Dans [29]:
```
import pandas as pd
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
```

Dans [12]:
```
bom_data = pd.read_csv(r'C:\Users\Saint Germain Emode\Downloads\bom.movie_

conn = sqlite3.connect(r'C:\Users\Saint Germain Emode\Downloads\dsc-phase-
imdb_data = pd.read_sql_query("SELECT * FROM movie_basics", conn)
```

Dans [18]:
```
print(bom_data.head())
```

```
                                              Titre Studio Domestic_gross \
0 Toy Story 3 BV 415000000.0
1 Alice au pays des merveilles (2010) BV 334200000.0
2 Harry Potter et les reliques de la mort, partie 1 WB 296000000.0
3 Inception WB 292600000.0
4 Shrek 4 : La Fin du monde P/DW 238700000.0

  Foreign_gross Année
0 652000000 2010
1 691300000 2010
2 664300000 2010
3 535700000 2010
4 513900000 2010
```

Dans [20]:
```
print(imdb_data.head())
```

```
    movie_id primary_title original_title \
0 tt0063540 Sunghursh Sunghursh
1 tt0066787 Un jour avant la saison des pluies Ashad Ka Ek Din
2 tt0069049 L'autre côté du vent L'autre côté du vent
3 tt0069204 Sabse Bada Sukh Sabse Bada Sukh
4 tt0100275 Le feuilleton errant La Telenovela Errante

   start_year runtime_minutes genres
0 2013 175.0 Action, Policier, Drame
1 2019 114.0 Biographie, Drame
2 2018 122.0 Drame
3 2018 NaN Comédie, Drame
4 2017 80.0 Comédie, Drame, Fantastique
```

Dans [23]:
```
print("Valeurs manquantes dans bom_data :")
print(bom_data.isnull().sum())

print("\nValeurs manquantes dans imdb_data :")
print(imdb_data.isnull().sum())
```

```
Valeurs manquantes dans bom_data :
title 0
studio 5
domestic_gross 28
Foreign_gross 1350
year 0
dtype: int64

Valeurs manquantes dans imdb_data :
movie_id 0
Primary_title 0
original_title 21
start_year 0
runtime_minutes 31739
genres 5408
dtype: int64
```

Dans [27]:
```
bom_data = bom_data.drop_duplicates()
imdb_data = imdb_data.drop_duplicates()
```

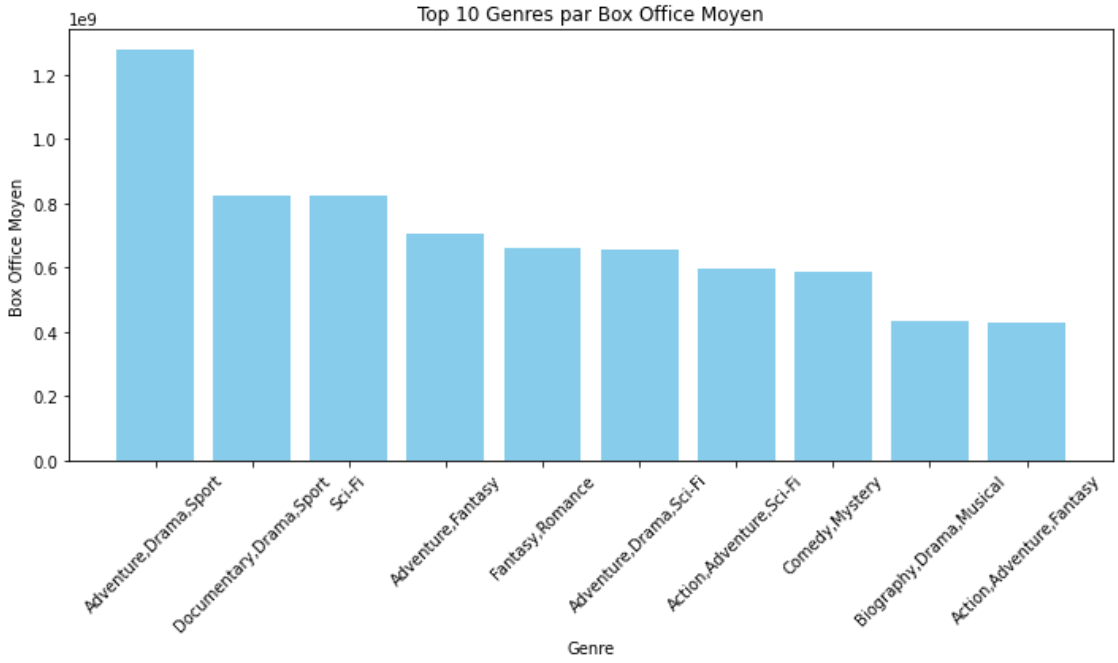
Dans [25]:
```
print(imdb_data.describe())
```

```
          début_année durée_minutes
nombre 146144,000000 114405,000000
moyenne 2014,621798 86,187247
écart-type 2,733583 166,360590
min 2010,000000 1,000000
25 % 2012,000000 70,000000
50 % 2015,000000 87,000000
75 % 2017,000000 99,000000
max 2115,000000 51420,000000
```

