

# Содержание

- 1 Исследование рынка российского кинопроката
  - 1.0.1 Откроем файлы с данными и изучим информацию
  - 1.0.2 Предобработка данных
    - 1.0.2.1 Пропуски
    - 1.0.2.2 Изменение типов данных
    - 1.0.2.3 Неявные дубликаты
    - 1.0.2.4 Изучение параметров
  - 1.0.3 Добавление параметров
  - 1.0.4 Исследовательский анализ данных
  - 1.0.5 Исследовательский анализ данных фильмов с государственной поддержкой
  - 1.0.6 Вывод:

Исследование рынка российского кинопроката =

Изучим рынок российского кинопроката и выявим текущие тренды. Уделим внимание фильмам, которые получили государственную поддержку. Попробуем ответить на вопрос, насколько такие фильмы интересны зрителю. Набор данных содержит информацию о прокатных удостоверениях, сборах и государственной поддержке фильмов, а также информацию с сайта КиноПоиск.

## Откроем файлы с данными и изучим информацию

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings(action = 'ignore')
pd.options.display.max_columns = 40
pd.options.display.float_format = '{:,.2f}'.format
```

```
In [2]: # загрузим первый набор с данными, посмотрим информацию о датасете.
df = pd.read_csv('C:/Users/User/Downloads/mkrf_movies.csv')
```

```
In [3]: df.head(5)
```

```
Out[3]:
```

	title	puNumber	show_start_date	type	film_studio	production_country	director	producer	age_restriction	refund
0	Открытый простор	221048915	2015-11-27T12:00:00.000Z	Художественный	Тачстоун Пикчерз, Кобальт Пикчерз, Бикон Пикче...	США	Кевин Костнер	Дэвид Валдес, Кевин Костнер, Джейк Эбертс	«18+» - запрещено для детей	
1	Особо важное задание	111013716	2016-09-13T12:00:00.000Z	Художественный	Киностудия "Мосфильм"	СССР	Е.Матвеев	NaN	«6+» - для детей старше 6 лет	
2	Особо опасен	221038416	2016-10-10T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Тимур Бекмамбетов	Джим Лемли, Джейсон Нетер, Марк Е.Платт, Яйн Смит	«18+» - запрещено для детей	
3	Особо опасен	221026916	2016-06-10T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Тимур Бекмамбетов	Джим Лемли, Джейсон Нетер, Марк Е.Платт, Яйн Смит	«18+» - запрещено для детей	
4	Особо опасен	221030815	2015-07-29T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Тимур Бекмамбетов	Джим Лемли, Джейсон Нетер, Марк Е.Платт, Яйн Смит	«18+» - запрещено для детей	

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7486 entries, 0 to 7485
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   title                 7486 non-null   object
1   puNumber              7486 non-null   object
2   show_start_date       7486 non-null   object
3   type                  7486 non-null   object
4   film_studio           7468 non-null   object
5   production_country     7484 non-null   object
6   director              7477 non-null   object
7   producer              6918 non-null   object
8   age_restriction        7486 non-null   object
9   refundable_support     332 non-null    float64
10  nonrefundable_support  332 non-null    float64
11  budget                 332 non-null    float64
12  financing_source       332 non-null    object
13  ratings                6519 non-null   object
14  genres                 6510 non-null   object
dtypes: float64(3), object(12)
memory usage: 877.4+ KB
```

```
In [5]: # загрузим второй датасет с данными, посмотрим информацию о датасете
df1 = pd.read_csv('C:/Users/User/Downloads/mkrf_shows.csv')
```

```
In [6]: df1.head(10)
```

```
Out[6]:
```

	puNumber	box_office
0	111000113	2,450.00
1	111000115	61,040.00
2	111000116	153,030,013.40
3	111000117	12,260,956.00
4	111000118	163,684,057.79
5	111000119	4,293,649.51
6	111000212	200.00
7	111000216	355,567.00
8	111000313	710.00
9	111000314	1,607,970.00

```
In [7]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3158 entries, 0 to 3157
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   puNumber    3158 non-null   int64
1   box_office  3158 non-null   float64
dtypes: float64(1), int64(1)
memory usage: 49.5 KB
```

Очевидно, что объединение датасетов будет происходить по столбцу `puNumber`. Так как данные указанного столбца имеют разные типы данных, перед объединением приведем столбцы к числовому формату, где необходимо

```
In [8]: #изменение формата на числовой
df['puNumber'] = pd.to_numeric(df['puNumber'], errors = 'coerce')
```

Объединим данные в один датасет:

```
In [9]: data = df.merge(df1, on = 'puNumber', how = 'left')
```

```
In [10]: # проверим информацию
data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 7486 entries, 0 to 7485
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   title                  7486 non-null   object
1   puNumber                7485 non-null   float64
2   show_start_date        7486 non-null   object
3   type                   7486 non-null   object
4   film_studio            7468 non-null   object
5   production_country     7484 non-null   object
6   director               7477 non-null   object
7   producer               6918 non-null   object
8   age_restriction        7486 non-null   object
9   refundable_support     332 non-null    float64
10  nonrefundable_support  332 non-null    float64
11  budget                 332 non-null    float64
12  financing_source       332 non-null    object
13  ratings                6519 non-null   object
14  genres                 6510 non-null   object
15  box_office             3158 non-null   float64
dtypes: float64(5), object(11)
memory usage: 994.2+ KB

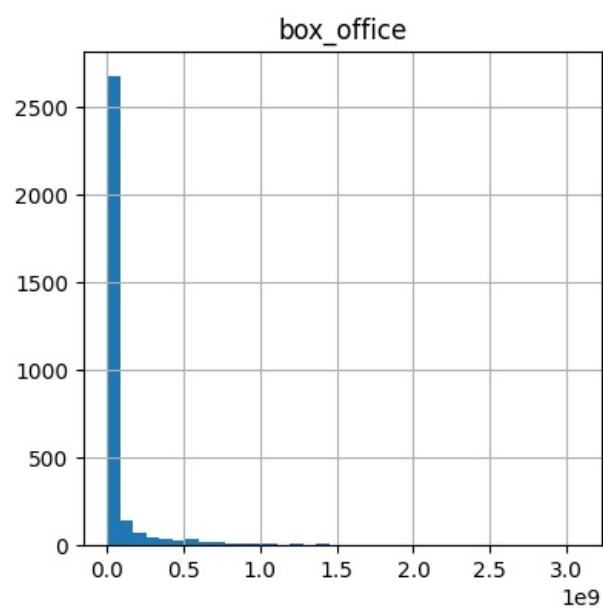
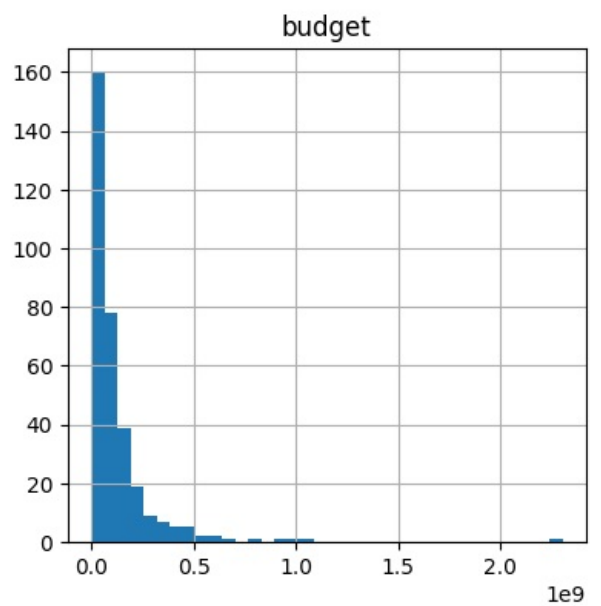
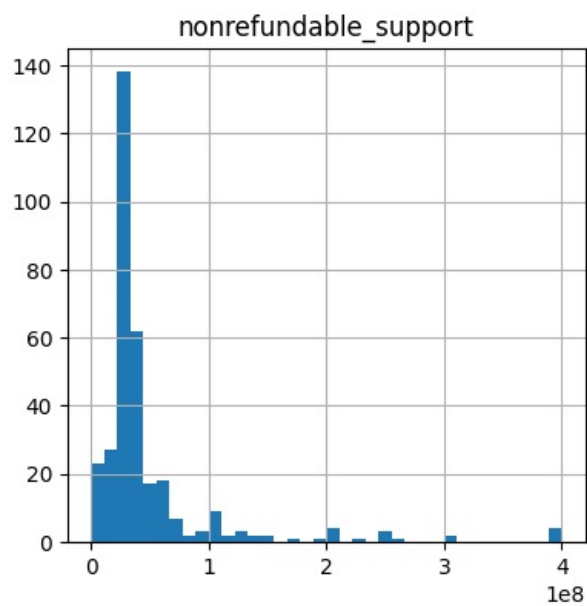
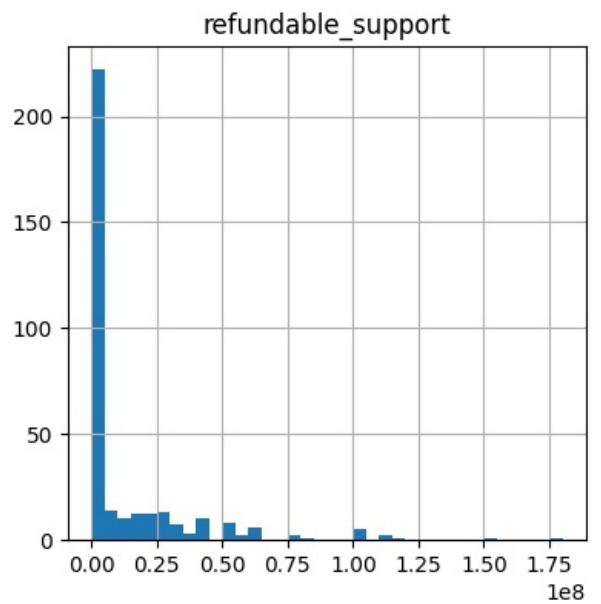
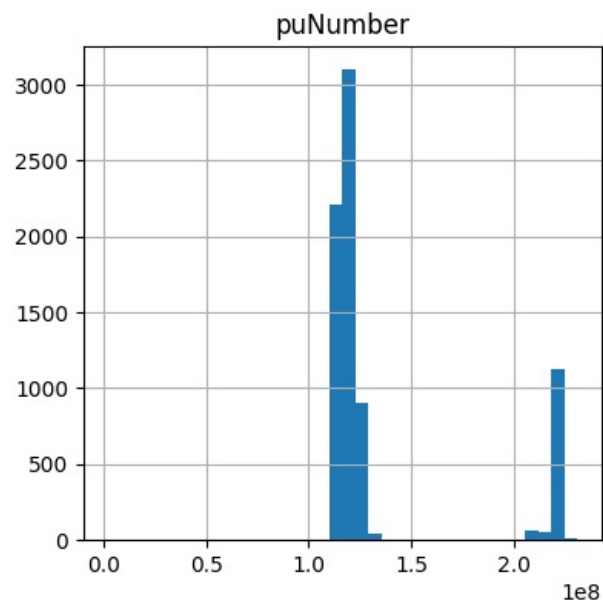
```

Произошло успешное слияние данных. Дальнейшая работа будет осуществляться над датасетом `data`

In [11]: `data.head()`

	title	puNumber	show_start_date	type	film_studio	production_country	director	producer	age_restriction	ref
0	Открытый простор	221,048,915.00	2015-11-27T12:00:00.000Z	Художественный	Тачстоун Пикчерз, Кобальт Пикчерз, Бикон Пикче...	США	Кевин Костнер	Дэвид Валдес, Кевин Костнер, Джейк Эбертс	«18+» - запрещено для детей	
1	Особо важное задание	111,013,716.00	2016-09-13T12:00:00.000Z	Художественный	Киностудия "Мосфильм"	СССР	Е.Матвеев	NaN	«6+» - для детей старше 6 лет	
2	Особо опасен	221,038,416.00	2016-10-10T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Тимур Бекмамбетов	Джим Лемли, Джейсон Нетер, Марк Е.Платт, Яйн Смит	«18+» - запрещено для детей	
3	Особо опасен	221,026,916.00	2016-06-10T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Тимур Бекмамбетов	Джим Лемли, Джейсон Нетер, Марк Е.Платт, Яйн Смит	«18+» - запрещено для детей	
4	Особо опасен	221,030,815.00	2015-07-29T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Тимур Бекмамбетов	Джим Лемли, Джейсон Нетер, Марк Е.Платт, Яйн Смит	«18+» - запрещено для детей	

In [12]: `# построим гистограммы для каждого числового столбца`  
`data.hist(figsize = (10,15), bins = 36);`



Видны некоторые аномалии, возвратная и невозвратная части бюджета имеют показатели около 0, так же как и данные общего бюджета, возможно это типичные значения для указанных столбцов или ошибочные значения

## Предобработка данных

### Пропуски

In [13]: # запишем в переменную количество строк до обработки

```
start_row, a = data.shape
```

```
In [14]: # исследуем количество пропусков
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7486 entries, 0 to 7485
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   title                  7486 non-null   object
1   puNumber               7485 non-null   float64
2   show_start_date        7486 non-null   object
3   type                   7486 non-null   object
4   film_studio            7468 non-null   object
5   production_country      7484 non-null   object
6   director               7477 non-null   object
7   producer               6918 non-null   object
8   age_restriction        7486 non-null   object
9   refundable_support     332 non-null    float64
10  nonrefundable_support  332 non-null    float64
11  budget                 332 non-null    float64
12  financing_source        332 non-null    object
13  ratings                6519 non-null   object
14  genres                 6510 non-null   object
15  box_office              3158 non-null   float64
dtypes: float64(5), object(11)
memory usage: 994.2+ KB
```

```
In [15]: def null_value(data):
        try:
            (
                (data.isna().mean()*100)
                .to_frame()
                .rename(columns = {0: 'space'})
                .query('space > 0')
                .sort_values(by = 'space', ascending = True)
                .plot(kind= 'barh', figsize=(20,5), rot = 0, legend = False, fontsize = 16)
                .set_title('Пропуски данных' + "\n", fontsize = 22, color = 'SteelBlue')
            );
        except:
            print('Пропусков нет')
```

```
In [16]: # визуализируем количество пропусков
null_value(data)
```



Внушительное количество пропусков наблюдается в столбце `financing_source`, и связанных с ним столбцов `budget` и `nonrefundable_support` (возвратной и невозвратной), что является ожидаемым результатом исходя из специфики анализируемой информации. Данные о кассовых сборах есть лишь у менее половины фильмов. Пропуски в столбца `ratings` и `genres` составляют менее 15%. Остальные столбцы не имеют или имеют минимальные пропуски в данных. Некоторые пропуски можно заполнить синтетическими данными.

```
In [17]: # найдем и посчитаем количество явных дубликатов, 0 - явных дубликатов нет.
data.duplicated().sum()
```

```
Out[17]: 0
```

- Заменяем пропуски в столбце `financing_source`

```
In [18]: # выведем список уникальных значений столбца
data['financing_source'].unique()
```

```
Out[18]: array([nan, 'Министерство культуры', 'Фонд кино',
                'Министерство культуры, Фонд кино'], dtype=object)
```

Очевидно пропуски связаны с отсутствием финансирования посредством госструктур. Заменяем пропуски на "Иное"

финансирование"

```
In [19]: data['financing_source'] = data['financing_source'].fillna('Иное финансирование')
```

- Пропуски в столбце `genres`

Пропуски в столбце можно заменить с поправкой на тип фильма и возрастные ограничения.

