**گزارش پروژۀ علم داده**

**دکتر نادری**

**زهرا دهقان**

**هلیا شمس زاده**

**الف) بخش EDA**

* **برای رسم نمودارها از plotly استفاده شده است تا فهم و خواندن نمودارها راحت‌تر باشد.**

**ایجاد توابع کمکی برای ستون‌هایی با مقادیر از نوع دیکشنری**

تابع parse\_list\_column برای تبدیل رشته‌هایی که نمایش‌دهنده یک لیست هستند به یک لیست واقعی استفاده می‌شود. این تابع ابتدا بررسی می‌کند که مقدار موجود در ستون از نوع رشته‌ای باشد و در صورت تأیید، با استفاده از ast.literal\_eval آن را به یک لیست تبدیل می‌کند. تابع extract\_dict\_field برای استخراج مقدار خاصی از یک دیکشنری که در یک ستون قرار دارد، استفاده می‌شود. این تابع مقدار موردنظر را بر اساس کلید مشخص‌شده (field\_name) از دیکشنری استخراج کرده و مقدار آن را بازمی‌گرداند. اگر مقدار موجود در ستون از نوع دیکشنری نباشد، مقدار None برگردانده می‌شود. این روش برای پردازش ستون‌هایی که شامل اطلاعاتی به‌صورت دیکشنری (مانند جزئیات امتیاز فیلم‌ها یا اطلاعات مربوط به کارگردان) هستند، کاربرد دارد.

**پیدا کردن تعداد ژانرهای منحصر به فرد**

ابتدا، با استفاده از apply و ast.literal\_eval، مقادیر رشته‌ای که نمایش‌دهنده یک لیست هستند، به لیست واقعی تبدیل می‌شوند. این کار فقط در صورتی انجام می‌شود که مقدار موردنظر از نوع رشته‌ای بوده و با [ شروع شده باشد؛ در غیر این صورت، مقدار به‌صورت یک لیست خالی تنظیم می‌شود تا از بروز خطا جلوگیری شود.

پس از تبدیل رشته‌ها به لیست، با استفاده از تابع explode، مقادیر لیست در هر سطر از DataFrame گسترش داده می‌شوند، به‌طوری که هر ژانر به‌عنوان یک مقدار مستقل در یک ردیف جداگانه قرار می‌گیرد. سپس، مقدار dropna اعمال می‌شود تا مقادیر NaN حذف شوند.

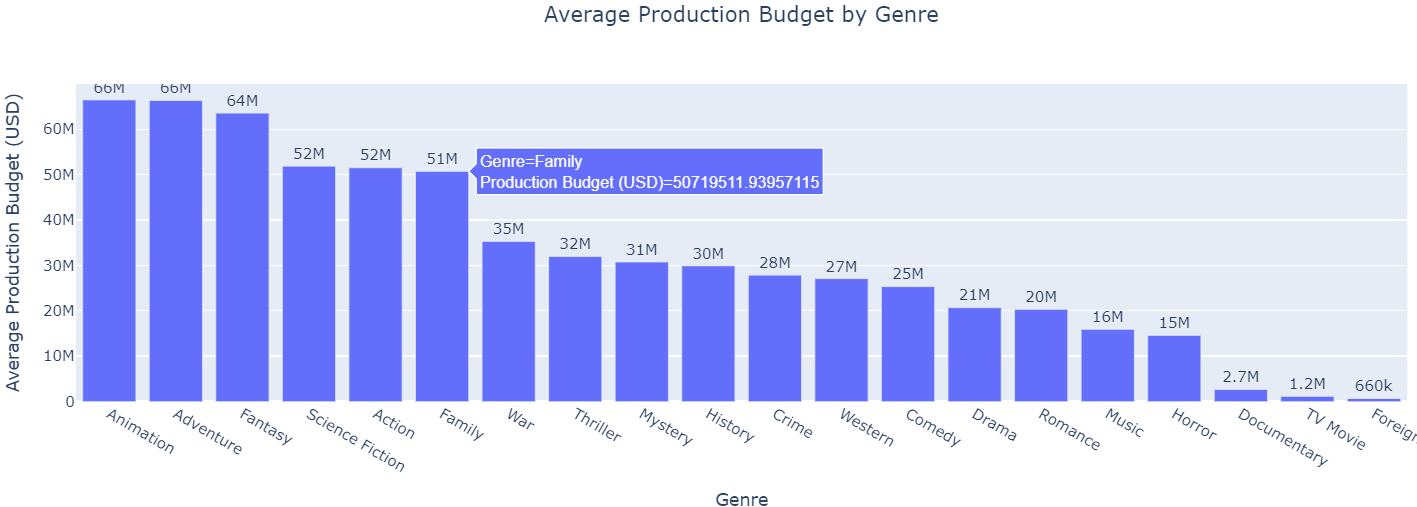
در مرحله بعد، از یک apply دیگر استفاده شده است تا نام ژانرها از دیکشنری استخراج شود. در صورتی که مقدار موجود از نوع دیکشنری باشد، مقدار مربوط به کلید 'name' دریافت می‌شود؛ در غیر این صورت، مقدار None برگردانده می‌شود. مقادیر None نیز با dropna حذف شده و در نهایت، مقادیر منحصربه‌فرد (unique()) استخراج می‌شوند.

1. **متوسط هزینه برای هر ژانر فیلم چقدر است؟**

ابتدا، یک کپی از DataFrame اصلی فیلم‌ها (df\_movies) با نام df\_genres\_expanded ایجاد می‌شود تا از تغییر داده‌های اصلی جلوگیری شود. سپس، تابع parse\_list\_column روی ستون rt\_genres اعمال می‌شود تا مقادیر رشته‌ای که نمایش‌دهنده لیست هستند، به لیست‌های واقعی تبدیل شوند.

در مرحله بعد، از متد explode برای گسترش ژانرها استفاده می‌شود. این روش باعث می‌شود که هر فیلمی که دارای چند ژانر است، در چندین ردیف نمایش داده شود، به‌طوری که هر ردیف نمایانگر یکی از ژانرهای مربوط به فیلم باشد. سپس، تابع extract\_dict\_field برای استخراج نام ژانرها از دیکشنری‌های موجود در ستون rt\_genres استفاده می‌شود و مقدار استخراج‌شده در یک ستون جدید با نام genre\_name ذخیره می‌شود. پس از آن، تمامی ردیف‌هایی که مقدار genre\_name آن‌ها مقدار NaN دارد، حذف می‌شوند.

در نهایت، میانگین بودجه تولید فیلم‌ها برای هر ژانر محاسبه می‌شود. این کار با استفاده از groupby('genre\_name') روی ستون rt\_production\_budget انجام شده و میانگین بودجه با mean() محاسبه می‌شود. نتیجه‌ی این عملیات در قالب یک DataFrame جدید ذخیره می‌شود که دارای دو ستون genre\_name و rt\_production\_budget است، جایی که هر ردیف میانگین بودجه تولید برای یک ژانر خاص را نشان می‌دهد.