Dans [107]:

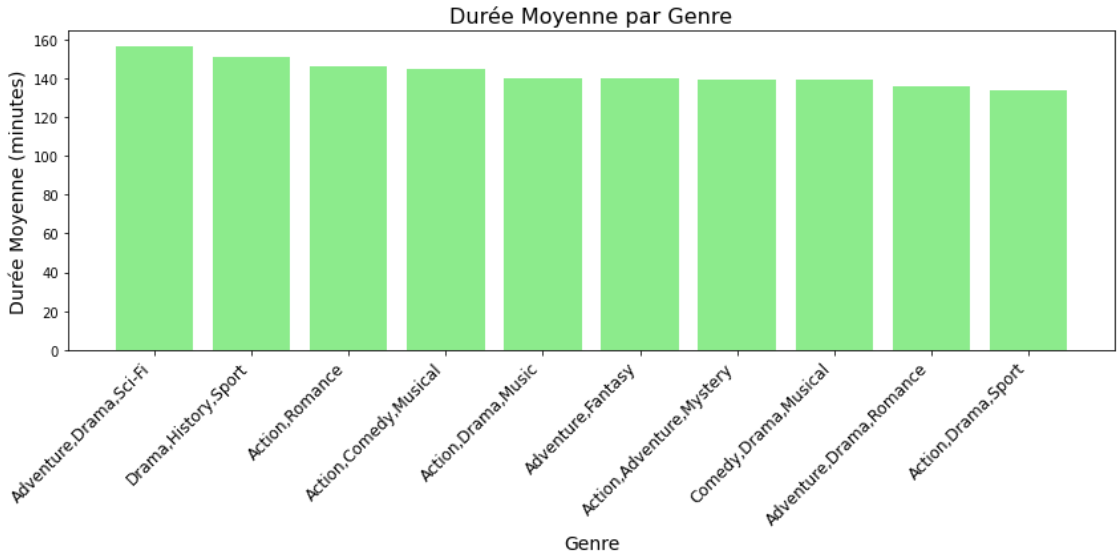
```
merged_data['domestic_gross'] = pd.to_numeric(merged_data['domestic_gross'
merged_data['foreign_gross'] = pd.to_numeric(merged_data['foreign_gross'],
merged_data['box_office'] = merged_data['domestic_gross'] + merged_data['f

box_office_moyen_par_genre = merged_data.groupby('genres')['box_office'].m

seuil_box_office = 1_000_000
box_office_filtré = box_office_moyen_par_genre[box_office_moyen_par_genre[

def regrouper_genres(genre):
    if 'Drame' in genre:
        return 'Drame'
    return genre

box_office_filtré.loc[:, 'genres'] = box_office_filtré['genres'].apply(reg

box_office_regroupé = box_office_filtré.groupby('genres')['box_office'].me

print("Durée Moyenne par Genres :")
print(durée_moyenne_par_genre)

top_genres = box_office_regroupé.nlargest(10, 'box_office')

plt.figure(figsize=(10, 6))
plt.bar(top_genres['genres'], top_genres['box_office'], color='skyblue')
plt.title('Top 10 Genres par Box Office Moyen')
plt.xlabel('Genre')
plt.ylabel('Box Office Moyen')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Durée Moyenne par Genres :
                        genres runtime_minutes
110 Aventure, Drame, Science-fiction 156,5
269 Drame, Histoire, Sport 151,0
74 Action, Romance 146,0
31 Action, Comédie, Comédie musicale 145,0
52 Action, Drame, Musique 140,0
.. ... ...
46 Action, Documentaire, Sport 2,0
79 Action, Sport, Guerre NaN
121 Aventure, Comédie musicale NaN
205 Comédie, Comédie musicale NaN
330 Western NaN

[331 lignes x 2 colonnes]
```


Top 10 Genres par Box Office Moyen
1e9
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Box Office Moyen
Adventure,Drama,Sport
Documentary,Drama,Sport
Sci-Fi
Adventure,Fantasy
Fantasy,Romance
Adventure,Drama,Sci-Fi
Action,Adventure,Sci-Fi
Comedy,Mystery
Biography,Drama,Musical
Action,Adventure,Fantasy
Genre