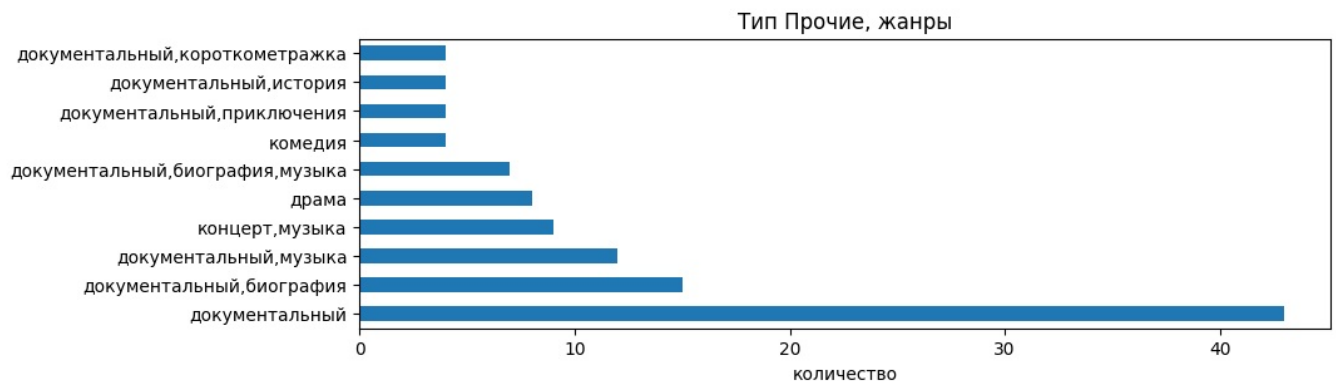
```
In [20]: data['type'].unique()
```

```
Out[20]: array(['Художественный', 'Анимационный', 'Прочие', 'Документальный',  
        'Научно-популярный', 'Художественный', 'Анимационный',  
        'Музыкально-развлекательный'], dtype=object)
```

```
In [21]: # есть ошибки, исправим правописание  
data.loc[data['type'] == " Художественный", 'type'] = 'Художественный'  
data.loc[data['type'] == " Анимационный", 'type'] = 'Анимационный'
```

```
In [22]: #если тип фильма анимационный, можно с уверенностью заполнить пропуски общим словом "мультфильм"  
data.loc[(data['genres'].isna())&(data['type'] == "Анимационный"), 'genres'] = 'мультфильм'
```

```
In [23]: # проверим какие категории жанра чаще всего встречаются в тип "Прочие"  
data.query('type == "Прочие")')['genres'].value_counts().head(10).plot(kind = 'barh', figsize = (10,3))  
plt.xlabel('количество')  
plt.title('Тип Прочие, жанры');
```

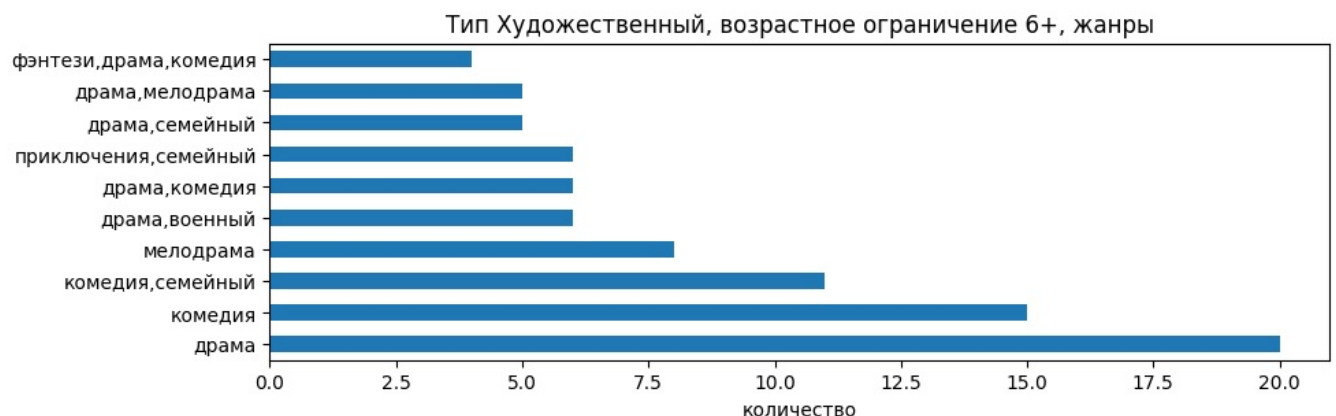


```
In [24]: # заменим пропуски в типе фильмов "Прочие" на подходящий жанр - "документальный"  
data.loc[(data['genres'].isna())&(data['type'] == "Прочие"), 'genres'] = 'документальный'
```

Пропуски в типе 'Художественный' в связи с жанровым разнообразием стоит заполнить с поправкой на возрастной ценз.

```
In [25]: # для типа "художественный" и возрастного ценза - для любой аудитории заменим пропуски на жанр "семейный"  
data.loc[(data['genres'].isna())&(data['type'] == "Художественный")&  
        (data['age_restriction'] == "«0+» - для любой зрительской аудитории"), 'genres'] = 'семейный'
```

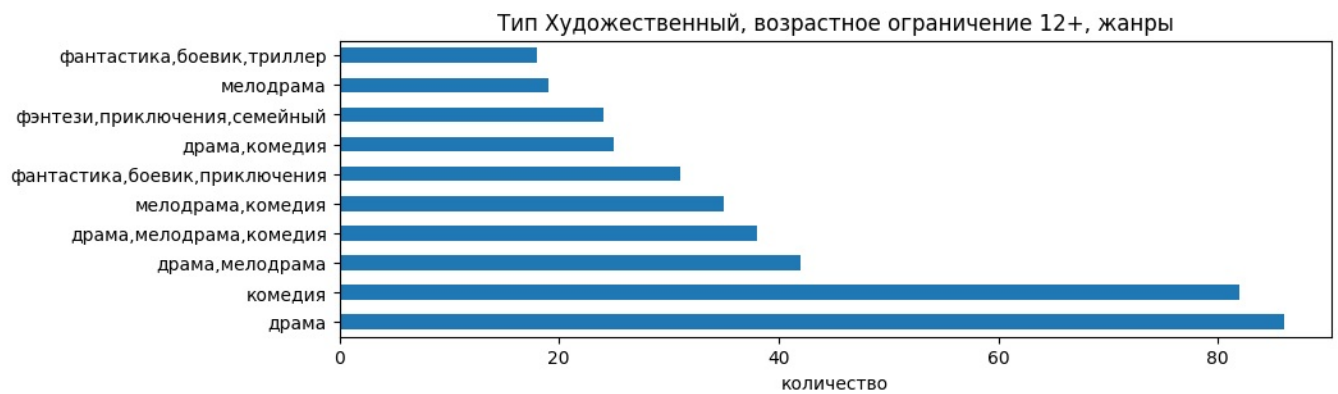
```
In [26]: # посмотрим типичные жанры для типа "Художественный" и с возрастным ограничением старше 6 лет  
data.query('type == "Художественный" and age_restriction == "«6+» - для детей старше 6 лет")')['genres'].value_counts().plot(kind = 'barh',figsize = (10,3))  
plt.xlabel('количество')  
plt.title('Тип Художественный, возрастное ограничение 6+, жанры');
```



```
In [27]: # для типа "художественный" и возрастного ценза - для детей старше 6 лет заменим пропуски на жанр "драма, комедия"  
data.loc[(data['genres'].isna())&(data['type'] == "Художественный")&  
        (data['age_restriction'] == "«6+» - для детей старше 6 лет"), 'genres'] = 'драма, комедия'
```

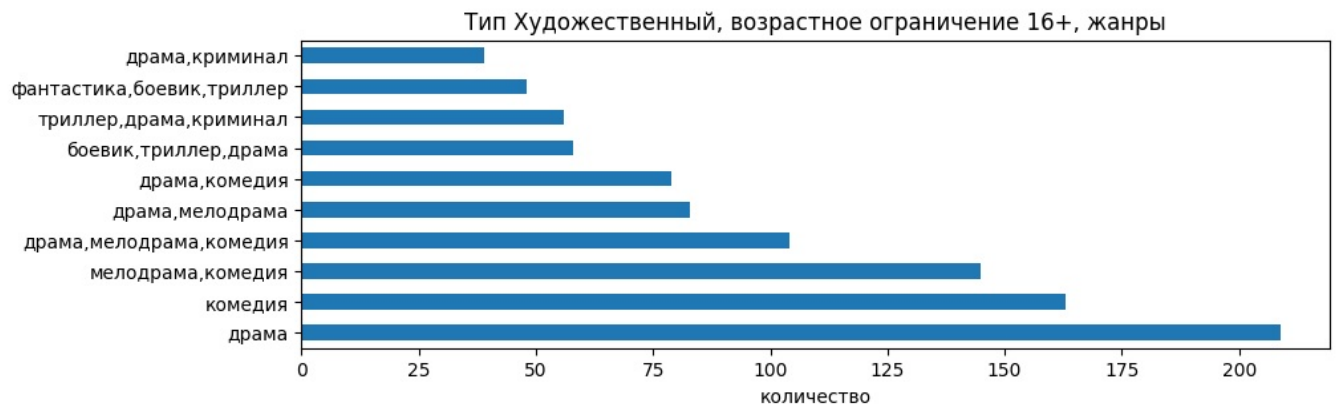
```
In [28]: # посмотрим типичные жанры для типа "Художественный" и возрастного ограничения старше 12 лет
```

```
data.query('type == "Художественный" and age_restriction == "«12+» - для детей старше 12 лет")['genres'].value_counts().head(10).plot(kind = 'barh',figsize = (10,3))
plt.xlabel('количество')
plt.title('Тип Художественный, возрастное ограничение 12+, жанры');
```



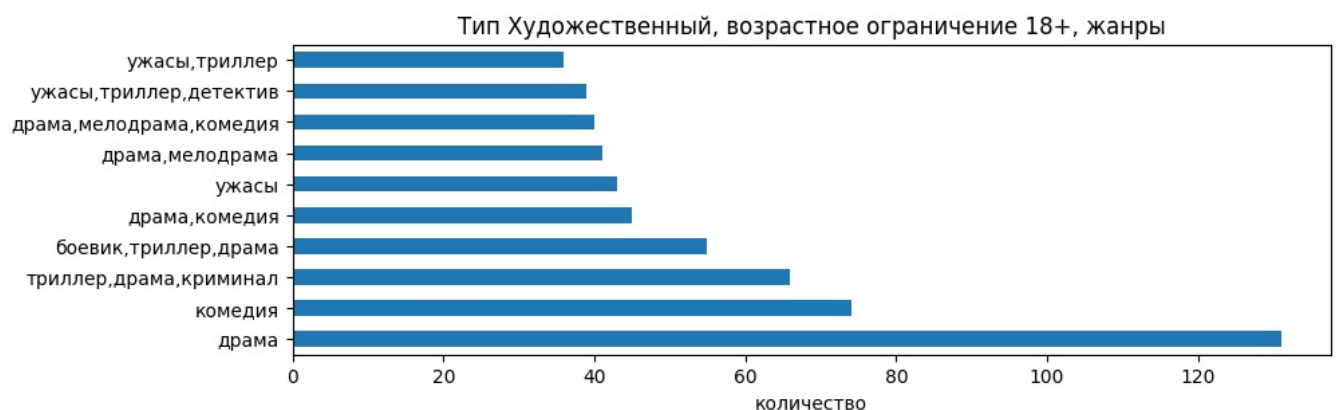
```
In [29]: # для типа "художественный" и возрастного ценза - для детей старше 12 лет заменим пропуски на жанр "драма, коме
data.loc[(data['genres'].isna())&(data['type'] == "Художественный")&\
        (data['age_restriction'] == "«12+» - для детей старше 12 лет"),'genres'] = 'драма, комедия'
```

```
In [30]: # посмотрим типичные жанры для типа "Художественный" и возрастного ограничения 16+ лет
data.query('type == "Художественный" and age_restriction == "«16+» - для детей старше 16 лет")['genres'].value_counts().head(10).plot(kind = 'barh',figsize = (10,3))
plt.xlabel('количество')
plt.title('Тип Художественный, возрастное ограничение 16+, жанры');
```



```
In [31]: # для типа "художественный" и возрастного ценза - для детей старше 16 лет заменим пропуски на жанр "драма, коме
data.loc[(data['genres'].isna())&(data['type'] == "Художественный")&\
        (data['age_restriction'] == "«16+» - для детей старше 16 лет"),'genres'] = 'драма, комедия'
```

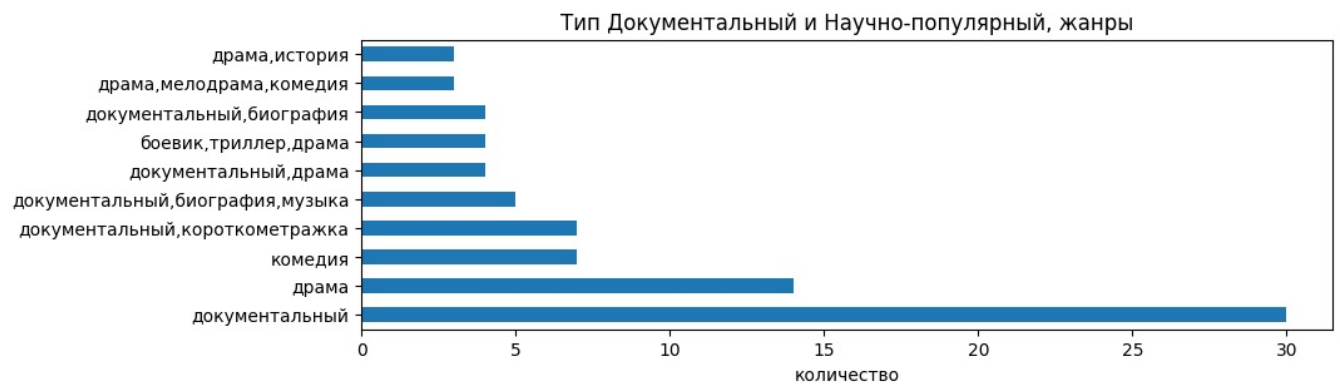
```
In [32]: # посмотрим типичные жанры для типа "Художественный" и запрещенного для детей, 18+
data.query('type == "Художественный" and age_restriction == "«18+» - запрещено для детей")['genres'].value_counts().head(10).plot(kind = 'barh',figsize = (10,3))
plt.xlabel('количество')
plt.title('Тип Художественный, возрастное ограничение 18+, жанры');
```



```
In [33]: # для типа "художественный" и возрастного ценза - 18+ заполним пропуски на жанр "драма"
data.loc[(data['genres'].isna())&(data['type'] == "Художественный")&\
        (data['age_restriction'] == "«18+» - запрещено для детей"),'genres'] = 'драма'
```

```
In [34]: data.query('type == "Документальный" or type == "Научно-популярный")['genres'].value_counts().head(10).plot(kind = 'barh',figsize = (10,3))
plt.xlabel('количество')
```

```
plt.title('Тип Документальный и Научно-популярный, жанры');
```



```
In [35]: # для типа "документальный" и 'научно-популярный'заполним пропуски на жанр "документальный"
data.loc[(data['genres'].isna())&((data['type'] == "Документальный")|(data['type'] == "Научно-популярный")), 'genres'] = 'документальный'
```

- Удалим пропуски в столбцах `production_country` и `puNumber`.

```
In [36]: data = data.dropna(subset = ['production_country'])
data = data.dropna(subset = ['puNumber'])
```

- Рассмотрим пропуски в столбце `producer` в зависимости от страны-производителя

```
In [37]: data.query('producer.isna()')['production_country'].value_counts().head(5)
```

```
Out[37]: СССР          342
Россия             37
Великобритания     33
США                24
СССР              20
Name: production_country, dtype: int64
```

Пропуски в некоторых данных скорее всего связаны с годом производства фильмов. Например пропуски в имени продюсера в СССР связаны с отсутствием одного как отдельного исполняющего. Режиссер исполнял функции продюсирования, которые как отдельный вид деятельности зародились лишь в 90-е 20 века уже в постсоветское время. Произведем заполнение пропусков в столбце продюсер для страны-производителя СССР (или совместно с СССР) именем режиссера.

```
In [38]: # произведем заполнение пропусков, возьмем в расчет ошибочное правописание
data.loc[data["production_country"].str.contains("СССР","СССР")\
&(data['producer'].isna()), 'producer'] = data['director']
```

Вывод:

В данных `financing_source` пропуски связаны с отсутствием категории иного финансирования, соответственно добавлена новая категория "Иное финансирование". Пропуски в столбце `genres` заполнены типичными жанрами для определенного типа, в случае с типом "Художественный" проведено дополнительное ранжирование по возрастному цензу. Столбец `box_office` содержит данные о сборах в рублях. Оценить предполагаемый сбор невозможно, пропуски остаются. Данные `rating` предполагают потребительскую оценку и не связаны с другими данными. Имеющиеся пропуски в столбцах `production_country` и `puNumber` удалены ввиду малого количества

## Изменение типов данных

```
In [39]: # приведем тип столбца к datetime
data['show_start_date'] = pd.to_datetime(data['show_start_date'])
```

```
In [40]: # сменим тип данных на строковый для удобства работы
data['production_country'] = data['production_country'].astype('str')
data['genres'] = data['genres'].astype('str')
data['director'] = data['director'].astype('str')
```

```
In [ ]:
```

```
In [41]: # проверим уникальные значения столбца ratings
data['ratings'].unique()
```



```
Out[41]: array(['7.2', '6.6', '6.8', '7.7', '8.3', '8.0', '7.8', '8.1', '7.1',
               '6.0', '7.4', '5.8', '8.7', '6.3', '6.9', '5.0', '4.3', '7.3',
               '7.0', '6.4', nan, '8.2', '7.5', '6.7', '7.9', '5.9', '6.2', '5.6',
               '6.5', '2.4', '7.6', '6.1', '8.6', '8.5', '8.8', '5.5', '5.1',
               '5.7', '5.4', '99%', '4.4', '4.5', '5.3', '4.1', '8.4', '2.6',
               '3.8', '4.6', '4.8', '4.0', '3.0', '1.6', '4.2', '5.2', '4.7',
               '4.9', '3.9', '2.7', '3.3', '2.9', '28%', '3.7', '1.4', '3.1',
               '97%', '3.5', '3.2', '2.8', '1.5', '2.1', '2.5', '9.2', '3.4',
               '1.1', '3.6', '83%', '64%', '91%', '94%', '62%', '79%', '90%',
               '19%', '88%', '1.0', '89%', '1.3', '1.9', '1.8', '1.2', '1.7',
               '9.0', '98%', '8.9', '9.1'], dtype=object)
```

Рейтинг имеет некоторые строковые значения с символом процента, указывающего на формат. Переведем значения процентов в баллы и преобразуем ошибочную запись в числовой формат.

```
In [42]: # в столбце rating имеются строки, напомним функцию для перевода в числовой формат
def converts(digit):
    try:
        digit = float(digit)
        return digit
    except:
        digit = float(digit[0] + '.' + digit[1])
        return digit
```

```
In [43]: # сменим формат столбца
data['ratings'] = data['ratings'].apply(converts)
```

### Неявные дубликаты

Рассмотрим категориальные данные датасета

- Столбец `title`.

