Movie-Dataset

May 27, 2019

Dataset (TMDb Movie Dataset) Prueba de Conocimientos

0.1 Tabla de contenido

In [1]:
```
# importando todas las bibliotecas / archivos necesarios
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
import seaborn as sns

# Esto asegura que todas las visualizaciones de datos estén trazadas en el cuaderno y no en una ventana se
% matplotlib inline
```

In [ ]:
```
<a id='wrangling'></a>
## Data Wrangling

>los datos se cargarán y luego podremos explorar los datos y eliminar las columnas / filas que no son neces
### General Properties
```

In [2]:
```
# Cargando los datos y almacenándolos en 'df'
df = pd.read_csv('tmdb-movies.csv')

# Echando un vistazo a las primeras 5 filas del conjunto de datos con las columnas proporcionadas intactas
df.head()
```

Out[2]:

	id	imdb_id	popularity	budget	revenue	\
0	135397	tt0369610	32.985763	150000000	1513528810	
1	76341	tt1392190	28.419936	150000000	378436354	

```
2  262500  tt2908446   13.112507  110000000   295238201
3  140607  tt2488496   11.173104  200000000  2068178225
4  168259  tt2820852    9.335014  190000000  1506249360

                 original_title  \
0                Jurassic World
1             Mad Max: Fury Road
2                      Insurgent
3  Star Wars: The Force Awakens
4                      Furious 7

                                          cast  \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...
1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...
2  Shailene Woodley|Theo James|Kate Winslet|Ansel...
3  Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...
4  Vin Diesel|Paul Walker|Jason Statham|Michelle ...

                                    homepage           director  \
0                  http://www.jurassicworld.com/    Colin Trevorrow
1                    http://www.madmaxmovie.com/       George Miller
2  http://www.thedivergentseries.movie/#insurgent   Robert Schwentke
3  http://www.starwars.com/films/star-wars-episod...        J.J. Abrams
4                        http://www.furious7.com/          James Wan

                      tagline  ...        \
0              The park is open.  ...
1             What a Lovely Day.  ...
2      One Choice Can Destroy You  ...
3  Every generation has a story.  ...
4            Vengeance Hits Home  ...

                                       overview runtime  \
0  Twenty-two years after the events of Jurassic ...     124
1  An apocalyptic story set in the furthest reach...     120
2  Beatrice Prior must confront her inner demons ...     119
3  Thirty years after defeating the Galactic Empi...     136
4  Deckard Shaw seeks revenge against Dominic Tor...     137

                                    genres  \
0  Action|Adventure|Science Fiction|Thriller
1  Action|Adventure|Science Fiction|Thriller
2         Adventure|Science Fiction|Thriller
3  Action|Adventure|Science Fiction|Fantasy
4                  Action|Crime|Thriller

                                production_companies release_date vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda...       6/9/15       5562
```

```
1  Village Roadshow Pictures|Kennedy Miller Produ...      5/13/15       6185
2  Summit Entertainment|Mandeville Films|Red Wago...      3/18/15       2480
3        Lucasfilm|Truenorth Productions|Bad Robot     12/15/15       5292
4  Universal Pictures|Original Film|Media Rights ...      4/1/15       2947

   vote_average  release_year    budget_adj   revenue_adj
0           6.5          2015  1.379999e+08  1.392446e+09
1           7.1          2015  1.379999e+08  3.481613e+08
2           6.3          2015  1.012000e+08  2.716190e+08
3           7.5          2015  1.839999e+08  1.902723e+09
4           7.3          2015  1.747999e+08  1.385749e+09

[5 rows x 21 columns]
```

In [3]:
```
# Inspeccionar los tipos de datos y buscar instancias de datos faltantes o posiblemente errantes
df.info()
```

```
<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
```

In [4]:
```
rows, col = df.query('budget == 0').shape
print('There are {} rows and {} columns'.format(rows, col))
```

```
There are 5696 rows and 21 columns
```

In [5]:
```
rows, col = df.query('revenue == 0').shape
print('There are {} rows and {} columns'.format(rows, col))
```

There are 6016 rows and 21 columns

In [6]:
```
rows, col = df.query('runtime == 0').shape
print('There are {} rows and {} columns'.format(rows, col))
```

There are 31 rows and 21 columns

Limpieza de datos

En primer lugar, eliminaremos las columnas que no son necesarias para nuestro análisis. Son id,imdb_id, budget_adj,revenue_adj, homepage,tagline, keywords yoverview.

En segundo lugar, todas las filas duplicadas presentes, si las hay, serán eliminadas.

Luego, cambiaremos el tipo de datos de la columna release_date de cadena a fecha y hora.

Actualmente, hay 5696 filas donde la columna presupuesto tiene un valor de 0, 6016 filas donde la columnaingreso tiene un valor de 0 y 31 filas donde la columna tiempo de ejecución tiene un valor de 0. Todos estos 0 Los valores se convertirán a NaN.