**نمودار:**

**نتیجه:** طبق نمودار می‌توان فهمید پرخرج‌ترین ژانرها به ترتیب انیمیشن، ماجراجویی، فانتزی، علمی-تخیلی، اکشن و خانوادگی هستند. کم‌خرج‌ترین ژانرها هم خارجی، تلویزیون و مستند هستند.

1. **سهم هر کشور در مجموع هزینۀ هر ژانر فیلم چقدر است؟ (برای 5 تا از پرخرج‌ترین ژانرها به دست آورید.)**

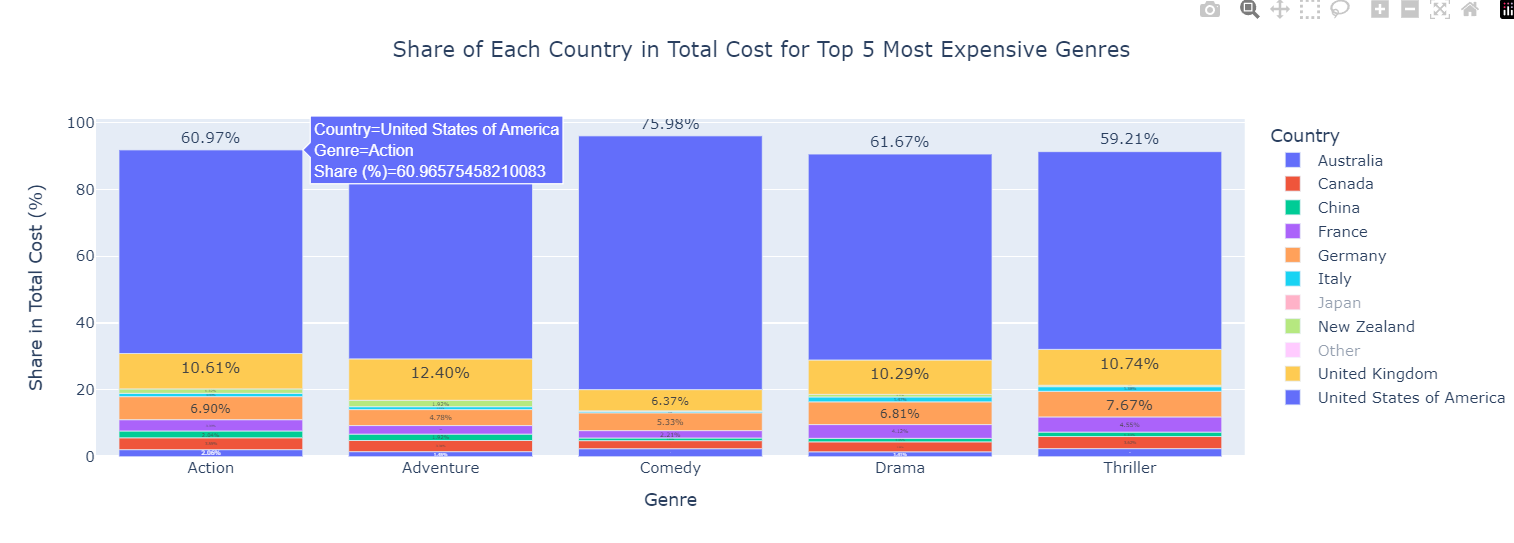
ابتدا، ستون‌های rt\_genres و rt\_production\_countries در DataFrame temp\_df با استفاده از تابع parse\_list\_column پردازش می‌شوند تا مقادیر رشته‌ای که نمایش‌دهنده لیست هستند، به لیست‌های واقعی تبدیل شوند.

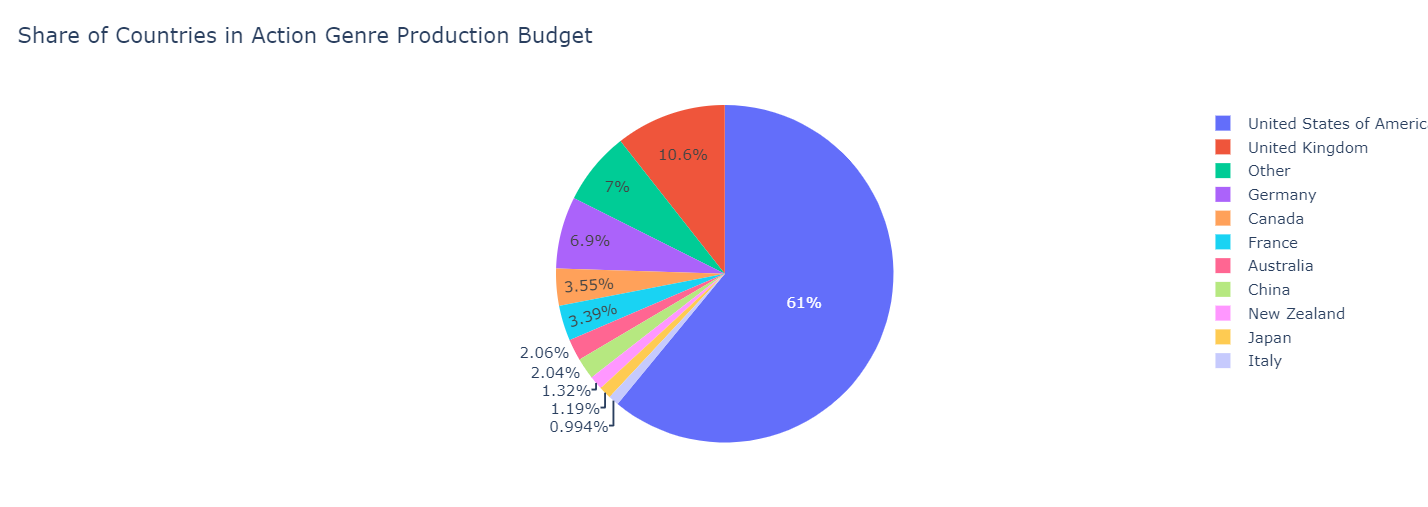
سپس، DataFrame با استفاده از explode گسترش داده می‌شود تا هر فیلمی که دارای چند ژانر یا چند کشور تولیدکننده است، در چندین ردیف نمایش داده شود. این باعث می‌شود که هر ردیف فقط یک ژانر و یک کشور را شامل شود. در ادامه، نام ژانرها و نام کشورها از دیکشنری‌های موجود در ستون‌های rt\_genres و rt\_production\_countries با استفاده از تابع extract\_dict\_field استخراج شده و در دو ستون جدید genre\_name و country\_name ذخیره می‌شوند. سپس، تمامی ردیف‌هایی که فاقد مقدار معتبر در هر یک از این دو ستون هستند، حذف می‌شوند.