Dans [106]:
```
durée_moyenne_par_genre = merged_data.groupby('genres')['runtime_minutes']

durée_moyenne_par_genre = durée_moyenne_par_genre.sort_values(by='runtime_

print("Durée Moyenne par Genres :")
print(durée_moyenne_par_genre)

top_genres = durée_moyenne_par_genre.nlargest(10, 'runtime_minutes')

plt.figure(figsize=(12, 6))
plt.bar(top_genres['genres'], top_genres['runtime_minutes'], color='lightg
plt.title('Durée Moyenne par Genre', fontsize=16)
plt.xlabel('Genre', fontsize=14)
plt.ylabel('Durée Moyenne (minutes)', fontsize=14)
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.tight_layout()
plt.show()
```

```
Durée Moyenne par Genres :
                         genres runtime_minutes
110 Aventure, Drame, Science-fiction 156,5
269 Drame, Histoire, Sport 151,0
74 Action, Romance 146,0
31 Action, Comédie, Comédie musicale 145,0
52 Action, Drame, Musique 140,0
.. ... ...
46 Action, Documentaire, Sport 2,0
79 Action, Sport, Guerre NaN
121 Aventure, Comédie musicale NaN
205 Comédie, Comédie musicale NaN
330 Western NaN

[331 lignes x 2 colonnes]
```



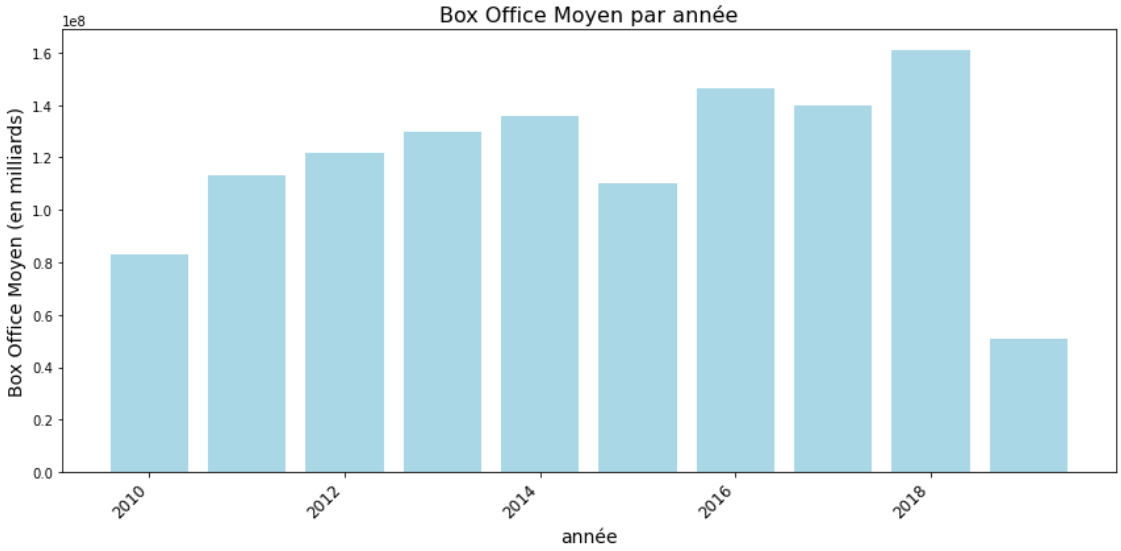
Dans [116]:
```
print(merged_data.columns)
```

```
Index(['movie_id', 'primary_title', 'original_title', 'start_year',
       'runtime_minutes', 'genres', 'title', 'studio', 'domestic_gross',
       'foreign_gross', 'year', 'box_office', 'succès'],
      dtype='object')
```

Dans [125]:

```
box_office_moyen = merged_data.groupby('start_year')['box_office'].mean().

box_office_moyen_par_années = box_office_moyen_par_années.sort_values(by='

print("Box Office Moyen par année :")
print(box_office_moyen_par_années)

top_years = box_office_moyen.nlargest(10, 'box_office')

plt.figure(figsize=(12, 6))
plt.bar(top_years['start_year'], top_years['box_office'], color='lightblue
plt.title('Box Office Moyen par année', fontsize=16)
plt.xlabel('année', fontsize=14)
plt.ylabel('Box Office Moyen (en milliards)', fontsize=14)
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.tight_layout()
plt.show()
```

