pixar_films

March 13, 2025

1 Análisis de Datos Exploratorio - DataFrame sobre Películas Pixar

Mi proyecto de análisis exploratorio de datos de películas Pixar comienza importando las librerías esenciales: numpy, pandas y matplotlib. Leí el archivo CSV con los datos de las películas de Pixar y exploré su estructura mediante las funciones head() y tail(). Verifiqué la información del dataframe con info() y detecté valores nulos, los cuales reemplacé en la columna cinema_score con "NC" (No Calificado). Mejoré los nombres de las columnas eliminando espacios y caracteres especiales. Para un análisis más enfocado, eliminé las columnas 'box_office_us_canada' y 'box_office_other', quedándome solo con 'box_office_worldwide'. Agregué una columna 'Revenue' calculando la diferencia entre los ingresos mundiales y el presupuesto. Corregí un error en el presupuesto de 'Luca' que aparecía como 0, reemplazándolo por 175 millones. Convertí la columna 'release_date' a formato fecha y extraje el año de lanzamiento. Guardé los datos limpios en un archivo Excel 'data_clean.xlsx'. Finalmente, elaboré un gráfico de barras ordenado cronológicamente que muestra la evolución de los presupuestos de las películas Pixar a lo largo del tiempo.

https://www.kaggle.com/datasets/willianoliveiragibin/pixar-films

[2]:
```
#Importamos las librerías numpy, pandas y matplotlib
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[3]:
```
#Leemos el archivo csv
df = pd.read_csv('./pixar_films new.csv')
```

[4]:
```
#Mostramos las primeras filas del data frame
df.head()
```

[4]:

	ID	film	film_rating	cinema_score	release_date	run_time	budget	box_office_us_canada	box_office_other	box_office_worldwide
0	1	Toy Story	G	A	1995-11-22	81	30000000	223225679	171210907	394436586
1	2	A Bug's Life	G	A	1998-11-25	95				
2	3	Toy Story 2	G	A+	1999-11-24	92				
3	4	Monsters, Inc.	G	A+	2001-11-02	92				
4	5	Finding Nemo	G	A+	2003-05-30	100				

```
1  120000000             162798565         200460294             363258859
2   90000000             245852179         265506097             511358276
3  115000000             255873250         272900000             528773250
4   94000000             339714978         531300000             871014978

   rotten_tomatoes_score  rotten_tomatoes_counts  metacritic_score  \
0                    100                      96                95
1                     92                      91                78
2                    100                     172                88
3                     96                     199                79
4                     99                     270                90

   metacritic_counts  imdb_score  imdb_counts
0                 26         8.3      1089101
1                 23         7.2       319596
2                 34         7.9       630573
3                 35         8.1      1000657
4                 38         8.2      1132877
```

[5]:
```
#Mostramos las últimas filas del data frame
df.tail()
```

[5]:
```
    ID          film film_rating cinema_score release_date  run_time  \
23  24          Luca          PG          NaN   2021-06-18        95
24  25   Turning Red          PG          NaN   2022-03-11       100
25  26     Lightyear          PG           A-   2022-06-17       105
26  27     Elemental          PG            A   2023-06-16       101
27  28  Inside Out 2          PG            A   2024-06-14        96

       budget  box_office_us_canada  box_office_other  box_office_worldwide  \
23          0               1324302          49788012              51112314
24  175000000               1399001          20414357              21813358
25  200000000             118307188         108118232             226425420
26  200000000             154426697         342017611             496444308
27  200000000             652980194        1045050771            1698030965

    rotten_tomatoes_score  rotten_tomatoes_counts  metacritic_score  \
23                     91                     303                71
24                     95                     289                83
25                     74                     319                60
26                     73                     262                58
27                     90                     313                73

    metacritic_counts  imdb_score  imdb_counts
23                 52         7.4       202404
24                 52         7.0       158649
25                 57         6.1       127045
```

```
26                  45          7.0        140174
27                  59          7.6        168090
```

[6]:
```
#Visualizamos la información del data frame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28 entries, 0 to 27
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ID                      28 non-null     int64
 1   film                    28 non-null     object
 2   film_rating             28 non-null     object
 3   cinema_score            25 non-null     object
 4   release_date            28 non-null     object
 5   run_time                28 non-null     int64
 6   budget                  28 non-null     int64
 7   box_office_us_canada    28 non-null     int64
 8   box_office_other        28 non-null     int64
 9   box_office_worldwide    28 non-null     int64
 10  rotten_tomatoes_score   28 non-null     int64
 11  rotten_tomatoes_counts  28 non-null     int64
 12  metacritic_score        28 non-null     int64
 13  metacritic_counts       28 non-null     int64
 14  imdb_score              28 non-null     float64
 15  imdb_counts             28 non-null     int64
dtypes: float64(1), int64(11), object(4)
memory usage: 3.6+ KB
```

[7]:
```
#Verificamos si hay elementos nulos
df.isnull().sum()
```

[7]:
```
ID                      0
film                    0
film_rating             0
cinema_score            3
release_date            0
run_time                0
budget                  0
box_office_us_canada    0
box_office_other        0
box_office_worldwide    0
rotten_tomatoes_score   0
rotten_tomatoes_counts  0
metacritic_score        0
metacritic_counts       0
imdb_score              0
```