Посчитаем количество уникальных значений

```
In [44]: len(data['title'].unique())
```

```
Out[44]: 6769
```

Столбец имеет некоторое количество повторений в названии фильма, что совершенно нормально при разных номерах прокатных удостоверений.

- В столбце `production_country` находятся данные о стране-происхождении киноленты, проведем работу по выявлению неявных дубликатов.

```
In [45]: # посчитаем количество значений
data['production_country'].nunique()
```

```
Out[45]: 950
```

```
In [46]: # выведем список уникальных значений, отсортировав по длине строки
sorted(data['production_country'].unique(), key = len)
```

```
Out[46]: ['США',
          'КНР',
          'ЮАР',
          'США',
          'США',
          'СССР',
          'Чили',
          'Иран',
          'США',
          'Перу',
          '2019',
          'Чехия',
          'Дания',
          'Индия',
          'СССР',
          'СССР',
          'Иран',
          'Корея',
          'Китай',
          'Япония',
          'Италия',
          'Россия',
          'Швеция',
          'Канада',
```

'Турция',  
'СССР',  
'СССР',  
'Грузия',  
'Греция',  
'Польша',  
'Франция',  
'Австрия',  
'Гонконг',  
'Испания',  
'Украина',  
'США-ОАЭ',  
'США-ЮАР',  
'Израиль',  
'Италия',  
'Армения',  
'Бельгия',  
'Таиланд',  
'Мексика',  
'СССР',  
'Венгрия',  
'США-КНР',  
'Румыния',  
'Эстония',  
'КНР-США',  
'Германия',  
'Ирландия',  
'Бразилия',  
'Норвегия',  
'Норвегия',  
'Киргизия',  
'Парагвай',  
'Франция',  
'СССР',  
'СССР',  
'Хорватия',  
'Исландия',  
'Малайзия',  
'Монголия',  
'Болгария',  
'КНР, США',  
'Колумбия',  
'Пакистан',  
'США - ОАЭ',  
'Аргентина',  
'Австралия',  
'США - ЮАР',  
'Швейцария',  
'США - КНР',  
'Индонезия',  
'США-Китай',  
'Финляндия',  
'СССР',  
'Венесуэла',  
'ЮАР - США',  
'СССР, ГДР',  
'КНР - США',  
'ОАЭ - США',  
'СССР, ВНР',  
'США-Индия',  
'Голландия',  
'Казахстан',  
'СССР, ФРГ',  
'Ливан-США',  
'США-Канада',  
'Нидерланды',  
'Канада-ЮАР',  
'США-Россия',  
'США-Мальта',  
'США - Чили',  
'Канада-США',  
'Россия-США',  
'США-Япония',  
'Португалия',  
'Китай, США',  
'США, Индия',  
'Италия-США',  
'КНР - США',  
'Южная Корея',  
'США-Франция',  
'Швеция-Куба',  
'США - Индия',  
'США-Испания',

'Франция-США',  
'Румыния-США',  
'США-Мексика',  
'Россия-Кипр',  
'Индия - США',  
'США - Китай',  
'Франция-КНР',  
'США - Дания',  
'Индия - КНР',  
'США-Бельгия',  
'Дания - США',  
'США-Венгрия',  
'США, Япония',  
'Япония, США',  
'Канада, США',  
'США, Россия',  
'Сша, Канада',  
'США, Канада',  
'Чехословакия',  
'США-Германия',  
'СССР, Швеция',  
'СССР, Италия',  
'Гонконг, КНР',  
'СССР, Польша',  
'США - Канада',  
'Канада - США',  
'Франция-Перу',  
'Германия-США',  
'Канада - КНР',  
'США - Япония',  
'США - Россия',  
'США- Ю.Корея',  
'Швеция - США',  
'США-Ирландия',  
'Турция - США',  
'СССР, Россия',  
'США - Италия',  
'Италия - США',  
'США-Колумбия',  
'Россия - США',  
'США, Франция',  
'США - Мальта',  
'Япония - США',  
'США - Швеция',  
'Россия-Китай',  
'США, Украина',  
'СССР, Венгрия',  
'США - Франция',  
'Франция - США',  
'США-Австралия',  
'США - Бельгия',  
'США - Испания',  
'КНР - Гонконг',  
'США-Индонезия',  
'Гонконг - КНР',  
'Франция-Индия',  
'США -Германия',  
'США - Украина',  
'США - Мексика',  
'Бельгия - США',  
'Испания - США',  
'СССР, Албания',  
'Мексика - США',  
'Франция-Китай',  
'Грузия-Россия',  
'Китай-Гонконг',  
'США - Вьетнам',  
'Иран, Франция',  
'США - Гонконг',  
'Россия-Италия',  
'СССР - Италия',  
'США-Аргентина',  
'Россия-Польша',  
'Австралия-США',  
'Таиланд - США',  
'Германия, США',  
'США, Гонконг',  
'Франция - КНР',  
'США, Германия',  
'КНР - Франция',  
'Новая Зеландия',  
'Великобритания',  
'США - Германия',

'Канада-Франция',  
'Германия - США',  
'Мексика - Чили',  
'США-Нидерланды',  
'Швеция - Дания',  
'Испания - Куба',  
'Дания - Швеция',  
'Франция-Канада',  
'Малайзия - США',  
'Чили - Испания',  
'Франция-Польша',  
'Бразилия - США',  
'Франция - Чили',  
'Россия-Украина',  
'Франция-Россия',  
'СССР, Монголия',  
'Россия, Италия',  
'КНР - Сингапур',  
'Австралия, США',  
'Армения-Россия',  
'Грузия-Франция',  
'Россия-Франция',  
'Италия-Франция',  
'США - Норвегия',  
'Испания-Канада',  
'США-ФРГ-Россия',  
'США - Колумбия',  
'Германия-Кения',  
'Франция-Италия',  
'Швеция-Франция',  
'Болгария - США',  
'Норвегия - США',  
'Россия, Сербия',  
'Ирландия - США',  
'Колумбия - США',  
'СССР, Финляндия',  
'Франция-Бельгия',  
'Франция-Испания',  
'Австралия - США',  
'Япония - Канада',  
'Бельгия-Франция',  
'США - Австралия',  
'Индия - Мексика',  
'Россия - Италия',  
'Греция - Россия',  
'Италия - Россия',  
'Россия-Ирландия',  
'Россия, Румыния',  
'Республика Кипр',  
'США - Финляндия',  
'Россия, Испания',  
'Россия, Украина',  
'Канада-Испания',  
'Китай - Гонконг',  
'Испания-Мексика',  
'Россия-Германия',  
'Россия - Польша',  
'Великобритания',  
'Франция-Украина',  
'Латвия - Россия',  
'Канада - Италия',  
'Литва - Венгрия',  
'Италия - Канада',  
'Франция - Дания',  
'Германия-Швеция',  
'Канада-Германия',  
'Киргизия-Россия',  
'Россия, Армения',  
'Италия, Франция',  
'Россия-Беларусь',  
'Россия, Австрия',  
'Кения, Германия',  
'Россия - Грузия',  
'Франция, Канада',  
'Италия, Испания',  
'Бельгия, Канада',  
'Франция - Италия',  
'Чехия - Германия',  
'Италия - Франция',  
'Германия-Австрия',  
'Испания - Канада',  
'Канада - Франция',  
'Франция - Россия',

'Франция - Канада',  
'Канада - Испания',  
'Ирландия, Канада',  
'Грузия - Украина',  
'СССР, Афганистан',  
'Эстония - Россия',  
'Россия, Германия',  
'Испания-Колумбия',  
'Франция-Германия',  
'Польша - Франция',  
'Россия, Испания',  
'Франция -Бельгия',  
'Франция - Монако',  
'Дания - Германия',  
'Германия-Израиль',  
'Россия-Казахстан',  
'Испания - Мальта',  
'Италия-Швейцария',  
'Германия - Китай',  
'Республика Корея',  
'Франция - Греция',  
'Канада - Мексика',  
'Франция, Бельгия',  
'Дания - Исландия',  
'Латвия - Франция',  
'Италия, Германия',  
'Израиль, Украина',  
'Германия, Россия',  
'Франция - Испания',  
'Франция - Бельгия',  
'США - Индия - ОАЭ',  
'Испания - Франция',  
'Южная Корея - США',  
'Мексика - Франция',  
'Израиль - Франция',  
'Норвегия - Швеция',  
'Германия - Россия',  
'США - Южная Корея',  
'Франция - Австрия',  
'Германия - Италия',  
'Германия - Канада',  
'Швеция - Германия',  
'Канада - Германия',  
'Мексика - Испания',  
'Южная Корея - КНР',  
'Китай-Гонконг-США',  
'Польша - Ирландия',  
'Россия - Германия',  
'Россия - Болгария',  
'Мексика-Аргентина',  
'Франция-Австралия',  
'Швейцария-Франция',  
'Россия-Белоруссия',  
'Бельгия - Франция',  
'Испания - Мексика',  
'США-Гонконг-Китай',  
'США-Канада-Италия',  
'Швейцария-Австрия',  
'Франция-Аргентина',  
'США-Италия-Греция',  
'Испания-Аргентина',  
'Исландия, Бельгия',  
'Франция, Германия',  
'США - Пуэрто Рико',  
'Швеция - Норвегия',  
'Аргентина-Испания',  
'Германия, Бельгия',  
'Австралия, Канада',  
'Германия, Франция',  
'Франция, Норвегия',  
'Канада - Норвегия',  
'Финляндия, Латвия',  
'Россия - Казахстан',  
'Великобритания-США',  
'Германия - Франция',  
'США-Великобритания',  
'Франция - Германия',  
'США-Канада-Бельгия',  
'США-Италия-Испания',  
'Бельгия-Нидерланды',  
'Азербайджан-Россия',  
'Финляндия - Канада',  
'Франция - Бельгия',

'Россия - Азербайджан',  
'Бразилия - Испания',  
'Уругвай - Колумбия',  
'Армения - Германия',  
'Франция - Бразилия',  
'США - Канада - КНР',  
'США-Франция-Канада',  
'Австрия - Германия',  
'Франция - Бельгия',  
'Австралия-Ирландия',  
'Германия - Австрия',  
'Бразилия - Канада',  
'США-Канада-Франция',  
'Япония-США-Франция',  
'КНР - Канада - США',  
'Финляндия - Польша',  
'Германия - Бельгия',  
'Китай, Канада, США',  
'Аргентина, Испания',  
'Республика Армения',  
'США-Франция-Испания',  
'КНР - Гонконг - США',  
'Республика Беларусь',  
'США, Великобритания',  
'Испания - Аргентина',  
'Франция - Швейцария',  
'Армения - Казахстан',  
'США - Франция - ЮАР',  
'Германия - Исландия',  
'Аргентина - Испания',  
'Россия, Таджикистан',  
'Франция - Македония',  
'Швейцария - Франция',  
'Израиль-США-Франция',  
'США-Германия-Канада',  
'Швейцария - Австрия',  
'Нидерланды - Россия',  
'США - ЮАР - Гонконг',  
'Великобритания, США',  
'Германия, Австралия',  
'Нидерланды, Бельгия',  
'Таиланд - КНР - США',  
'Франция - Финляндия',  
'США - Великобритания',  
'Великобритания - США',  
'Новая Зеландия - США',  
'Аргентина - Бразилия',  
'США - Новая Зеландия',  
'Португалия - Франция',  
'Швеция-Франция-Дания',  
'Чили - Франция - США',  
'Великобритания - ЮАР',  
'Канада - Южная Корея',  
'Франция - Португалия',  
'Исландия - Финляндия',  
'Австралия-Турция-США',  
'Австралия - Германия',  
'Украина - Нидерланды',  
'США - Гонконг - КНР',  
'Австралия - Ирландия',  
'США - Германия - КНР',  
'Республика Казахстан',  
'Германия - Австралия',  
'США-Франция-Ирландия',  
'Литва-Россия-Украина',  
'США-Канада-Австралия',  
'Россия - Азербайджан',  
'Германия - Швейцария',  
'Австрия - Люксембург',  
'Великобритания - КНР',  
'Франция - Нидерланды',  
'Австралия - Колумбия',  
'Финляндия - Германия',  
'Франция - Люксембург',  
'Бельгия - Люксембург',  
'США, Франция, Япония',  
'Новая Зеландия - КНР',  
'Япония-Великобритания',  
'Республика Узбекистан',  
'США - Великобритания',  
'США - Чехия - Франция',  
'Италия-Великобритания',  
'Франция - Дания - США',

'Канада-Великобритания',  
'Россия-Словакия-Чехия',  
'Чехия - Испания - США',  
'Канада, США, Норвегия',  
'Бразилия - США - КНР',  
'Норвегия - Нидерланды',  
'США - Франция - Турция',  
'Россия, Казахстан, США',  
'Франция-Великобритания',  
'Бельгия-Франция-Италия',  
'США - Россия - Франция',  
'Индия - Великобритания',  
'Великобритания, Канада',  
'Швейцария-Германия-ЮАР',  
'Великобритания-Франция',  
'СССР, Румыния, Франция',  
'Франция-Бельгия-Канада',  
'Великобритания - Дания',  
'Канада-Бразилия-Япония',  
'Великобритания-Венгрия',  
'США-Германия-Австралия',  
'Швейцария - Люксембург',  
'Канада-Южная Корея-США',  
'Канада, Великобритания',  
'Япония, Великобритания',  
'Республика Кипр-Россия',  
'Германия, Франция, США',  
'Германия-Великобритания',  
'Великобритания - Италия',  
'Франция - Мексика - США',  
'Германия - Италия - США',  
'США - Канада - Германия',  
'Германия-Франция-Польша',  
'США - Австралия - Дания',  
'Россия, Франция, Латвия',  
'США - Испания - Франция',  
'Ирландия-Великобритания',  
'Россия, Грузия, Испания',  
'Россия, Украина, Польша',  
'Великобритания, Франция',  
'Россия, Республика Кипр',  
'Великобритания - Канада',  
'Мексика - Тайвань - США',  
'США - Германия - Канада',  
'Россия-Украина-Германия',  
'Франция-Германия-Италия',  
'Франция - Италия - Иран',  
'США - Япония - Германия',  
'Бразилия - Канада - США',  
'Франция - США - Бельгия',  
'США - Германия - Япония',  
'Великобритания - Польша',  
'США - Австралия - Индия',  
'Норвегия, Швеция, Дания',  
'Республика Кипр, Россия',  
'Франция, Новая Зеландия',  
'Великобритания - Франция',  
'Великобритания - Испания',  
'США-Великобритания-Индия',  
'Испания - США - Колумбия',  
'Австрия-Германия-Франция',  
'Австралия-Великобритания',  
'США - Канада - Индонезия',  
'США - Испания - Болгария',  
'Польша - Италия - Россия',  
'СССР, Венгрия, ЧССР, ГДР',  
'Россия - Республика Кипр',  
'США - Колумбия - Испания',  
'Франция - Великобритания',  
'США - Германия - Франция',  
'Франция - Германия - США',  
'Чехия-Великобритания-США',  
'США - Франция - Германия',  
'Австрия-Франция-Германия',  
'Франция-Гонконг-Ирландия',  
'Франция-Испания-Германия',  
'Великобритания - Румыния',  
'Франция-Германия-Австрия',  
'Испания - Болгария - США',  
'Испания, Франция, Италия',  
'США - Канада - Аргентина',  
'США - Болгария - Мексика',  
'Великобритания - Израиль',