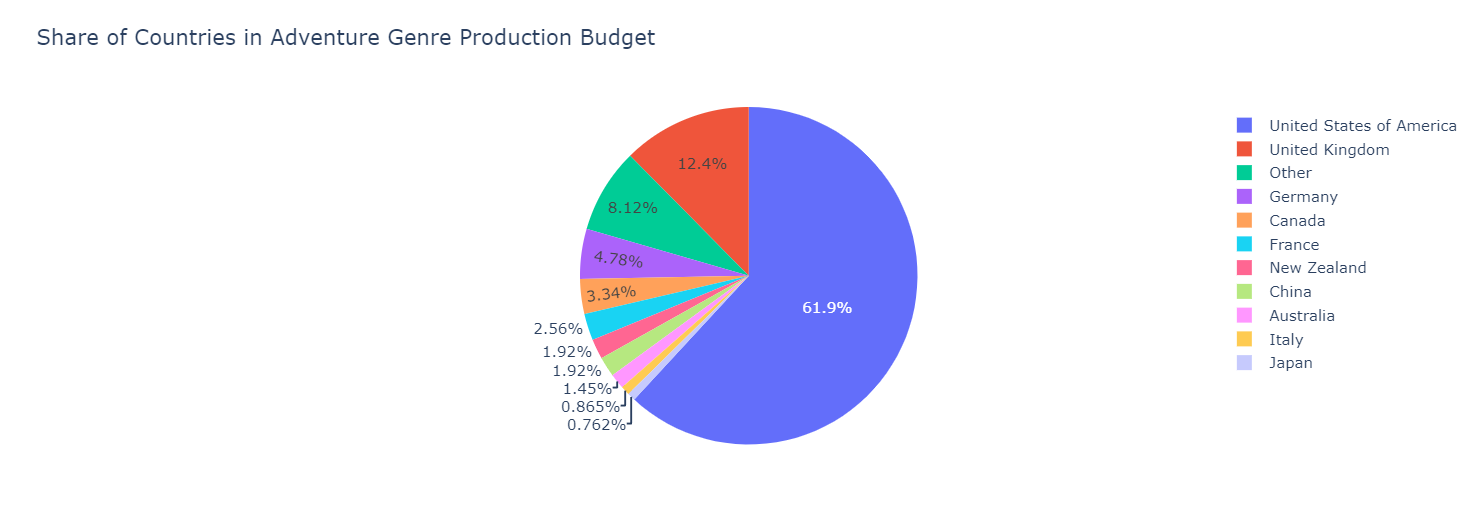
پس از پاک‌سازی داده‌ها، مجموع بودجه تولید فیلم‌ها برای هر ترکیب کشور-ژانر با استفاده از groupby محاسبه می‌شود. نتیجه این گروه‌بندی در DataFrame country\_genre\_budget ذخیره شده است. همچنین، مجموع بودجه تولید برای هر ژانر در سطح کلی محاسبه شده و در DataFrame total\_budget\_by\_genre ذخیره می‌شود. این اطلاعات به DataFrame اولیه ترکیب کشور-ژانر اضافه می‌شود تا امکان محاسبه سهم هر کشور از کل بودجه تولید ژانر مربوطه فراهم شود. سهم هر کشور برای یک ژانر خاص بر اساس نسبت بودجه آن کشور به کل بودجه ژانر در سطح جهانی، در ستون share به‌صورت درصدی ذخیره می‌شود.

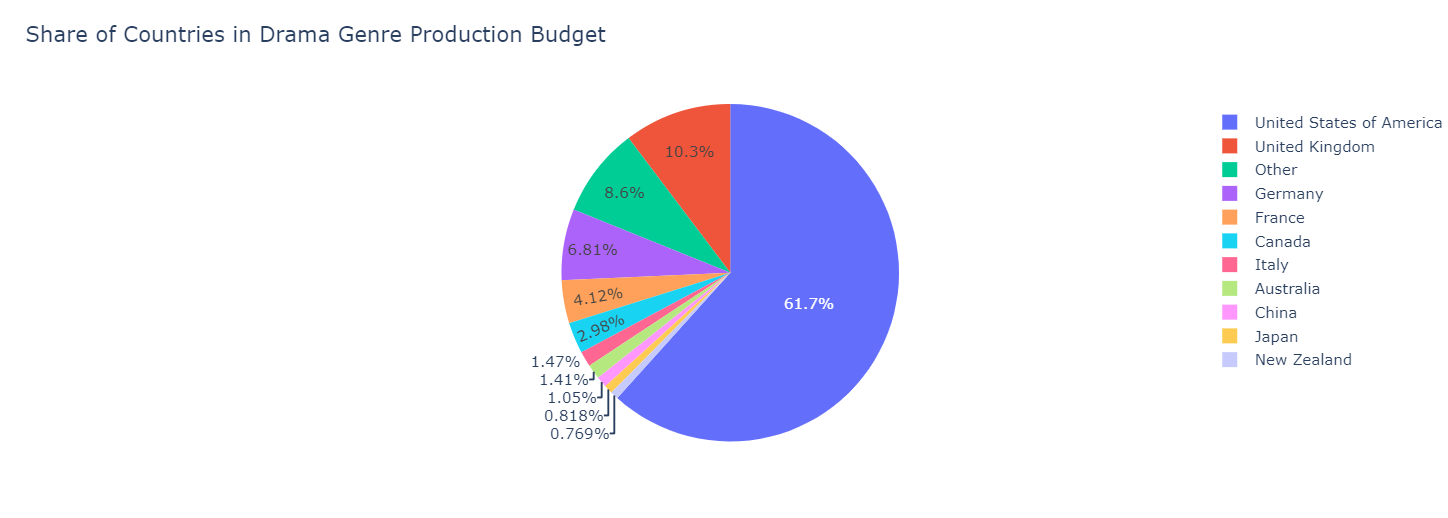
در نهایت، برای هر کشور، پنج ژانر با بیشترین بودجه تولید شناسایی شده‌اند. این کار از طریق مرتب‌سازی داده‌ها بر اساس country\_name و rt\_production\_budget (به ترتیب صعودی برای کشورها و نزولی برای بودجه) انجام شده و سپس با استفاده از groupby و head(5)، پنج ژانر با بیشترین بودجه برای هر کشور استخراج شده‌اند.