```
imdb_counts               0
dtype: int64
```

[8]:
```
#Reemplazamos los elementos nulos de la columna cinema_score por NC, que␣
 ↪significa NO CALIFICADO
df = df.fillna('NC')
#Mostramos los nuevos datos del data frame
df.tail(7)
```

[8]:

	ID	film	film_rating	cinema_score	release_date	run_time
21	22	Onward	PG	A-	2020-03-06	102
22	23	Soul	PG	NC	2020-12-25	100
23	24	Luca	PG	NC	2021-06-18	95
24	25	Turning Red	PG	NC	2022-03-11	100
25	26	Lightyear	PG	A-	2022-06-17	105
26	27	Elemental	PG	A	2023-06-16	101
27	28	Inside Out 2	PG	A	2024-06-14	96

	budget	box_office_us_canada	box_office_other	box_office_worldwide
21	175000000	61555145	80384897	141940042
22	150000000	946154	120957731	121903885
23	0	1324302	49788012	51112314
24	175000000	1399001	20414357	21813358
25	200000000	118307188	108118232	226425420
26	200000000	154426697	342017611	496444308
27	200000000	652980194	1045050771	1698030965

	rotten_tomatoes_score	rotten_tomatoes_counts	metacritic_score
21	88	350	64
22	95	360	83
23	91	303	71
24	95	289	83
25	74	319	60
26	73	262	58
27	90	313	73

	metacritic_counts	imdb_score	imdb_counts
21	56	7.4	174917
22	55	8.0	392783
23	52	7.4	202404
24	52	7.0	158649
25	57	6.1	127045
26	45	7.0	140174
27	59	7.6	168090

[9]:
```
df.columns = df.columns.str.replace(" ", "_").str.replace('%', '')
```

