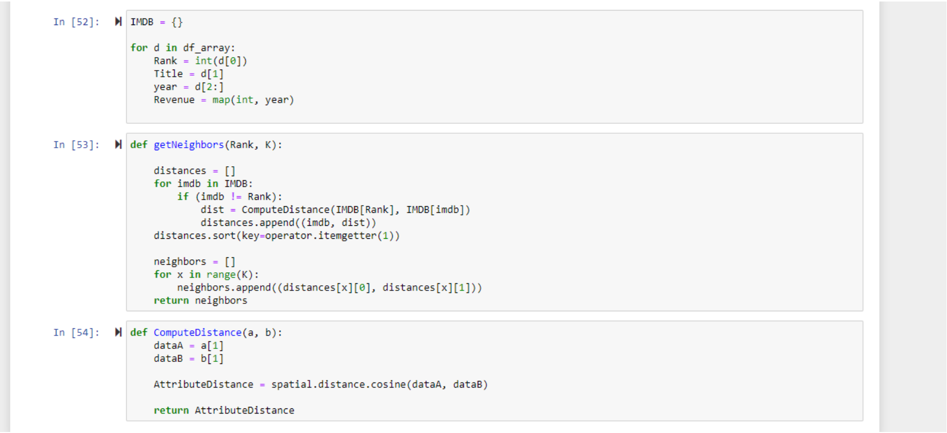


kNN (k- Nearest Neighbors)

```
In [46]:  pip install scipy

          Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\site-packages (1.4.1)
          Requirement already satisfied: numpy>=1.13.3 in c:\programdata\anaconda3\lib\site-packages (from scipy) (1.18.1)
          Note: you may need to restart the kernel to use updated packages.

In [47]:  pd_df0 = df.iloc[:, [0, 2]]

In [48]:  pd_df1 = df.iloc[:, [3, 4, 5, 6, 7, 8]]

In [49]:  pd_df1 = pd.get_dummies(pd_df1)

In [50]:  pd_df2 = pd.concat([pd_df0, pd_df1], axis=1, sort=False)

In [51]:  df_array = pd_df2.to_numpy()
```

SciPy is a free and open-source Python library used for scientific computing and technical computing.

SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.



```
In [52]:  IMDB = {}

          for d in df_array:
              Rank = int(d[0])
              Title = d[1]
              year = d[2:]
              Revenue = map(int, year)

In [53]:  def getNeighbors(Rank, K):

              distances = []
              for imdb in IMDB:
                  if (imdb != Rank):
                      dist = ComputeDistance(IMDB[Rank], IMDB[imdb])
                      distances.append((imdb, dist))
              distances.sort(key=operator.itemgetter(1))

              neighbors = []
              for x in range(K):
                  neighbors.append((distances[x][0], distances[x][1]))
              return neighbors

In [54]:  def ComputeDistance(a, b):
              dataA = a[1]
              dataB = b[1]

              AttributeDistance = spatial.distance.cosine(dataA, dataB)

              return AttributeDistance
```