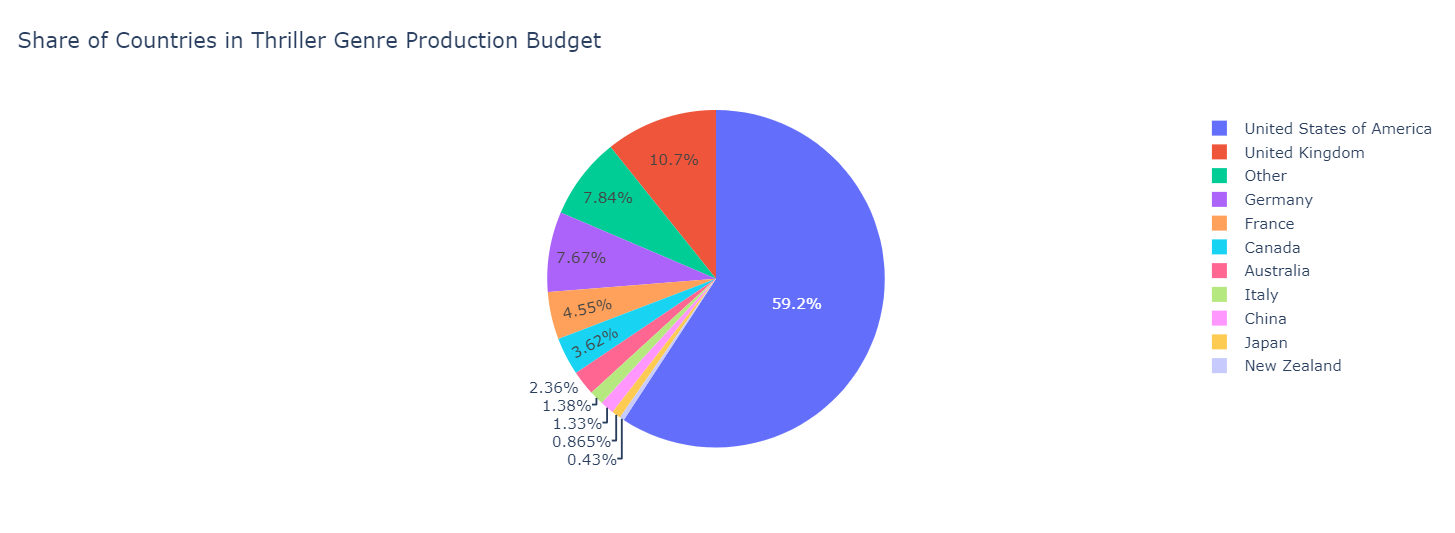
**نمودار:**

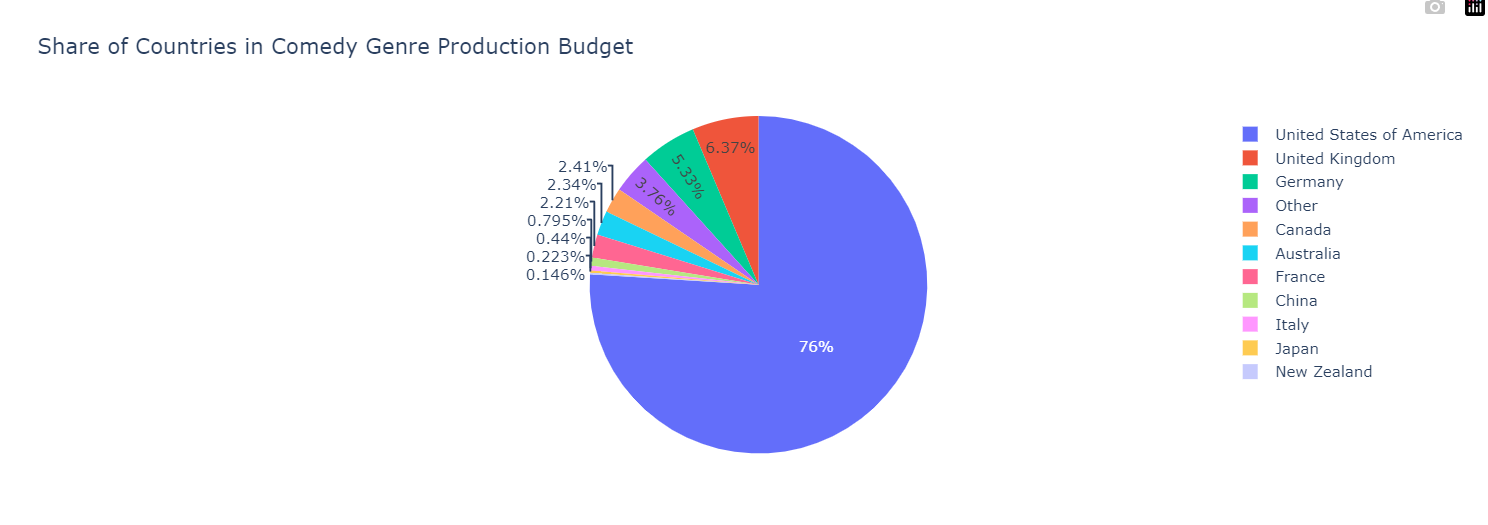












**نتیجه:** در همۀ 5 تا ژانر، بیشترین سهم برای آمریکا و سپس انگلستان و آلمان می‌باشد.

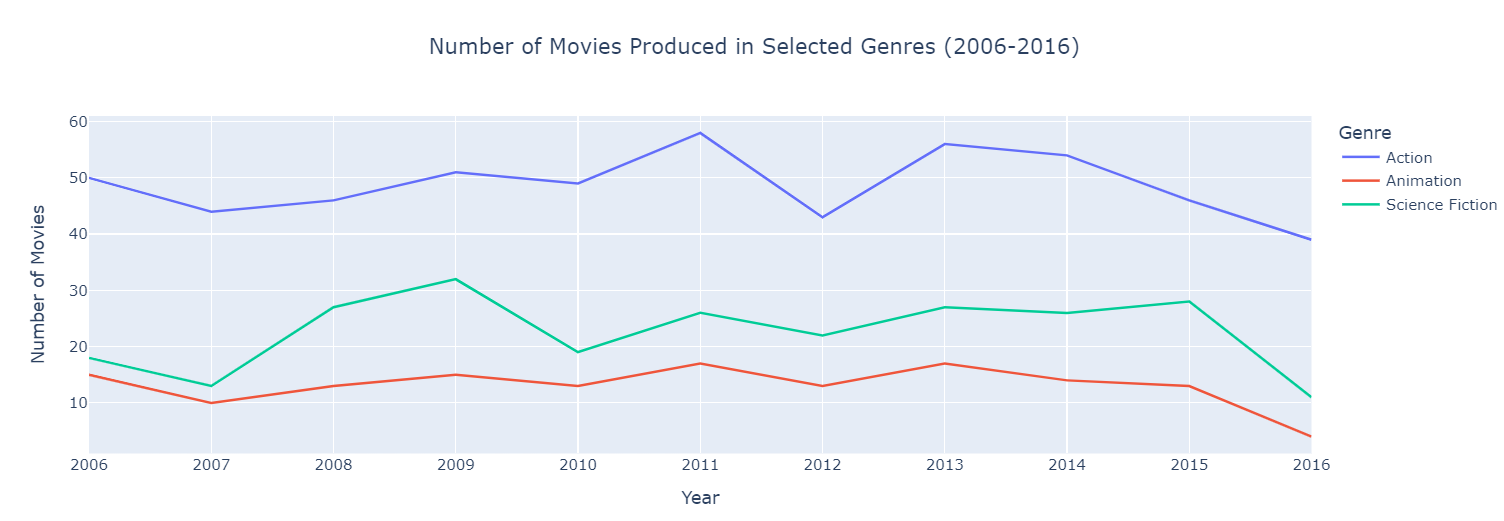
1. **تعداد فیلم های ساخته شده در 3 ژانر را در 10 سال گذشته مقایسه کنید.**

ابتدا، ستون rt\_release\_date که شامل تاریخ انتشار فیلم‌ها است، با استفاده از pd.to\_datetime به یک نوع داده‌ی زمانی (datetime) تبدیل می‌شود. برای جلوگیری از بروز خطا هنگام تبدیل مقادیر نامعتبر، گزینه errors='coerce' استفاده شده است که مقدارهای نامعتبر را به NaT تبدیل می‌کند. سپس، سال انتشار هر فیلم از این داده‌ی زمانی استخراج شده و در یک ستون جدید با نام release\_year ذخیره می‌شود.