1. Eliminar las columnas que no son necesarias

In [7]:
```
# Lista de columnas que se van a eliminar / eliminar
col = ['id', 'imdb_id', 'budget_adj', 'revenue_adj', 'homepage',  'tagline', 'keywords', 'overview']

# Eliminando las columnas
df.drop(col, axis = 1, inplace = True)

#comprobando si las columnas han sido eliminadas
df.head()
```

Out[7]:

	popularity	budget	revenue	original_title	cast	director
0	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow
1	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller
2	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	Robert Schwentke
3	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	J.J. Abrams
4	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	James Wan

```
   runtime                                genres  \
0      124  Action|Adventure|Science Fiction|Thriller
1      120  Action|Adventure|Science Fiction|Thriller
2      119         Adventure|Science Fiction|Thriller
3      136   Action|Adventure|Science Fiction|Fantasy
4      137                   Action|Crime|Thriller

                        production_companies release_date  vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda...       6/9/15        5562
1  Village Roadshow Pictures|Kennedy Miller Produ...      5/13/15        6185
2  Summit Entertainment|Mandeville Films|Red Wago...      3/18/15        2480
3        Lucasfilm|Truenorth Productions|Bad Robot     12/15/15        5292
4  Universal Pictures|Original Film|Media Rights ...       4/1/15        2947

   vote_average  release_year
0           6.5          2015
1           7.1          2015
2           6.3          2015
3           7.5          2015
4           7.3          2015
```

2. liminar filas duplicadas (si las hay)

In [8]:
```
# eliminando filas duplicadas
# la primera entrada se mantiene por defecto
df.drop_duplicates(inplace=True)

rows, col = df.shape
print('There are now {} columns and {} entries of movie data'.format(col, rows-1))
```

There are now 13 columns and 10864 entries of movie data

3. Cambiar el tipo de datos de la columna release_date

In [9]:
```
df['release_date'] = pd.to_datetime(df['release_date'])
```

In [10]:
```
# Compruebe si el cambio ha tenido lugar con éxito
df.dtypes
```

Out[10]:
```
popularity                float64
budget                      int64
revenue                     int64
original_title             object
cast                       object
director                   object
runtime                     int64
genres                     object
production_companies       object
```

```
release_date          datetime64[ns]
vote_count                 int64
vote_average             float64
release_year               int64
dtype: object
```

4. **Manejo de 0 valores en las columnas** presupuesto,ingresos y tiempo de ejecución

```
In [11]: # Hacer una lista de las 3 columnas.
         temp_col = ['budget', 'revenue', 'runtime']

         # Reemplazando todos los valores 0 con NaN
         df[temp_col] = df[temp_col].replace(0, np.NAN)
```

```
In [12]: # Eliminar / Borrar todos los valores de NaN
         # El subconjunto ayuda a definir en qué columnas buscar valores perdidos
         df.dropna(subset = temp_col, inplace = True)
         rows, col = df.shape

         print('Now there are only {} entries'.format(rows-1))
```

```
Now there are only 3853 entries
```

Ese es el final de nuestro proceso de limpieza de datos. Echemos un vistazo al conjunto de datos ahora.

```
In [13]: df.head()
```

```
Out[13]:
   popularity       budget       revenue              original_title  \
0   32.985763  150000000.0  1.513529e+09                Jurassic World
1   28.419936  150000000.0  3.784364e+08           Mad Max: Fury Road
2   13.112507  110000000.0  2.952382e+08                    Insurgent
3   11.173104  200000000.0  2.068178e+09  Star Wars: The Force Awakens
4    9.335014  190000000.0  1.506249e+09                    Furious 7

                                                cast         director  \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...   Colin Trevorrow
1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...     George Miller
2  Shailene Woodley|Theo James|Kate Winslet|Ansel...  Robert Schwentke
3  Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...      J.J. Abrams
4  Vin Diesel|Paul Walker|Jason Statham|Michelle ...       James Wan

   runtime                                     genres  \
0    124.0  Action|Adventure|Science Fiction|Thriller
1    120.0  Action|Adventure|Science Fiction|Thriller
2    119.0         Adventure|Science Fiction|Thriller
3    136.0   Action|Adventure|Science Fiction|Fantasy
4    137.0                    Action|Crime|Thriller
```

```
                                production_companies release_date  vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda...    2015-06-09        5562
1  Village Roadshow Pictures|Kennedy Miller Produ...    2015-05-13        6185
2  Summit Entertainment|Mandeville Films|Red Wago...    2015-03-18        2480
3           Lucasfilm|Truenorth Productions|Bad Robot    2015-12-15        5292
4  Universal Pictures|Original Film|Media Rights ...    2015-04-01        2947

   vote_average  release_year
0           6.5          2015
1           7.1          2015
2           6.3          2015
3           7.5          2015
4           7.3          2015
```

Análisis exploratorio de datos ąAhora que todos los datos se han limpiado a nuestro gusto, podemos seguir adelante, analizar el conjunto de datos y hacer algunos descubrimientos!