'Ирландия, Великобритания',  
'Республика Кипр - Россия',  
'КНР, Индия, Гонконг, США',  
'Великобритания - Германия',  
'Великобритания - Ирландия',  
'Испания - Канада - Япония',  
'США - Швейцария - Франция',  
'США - Австралия - Франция',  
'Дания - Германия - Швеция',  
'Уругвай-Мексика-Германия',  
'Германия - Великобритания',  
'Норвегия - Дания - Швеция',  
'Франция - Чехия - Бельгия',  
'Франция - Бельгия - Чехия',  
'Франция-Испания-Швейцария',  
'США - Австралия - Мексика',  
'Германия-Италия-Швейцария',  
'Великобритания, Австралия',  
'Италия, Франция, Германия',  
'Россия, Польша, Финляндия',  
'Канада - Франция - Испания',  
'СССР - Швейцария - Франция',  
'Исландия-Ирландия-Германия',  
'Германия-Италия - Шри-Ланка',  
'Франция - Канада - Бельгия',  
'СССР, ФРГ, Западный Берлин',  
'Южная Корея - США - Канада',  
'Испания - Италия - Франция',  
'Турция - Германия - Франция',  
'США - Великобритания - КНР',  
'Франция - Бельгия - Канада',  
'Япония - КНР - Южная Корея',  
'Великобритания-США-Франция',  
'Швейцария - Великобритания',  
'Франция - Бельгия - Япония',  
'США-Великобритания-Франция',  
'Швеция - Дания - Финляндия',  
'Франция - Польша - Бельгия',  
'Норвегия - Швеция - Россия',  
'Италия - Бельгия - Франция',  
'Россия, Бельгия, Финляндия',  
'Греция - Германия - Франция',  
'США-Великобритания-Ирландия',  
'Румыния - Франция - Бельгия',  
'Болгария - Франция - Израиль',  
'Россия, Республика Беларусь',  
'Франция - Ирландия - Швеция',  
'Сербия - Германия - Венгрия',  
'Бельгия - Италия - Франция',  
'Венгрия - Германия - Швеция',  
'Франция - Бразилия - Италия',  
'Великобритания - Люксембург',  
'США - Канада - Япония - КНР',  
'Франция - Бельгия - Испания',  
'Княжество Андорра - Украина',  
'Германия - Дания - Норвегия',  
'Франция - Испания - Тайвань',  
'Германия - Италия - Франция',  
'США - Германия - Нидерланды',  
'Германия - Франция - Канада',  
'Германия-Украина-Нидерланды',  
'США-Германия-Великобритания',  
'Россия - Германия - Украина',  
'Германия - Нидерланды - ЮАР',  
'Германия - Франция - Польша',  
'Италия - Франция - Германия',  
'Франция - Германия - Италия',  
'Чехия - Словакия - Хорватия',  
'Великобритания, Швеция, США',  
'США - КНР - Индия - Ю.Корея',  
'Великобритания, Япония, США',  
'Великобритания, Канада, США',  
'Франция - Сенегал - Бельгия',  
'США - Великобритания - Чехия',  
'США-Италия-Испания-Аргентина',  
'Франция - Италия - Швейцария',  
'Австрия - Германия - Франция',  
'Финляндия - Австрия - Россия',  
'Германия - Франция - Австрия',  
'Франция - Германия - Бельгия',  
'Франция-Великобритания-Чехия',  
'Великобритания - Пуэрто-Рико',  
'Великобритания - США - Индия',



'Дания - Великобритания - ЮАР',  
'Россия-США-Канада-Люксембург',  
'Италия - Франция - Швейцария',  
'Германия - Франция - Бельгия',  
'США - Пуэрто-Рико - Франция',  
'СССР - ГДР - Польша - Италия',  
'Франция - Германия - Австрия',  
'США-Великобритания-Австралия',  
'Бельгия, Великобритания, США',  
'Великобритания, США, Испания',  
'Германия - Австрия - Франция',  
'США, Великобритания, Франция',  
'Франция - Израиль - Германия',  
'Франция, Люксембург, Бельгия',  
'Франция, Бельгия, Люксембург',  
'Республика Кипр, США, Россия',  
'Грузия - Франция - Люксембург',  
'США - Великобритания - Канада',  
'США - Канада - Великобритания',  
'Испания - Уругвай - Аргентина',  
'Великобритания - США - Россия',  
'Финляндия - Исландия - Швеция',  
'Италия-Франция-Великобритания',  
'Германия - Австрия - Ирландия',  
'Аргентина - Франция - Испания',  
'Великобритания - Канада - США',  
'Финляндия - Швеция - Германия',  
'Финляндия - Швеция - Норвегия',  
'Сербия - Великобритания - США',  
'США - Великобритания - Япония',  
'Великобритания - США - Канада',  
'Россия - Германия - Швейцария',  
'США - Новая Зеландия - Япония',  
'Дания-Швеция-Франция-Германия',  
'США - Великобритания - Швеция',  
'Дания - Латвия - Россия - США',  
'Италия - Швейцария - Германия',  
'Аргентина - Испания - Уругвай',  
'Россия, Нидерланды, Финляндия',  
'Германия - Люксембург - Дания',  
'Великобритания - Япония - США',  
'Великобритания-Франция-Италия',  
'Уругвай - Аргентина - Испания',  
'Великобритания - Россия - КНР',  
'Великобритания - Франция - США',  
'США - Франция - Великобритания',  
'Франция - Бельгия - Люксембург',  
'США - Венгрия - Великобритания',  
'Великобритания-Австралия - США',  
'Финляндия - Франция - Германия',  
'Нидерланды - Россия - Германия',  
'Франция - Люксембург - Бельгия',  
'Франция-Италия-Испания-Венгрия',  
'США - Румыния - Великобритания',  
'Великобритания - США - Франция',  
'Франция - Швейцария - Германия',  
'США - Великобритания - Испания',  
'Бельгия - Франция - Люксембург',  
'Франция-Испания-Бельгия-Панама',  
'Италия - Аргентина - Словения',  
'Франция - Дания - Швеция - КНР',  
'США - Бельгия - Великобритания',  
'США-Южная Корея-Новая Зеландия',  
'США - Великобритания - Франция',  
'Бельгия - Нидерланды - Франция',  
'Люксембург - Бельгия - Франция',  
'Великобритания, Новая Зеландия',  
'Великобритания - Мексика - США',  
'Канада, США, Каймановы острова',  
'США - Канада - Франция - Индия',  
'Великобритания-США-Германия-КНР',  
'США - Ирландия - Великобритания',  
'Великобритания - Германия - США',  
'Мексика - Испания - Дания - США',  
'США - Великобритания - Ирландия',  
'США - Германия - Великобритания',  
'Германия - Франция - Люксембург',  
'Бельгия - Германия - Люксембург',  
'Великобритания - Ирландия - США',  
'Великобритания - США - Германия',  
'Германия - Люксембург - Франция',  
'США - Великобритания - Болгария',  
'США - Великобритания - Исландия',

'Франция - Германия - Нидерланды',  
'Великобритания - Новая Зеландия',  
'Нидерланды - Бельгия - Болгария',  
'Германия - Люксембург - Бельгия',  
'Германия-Австрия-Франция-Италия',  
'США -Великобритания - Австралия',  
'Германия-Канада-Франция-Бельгия',  
'Франция-Германия-Великобритания',  
'Великобритания-Франция-Германия',  
'Германия-Франция-Великобритания',  
'США - Великобритания - Германия',  
'США, Великобритания, Люксембург',  
'Бразилия - Португалия - Франция',  
'Дания, Норвегия, Венгрия, Чехия',  
'Ирландия - Великобритания - США',  
'Нидерланды, Бельгия, Люксембург',  
'Швеция, Франция, Великобритания',  
'Испания - Германия - Нидерланды',  
'Германия - Великобритания - США',  
'СССР, Франция, Англия, Куба, ГДР',  
'СССР, ЧССР, Западный Берлин, ПНР',  
'Ирландия-Великобритания-Германия',  
'Германия - Финляндия - Австралия',  
'США - Великобритания - Австралия',  
'Россия-Польша-Голландия-Словакия',  
'США - Франция - Бельгия - Италия',  
'Великобритания - США - Швейцария',  
'Франция - Италия - Бельгия - КНР',  
'США-Великобритания-Чехия-Румыния',  
'Россия, Германия, Великобритания',  
'Франция - США - Норвегия - Дания',  
'Великобритания - Чехия - Франция',  
'Франция - Чехия - Великобритания',  
'Великобритания - Ирландия - США',  
'Португалия, Франция, Польша, США',  
'Пуэрто-Рико, Великобритания, США',  
'Великобритания - Австралия - США',  
'Великобритания, Франция, Бельгия',  
'Канада - Франция - Великобритания',  
'Франция-Великобритания - Германия',  
'Германия - Бельгия - США -Франция',  
'Япония - Великобритания - Франция',  
'Германия - США - Швеция - Франция',  
'Испания - Аргентина - Индия - США',  
'Канада - США - Германия - Франция',  
'Дания - Норвегия - Великобритания',  
'Великобритания - Италия - Испания',  
'Россия, Грузия, Хорватия, Испания',  
'США - Канада - Германия - Франция',  
'Швеция-Финляндия-Франция-Норвегия',  
'Великобритания - Франция - Италия',  
'США-Великобритания-Новая Зеландия',  
'Финляндия-Дания-Германия-Ирландия',  
'Нидерланды - Бельгия - Люксембург',  
'Германия, Франция, Великобритания',  
'Польша - Франция - Великобритания',  
'Австрия, Великобритания, Германия',  
'Дания, Норвегия, Швеция, Исландия',  
'США-Франция-Великобритания-Австрия',  
'Россия, Германия, Франция, Бельгия',  
'Великобритания-Германия-Канада-ЮАР',  
'Испания - Великобритания - Франция',  
'Ирландия - Великобритания - Канада',  
'Великобритания - Франция - Венгрия',  
'Россия, Франция, Германия, Бельгия',  
'Великобритания - Франция - Бельгия',  
'США-Япония-Германия-Великобритания',  
'Германия - США - Франция - Испания',  
'Россия, Латвия, Чешская Республика',  
'Италия-Швейцария-Франция-Германия',  
'Германия - Великобритания - Австрия',  
'США - Германия - Гонконг - Сингапур',  
'Великобритания - Италия - Швейцария',  
'США - Великобритания - Канада - КНР',  
'Швеция - Франция - Норвегия - Дания',  
'Франция - Германия - Турция - Катар',  
'Великобритания - Австрия - Германия',  
'США - Хорватия - Босния-Герцеговина',  
'Германия-Франция-Великобритания-США',  
'США - Объединенные Арабские Эмираты',  
'Великобритания - Франция - Германия',  
'Швеция - Германия - Франция - Дания',  
'Франция - Швеция - Дания - Норвегия',

'Испания, Ирландия, Бельгия, Франция',  
'Великобритания - Испания - Германия',  
'Дания - Франция - Германия - Швеция',  
'Германия - Бельгия - Великобритания',  
'Норвегия - Швеция - Дания - Германия',  
'Мексика - Аргентина - Великобритания',  
'Франция - Украина - Грузия - Армения',  
'Македония - Франция - Великобритания',  
'Швеция - Германия - Дания - Норвегия',  
'Швеция, Норвегия, Финляндия, Франция',  
'Великобритания, Германия, Дания, США',  
'Дания - Швейцария - Бельгия - Франция',  
'США - Сингапур - Малайзия - Индонезия',  
'Финляндия - Великобритания - Германия',  
'Нидерланды - Великобритания - Бельгия',  
'США - Великобритания - Новая Зеландия',  
'Франция-Россия-Румыния-Италия-Бельгия',  
'Украина - Германия - Латвия - Эстония',  
'Италия - Франция - Бразилия - Германия',  
'США - Великобритания - Канада - Швеция',  
'США - Великобритания- Франция - Швеция',  
'Швеция - Германия - Франция - Норвегия',  
'США - Япония - Франция - Великобритания',  
'Франция - Италия - Великобритания - США',  
'Великобритания - Франция - Италия - США',  
'Финляндия - Дания - Германия - Ирландия',  
'Бельгия - Испания - Канада - Нидерланды',  
'Франция - Италия - Бельгия - Люксембург',  
'Россия - Эстония - Финляндия - Беларусь',  
'Великобритания - Россия - Украина - США',  
'США - Франция - Канада - Великобритания',  
'Испания - США - Великобритания - Канада',  
'Канада, Франция, Италия, Швейцария, США',  
'Дания, Канада, Норвегия, Австралия, США',  
'США - Франция - Испания - Великобритания',  
'Великобритания - Австрия - Франция - США',  
'Германия - Аргентина - Нидерланды - Чили',  
'США - Великобритания - Франция - Гонконг',  
'США - Испания - Франция - Великобритания',  
'Великобритания - Франция - Бельгия - США',  
'Израиль - Германия - Франция - Швейцария',  
'США - Бразилия - Великобритания - Канада',  
'Мексика - Франция - Нидерланды - Германия',  
'Германия - США - Великобритания - Израиль',  
'Великобритания - США - Германия - Бельгия',  
'Мексика - Нидерланды - Германия - Франция',  
'Великобритания - Франция - Германия - США',  
'Ирландия - Великобритания - Франция - США',  
'США - Ирландия - Великобритания - Франция',  
'Великобритания - Чехия - Франция - Италия',  
'Бельгия - Германия - Люксембург - Франция',  
'Мексика, Франция, Германия, Дания, Швеция',  
'США, Великобритания, Ирландия, Люксембург',  
'США - Франция - Великобритания - Бразилия',  
'Бельгия - Люксембург - Франция - Швейцария',  
'Германия - Бельгия - Люксембург - Ирландия',  
'Нидерланды - Бельгия - Германия - Ирландия',  
'Германия - Бельгия - Люксембург - Норвегия',  
'Германия - США - Великобритания - Ирландия',  
'Республика Беларусь, Германия, США, Россия',  
'Россия - Франция - Великобритания - Латвия',  
'Россия, Германия, Казахстан, Польша, Китай',  
'США - Великобритания - Швейцария - Франция',  
'Франция - Австрия - Германия - Италия - США',  
'Великобритания - Франция - Бельгия - Италия',  
'Великобритания - Франция - Швеция - Бельгия',  
'Норвегия - Исландия - США - Великобритания',  
'Россия, Украина, Республика Беларусь, Литва',  
'Великобритания - Франция - Австрия - Швеция',  
'Великобритания - Нидерланды - Франция - США',  
'Франция - Испания - Румыния - США - Бельгия',  
'Бразилия - Германия - Португалия - Франция',  
'Великобритания - Норвегия - Дания - Германия',  
'Дания - Швеция - Италия - Франция - Германия',  
'Австралия - Великобритания - Бельгия - Индия',  
'Германия - Великобритания - Италия - Испания',  
'Великобритания - Норвегия - Кения - Хорватия',  
'Гонконг - Сингапур - Таиланд - Великобритания',  
'Япония - Франция - США - Южная Корея - Турция',  
'Канада - Франция - США - ОАЭ - Великобритания',  
'Дания-Франция-Бельгия-Германия-Великобритания',  
'США - Канада - Афганистан - Бельгия - Франция',  
'Италия - Франция - Швейцария - Великобритания',