پس از آن، یک فیلتر روی داده‌ها اعمال می‌شود تا فقط فیلم‌هایی که بین سال‌های ۲۰۰۷ تا ۲۰۱۶ منتشر شده‌اند، در DataFrame last\_10\_years نگه‌داشته شوند. سپس، ستون rt\_genres که شامل لیستی از ژانرهای هر فیلم است، با استفاده از متد explode گسترش داده می‌شود، به‌طوری که هر فیلم در چندین ردیف قرار گیرد، به‌ازای هر ژانری که به آن تعلق دارد. نام ژانرها با استفاده از تابع extract\_dict\_field استخراج شده و در ستون genre\_name ذخیره می‌شود.

در مرحله بعد، از بین تمام ژانرها، فقط سه ژانر اکشن، انیمیشن و علمی-تخیلی انتخاب می‌شوند. این کار از طریق اعمال یک فیلتر روی ستون genre\_name و استفاده از متد isin(selected\_genres) انجام می‌شود. در نهایت، تعداد فیلم‌های مربوط به هر ژانر در هر سال با استفاده از groupby(['release\_year', 'genre\_name']) محاسبه شده و تعداد فیلم‌های موجود در هر گروه با size() شمارش می‌شود. نتیجه در قالب یک DataFrame جدید با نام genre\_counts ذخیره شده که شامل سه ستون release\_year، genre\_name و movie\_count است.

**نمودار:**



**نتیجه:** سه ژانر انتخاب شده اند. در بین این سه ژانر، در تمام این سال‌ها تعداد فیلم‌های بیشتری در ژانر اکشن ساخته شده است. اما هر سه از سال 2013 افت داشته اند. ژانر اکشن در این بازه زمانی پرطرفدار بوده اما از 2011 روند نزولی گرفته است. ژانر علمی تخیلی رشد داشته اما در سال 2016 افت کرده است. تولید فیلم‌های انیمیشنی به‌طور کلی کمتر از سایر ژانرها بوده اما تغییرات محسوسی نداشته است.

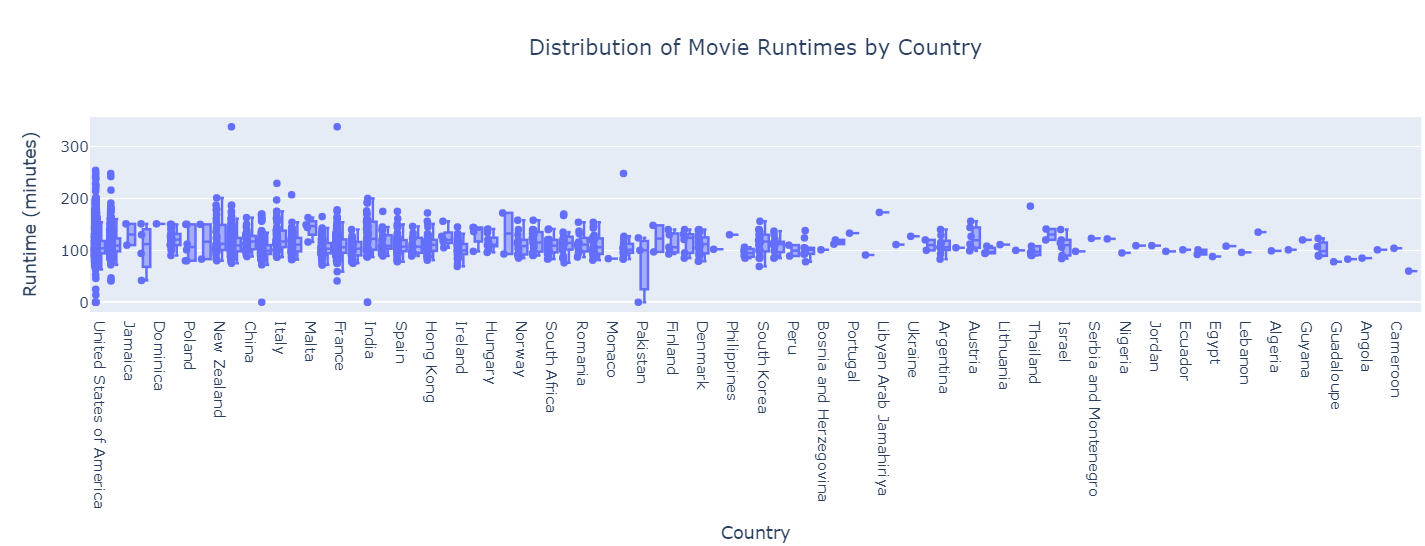
1. **به طور متوسط كدام كشور ها طولانى ترين فيلم ها و كوتاه ترين فيلم ها را مي‌سازند؟**

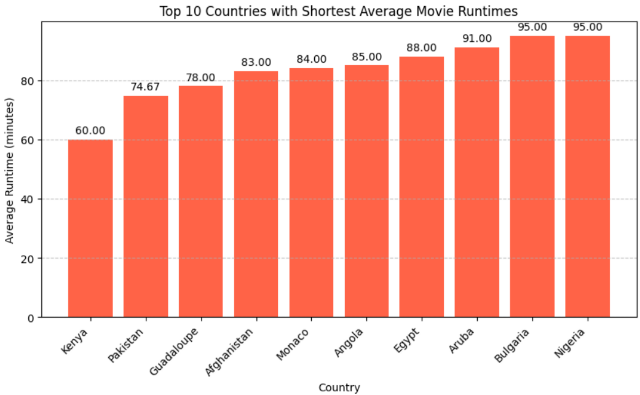
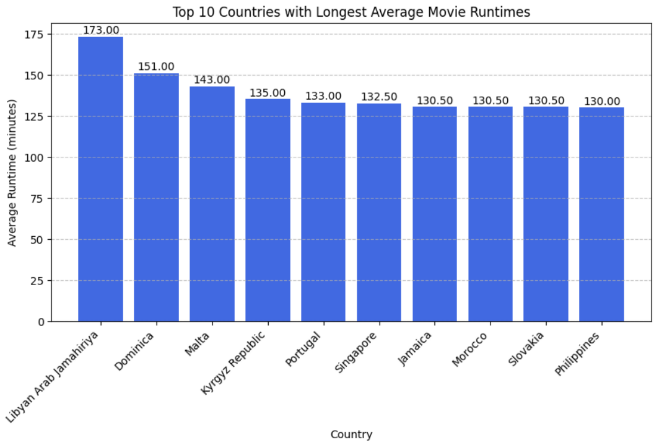
ابتدا، یک کپی از DataFrame اصلی فیلم‌ها (df\_movies) در متغیر temp\_df ذخیره می‌شود تا داده‌های اصلی بدون تغییر باقی بمانند. سپس، ستون rt\_production\_countries که شامل اطلاعات کشورها به‌صورت رشته‌ای (لیست دیکشنری‌ها) است، پردازش می‌شود. با استفاده از ast.literal\_eval، مقادیر رشته‌ای به لیست‌های واقعی از دیکشنری‌ها تبدیل می‌شوند. این کار فقط در صورتی انجام می‌شود که مقدار موجود از نوع رشته باشد؛ در غیر این صورت، مقدار به‌صورت یک لیست خالی قرار داده می‌شود.