0.1.1 I.) 1. £Cuáles son las popularidades promedio de las películas según los niveles del presupuesto?

```
In [14]: # Primero tenemos que hacer columnas para rangos de presupuesto
         # Utilizamos los métodos de corte de la biblioteca de pandas para hacerlo.
         df['budget_ranges'] = pd.cut(df['budget'], df['budget'].describe()[3:8], labels = ['Low', 'Medium', 'Moderat

In [15]: # Echando un vistazo a nuestro conjunto de datos después de agregar una nueva columna
         df.head(2)

Out[15]:    popularity       budget       revenue      original_title  \
         0   32.985763  150000000.0  1.513529e+09      Jurassic World
         1   28.419936  150000000.0  3.784364e+08  Mad Max: Fury Road

                                               cast         director  \
         0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...   Colin Trevorrow
         1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...     George Miller

            runtime                                genres  \
         0    124.0  Action|Adventure|Science Fiction|Thriller
         1    120.0  Action|Adventure|Science Fiction|Thriller

                                production_companies release_date  vote_count  \
         0  Universal Studios|Amblin Entertainment|Legenda...    2015-06-09        5562
         1  Village Roadshow Pictures|Kennedy Miller Produ...    2015-05-13        6185

            vote_average  release_year budget_ranges
         0           6.5          2015          High
         1           7.1          2015          High
```

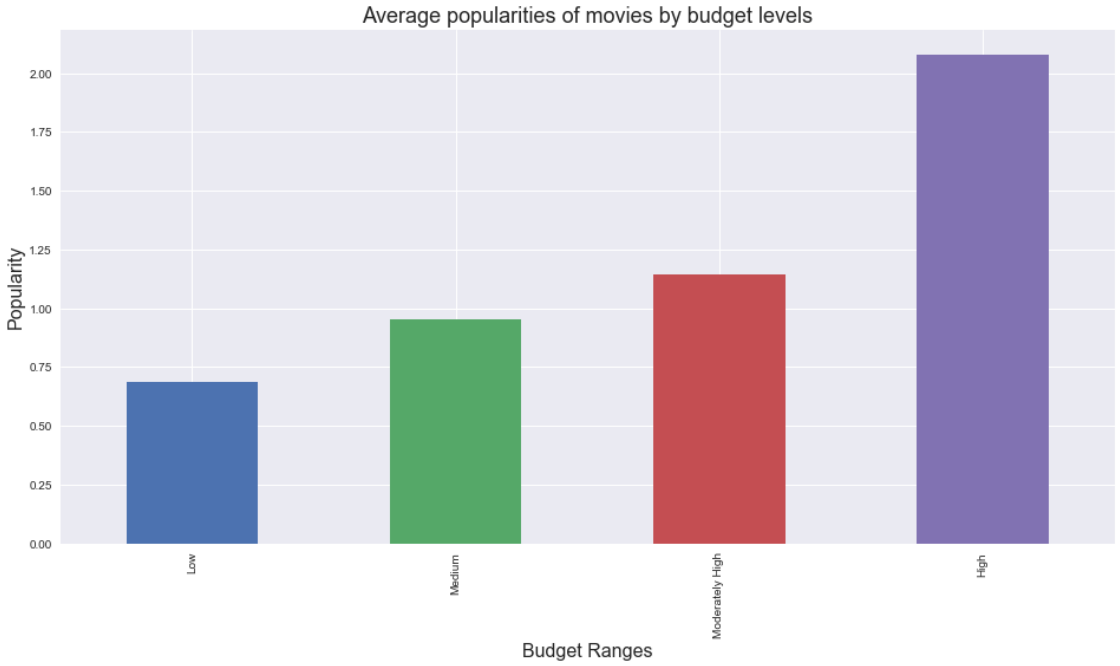
In [16]: # Ahora nos enteramos de las popularidades promedio de cada nivel.
df.groupby('budget_ranges')['popularity'].mean()

Out[16]: budget_ranges
Low 0.686413
Medium 0.951718
Moderately High 1.142414
High 2.080911
Name: popularity, dtype: float64

In [17]: # Trazando la información anterior en un gráfico de barras
sns.set()
df.groupby('budget_ranges')['popularity'].mean().plot(kind = 'bar', figsize = (16, 8))

# Configuración del título de la trama.
plt.title('Average popularities of movies by budget levels', fontsize = 18)

# Configurando las etiquetas de los ejes xy y
plt.xlabel('Budget Ranges', fontsize = 16)
plt.ylabel('Popularity', fontsize = 16);



Se puede observar que las películas con un rango de presupuesto más alto tienden a ser más populares entre la audiencia.