```
#Eliminamos las columnas 'box_office_us_canada' , 'box_office_other', ya que␣
 ↪evaluaremos sólo con 'box_office_worldwide'
df = df.drop(columns=['box_office_us_canada', 'box_office_other'])
```

[10]:
```
#Modificamos el valor de budget de la película 'Luca', ya que por error de la␣
 ↪bd, sale 0
df['budget'] = df['budget'].replace(0, 175000000)
```

[11]:
```
#Agregamos una columna 'Revenue' que hace referencia a las ganancias de acuerdo␣
 ↪al presupuesto de cada película
df['Revenue'] = df['box_office_worldwide'] - df['budget']
```

[12]:
```
#Mostramos la columna 'Revenue'
df['Revenue']
```

[12]:
```
0      364436586
1      243258859
2      421358276
3      413773250
4      777014978
5      539442092
6      341983149
7      473726085
8      341311860
9      560099082
10     866969703
11     359852396
12     353983207
13     543559607
14     682611174
15     157207671
16     828570889
17     208930656
18     639641172
19    1042805359
20     873394593
21     -33059958
22     -28096115
23    -123887686
24    -153186642
25      26425420
26     296444308
27    1498030965
Name: Revenue, dtype: int64
```

[13]:
```
#Cambiamos la columna 'release_date' a tipo fecha
df['release_date'] = pd.to_datetime(df['release_date'])
```

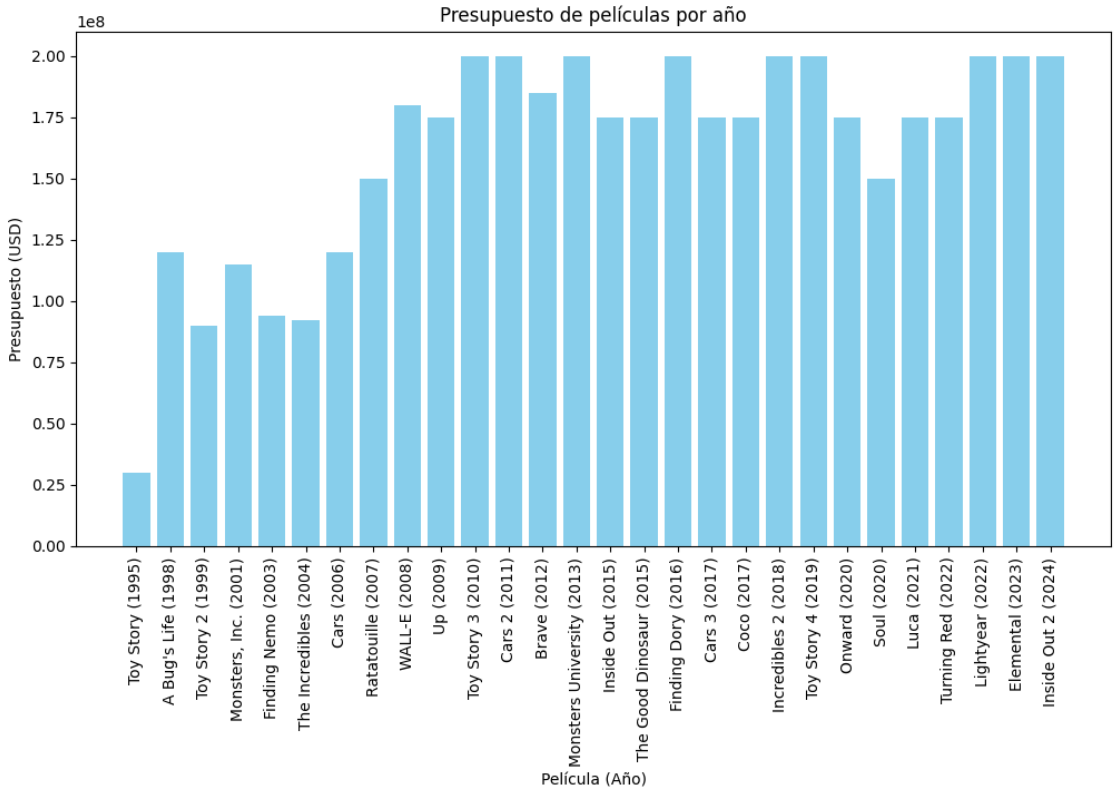
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28 entries, 0 to 27
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ID                      28 non-null     int64
 1   film                    28 non-null     object
 2   film_rating             28 non-null     object
 3   cinema_score            28 non-null     object
 4   release_date            28 non-null     datetime64[ns]
 5   run_time                28 non-null     int64
 6   budget                  28 non-null     int64
 7   box_office_worldwide    28 non-null     int64
 8   rotten_tomatoes_score   28 non-null     int64
 9   rotten_tomatoes_counts  28 non-null     int64
 10  metacritic_score        28 non-null     int64
 11  metacritic_counts       28 non-null     int64
 12  imdb_score              28 non-null     float64
 13  imdb_counts             28 non-null     int64
 14  Revenue                 28 non-null     int64
dtypes: datetime64[ns](1), float64(1), int64(10), object(3)
memory usage: 3.4+ KB
```

[14]:
```
#Exportamos la data limpia a un excel
df.to_excel('data_clean.xlsx', index = False)
```

[15]:
```
# Extraer el año
df['year'] = df['release_date'].dt.year

# Crear una nueva columna combinando película y año
df['label'] = df['film'] + " (" + df['year'].astype(str) + ")"

# Ordenar el DataFrame por año
df_sorted = df.sort_values(by='year')

# Crear el gráfico de barras
plt.figure(figsize=(12, 6))
plt.bar(df_sorted['label'], df_sorted['budget'], color='skyblue')

# Personalizar el gráfico
plt.xlabel("Película (Año)")
plt.ylabel("Presupuesto (USD)")
plt.title("Presupuesto de películas por año")
plt.xticks(rotation=90)  # Rotar los nombres de películas para mejor␣
 ↪visualización
```

```
# Mostrar el gráfico
plt.show()
```



1.1 Conclusión General:

- El análisis de presupuestos de películas Pixar muestra una clara tendencia al alza desde sus inicios, con un punto de inflexión significativo alrededor de 2008-2010.
- Se observa que los presupuestos se han estabilizado en aproximadamente 200 millones de dólares desde 2010, lo que indica una estandarización en la inversión por película.
- Las primeras producciones (1995-2006) tenían presupuestos considerablemente menores, con "Toy Story" (1995) siendo la de menor presupuesto con apenas 30 millones.
- La inversión en producciones ha aumentado aproximadamente un 566% desde la primera película hasta las más recientes.
- No se observa una correlación directa entre mayores presupuestos y éxito comercial (podría añadirse esta información si tienes datos de ingresos).
- El período 2022-2024 muestra presupuestos consistentemente altos, sugiriendo que Pixar mantiene su estrategia de inversión considerable en sus producciones más recientes.