پس از تبدیل، متد explode برای گسترش داده‌ها استفاده می‌شود، به‌طوری که هر فیلم که در چند کشور تولید شده باشد، در چندین ردیف جداگانه نمایش داده شود، به ازای هر کشور یک ردیف. سپس، نام کشورها از دیکشنری‌های موجود در ستون rt\_production\_countries استخراج شده و در ستون country\_name ذخیره می‌شود. اگر مقدار موجود از نوع دیکشنری نباشد، مقدار None در نظر گرفته می‌شود.

در ادامه، ردیف‌هایی که مقدار country\_name یا rt\_runtime آن‌ها مقدار نامعتبر (NaN) دارند، حذف می‌شوند تا فقط داده‌های معتبر باقی بمانند. سپس، مدت‌زمان میانگین فیلم‌ها برای هر کشور محاسبه می‌شود. این کار با استفاده از groupby('country\_name') روی ستون rt\_runtime انجام شده و مقدار میانگین با mean() محاسبه می‌شود. نتیجه این محاسبات در DataFrame average\_runtime\_by\_country ذخیره می‌شود.

در مرحله‌ی آخر، کشور با بیشترین و کمترین میانگین زمان پخش فیلم‌ها شناسایی می‌شود. این کار از طریق idxmax() و idxmin() روی ستون rt\_runtime انجام می‌شود که به ترتیب بیشترین و کمترین مقدار میانگین را مشخص می‌کنند. اطلاعات مربوط به این دو کشور در متغیرهای longest\_runtime\_country و shortest\_runtime\_country ذخیره می‌شوند.

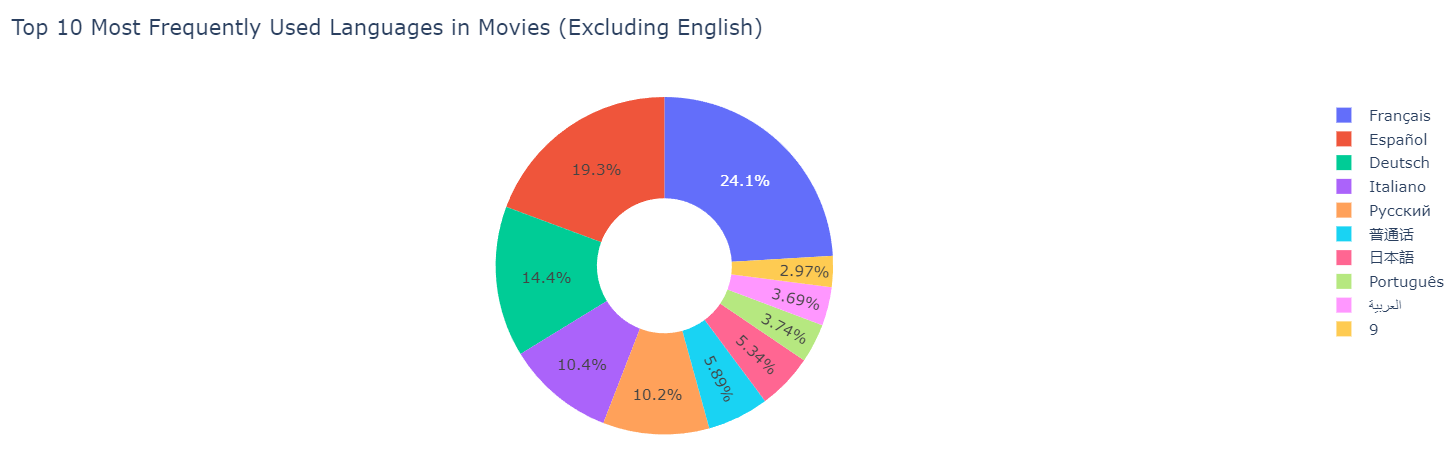
**نمودار:**



**نتیجه:** طولانی‌ترین فیلم‌ها را به طور میانگین به ترتیب لیبی، دومینیکا و مالتا ساخته اند، و کوتاه‌ترین فیلم‌ها را به ترتیب کنیا، پاکستان و گوادالوپ ساخته‌اند.

1. **به غیر از انگلیسی، پر تکرار ترین زبان ها در فیلم ها چه هستند؟**

ابتدا یک کپی از DataFrame اصلی ساخته می‌شود تا داده‌ها بدون تغییر باقی بمانند. سپس، ستون rt\_languages که شامل لیستی از زبان‌های هر فیلم است، به صورت دیکشنری‌های قابل پردازش تبدیل می‌شود. پس از آن، با استفاده از متد explode، هر زبان به ردیف جداگانه‌ای تبدیل می‌شود و نام زبان‌ها از دیکشنری‌ها استخراج می‌شود. ردیف‌هایی که دارای مقادیر نامعتبر در ستون زبان هستند، حذف می‌شوند. همچنین، فیلم‌های به زبان انگلیسی از تحلیل کنار گذاشته می‌شوند. در نهایت، تعداد فیلم‌هایی که به هر زبان غیر انگلیسی تولید شده‌اند، شمارش شده و در یک DataFrame جدید ذخیره می‌شود.