0.1.2 I.) 2. £Cuáles son las tendencias de ganancias de un año a otro?

In [18]: # Primero necesitamos insertar una columna para el valor de ganancia / pérdida de cada película
df.insert(3, 'profit_loss', df['revenue'] - df['budget'])

In [19]:
```
# Comprobar para ver si la columna fue insertada
df.head(2)
```

Out[19]:

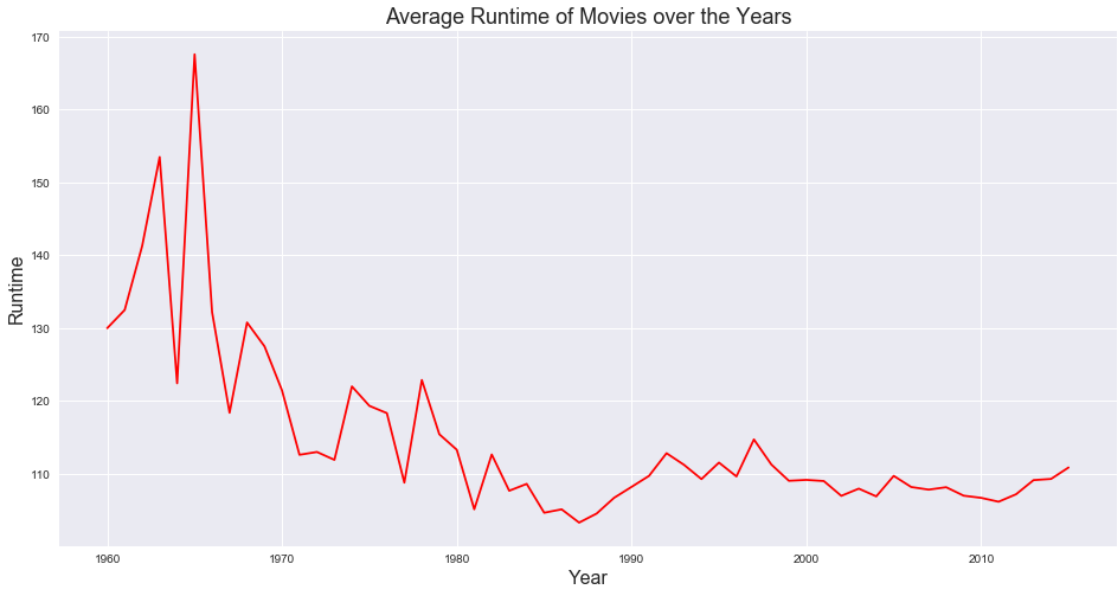
	popularity	budget	revenue	profit_loss	original_title	cast	director	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year	budget_ranges
0	32.985763	150000000.0	1.513529e+09	1.363529e+09	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124.0	Action|Adventure|Science Fiction|Thriller	Universal Studios|Amblin Entertainment|Legenda...	2015-06-09	5562	6.5	2015	High
1	28.419936	150000000.0	3.784364e+08	2.284364e+08	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	120.0	Action|Adventure|Science Fiction|Thriller	Village Roadshow Pictures|Kennedy Miller Produ...	2015-05-13	6185	7.1	2015	High

In [20]:
```
df.groupby('release_year')['profit_loss'].sum().plot(kind = 'line', figsize = (16, 8), color = 'g')
plt.title('Profit Trends from year to year', fontsize = 18)
plt.xlabel('Year', fontsize = 16)
plt.ylabel('Profit', fontsize = 16);
```

Las ganancias han aumentado exponencialmente con cada año que pasa, especialmente después del comienzo del siglo XXI.

0.1.3 I.) 3. £Cuáles son los tiempos de ejecución promedio de las películas a lo largo de los años?

```
In [21]: df.groupby('release_year')['runtime'].mean().plot(kind='line', figsize = (16, 8), color = 'r')
         plt.title('Average Runtime of Movies over the Years', fontsize = 18)
         plt.xlabel('Year', fontsize = 16)
         plt.ylabel('Runtime', fontsize = 16);
```



El tiempo de ejecución de las películas ha disminuido con cada año que pasa. Experimentó una subida durante los años 60, pero luego ha disminuido constantemente a lo largo de los años. El más bajo fue de alrededor de 100-110 minutos. Actualmente, las películas tienden a durar alrededor de los 110 minutos.