'Израиль - Франция - Великобритания - Германия',  
'США-Япония-Германия-Великобритания-Нидерланды',  
'Польша - Португалия - Франция - Великобритания',  
'Швейцария - Израиль - Франция - Великобритания',  
'США - Россия - Польша - Германия - Пуэрто Рико',  
'Аргентина - Мексика - Бразилия - Франция - США',  
'Ирландия-Нидерланды-Франция-США-Великобритания',  
'США - Франция - Германия - Канада - Австралия',  
'Франция-Грузия-Германия-Россия-Украина-Бельгия',  
'Великобритания - Германия - Нидерланды - Дания',  
'Польша - Франция - Испания - Бразилия - Швеция',  
'Франция - Испания - Дания - Венгрия - Швейцария',  
'Нидерланды-Великобритания-Франция-Италия-Япония',  
'Грузия - Россия - Украина - Хорватия - Германия',  
'Франция - Великобритания - Камбоджа - США - КНР',  
'Дания - Франция - Италия - Бельгия - Нидерланды',  
'Германия-Канада-Великобритания-Швейцария-Франция',  
'Великобритания - Нидерланды - Франция - Хорватия',  
'Великобритания - Польша - КНР - Индия - Норвегия',  
'Великобритания - Германия - Франция - Кипр - США',  
'Франция - Германия - Литва - Нидерланды - Россия',  
'Франция - Россия - Швейцария - Румыния - Венгрия',  
'Бразилия, Уругвай, Дания, Норвегия, Чили, Швеция',  
'Великобритания - Израиль - Франция - Япония - США',  
'Ирландия - Дания - Бельгия - Люксембург - Франция',  
'США - Канада - Россия - Франция - Чили - Ирландия',  
'США - Великобритания - Германия - Бельгия - Дания',  
'США - Австралия - Новая Зеландия - Великобритания',  
'Швейцария, Великобритания, Франция, США, Ирландия',  
'Германия - Нидерланды - Беларусь - Россия - Латвия',  
'Япония, Великобритания, Австрия, Германия, Ю.Корея',  
'Германия - Швейцария - Франция - Южная Корея - США',  
'Испания - Франция - США - Великобритания - Бельгия',  
'Франция - Великобритания - Багамские острова - США',  
'Франция - Великобритания - Нидерланды - Люксембург',  
'Бельгия-Германия-Канада-Франция-США-Великобритания',  
'Германия - Франция - Великобритания - Польша - США',  
'США - Германия - Франция - Великобритания - Канада',  
'Великобритания - Гонконг - Венгрия - США - Ирландия',  
'США - Великобритания - Германия - Швеция - Канада',  
'Румыния, ЮАР, Иран, Франция, Канада, Великобритания',  
'США - Финляндия - Испания - Великобритания - Франция',  
'Великобритания - США - Канада - Нидерланды - Франция',  
'Великобритания - США - Германия - Канада - Австралия',  
'Сингапур - Великобритания - Индонезия - Канада - США',  
'Сербия - Словения - Хорватия - Черногория - Македония',  
'Великобритания - Исландия - Испания - Швейцария - США',  
'США - Франция - Канада - Германия - Австралия - Индия',  
'США - Нидерланды - Бельгия - Венгрия - Греция - Канада',  
'США - Нидерланды - Финляндия - Великобритания - Италия',  
'Канада - США - Нидерланды - Великобритания - Аргентина',  
'Испания-Франция-Греция-Индия-США-Россия-Великобритания',  
'Великобритания, Нидерланды, Польша, Швейцария, Франция',  
'США - Великобритания - Италия - Израиль - Сербия - Индия',  
'Испания - Бельгия - Франция - Португалия - Великобритания',  
'Таиланд-Великобритания-Франция-Германия-Испания-Нидерланды',  
'Испания, Франция, Великобритания, Дания, Бельгия, Германия',  
'США - Япония - Канада - Великобритания - Германия - Франция',  
'Израиль - Германия - Польша - Бельгия - Франция - Люксембург',  
'Франция - США - Великобритания - Колумбия - Бельгия - Россия',  
'Франция - Бельгия - Великобритания - Испания - Германия - США',  
'Великобритания - Аргентина - США - Франция - Россия - Испания',  
'Франция - Бельгия - Великобритания - США - Нидерланды - Канада',  
'Аргентина - Уругвай - Россия - Германия - Франция - Нидерланды',  
'Великобритания - Испания - Италия - Латвия - Франция - Эстония',  
'Великобритания, Южная Корея, Канада, США, Индия, Китай, Япония',  
'США - Бразилия - Франция - Австралия - Великобритания - Германия',  
'Германия - Дания - Испания - Швеция - Канада - Эстония - Франция',  
'Люксембург - Нидерланды - Испания - Великобритания - США - Италия',  
'Ирландия - Финляндия - Бельгия - Великобритания - США - Швейцария',  
'Германия - Франция - Польша - Турция - Канада - Италия - Россия',  
'Ирландия - Великобритания - Франция - США - Германия - Нидерланды',  
'Швейцария - КНР - США - Россия - Республика Корея - Великобритания',  
'Нидерланды - Франция - Германия - Бельгия - Швеция - Великобритания',  
'Дания - Канада - Швеция - Франция - Германия - Великобритания - США',  
'США - Великобритания - Германия - Новая Зеландия - Бельгия - Франция',  
'Великобритания - Франция - Республика Кипр - Швейцария - США - Сербия',  
'Великобритания - Нидерланды - Германия - Франция - Бельгия - Австрия',  
'Великобритания - США - Австралия - Ирландия - Германия - Куба - Канада',  
'Франция - Люксембург - Германия - Бельгия - Швейцария - Великобритания',  
'Дания - Австрия - Ирландия - Финляндия - Норвегия - Швеция - Нидерланды',  
'Великобритания - США - Канада - Бельгия - Нидерланды - Австрия - Германия',  
'Израиль - Франция - Германия - Палестина - США - Австрия - Великобритания',

'Нидерланды - США - Германия - Канада - Франция - Ирландия - Великобритания',  
 'Испания - Франция - Нидерланды - Германия - Бельгия - Великобритания - Канада',  
 'Франция - Канада - Н.Зеландия - США - Нидерланды - Германия - Швеция - Россия',  
 'Франция - Германия - Швеция - США - Чехия - Словакия - Великобритания - Нидерланды',  
 'Испания - Швейцария - Великобритания - Германия - Новая Зеландия - Нидерланды - Канада',  
 'Босния и Герцеговина - Франция - Великобритания - Германия - Словения - Бельгия - Сербия',  
 'Дания - Швеция - Великобритания - Франция - Германия - Нидерланды - Норвегия - Финляндия',  
 'Великобритания - Испания - Непал - Индия - Венгрия - Гонконг - Германия - Дания - Бахрейн',  
 'Дания, Швеция, Великобритания, Франция, Германия, Норвегия, Финляндия, Нидерланды, Италия',  
 'Мексика - Эквадор - Канада - США - Франция - Малайзия - Италия - Аргентина - Германия - Индия ',  
 'Великобритания - Франция - Италия - Индия - Дания - КНР - Бангладеш - Камбоджа - Гаити - Уганда',  
 'Япония - Великобритания - Швейцария - Ирландия - Дания - Франция - Польша - Австралия - Канада ',  
 'Норвегия - Азербайджан - Россия - Колумбия - Великобритания - Венгрия - Румыния - Франция - Грузия']

Видим ошибки правописания, использование множества пробелов.

```
In [47]: # уберем пробелы в начале и конце строки
data['production_country'] = data['production_country'].str.strip()
```

```
In [48]: # произведем замену ошибочного написания
for i in ['США', 'США', 'СЩА', 'Сша']:
    data['production_country'] = data['production_country'].str.replace(i, "США")
```

```
In [49]: data['production_country'] = data['production_country'].replace('Норвения', 'Норвегия')
```

```
In [50]: data['production_country'] = data['production_country'].str.replace("КНР", "Китай")
```

```
In [51]: data['production_country'] = data['production_country'].replace('2019', 'Германия')
```

```
In [52]: data['production_country'] = data['production_country'].str.replace('Объединенные Арабские Эмираты', 'ОАЭ')
```

```
In [53]: data['production_country'] = data['production_country'].str.replace('Белоруссия', 'Беларусь')
```

```
In [54]: data['production_country'] = data['production_country'].str.replace('Германя', 'Германия')
```

```
In [55]: data['production_country'] = data['production_country'].str.replace('Республика', '')
```

Данные имеют разные стили написания, приведем все к одному виду. С помощью цикла на основе каждого уникального значения столбца `production_country` создадим список стран, входящих в строку, предварительно убрав все разделители. Отсортируем список в алфавитном порядке, и соединим заново в строку с помощью выбранного разделителя. В цикле произведем замену значений столбца с помощью логической индексации

```
In [56]: for symbol in [',', '-']:
    for name in data['production_country'].unique():
        res = []
        for tot in [symbol.strip(' -, --') for symbol in name.split(symbol)]:
            res.append(tot)
        data.loc[data['production_country'] == name, 'production_country'] = ' - '.join(sorted(res))
```

```
In [57]: # отсортируем список по превым 6 символам и проверим данные еще раз
sorted(data['production_country'].unique(), key = lambda x: x[:6])
```

```
Out[57]: [' - Великобритания - Германия - Канада - США - Швеция',
'Австралия - США',
'Австралия',
'Австралия - Бразилия - Великобритания - Германия - США - Франция',
'Австралия - Великобритания - США',
'Австралия - Германия - Финляндия',
'Австралия - США - Франция',
'Австралия - Великобритания',
'Австралия - Дания - США',
'Австралия - США - Турция',
'Австралия - Великобритания - Дания - Ирландия - Канада - Польша - Франция - Швейцария - Япония',
'Австралия - Германия',
'Австралия - Бельгия - Великобритания - Индия',
'Австралия - Великобритания - Германия - Ирландия - Канада - Куба - США',
'Австралия - Великобритания - Германия - Канада - США',
'Австралия - Ирландия',
'Австралия - Германия - Индия - Канада - США - Франция',
'Австралия - Великобритания - Новая Зеландия - США',
'Австралия - Германия - Канада - США - Франция',
'Австралия - Франция',
'Австралия - Канада - США',
'Австралия - Германия - США',
'Австралия - Мексика - США',
'Австралия - Колумбия',
'Австралия - Дания - Канада - Норвегия - США',
'Австралия - Индия - США',
'Австралия - Канада',
```

'Австрия - Великобритания - США - Франция',  
'Австрия - Германия - Италия - США - Франция',  
'Австрия',  
'Австрия - Великобритания - Германия',  
'Австрия - Германия',  
'Австрия - Германия - Франция',  
'Австрия - Россия - Финляндия',  
'Австрия - Франция',  
'Австрия - Великобритания - Германия - Ю.Корея - Япония',  
'Австрия - Бельгия - Великобритания - Германия - Нидерланды - Франция',  
'Австрия - Германия - Ирландия',  
'Австрия - Дания - Ирландия - Нидерланды - Норвегия - Финляндия - Швеция',  
'Австрия - Бельгия - Великобритания - Германия - Канада - Нидерланды - США',  
'Австрия - Великобритания - Германия - Израиль - Палестина - США - Франция',  
'Австрия - Великобритания - Франция - Швеция',  
'Австрия - Германия - Италия - Франция',  
'Австрия - Швейцария',  
'Австрия - Люксембург',  
'Австрия - Россия',  
'Азербайджан - Россия',  
'Азербайджан - Великобритания - Венгрия - Грузия - Колумбия - Норвегия - Россия - Румыния - Франция',  
'Албания - СССР',  
'Англия - ГДР - Куба - СССР - Франция',  
'Аргентина - Бразилия',  
'Аргентина - Испания - Италия - США',  
'Аргентина',  
'Аргентина - Великобритания - Мексика',  
'Аргентина - Испания',  
'Аргентина - Германия - Нидерланды - Чили',  
'Аргентина - Германия - Индия - Италия - Канада - Малайзия - Мексика - США - Франция - Эквадор',  
'Аргентина - Испания - Уругвай',  
'Аргентина - Индия - Испания - США',  
'Аргентина - Германия - Нидерланды - Россия - Уругвай - Франция',  
'Аргентина - Бразилия - Мексика - США - Франция',  
'Аргентина - Италия - Словения',  
'Аргентина - Испания - Франция',  
'Аргентина - Великобритания - Испания - Россия - США - Франция',  
'Аргентина - Великобритания - Канада - Нидерланды - США',  
'Аргентина - Мексика',  
'Аргентина - США',  
'Аргентина - Франция',  
'Аргентина - Канада - США',  
'Армения',  
'Армения - Казахстан',  
'Армения - Грузия - Украина - Франция',  
'Армения - Германия',  
'Армения - Россия',  
'Афганистан - Бельгия - Канада - США - Франция',  
'Афганистан - СССР',  
'Багамские острова - Великобритания - США - Франция',  
'Бангладеш - Великобритания - Гаити - Дания - Индия - Италия - Камбоджа - Китай - Уганда - Франция',  
'Бахрейн - Великобритания - Венгрия - Германия - Гонконг - Дания - Индия - Испания - Непал',  
'Беларусь - Германия - Латвия - Нидерланды - Россия',  
'Беларусь',  
'Беларусь - Россия',  
'Беларусь - Россия - Финляндия - Эстония',  
'Беларусь - Литва - Россия - Украина',  
'Беларусь - Германия - Россия - США',  
'Бельгия - США',  
'Бельгия - Дания - Франция - Швейцария',  
'Бельгия - Франция',  
'Бельгия - Люксембург - Франция',  
'Бельгия - Италия - Франция',  
'Бельгия - Босния и Герцеговина - Великобритания - Германия - Сербия - Словения - Франция',  
'Бельгия - Великобритания - Италия - Франция',  
'Бельгия - Канада - Франция',  
'Бельгия - Канада - США',  
'Бельгия - Люксембург - Франция - Швейцария',  
'Бельгия - Великобритания - Германия - Люксембург - Франция - Швейцария',  
'Бельгия',  
'Бельгия - Германия - Франция',  
'Бельгия - Румыния - Франция',  
'Бельгия - Германия - Израиль - Люксембург - Польша - Франция',  
'Бельгия - Германия - Россия - Франция',  
'Бельгия - Нидерланды',  
'Бельгия - Великобритания - Германия - Дания - Франция',  
'Бельгия - Великобритания - Германия - Новая Зеландия - США - Франция',  
'Бельгия - Великобритания - Ирландия - США - Финляндия - Швейцария',  
'Бельгия - Италия - США - Франция',  
'Бельгия - Германия - США - Франция',  
'Бельгия - Испания - Канада - Нидерланды',  
'Бельгия - Великобритания - Франция - Швеция',  
'Бельгия - Италия - Люксембург - Франция',