**نمودار:**

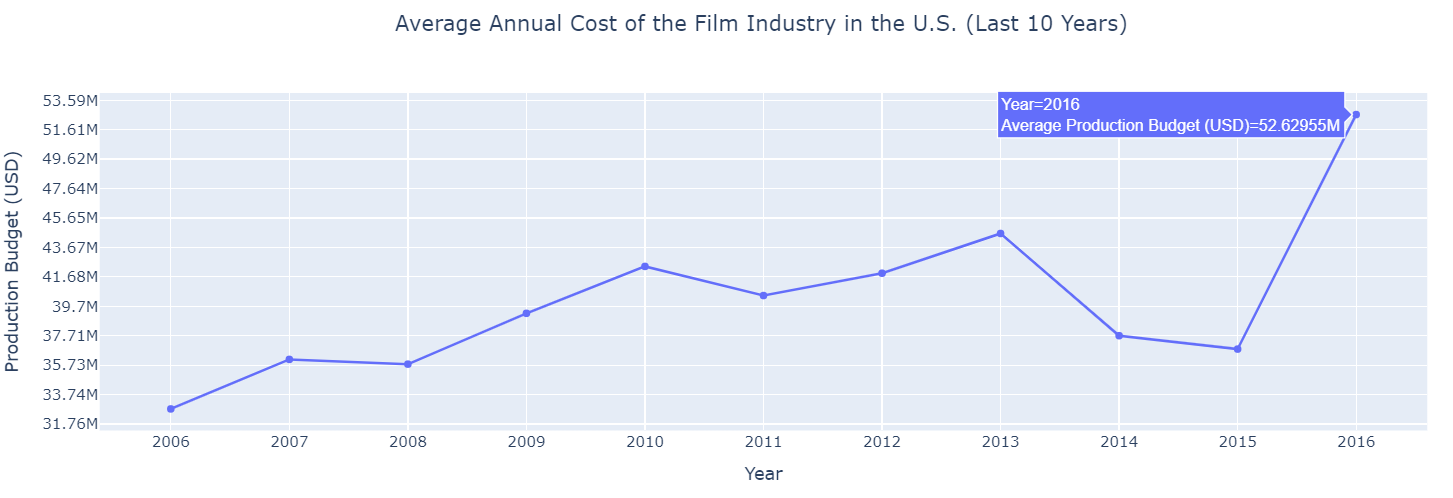
**نتیجه:** به غیر از انگلیسی، به ترتیب فرانسوی، اسپانیایی، آلمانی (داچ) و ایتالیایی استفاده شده اند.

1. **آمریکا در 10 سال گذشته، به طور متوسط در هر سال چقدر در صنعت فیلم‌سازی هزینه کرده است؟ (به تفکیک سال)**

در ابتدا، تاریخ انتشار فیلم‌ها از ستون rt\_release\_date به فرمت تاریخ و زمان تبدیل می‌شود، سپس فقط سال انتشار فیلم‌ها از آن استخراج می‌شود و در ستون جدید release\_year ذخیره می‌گردد. پس از آن، فیلم‌های تولید شده در ایالات متحده آمریکا فیلتر می‌شوند با استفاده از country\_name == 'United States of America' . سپس، برای محدود کردن تحلیل به فیلم‌های ۱۰ سال اخیر، از تابع between استفاده می‌شود تا تنها فیلم‌هایی که در بازه زمانی ۱۰ سال گذشته (از سال جاری تا سال ۱۰ سال پیش) منتشر شده‌اند، در نظر گرفته شوند.

در مرحله بعد، برای هر سال، میانگین هزینه تولید محاسبه می‌شود. این کار با استفاده از groupby('release\_year') و محاسبه میانگین rt\_production\_budget انجام می‌شود. نتیجه این محاسبات در yearly\_cost ذخیره می‌شود. در نهایت، برای اطمینان از عدم وجود مقادیر NaN در ستون‌های هزینه تولید، از fillna(0) استفاده می‌شود تا این مقادیر به صفر تبدیل شوند.

**نمودار:**

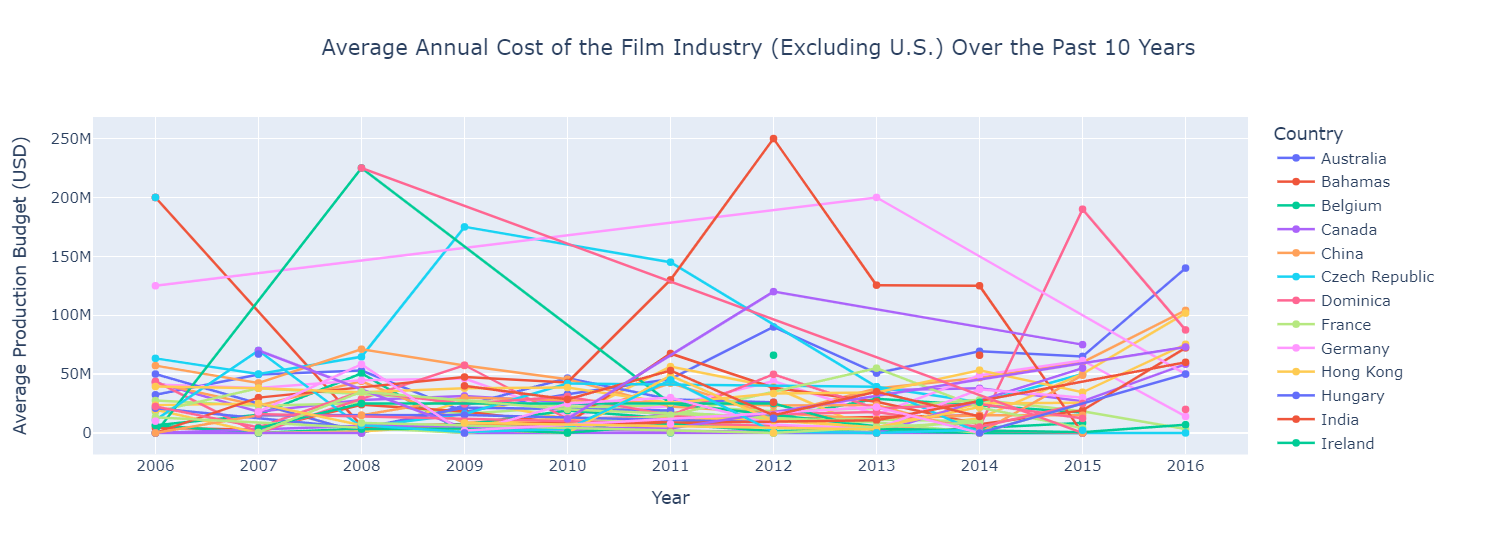


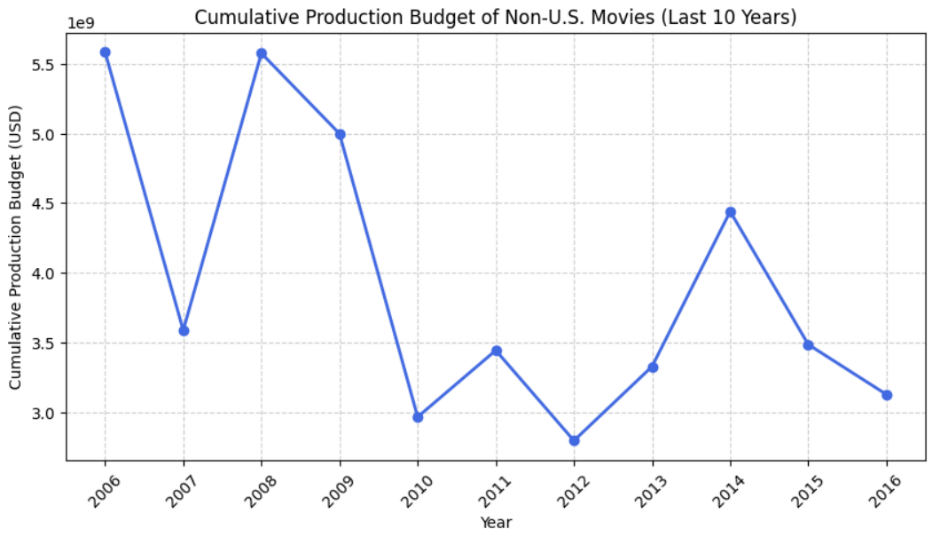
**نتیجه:** به طور کلی سرمایه‌گذاری آمریکا از سال 2006 روند صعودی داشته است (با کاهش در سال‌های 2014 و 2015) اما در نهایت به بیشترین میزان خود در سال 2016 رسیده است.