0.1.4 I.) 4. £Cuáles son las 5 películas rentables más baratas y caras de todos los tiempos?

Para esto estableceremos un valor estándar de ganancia que se debe cumplir, ese valor será de $ 50,000,000

```
In [22]: # creando una lista de columnas que serán vistas
         col = ['original_title', 'cast', 'director', 'budget', 'revenue', 'profit_loss']

         # Uso de la función de consulta para mostrar registros de películas que tienen un beneficio de más de $ 50
         # También utiliza la función sort_values para asegurarse de que esté ordenada de acuerdo con la columna

         df.query('profit_loss>50000000')[col].sort_values('budget', ascending = False).head(5)
```

```
Out[22]:                                original_title  \
         3375  Pirates of the Caribbean: On Stranger Tides
```

```
7387     Pirates of the Caribbean: At World's End
14                     Avengers: Age of Ultron
6570                        Superman Returns
1929                                Tangled

                                                 cast  \
3375  Johnny Depp|PenÃ¯lope Cruz|Geoffrey Rush|Ian M...
7387  Johnny Depp|Orlando Bloom|Keira Knightley|Geof...
14    Robert Downey Jr.|Chris Hemsworth|Mark Ruffalo...
6570  Brandon Routh|Kevin Spacey|Kate Bosworth|James...
1929  Zachary Levi|Mandy Moore|Donna Murphy|Ron Perl...

                    director       budget       revenue    profit_loss
3375             Rob Marshall  380000000.0  1.021683e+09   6.416830e+08
7387            Gore Verbinski  300000000.0  9.610000e+08   6.610000e+08
14                Joss Whedon  280000000.0  1.405036e+09   1.125036e+09
6570              Bryan Singer  270000000.0  3.910812e+08   1.210812e+08
1929  Nathan Greno|Byron Howard  260000000.0  5.917949e+08   3.317949e+08
```

In [23]: df.query('profit_loss>50000000')[col].sort_values('budget', ascending = True).head(5)

Out[23]:

```
                original_title  \
10495  The Karate Kid, Part II
7447       Paranormal Activity
2449   The Blair Witch Project
7057                Open Water
10759               Halloween

                                                 cast  \
10495  Ralph Macchio|Pat Morita|Martin Kove|Charlie T...
7447   Katie Featherston|Micah Sloat|Mark Fredrichs|A...
2449   Heather Donahue|Michael C. Williams|Joshua Leo...
7057   Blanchard Ryan|Daniel Travis|Saul Stein|Michae...
10759  Donald Pleasence|Jamie Lee Curtis|P.J. Soles|N...

                      director     budget       revenue   profit_loss
10495              John G. Avildsen      113.0  115103979.0  115103866.0
7447                   Oren Peli    15000.0  193355800.0  193340800.0
2449   Daniel Myrick|Eduardo SÃąnchez    25000.0  248000000.0  247975000.0
7057                  Chris Kentis   130000.0   54667954.0   54537954.0
10759                John Carpenter   300000.0   70000000.0   69700000.0
```

0.1.5 I.) 5. £Qué variables afectan los ingresos y la popularidad de una película?

In [24]:
```
# Use corr para calcular la correlación de columnas
df.corr()
```

Out[24]:

	popularity	budget	revenue	profit_loss	runtime	vote_count	vote_average	release_year
popularity	1.000000	0.446987	0.615535	0.596201	0.215092	0.780096	0.317866	0.173278
budget	0.446987	1.000000	0.688556	0.526818	0.260977	0.556937	0.024169	0.268040
revenue	0.615535	0.688556	1.000000	0.979133	0.250298	0.754567	0.227123	0.139140
profit_loss	0.596201	0.526818	0.979133	1.000000	0.220238	0.728348	0.259435	0.087971
runtime	0.215092	0.260977	0.250298	0.220238	1.000000	0.273771	0.351712	-0.112453
vote_count	0.780096	0.556937	0.754567	0.728348	0.273771	1.000000	0.387210	0.207191
vote_average	0.317866	0.024169	0.227123	0.259435	0.351712	0.387210	1.000000	-0.134246
release_year	0.173278	0.268040	0.139140	0.087971	-0.112453	0.207191	-0.134246	1.000000

En caso de ingresos.

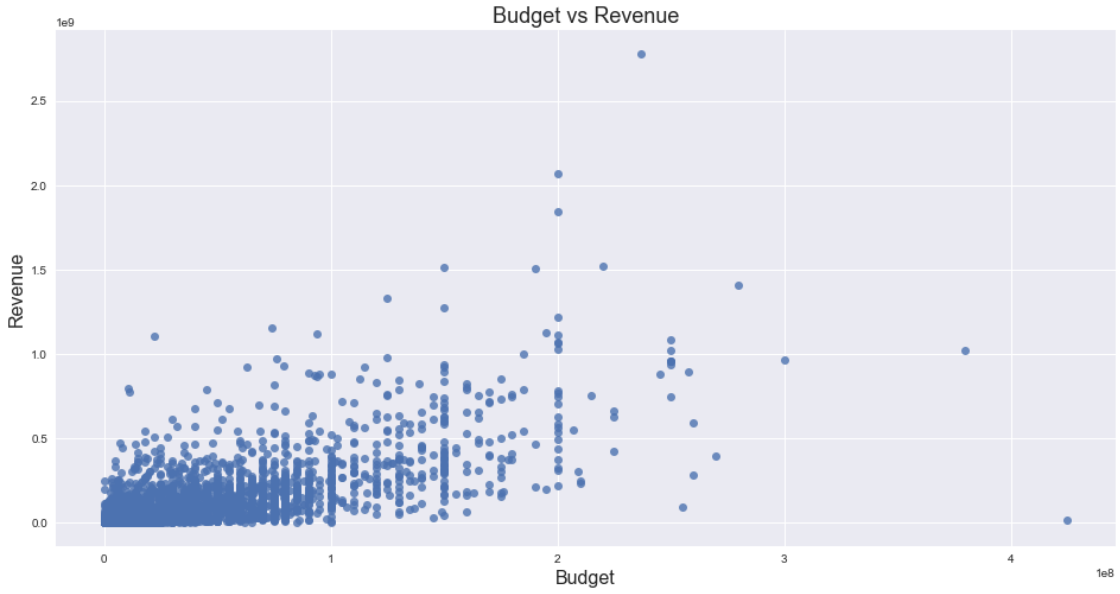
Fuerte correlación con la popularidad, el presupuesto y el conteo de votos.
Correlación débil con el tiempo de ejecución.