'Бельгия - Венгрия - Греция - Канада - Нидерланды - США',  
'Бельгия - Германия - Люксембург',  
'Бельгия - Великобритания - Колумбия - Россия - США - Франция',  
'Бельгия - Великобритания - Германия - США',  
'Бельгия - Великобритания - Германия - Испания - США - Франция',  
'Бельгия - Италия - Китай - Франция',  
'Бельгия - Германия - Ирландия - Люксембург',  
'Бельгия - Великобритания - Канада - Нидерланды - США - Франция',  
'Бельгия - Дания - Ирландия - Люксембург - Франция',  
'Бельгия - Германия - Ирландия - Нидерланды',  
'Бельгия - Великобритания - Нидерланды',  
'Бельгия - Испания - Панама - Франция',  
'Бельгия - Великобритания - Германия - Испания - Канада - Нидерланды - Франция',  
'Бельгия - Франция - Чехия',  
'Бельгия - Великобритания - Германия - Дания - США',  
'Бельгия - Болгария - Нидерланды',  
'Бельгия - Испания - Франция',  
'Бельгия - Германия - Люксембург - Норвегия',  
'Бельгия - Великобритания - Германия - Нидерланды - Франция - Швеция',  
'Бельгия - Великобритания - Франция',  
'Бельгия - Великобритания - США',  
'Бельгия - Великобритания - Испания - США - Франция',  
'Бельгия - Франция - Япония',  
'Бельгия - Германия - Канада - Франция',  
'Бельгия - Италия - Россия - Румыния - Франция',  
'Бельгия - Германия - Грузия - Россия - Украина - Франция',  
'Бельгия - Германия - Люксембург - Франция',  
'Бельгия - Великобритания - США - Франция',  
'Бельгия - США - Франция',  
'Бельгия - Нидерланды - Франция',  
'Бельгия - Германия',  
'Бельгия - Великобритания - Германия - Канада - США - Франция',  
'Бельгия - Люксембург - Нидерланды',  
'Бельгия - Исландия',  
'Бельгия - Ирландия - Испания - Франция',  
'Бельгия - Великобритания - Испания - Португалия - Франция',  
'Бельгия - Польша - Франция',  
'Бельгия - Испания - Румыния - США - Франция',  
'Бельгия - Люксембург',  
'Бельгия - Великобритания - Германия',  
'Бельгия - Великобритания - Германия - Дания - Испания - Франция',  
'Бельгия - Дания - Италия - Нидерланды - Франция',  
'Бельгия - Канада',  
'Бельгия - Россия - Финляндия',  
'Бельгия - Сенегал - Франция',  
'Бельгия - Франция',  
'Болгария - Израиль - Франция',  
'Болгария - Испания - США',  
'Болгария - Великобритания - США',  
'Болгария - Россия',  
'Болгария - США',  
'Болгария',  
'Болгария - Мексика - США',  
'Босния - Герцеговина - США - Хорватия',  
'Бразилия',  
'Бразилия - Великобритания - США - Франция',  
'Бразилия - Германия - Португалия - Франция',  
'Бразилия - Испания',  
'Бразилия - Франция',  
'Бразилия - США',  
'Бразилия - Италия - Франция',  
'Бразилия - Канада',  
'Бразилия - Канада - Япония',  
'Бразилия - Канада - США',  
'Бразилия - Португалия - Франция',  
'Бразилия - Великобритания - Канада - США',  
'Бразилия - Дания - Норвегия - Уругвай - Чили - Швеция',  
'Бразилия - Китай - США',  
'Бразилия - Германия - Италия - Франция',  
'Бразилия - Испания - Польша - Франция - Швеция',  
'ВНР - СССР',  
'Великобритания - Германия',  
'Великобритания - Италия',  
'Великобритания - США',  
'Великобритания',  
'Великобритания - Германия - Китай - США',  
'Великобритания - США - Франция - Япония',  
'Великобритания - Гонконг - Сингапур - Таиланд',  
'Великобритания - Италия - США - Франция',  
'Великобритания - Германия - Ирландия',  
'Великобритания - Ирландия - США',  
'Великобритания - Испания - Италия - Люксембург - Нидерланды - США',  
'Великобритания - США - Франция',

'Великобритания - США - Чехия',  
'Великобритания - Канада - Франция',  
'Великобритания - Германия - США',  
'Великобритания - Испания - США - Франция',  
'Великобритания - Франция',  
'Великобритания - Израиль - США - Франция - Япония',  
'Великобритания - Германия - Канада - Франция - Швейцария',  
'Великобритания - Венгрия - США',  
'Великобритания - Германия - Франция',  
'Великобритания - Испания',  
'Великобритания - Канада - США - Швеция',  
'Великобритания - ЮАР',  
'Великобритания - Ирландия',  
'Великобритания - Япония',  
'Великобритания - Германия - Дания - Норвегия',  
'Великобритания - Индия - США',  
'Великобритания - Нидерланды - Франция - Хорватия',  
'Великобритания - Индия - Китай - Норвегия - Польша',  
'Великобритания - Польша - Португалия - Франция',  
'Великобритания - Канада - США',  
'Великобритания - Франция - Чехия',  
'Великобритания - Пуэрто - Рико',  
'Великобритания - Канада - ОАЭ - США - Франция',  
'Великобритания - Германия - Кипр - США - Франция',  
'Великобритания - Испания - США - Финляндия - Франция',  
'Великобритания - Италия - Швейцария',  
'Великобритания - Румыния - США',  
'Великобритания - Индия',  
'Великобритания - Франция - Япония',  
'Великобритания - Германия - Ирландия - Канада - Нидерланды - США - Франция',  
'Великобритания - США - Франция - Швеция',  
'Великобритания - Канада - Китай - США',  
'Великобритания - Дания - ЮАР',  
'Великобритания - Германия - Канада - ЮАР',  
'Великобритания - Канада',  
'Великобритания - Россия - США',  
'Великобритания - Германия - Израиль - США',  
'Великобритания - Кипр - США - Сербия - Франция - Швейцария',  
'Великобритания - Исландия - Норвегия - США',  
'Великобритания - США - Швейцария',  
'Великобритания - Китай - США',  
'Великобритания - Испания - США',  
'Великобритания - Испания - Франция',  
'Великобритания - Дания - Норвегия',  
'Великобритания - Италия - Франция - Швейцария',  
'Великобритания - Израиль - Франция - Швейцария',  
'Великобритания - Канада - Нидерланды - США - Франция',  
'Великобритания - Германия - Финляндия',  
'Великобритания - Исландия - Испания - США - Швейцария',  
'Великобритания - Румыния - США - Чехия',  
'Великобритания - Италия - Франция',  
'Великобритания - Германия - Россия',  
'Великобритания - Испания - Италия',  
'Великобритания - Македония - Франция',  
'Великобритания - Италия - Нидерланды - США - Финляндия',  
'Великобритания - Люксембург',  
'Великобритания - Германия - США - Франция',  
'Великобритания - Германия - Ирландия - Нидерланды - США - Франция',  
'Великобритания - Исландия - США',  
'Великобритания - Германия - Нидерланды - США - Словакия - Франция - Чехия - Швеция',  
'Великобритания - Ирландия - Канада',  
'Великобритания - Новая Зеландия',  
'Великобритания - Россия - США - Украина',  
'Великобритания - Канада - США - Франция',  
'Великобритания - Венгрия - Франция',  
'Великобритания - Испания - Канада - США',  
'Великобритания - Ирландия - Нидерланды - США - Франция',  
'Великобритания - США - Сербия',  
'Великобритания - Гонконг - США - Франция',  
'Великобритания - Швейцария',  
'Великобритания - Новая Зеландия - США',  
'Великобритания - США - Япония',  
'Великобритания - Ирландия - США - Франция',  
'Великобритания - Дания',  
'Великобритания - Германия - Канада - США - Франция - Япония',  
'Великобритания - Германия - Израиль - Франция',  
'Великобритания - Германия - Испания - Италия',  
'Великобритания - Венгрия',  
'Великобритания - Италия - Франция - Чехия',  
'Великобритания - Германия - Канада - США - Франция',  
'Великобритания - Германия - Ирландия - США',  
'Великобритания - Германия - Нидерланды - США - Япония',  
'Великобритания - Германия - США - Япония',



'Великобритания - Германия - Испания - Нидерланды - Таиланд - Франция',  
'Великобритания - Германия - Дания - Нидерланды - Норвегия - Финляндия - Франция - Швеция',  
'Великобритания - Люксембург - Нидерланды - Франция',  
'Великобритания - Италия - Нидерланды - Франция - Япония',  
'Великобритания - Нидерланды - США - Франция',  
'Великобритания - Китай',  
'Великобритания - Индонезия - Канада - США - Сингапур',  
'Великобритания - Румыния',  
'Великобритания - Германия - Дания - Нидерланды',  
'Великобритания - Польша',  
'Великобритания - США - Швеция',  
'Великобритания - Германия - Испания - Канада - Нидерланды - Новая Зеландия - Швейцария',  
'Великобритания - Люксембург - США',  
'Великобритания - Венгрия - Гонконг - Ирландия - США',  
'Великобритания - Мексика - США',  
'Великобритания - Китай - Корея - Россия - США - Швейцария',  
'Великобритания - Камбоджа - Китай - США - Франция',  
'Великобритания - Германия - Испания',  
'Великобритания - Испания - Италия - Латвия - Франция - Эстония',  
'Великобритания - Германия - Дания - Италия - Нидерланды - Норвегия - Финляндия - Франция - Швеция',  
'Великобритания - Польша - Франция',  
'Великобритания - Латвия - Россия - Франция',  
'Великобритания - Ирландия - США - Франция - Швейцария',  
'Великобритания - Греция - Индия - Испания - Россия - США - Франция',  
'Великобритания - США - Франция - Швейцария',  
'Великобритания - Нидерланды - Польша - Франция - Швейцария',  
'Великобритания - Германия - Польша - США - Франция',  
'Великобритания - Израиль',  
'Великобритания - Кения - Норвегия - Хорватия',  
'Великобритания - Франция - Швеция',  
'Великобритания - Иран - Канада - Румыния - Франция - ЮАР',  
'Великобритания - Ирландия - Люксембург - США',  
'Великобритания - Пуэрто - Рико - США',  
'Великобритания - Израиль - Индия - Италия - США - Сербия',  
'Великобритания - Китай - Россия',  
'Великобритания - Германия - Дания - США',  
'Великобритания - Индия - Канада - Китай - США - Южная Корея - Япония',  
'Великобритания - Германия - Дания - Канада - США - Франция - Швеция',  
'Венгрия - СССР',  
'Венгрия - Испания - Италия - Франция',  
'Венгрия - Германия - Сербия',  
'Венгрия - ГДР - СССР - ЧССР',  
'Венгрия - Германия - Швеция',  
'Венгрия',  
'Венгрия - Дания - Испания - Франция - Швейцария',  
'Венгрия - США',  
'Венгрия - Литва',  
'Венгрия - Дания - Норвегия - Чехия',  
'Венгрия - Россия - Румыния - Франция - Швейцария',  
'Венесуэла',  
'Вьетнам - США',  
'ГДР - СССР',  
'ГДР - Италия - Польша - СССР',  
'Германия - США',  
'Германия',  
'Германия - Италия - США',  
'Германия - Канада - США',  
'Германия - Ирландия - Исландия',  
'Германия - Чехия',  
'Германия - Франция',  
'Германия - Польша - Франция',  
'Германия - Италия - Ланка - Шри',  
'Германия - Дания - Норвегия - Швеция',  
'Германия - Греция - Франция',  
'Германия - Гонконг - США - Сингапур',  
'Германия - Финляндия - Франция',  
'Германия - Мексика - Нидерланды - Франция',  
'Германия - Нидерланды - Россия',  
'Германия - Россия',  
'Германия - Дания - Ирландия - Финляндия',  
'Германия - Дания - Италия - Франция - Швеция',  
'Германия - Дания - Швеция',  
'Германия - Италия',  
'Германия - Канада',  
'Германия - Мексика - Уругвай',  
'Германия - Люксембург - Франция',  
'Германия - Швеция',  
'Германия - США - Франция - Швеция',  
'Германия - Турция - Франция',  
'Германия - Франция - Швейцария',  
'Германия - Швейцария - ЮАР',  
'Германия - Исландия',  
'Германия - Канада - США - Франция',

'Германия - США - Франция - Швейцария - Южная Корея',  
'Германия - Италия - Канада - Польша - Россия - Турция - Франция',  
'Германия - Польша - Пуэрто Рико - Россия - США',  
'Германия - Норвегия - Франция - Швеция',  
'Германия - Катар - Турция - Франция',  
'Германия - Нидерланды - Франция',  
'Германия - Канада - Н.Зеландия - Нидерланды - Россия - США - Франция - Швеция',  
'Германия - Китай - США',  
'Германия - США - Франция',  
'Германия - Дания - Испания - Канада - Франция - Швеция - Эстония',  
'Германия - Финляндия - Швеция',  
'Германия - Россия - Украина',  
'Германия - Кения',  
'Германия - Дания - Норвегия',  
'Германия - Дания',  
'Германия - Россия - Швейцария',  
'Германия - Италия - Франция',  
'Германия - Израиль',  
'Германия - Испания - Франция',  
'Германия - Нидерланды - США',  
'Германия - Канада - Франция',  
'Германия - США - Япония',  
'Германия - Швейцария',  
'Германия - Дания - Франция - Швеция',  
'Германия - Нидерланды - Украина',  
'Германия - Китай',  
'Германия - Нидерланды - ЮАР',  
'Германия - Италия - Швейцария',  
'Германия - Латвия - Украина - Эстония',  
'Германия - Литва - Нидерланды - Россия - Франция',  
'Германия - Финляндия',  
'Германия - Грузия - Россия - Украина - Хорватия',  
'Германия - Испания - США - Франция',  
'Германия - Израиль - Франция - Швейцария',  
'Германия - Дания - Люксембург',  
'Германия - Италия - Франция - Швейцария',  
'Германия - Дания - Мексика - Франция - Швеция',  
'Германия - Казахстан - Китай - Польша - Россия',  
'Германия - Израиль - Франция',  
'Германия - Испания - Нидерланды',  
'Голландия - Польша - Россия - Словакия',  
'Голландия',  
'Гонконг - Китай',  
'Гонконг',  
'Гонконг - Китай - США',  
'Гонконг - США',  
'Гонконг - Ирландия - Франция',  
'Гонконг - США - ЮАР',  
'Гонконг - Индия - Китай - США',  
'Греция - Россия',  
'Греция',  
'Греция - Франция',  
'Греция - Италия - США',  
'Грузия - Люксембург - Франция',  
'Грузия - Украина',  
'Грузия - Испания - Россия',  
'Грузия - Франция',  
'Грузия - Россия',  
'Грузия',  
'Грузия - Испания - Россия - Хорватия',  
'Дания',  
'Дания - Испания - Мексика - США',  
'Дания - Франция - Швеция',  
'Дания - Швеция',  
'Дания - Норвегия - Швеция',  
'Дания - Норвегия - Франция - Швеция',  
'Дания - Норвегия - США - Франция',  
'Дания - Китай - Франция - Швеция',  
'Дания - США',  
'Дания - США - Франция',  
'Дания - Франция',  
'Дания - Финляндия - Швеция',  
'Дания - Латвия - Россия - США',  
'Дания - Исландия',  
'Дания - Исландия - Норвегия - Швеция',  
'Западный Берлин - ПНР - СССР - ЧССР',  
'Западный Берлин - СССР - ФРГ',  
'Израиль',  
'Израиль - Франция',  
'Израиль - США - Франция',  
'Израиль - Украина',  
'Индия',  
'Индия - ОАЭ - США',

'Индия - США',  
'Индия - Франция',  
'Индия - Мексика',  
'Индия - Китай',  
'Индия - Китай - США - Ю.Корея',  
'Индия - Канада - США - Франция',  
'Индонезия - США',  
'Индонезия - Малайзия - США - Сингапур',  
'Индонезия',  
'Индонезия - Канада - США',  
'Иран',  
'Иран - Франция',  
'Иран - Италия - Франция',  
'Ирландия',  
'Ирландия - Франция - Швеция',  
'Ирландия - Канада',  
'Ирландия - Россия',  
'Ирландия - США',  
'Ирландия - Канада - Россия - США - Франция - Чили',  
'Ирландия - США - Франция',  
'Ирландия - Польша',  
'Исландия - Финляндия',  
'Исландия - Финляндия - Швеция',  
'Исландия',  
'Испания - Канада - Франция',  
'Испания - Франция',  
'Испания - США',  
'Испания',  
'Испания - США - Франция',  
'Испания - Куба',  
'Испания - Канада - Япония',  
'Испания - Канада',  
'Испания - Италия - США',  
'Испания - Колумбия - США',  
'Испания - Чили',  
'Испания - Италия - Франция',  
'Испания - Мексика',  
'Испания - Россия',  
'Испания - Колумбия',  
'Испания - Франция - Швейцария',  
'Испания - Тайвань - Франция',  
'Испания - Мальта',  
'Испания - США - Чехия',  
'Испания - Италия',  
'Италия - СССР',  
'Италия',  
'Италия - Франция - Швейцария',  
'Италия - Франция',  
'Италия - Польша - Россия',  
'Италия - Россия',  
'Италия - США',  
'Италия - Канада - США',  
'Италия - Швейцария',  
'Италия - Канада',  
'Италия - Канада - США - Франция - Швейцария',  
'Казахстан - Россия - США',  
'Казахстан - Россия',  
'Казахстан',  
'Каймановы острова - Канада - США',  
'Канада - США',  
'Канада - Франция',  
'Канада',  
'Канада - Южная Корея',  
'Канада - Япония',  
'Канада - ЮАР',  
'Канада - Китай',  
'Канада - США - Южная Корея',  
'Канада - Финляндия',  
'Канада - Китай - США',  
'Канада - Люксембург - Россия - США',  
'Канада - США - Франция',  
'Канада - Китай - США - Япония',  
'Канада - Мексика',  
'Канада - Норвегия - США',  
'Канада - Норвегия',  
'Кипр',  
'Кипр - Россия',  
'Кипр - Россия - США',  
'Киргизия',  
'Киргизия - Россия',  
'Китай',  
'Китай - США',  
'Китай - Сингапур',