1. **روند قبلی را بدون در نظر گرفتن کشور برای 10 سال گذشته مقایسه کنید.**

در ابتدا، یک کپی از DataFrame اصلی (df\_movies) ساخته می‌شود و سپس ستون rt\_release\_date که شامل تاریخ انتشار فیلم‌ها است، به فرمت تاریخ تبدیل می‌شود. فیلم‌هایی که تاریخ انتشار نامعتبر دارند، با استفاده از dropna حذف می‌شوند. سپس سال انتشار فیلم‌ها در release\_year ذخیره می‌شود. در مرحله بعد، ستون rt\_production\_countries که شامل لیستی از کشورهای تولید فیلم است، با استفاده از تابع parse\_list\_column به لیست‌های دیکشنری تبدیل می‌شود. سپس با استفاده از explode، لیست کشورهای تولید به ردیف‌های جداگانه تبدیل می‌شود و هر کشور در یک ردیف خاص قرار می‌گیرد. در نهایت، نام کشورهای تولید از دیکشنری‌ها استخراج شده و در ستون country\_name ذخیره می‌شود. سپس فیلم‌هایی که در ایالات متحده آمریکا تولید نشده‌اند فیلتر می‌شوند و در non\_us\_movies ذخیره می‌شوند. این داده‌ها در مرحله بعد فیلتر می‌شوند تا فقط فیلم‌هایی که در ۱۰ سال گذشته منتشر شده‌اند نگه‌داری شوند. در نهایت، تحلیل هزینه تولید برای فیلم‌های غیرآمریکایی انجام می‌شود. برای هر سال و هر کشور، میانگین rt\_production\_budget محاسبه شده و در ستون average\_production\_budget ذخیره می‌شود. این تحلیل به بررسی هزینه‌های تولید فیلم در کشورهای مختلف و روند تغییرات آن در طی دهه گذشته کمک می‌کند.

**نمودار:**





**نتیجه:**

1. **Johnny Depp در چه فیلم هایی بازی کرده است؟**

ستون rt\_actors که شامل اطلاعات بازیگران به صورت لیست دیکشنری است، با استفاده از تابع parse\_list\_column به لیست‌های واقعی تبدیل می‌شود.

بعد از آن، با استفاده از تابع apply، فیلتر می‌شود که آیا **نام جانی دپ** در هر کدام از دیکشنری‌های بازیگران موجود است یا خیر. این بررسی با استفاده از دستور any انجام می‌شود که از بین همه بازیگران هر فیلم، اگر حتی یک بازیگر با نام جانی دپ وجود داشته باشد، آن فیلم در نتیجه گنجانده می‌شود.

در نهایت، فقط ستون‌های rt\_title (عنوان فیلم) و rt\_movie\_id (شناسه فیلم) انتخاب می‌شود و برای نمایش نمایش داده می‌شود.

**نمودار:**

**نتیجه:**

**9) به طور متوسط چند درصد نقش اول تا پنجم فیلم ها(به تفکیک برای هر نقش) مرد، و چند درصد زن هستند؟**

ابتدا یک کپی از DataFrame اصلی (در اینجا df\_credit) گرفته می‌شود تا پردازش‌ها بدون تغییر در داده‌های اصلی انجام شوند. در مرحله بعد، ستون rt\_actors که شامل اطلاعات بازیگران به صورت رشته است، با استفاده از تابع parse\_list\_column تبدیل به لیست‌های دیکشنری می‌شود. سپس، از متد explode برای تبدیل لیست‌ها به ردیف‌های جداگانه استفاده می‌شود، به طوری که هر بازیگر در یک ردیف جداگانه قرار می‌گیرد.

در ادامه، اطلاعات مربوط به جنسیت و ترتیب بازیگر از هر دیکشنری استخراج می‌شود و در دو ستون جدید gender و order ذخیره می‌شود. سپس، فقط بازیگران نقش‌های اصلی (با ترتیب کمتر از ۵) فیلتر می‌شوند و داده‌ها در DataFrame lead\_roles ذخیره می‌شوند.

در نهایت، توزیع جنسیتی بازیگران در این نقش‌ها محاسبه می‌شود و در gender\_distribution ذخیره می‌شود. این توزیع به صورت درصدی نشان‌دهنده سهم هر جنسیت از بازیگران نقش‌های اصلی است. برای خوانایی بیشتر، مقادیر جنسیتی به Female، Male و Unknown تبدیل می‌شوند.

**نمودار:**

**نتیجه**:

**11) محبوب ترین ژانرهای فیلم در 10 سال گذشته به چه ترتیب بوده است؟(یکبار بر اساس تعداد review و يكبار بر اساس critics\_score مقايسه كنيد)**

ابتدا یک کپی از DataFrame اصلی (df\_movies) ساخته می‌شود تا تغییرات بدون اثر بر داده‌های اصلی انجام شود. سپس، ستون rt\_genres که شامل لیستی از ژانرها به صورت رشته است، با استفاده از ast.literal\_eval به لیست‌های دیکشنری تبدیل می‌شود. بعد از آن، با استفاده از explode، لیست ژانرها به ردیف‌های جداگانه تبدیل می‌شود و هر ژانر در یک ردیف جدید قرار می‌گیرد.

در مرحله بعد، نام ژانرها از دیکشنری‌ها استخراج شده و در ستون genre\_name ذخیره می‌شود. همچنین، تاریخ انتشار فیلم‌ها در rt\_release\_date به فرمت تاریخ و زمان تبدیل شده و تنها سال انتشار (در ستون release\_year) استخراج می‌شود. سپس، فیلم‌هایی که در بازه زمانی ۱۱ سال گذشته منتشر شده‌اند (بین سال‌های حداکثر سال انتشار تا یک سال پیش) فیلتر می‌شوند.

در نهایت، دو تحلیل انجام می‌شود:

1. محبوبیت بر اساس تعداد نظرات کاربران: برای هر ژانر، مجموع rt\_review\_count محاسبه شده و بر اساس آن ژانرها مرتب می‌شوند.
2. محبوبیت بر اساس امتیاز منتقدین: برای هر ژانر، میانگین rt\_critics\_score محاسبه شده و بر اساس آن ژانرها مرتب می‌شوند.

**نمودار:  
  
نتیجه:**