En caso de popularidad,

Correlación moderada con presupuesto.
Fuerte correlación con los ingresos, ganancias y pérdidas y conteo de votos.

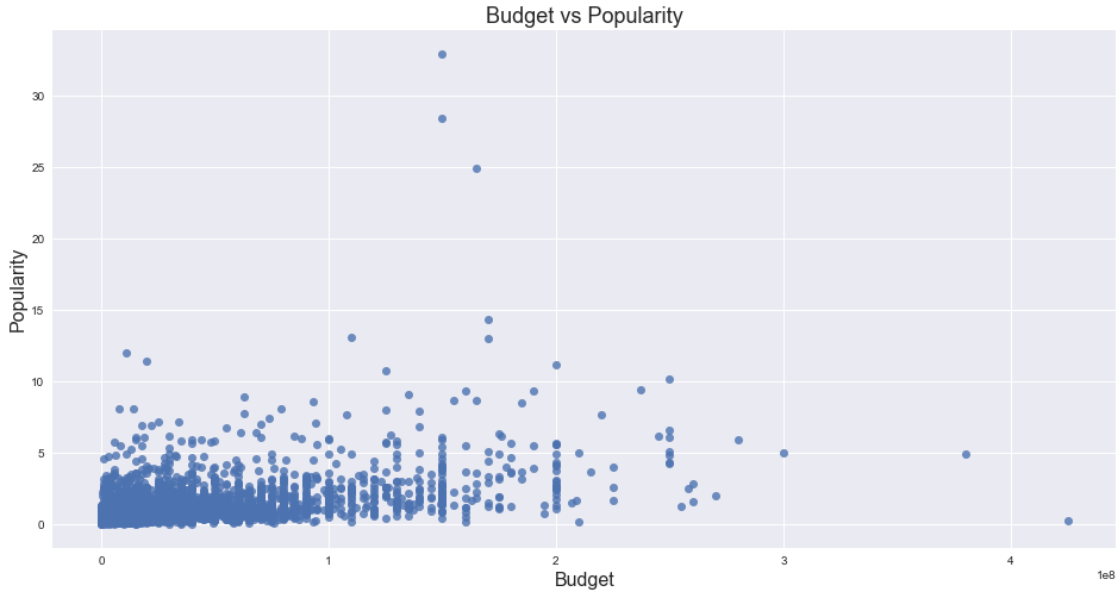
Ahora veremos uno de cada nivel de correlación.
Correlación Fuerte: Presupuesto vs Ingresos

In [38]:
```
# Trazar diagramas de dispersión para ver la correlación visualmente
sns.regplot(x = df['budget'], y = df['revenue'], fit_reg = False)
# Obtención del tamaño de la muestra.
fig_size = plt.rcParams["figure.figsize"]

# Cambiando el largo y ancho del grafico.
fig_size[0] = 16
fig_size[1] = 8
plt.rcParams["figure.figsize"] = fig_size

plt.title('Budget vs Revenue', fontsize = 18)
plt.xlabel('Budget', fontsize = 16)
plt.ylabel('Revenue', fontsize = 16);
```