'Китай - Южная Корея',  
'Китай - Франция',  
'Китай - Южная Корея - Япония',  
'Китай - США - Таиланд',  
'Китай - Россия',  
'Китай - Новая Зеландия',  
'Княжество Андорра - Украина',  
'Колумбия - Уругвай',  
'Колумбия - США',  
'Колумбия',  
'Корея',  
'Куба - Швеция',  
'Латвия - Россия - Франция',  
'Латвия - Россия',  
'Латвия - Россия - Чешская',  
'Латвия - Франция',  
'Латвия - Финляндия',  
'Ливан - США',  
'Литва - Россия - Украина',  
'Люксембург - Швейцария',  
'Люксембург - Франция',  
'Македония - Сербия - Словения - Хорватия - Черногория',  
'Македония - Франция',  
'Малайзия - США',  
'Малайзия',  
'Мальта - США',  
'Мексика - США - Франция',  
'Мексика - Чили',  
'Мексика - Франция',  
'Мексика',  
'Мексика - США',  
'Мексика - США - Тайвань',  
'Монако - Франция',  
'Монголия - СССР',  
'Монголия',  
'Нидерланды - США',  
'Нидерланды',  
'Нидерланды - Украина',  
'Нидерланды - Россия',  
'Нидерланды - Франция',  
'Нидерланды - Россия - Финляндия',  
'Нидерланды - Норвегия',  
'Новая Зеландия',  
'Новая Зеландия - США',  
'Новая Зеландия - США - Южная Корея',  
'Новая Зеландия - США - Япония',  
'Новая Зеландия - Франция',  
'Норвегия',  
'Норвегия - Швеция',  
'Норвегия - США',  
'Норвегия - Финляндия - Швеция',  
'Норвегия - Финляндия - Франция - Швеция',  
'Норвегия - Россия - Швеция',  
'Норвегия - Франция',  
'ОАЭ - США',  
'Пакистан',  
'Парагвай',  
'Перу',  
'Перу - Франция',  
'Польша - СССР',  
'Польша - Франция',  
'Польша - Россия - Украина',  
'Польша - Россия',  
'Польша',  
'Польша - Финляндия',  
'Польша - Россия - Финляндия',  
'Польша - Португалия - США - Франция',  
'Португалия - Франция',  
'Португалия',  
'Пуэрто - Рико - США - Франция',  
'Пуэрто Рико - США',  
'Россия',  
'Россия - Франция',  
'Россия - США',  
'Россия - США - Франция',  
'Россия - Украина',  
'Россия - Эстония',  
'Россия - Румыния',  
'Россия - СССР',  
'Россия - Таджикистан',  
'Россия - США - ФРГ',  
'Россия - Словакия - Чехия',  
'Россия - Сербия',

```
'Румыния - США',
'Румыния - СССР - Франция',
'Румыния',
'СССР',
'СССР - Швеция',
'СССР - Финляндия',
'СССР - Франция - Швейцария',
'СССР - ФРГ',
'США',
'США - Турция - Франция',
'США - Франция',
'США - ЮАР',
'США - Южная Корея',
'США - Франция - Чили',
'США - Турция - Франция - Южная Корея - Япония',
'США - Франция - Швейцария',
'США - Украина',
'США - Япония',
'США - Чили',
'США - Ю.Корея',
'США - Франция - Чехия',
'США - Франция - ЮАР',
'США - Швеция',
'США - Финляндия',
'США - Турция',
'США - Франция - Япония',
'США - Таиланд',
'Словакия - Хорватия - Чехия',
'Таиланд',
'Турция',
'Узбекистан',
'Украина',
'Украина - Франция',
'Финляндия',
'Финляндия - Франция',
'Франция',
'Франция - Швейцария',
'Франция - Чили',
'Франция - Швеция',
'Хорватия',
'Чехия',
'Чехословакия',
'Чили',
'Швейцария',
'Швеция',
'Эстония',
'ЮАР',
'Южная Корея',
'Япония']
```

```
In [58]: # посчитаем количество оставшихся значений
data['production_country'].nunique()
```

```
Out[58]: 657
```

- Проверим столбец `puNumber` на дубликаты

```
In [59]: data[data['puNumber'].duplicated()]
```

```
Out[59]:
```

	title	puNumber	show_start_date		type	film_studio	production_country	director	producer	age_restriction
4639	Иоанна - женщина на папском престоле / По роман...	221,154,310.00	2010-12-17 12:00:00+00:00	Художественный		Константин Фильм, А Эр Ди Дегето Фильм, Дюне ...	Великобритания - Германия - Испания - Италия	Зенке Вортманн	Оливер Бербен, Дорис Д.Хайнце, Фарук Элтан	«16+» - для детей старше 16 лет
5068	Анализируй то!	221,054,410.00	2010-05-25 12:00:00+00:00	Художественный		Уорнер Бразерс, Виллидж Рoadшоу Пикчерз, Эн-Пи...	США	Гарольд Реймис	Джейн Розенталь, Паул Уэйнштейн	«12+» - для детей старше 12 лет

```
In [60]: # выведем строки с одинаковым прокатным удостоверением
data.loc[(data['puNumber'] == 221154310) | (data['puNumber'] == 221054410)]
```

Out[60]:		title	puNumber	show_start_date		type	film_studio	production_country	director	producer	age_restriction
	4638	Как жениться и остаться холостым	221,154,310.00	2010-12-17 12:00:00+00:00	Художественный	Ше Вам, Скрипт Ассосье, Тэ Фэ 1 Фильм Продюксь...		Франция	Эрик Лартиго	Амандин Било, Алан Шаба	«16+» - для детей старше 16 лет
	4639	Иоанна - женщина на папском престоле / По роман...	221,154,310.00	2010-12-17 12:00:00+00:00	Художественный	Константин Фильм, А Эр Ди Дегето Фильм, Дюне ...		Великобритания - Германия - Испания - Италия	Зенке Вортманн	Оливер Бербен, Дорис Д.Хайнце, Фарук Элтан	«16+» - для детей старше 16 лет
	5067	Анализируй это!	221,054,410.00	2010-05-25 12:00:00+00:00	Художественный	Уорнер Бразерс, Вилладж Роудшоу Филмз ЛТД		Австралия - США	ХЭрольд Рэмис	Джейн Розенталь, Пола Уейнштейн	«16+» - для детей старше 16 лет
	5068	Анализируй то!	221,054,410.00	2010-05-25 12:00:00+00:00	Художественный	Уорнер Бразерс, Виллидж Роадшоу Пикчерз, Эн-Пи...		США	Гарольд Реймис	Джейн Розенталь, Паул Уэйнштейн	«12+» - для детей старше 12 лет

Фильмы с одинаковым номером прокатного удостоверения являются разными, что является некоторым допущением, удаление строк нецелесообразно.

Вывод:

Проведена работа по выявлению неявных дубликатов. В столбцах с названием фильма дубликаты допустимы, в связи со спецификой кинопроката. В столбцах указывающих на страну и тип фильмов, выявлены дубликаты, произведена замена. Информация о режиссере и продюсере содержит большой объем информации, обработка и поиск дубликатов не является целесообразным в рамках данного исследования.

Изучение параметров

```
In [61]: # напишем функцию для возврата предельных значений , характерных для столбца
def moustache(data):
    q3 = data.quantile(0.75)
    q1 = data.quantile(0.25)
    delta = q3 - q1
    if q1 - (delta*1.5) < data.min():
        return (data.min(), q3 + (delta*1.5))
    else:
        return (q1 - (delta*1.5), q3 + (delta*1.5))
```

- Данные столбца ratings

```
In [62]: data['ratings'].describe()
```

Out[62]: count 6,517.00
mean 6.49
std 1.11
min 1.00
25% 5.90
50% 6.60
75% 7.20
max 9.90
Name: ratings, dtype: float64

```
In [63]: data['ratings'].hist(bins = 105, legend = True, grid = True)
plt.title('Распределение значений ratings')
plt.xlabel('показатель рейтинга')
plt.ylabel('количество');
```



Чаще всего фильмы оценивают между 5,5 и 7 баллами. Максимальная оценка 9.9, минимальная 1 балл.

- Исследуем столбец `refundable_support`

Данные столбца включают информацию только для фильмов с поддержкой госбюджета и содержит информацию о возвратной части бюджета, которая в свою очередь может равняться 0

```
In [64]: # определим границы нормальных значений
moustache(data['refundable_support'])
```

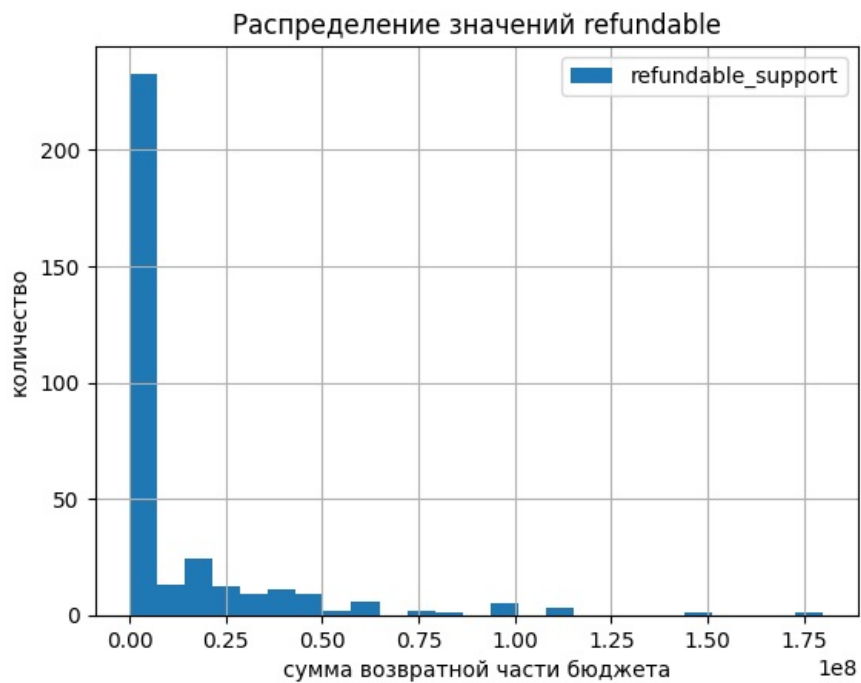
```
Out[64]: (0.0, 37500000.0)
```

```
In [65]: # посмотрим на медианное и среднее значения
data['refundable_support'].describe()
```

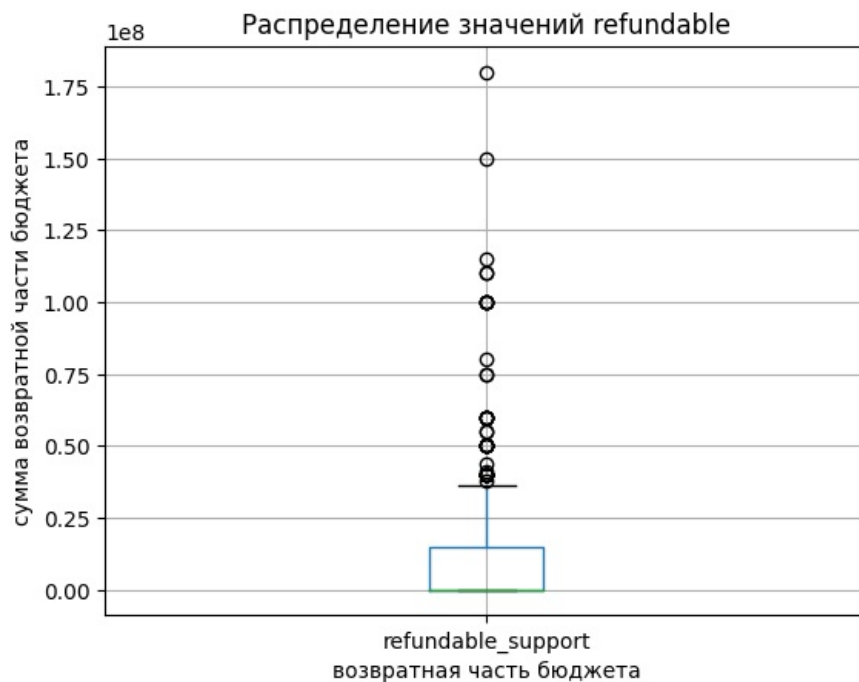
```
Out[65]: count      332.00
mean      11,864,457.83
std       24,916,555.26
min         0.00
25%         0.00
50%         0.00
75%      15,000,000.00
max      180,000,000.00
Name: refundable_support, dtype: float64
```

Медианное значение равно 0, значит большая часть данных равна 0

```
In [66]: # отобразим на гистограмме количественное распределение значений
data['refundable_support'].hist(bins = 25, legend = True);
plt.title('Распределение значений refundable')
plt.xlabel('сумма возвратной части бюджета')
plt.ylabel('количество');
```



```
In [67]: # построим боксплот для визуализации данных
data['refundable_support'].plot(kind = 'box', grid = True);
plt.title('Распределение значений refundable')
plt.xlabel('возвратная часть бюджета')
plt.ylabel('сумма возвратной части бюджета');
```



Минимальное значение для возвратной части бюджета это 0 рублей, максимальное значение 180 млн.рублей. Медианное значение составляет 0 рублей .Среднее арифметическое сильно отличается от медианы, что говорит о неоднородности данных.

- Исследуем столбец с информацией о невозвратной части бюджета `nonrefundable_support`

```
In [68]: # определим границы нормальных значений
moustache(data['nonrefundable_support'])
```

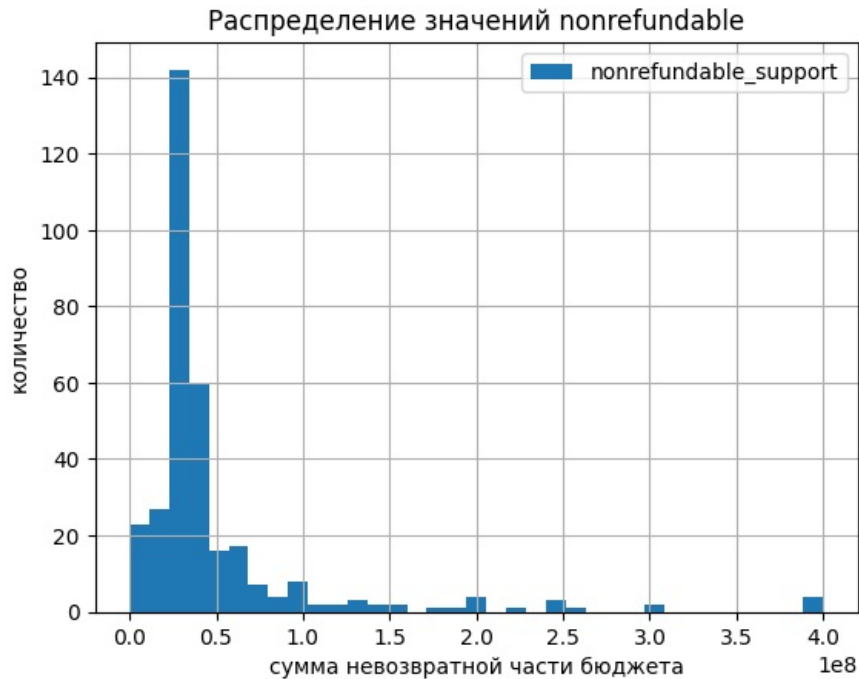
```
Out[68]: (1937500.0, 63437500.0)
```

```
In [69]: # описание основных метрик данных
data['nonrefundable_support'].describe()
```