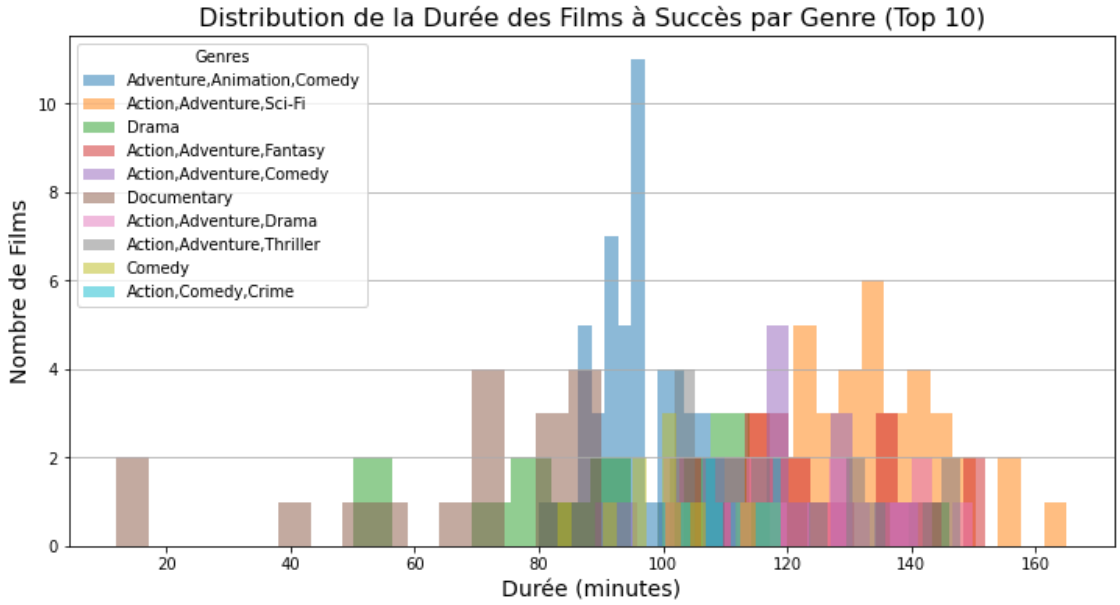
```
Box Office Moyenne par année :
    début_année box_office
0 2010 8.320715e+07
1 2011 1.131199e+08
2 2012 1.217887e+08
3 2013 1.297478e+08
4 2014 1.357221e+08
5 2015 1.100242e+08
6 2016 1.463142e+08
7 2017 1.396953e+08
8 2018 1.607877e+08
9 2019 5.101573e+07
10 2020 NaN
```



Dans [121]:

```
print(merged_data.columns)

seuil_succes = merged_data['box_office'].quantile(0.75)

merged_data['succès'] = merged_data['box_office'] >= seuil_succes

films_succes = merged_data[merged_data['succès']]

genre_counts = films_succes['genres'].value_counts().head(10)

plt.figure(figsize=(12, 6))
for genre in genre_counts.index:
    durée_genre = films_succes[films_succes['genres'] == genre]['runtime_m

    if not durée_genre.empty:
        plt.hist(durée_genre, bins=15, alpha=0.5, label=genre)

plt.title('Distribution de la Durée des Films à Succès par Genre (Top 10)'
plt.xlabel('Durée (minutes)', fontsize=14)
plt.ylabel('Nombre de Films', fontsize=14)
plt.legend(title='Genres')
plt.grid(axis='y')
plt.show()
```

```
Index(['movie_id', 'primary_title', 'original_title', 'start_year',
       'runtime_minutes', 'genres', 'title', 'studio', 'domestic_gross',
       'foreign_gross', 'year', 'box_office', 'succès'],
      dtype='object')
```



Voici trois recommandations commerciales concrètes basées sur les analyses que je fais concernant le box-office, la durée des films, et les genres

1: Cibler les Productions selon les Genres les Plus Rentables : Utilisez les données de box-office pour identifier les genres de films les plus lucratifs. Par exemple, si les films d'action et de comédie génèrent les meilleurs revenus, concentrez-vous sur le développement de projets dans ces genres spécifiques.

2: Optimiser la Durée des Films pour Maximiser les Revenus : Analysez la relation entre la durée des films et leur performance au box-office. Si les films d'une certaine durée (par exemple, entre 120 et 140 minutes) rapportent plus, ajustez les scripts et la production pour respecter ces durées.

3: Adapter les Stratégies de Marketing et de Distribution : En analysant les tendances de succès des films Cibler une durée de 90 à 120 minutes pour les nouveaux projets afin d'attirer un maximum de spectateurs.Explorer davantage les combinaisons de genre et de durée qui maximisent le succès au box-office.