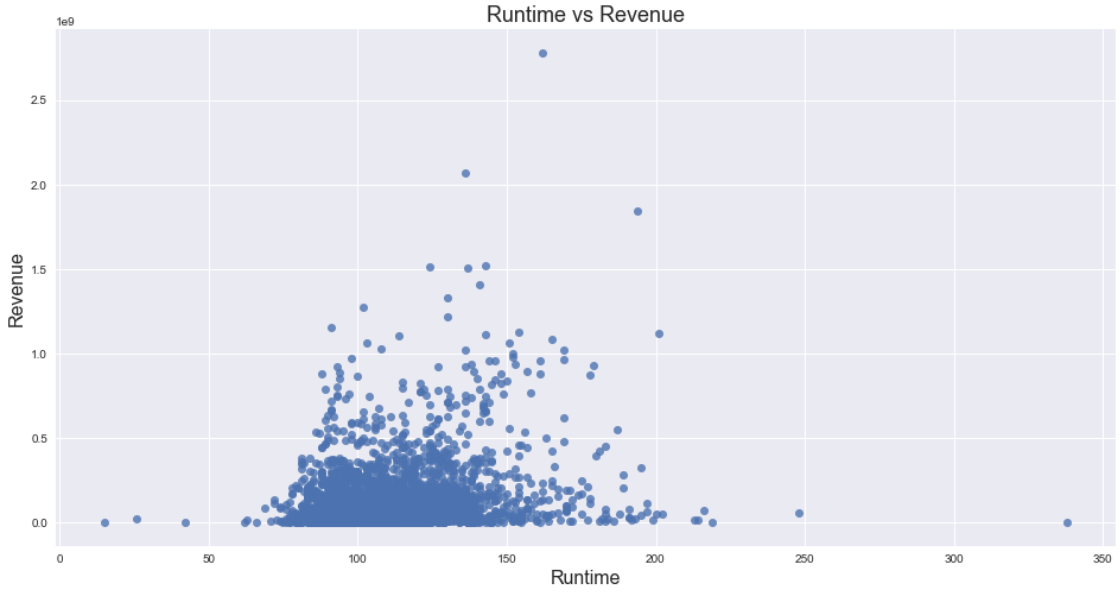
Correlación Moderada: Presupuesto vs Popularidad

In [26]:
```
sns.regplot(x = df['budget'], y = df['popularity'], fit_reg = False)

fig_size = plt.rcParams["figure.figsize"]
fig_size[0] = 16
fig_size[1] = 8

plt.rcParams["figure.figsize"] = fig_size
plt.title('Budget vs Popularity', fontsize = 18)
plt.xlabel('Budget', fontsize = 16)
plt.ylabel('Popularity', fontsize = 16);
```



Correlación débil: Runtime vs Ingresos

In [27]:
```
sns.regplot(x = df['runtime'], y = df['revenue'], fit_reg = False)

fig_size = plt.rcParams["figure.figsize"]
fig_size[0] = 16
fig_size[1] = 8

plt.rcParams["figure.figsize"] = fig_size
plt.title('Runtime vs Revenue', fontsize = 18)
plt.xlabel('Runtime', fontsize = 16)
plt.ylabel('Revenue', fontsize = 16);
```



0.1.6 II.) 1. Genero

In [28]:
```
# Primero, crearemos un marco de datos que contiene datos de todas las películas que han obtenido al m
# Crea una lista de columnas que son requeridas
profit_col = ['original_title', 'cast', 'director', 'production_companies', 'genres', 'budget', 'revenue', 'runti
profit_df = df.query('profit_loss>50000000')[profit_col]

# Ver el marco de datos recién creado
profit_df.head(2)
```

Out[28]:

	original_title	cast	\
0	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	

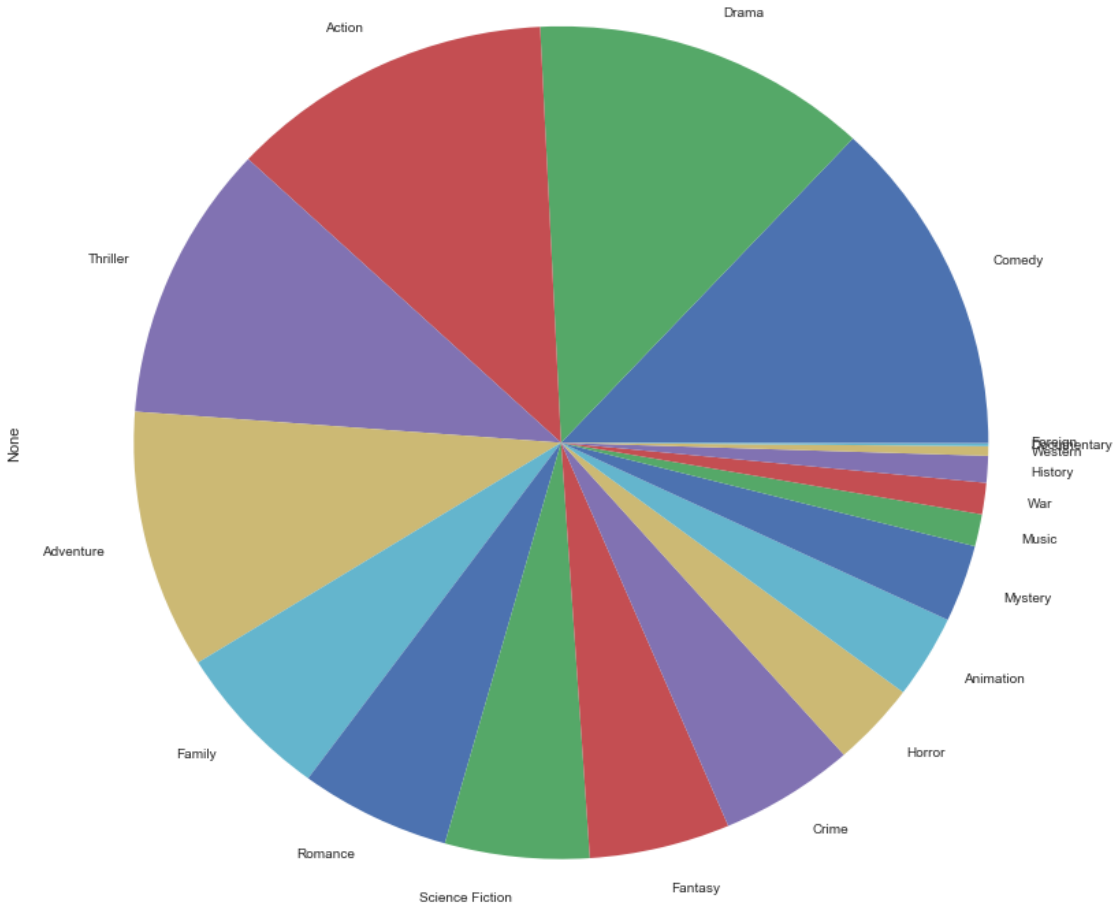
```
1  Mad Max: Fury Road  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...

          director                                production_companies  \
0  Colin Trevorrow  Universal Studios|Amblin Entertainment|Legenda...
1    George Miller  Village Roadshow Pictures|Kennedy Miller Produ...

                                  genres       budget       revenue  \
0  Action|Adventure|Science Fiction|Thriller  150000000.0  1.513529e+09
1  Action|Adventure|Science Fiction|Thriller  150000000.0  3.784364e+08

   runtime
0    124.0
1    120.0
```

In [29]:
```
#funcion que tomará cualquier columna como argumento de y mantendrá su pista
def calculate_count(column):
    # Convierta la columna en cadena y sepárela por '|'
    data = profit_df[column].str.cat(sep = '|')

    # Almacenando los valores por separado en una serie de Pandas
    data = pd.Series(data.split('|'))
    count = data.value_counts(ascending = False)

    return count
```

In [30]:
```
#variable to store the retured value#variabl
count = calculate_count('genres')

#printing top 5 values
count.head()
```

Out[30]:
```
Comedy       492
Drama        481
Action       464
Thriller     405
Adventure    379
dtype: int64
```

In [31]:
```
count.plot(kind='pie', figsize = (14, 14));
```



0.1.7 II.) 2. Actor/Actrices

In [32]:
```
#variable para almacenar el valor devuelto
count = calculate_count('cast')

# imprimiendo los 5 mejores valores
count.head()
```

Out[32]:
```
Tom Cruise          27
Brad Pitt           25
Tom Hanks           22
Sylvester Stallone  21
Cameron Diaz        20
dtype: int64
```

0.1.8 II.) 3. Director

In [33]:
```
#variable para almacenar el valor devuelto
count = calculate_count('director')

# imprimiendo los 5 mejores valores
count.head()
```

Out[33]:
```
Steven Spielberg    23
Robert Zemeckis     13
Clint Eastwood      12
Tim Burton          11
Tony Scott          10
dtype: int64
```

0.1.9 II.) 4. Empresas de produccion

In [34]:
```
#variable para almacenar el valor devuelto
count = calculate_count('production_companies')

# imprimiendo los 5 mejores valores
count.head()
```

Out[34]:
```
Universal Pictures                        156
Warner Bros.                              144
Paramount Pictures                        130
Twentieth Century Fox Film Corporation    118
Columbia Pictures                          93
dtype: int64
```

0.1.10 II.) 5. Presupuesto

In [35]:
```
# Recuperando el presupuesto promedio
profit_avg_budget = profit_df['budget'].mean()
print('The average budget of a succesful movie is ${0:.2f}'.format(profit_avg_budget))
```

The average budget of a succesful movie is $60444957.76

0.1.11 II.) 6. Tiempo de ejecución

In [36]:
```
# Recuperando el tiempo de ejecución promedio
profit_avg_runtime = profit_df['runtime'].mean()
print('The average runtime of a succesful movie is {0:.1f}'.format(profit_avg_runtime))
```

The average runtime of a succesful movie is 113.7

0.1.12 Ingresos que pueden ser explicados

In [37]:
```
# Recuperando el ingreso promedio
profit_avg_runtime = profit_df['revenue'].mean()
print('The average revenue of a succesful movie is {0:.1f}'.format(profit_avg_runtime))
```

The average revenue of a succesful movie is 254957662.6