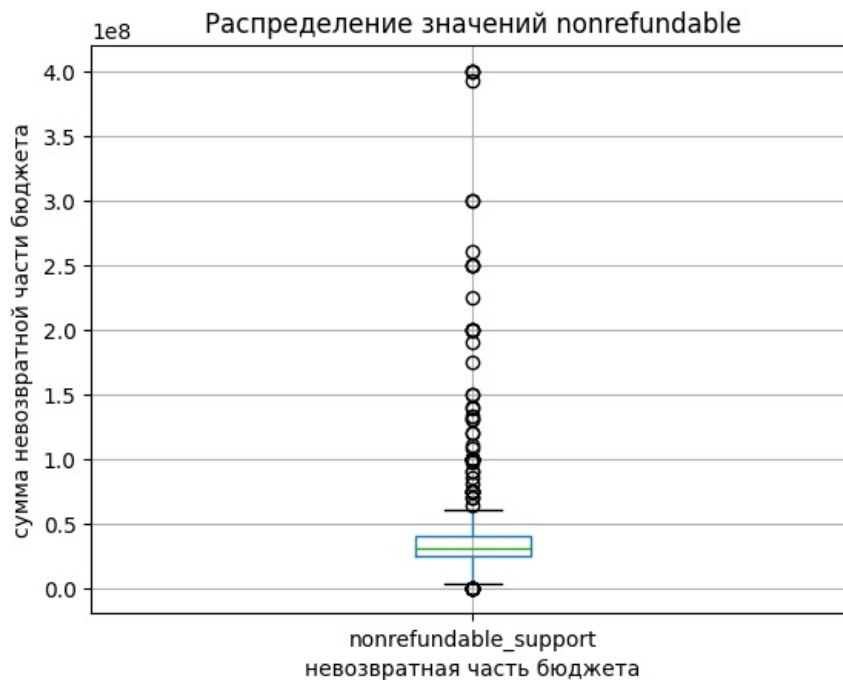


```
Out[69]: count      332.00
mean      48,980,988.89
std       59,980,117.92
min        0.00
25%       25,000,000.00
50%       30,000,000.00
75%       40,375,000.00
max       400,000,000.00
Name: nonrefundable_support, dtype: float64
```

```
In [70]: # построим гистограмму распределения значений
data['nonrefundable_support'].hist(bins = 35, legend = True);
plt.title('Распределение значений nonrefundable')
plt.xlabel('сумма невозвратной части бюджета')
plt.ylabel('количество');
```



```
In [71]: # построим график распределения значений невозвратной части бюджета
data['nonrefundable_support'].plot(kind = 'box', grid = True);
plt.title('Распределение значений nonrefundable')
plt.xlabel('невозвратная часть бюджета')
plt.ylabel('сумма невозвратной части бюджета');
```



Для данного набора данных минимальное значений равное 0 соответствует действительности и означает отсутствие суммы невозвратной части бюджета. Медианное значение составляет 30 млн, максимальная сумма - это 400 млн

```
In [72]: # сделаем проверку на корректность исходя из данных
data.query('nonrefundable_support + refundable_support > budget')['budget'].count()
```

Out[72]: 17

```
In [73]: # заменим ошибочные значения на сумму nonrefundable_support и refundable_support
data.loc[data['nonrefundable_support'] + data['refundable_support'] > data['budget'], 'budget'] = \
data['nonrefundable_support'] + data['refundable_support']
```

- Описание столбца `budget` - общий бюджет фильмов с господдержкой

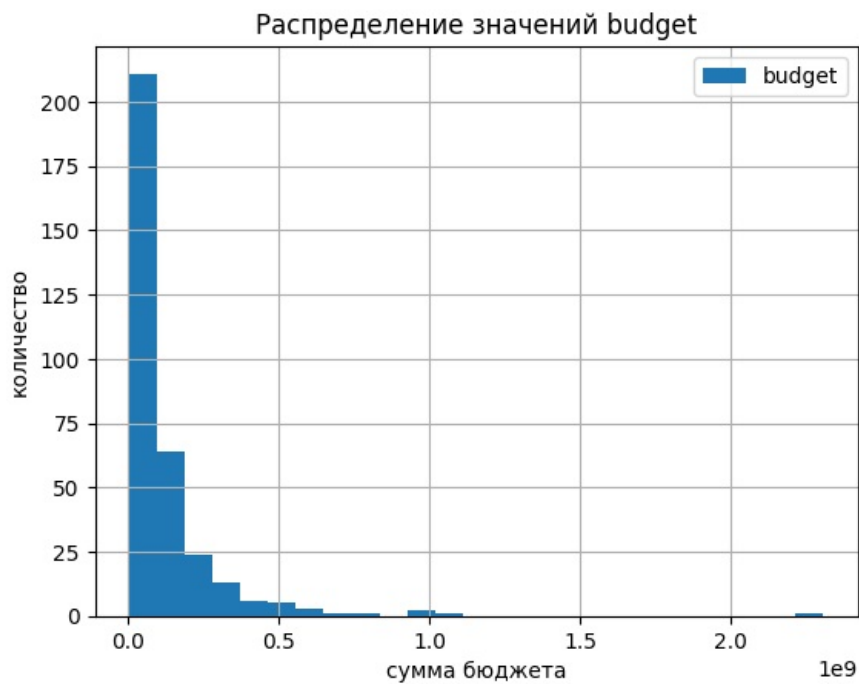
```
In [74]: data['budget'].describe()
```

```
Out[74]: count      332.00
mean    131,422,395.80
std     187,148,242.67
min       6,000,000.00
25%     45,524,799.50
50%     71,196,901.50
75%    150,000,000.00
max     2,305,074,303.00
Name: budget, dtype: float64
```

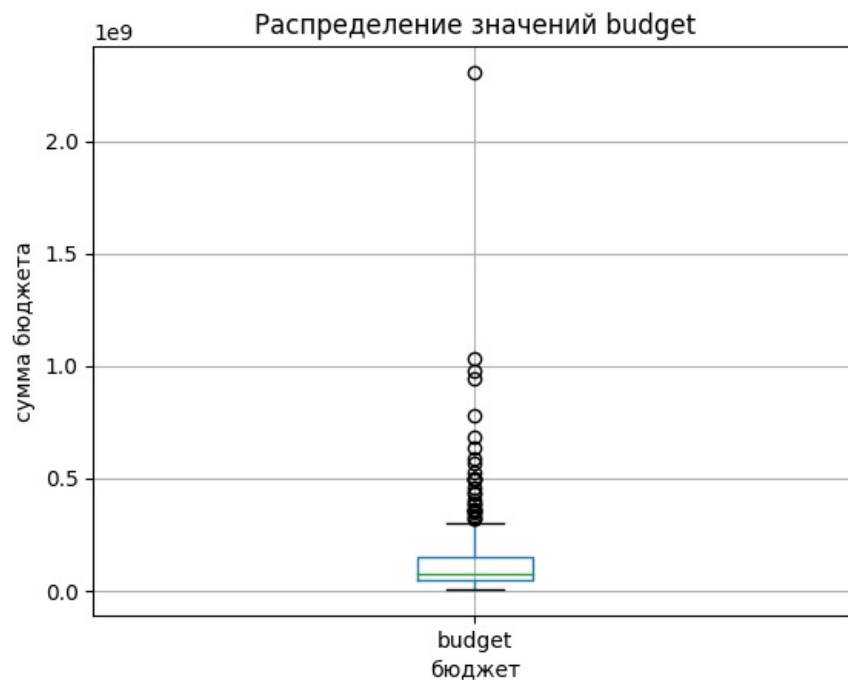
```
In [75]: # рассчитаем границы типичных значений
moustache(data['budget'])
```

```
Out[75]: (6000000.0, 306712800.75)
```

```
In [76]: # построим гистограмму распределения значений
data['budget'].hist(bins = 25, legend = True);
plt.title('Распределение значений budget')
plt.xlabel('сумма бюджета')
plt.ylabel('количество');
```



```
In [77]: # построим график для визуализации данных
data['budget'].plot(kind = 'box', grid = True);
plt.title('Распределение значений budget')
plt.xlabel('бюджет')
plt.ylabel('сумма бюджета');
```



Для фильмов с государственной поддержкой минимальный бюджет составляет 6 млн руб, максимальное же значение 2,30 млрд, что значительно превышает медиану. Верхняя граница нормальных значений - 306 млн.руб

- Рассмотрим распределение значений кассового сбора

```
In [78]: # границы типичных значений столбца
moustache(data['box_office'])
```

```
Out[78]: (0.0, 59828509.25)
```

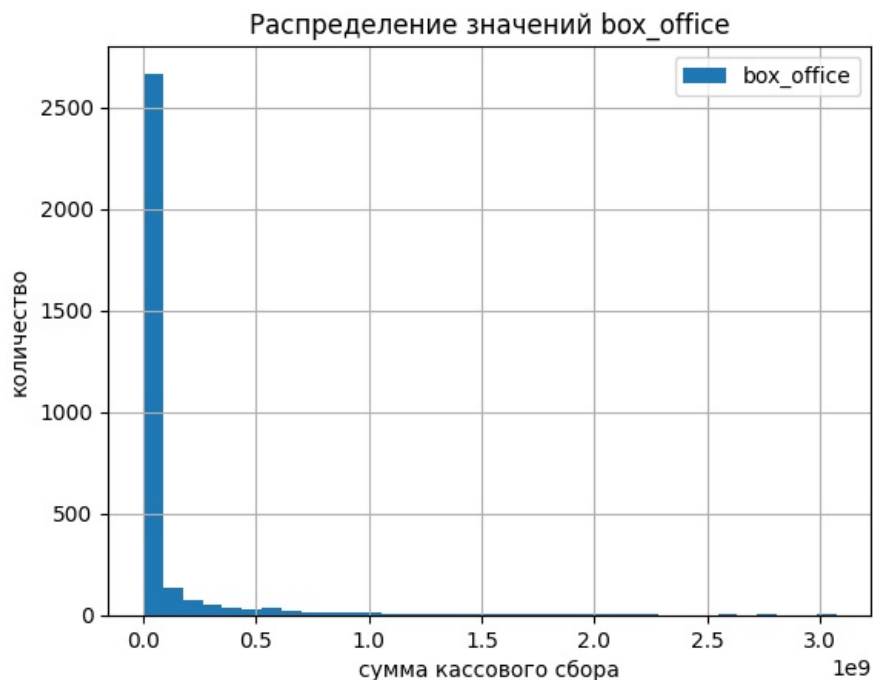
```
In [79]: data['box_office'].describe()
```

```
Out[79]: count      3,157.00
mean       76,502,859.66
std        240,387,362.16
min         0.00
25%         86,190.00
50%        2,330,104.00
75%        23,983,117.70
max        3,073,568,690.79
Name: box_office, dtype: float64
```

Видим минимальное значение равное 0, удалим такие строки

```
In [80]: # удаление строк со значением box_office равным 0
data = data.loc[(data['box_office'] > 0) | (data['box_office'].isna()),:]
```

```
In [81]: data['box_office'].hist(bins = 35, legend = True);
plt.title('Распределение значений box_office')
plt.xlabel('сумма кассового сбора')
plt.ylabel('количество');
```



Анализируя гистограмму и показатели столбца можно прийти к выводу о некоем разделении значений. Одновременно большой объем в самом начале гистограммы, резкое снижение в совокупности с длинным хвостом. Медиана и максимальное значение отличаются на 3 порядка.

Вывод:

Средняя оценка фильмов равна примерно 6,3 балла. Абсолютное большинство фильмов получает оценку от 5 до 7 баллов. Можно говорить о том, что большая часть фильмов с господдержкой не имеют возвратной стоимости бюджета. Невозвратная же часть в среднем находится на отметке 49 млн, финансирование невозвратной части больше 63 млн и ниже 2 млн уже редкость. Бюджет фильмов с господдержкой ожидаемо показывает большой разброс от 6 млн до 306 млн - типичный разброс, цифры ниже или выше этих границ встречаются реже. Суммы кассовых сборов показывают нетипичное распределение, возможно в данные включены разные способы исчисления.

```
In [82]: stop_row, a1 = data.shape
```

```
In [83]: print('Процент потерянных данных:', round(100 - (stop_row/start_row)*100, 2), '%')
```

Процент потерянных данных: 0.36 %

## Добавление параметров

```
In [84]: # добавим столбец с годом выдачи прокатного удостоверения
data['year_premier'] = data['show_start_date'].dt.year
```

```
In [85]: # напишем функцию для возврата ФИО главного режиссера
def direc(string):
    a = [i.strip() for i in string.split(',')]
    return (a[0])
```

```
In [86]: # создадим столбец с именем главного режиссера
data['main_director'] = data['director'].apply(direc)
```

```
In [87]: # создадим столбец с главным жанром фильма
data['main_genres'] = data['genres'].apply(direc)
```

```
In [88]: # создадим функцию для расчета доли государственного финансирования в бюджете, если госфинансирования нет, запо.
def perc(data):
    if data['budget'] > 0:
        return round((data['refundable_support'] + data['nonrefundable_support'])/data['budget'], 2)
    else:
        return 0
```

```
In [89]: # создадим столбец с долей государственного финансирования в общем бюджете
data['gov_part'] = data.apply(perc, axis = 1)
```

## Исследовательский анализ данных

- Исследуем количество выданных прокатных удостоверений

```
In [90]: # построим график зависимости количества удостоверений от года выдачи
data.pivot_table(index = 'year_premier', values = 'puNumber', aggfunc = 'count')\
.rename(columns = {'puNumber': 'Количество прокатных удостоверений'})\
.plot(title = 'Количество выданных прокатных удостоверений по годам', grid = True);
plt.xlabel('год премьеры')
plt.ylabel('количество');
```



Количество прокатных удостоверений меняется от года к году, наблюдается некоторое падение в 2017г, максимум в 2010

- Посчитаем количество и долю прокатных удостоверений с имеющейся информацией о кассовых сборах

```
In [91]: # посчитаем количество прокатных удостоверений с указанной информацией о кассовых сборах
box_quantity = data.query('box_office.isna() == False')['box_office'].count()
# посчитаем процент прокатных удостоверений с информацией о кассовых сборах от общего числа прокатных удостоверений
per_quantity = (box_quantity/data['puNumber'].count())*100

print("Количество прокатных удостоверений с указанной суммой кассовых сборов: ", box_quantity)
print()
print('Доля прокатных удостоверений с указанной суммой кассовых сборов от общего количества: ', round(per_quantity, 1))
```

Количество прокатных удостоверений с указанной суммой кассовых сборов: 3133

Доля прокатных удостоверений с указанной суммой кассовых сборов от общего количества: 42.0 %

Доля прокатных удостоверений с указанной суммой сборов составляет 42%. Посчитаем как менялось количество прокатных удостоверений с информацией по кассовым сборам и без нее в течении 10 лет.

```
In [92]: # создадим сводную таблицу с наличием кассовых сборов сгруппированную по годам и найдем количество прокатных удостоверений
x = data.query('box_office.isna() == False').pivot_table(index = 'year_premier', values = 'puNumber', aggfunc = 'count')
.rename(columns = {'puNumber': 'есть данные box_office'})
# создадим сводную таблицу без сумм кассовых сборов сгруппированную по годам и найдем количество прокатных удостоверений
b = data.query('box_office.isna()').pivot_table(index = 'year_premier', values = 'puNumber', aggfunc = 'count')
.rename(columns = {'puNumber': 'нет данных box_office'})
# соединим полученные таблицы по году выпуска прокатных удостоверений и построим столбчатую диаграмму
x.merge(b, on = 'year_premier', how = 'outer')\
.plot(kind = 'bar', title = 'Количество прокатных удостоверений с наличием/отсутствием данных о кассовых сборах',
      figsize = (15,5));
plt.xlabel('год')
plt.ylabel('количество');
```

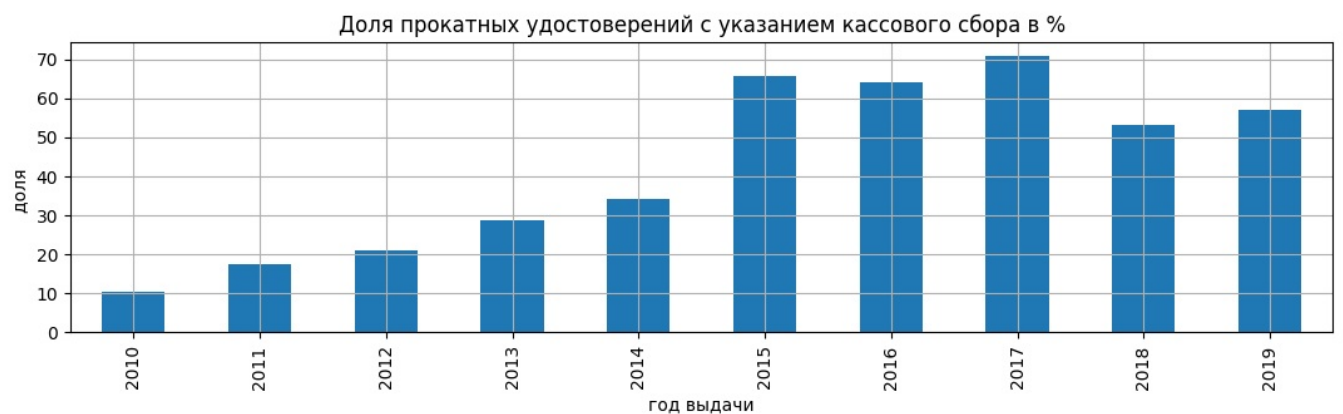


Видим, что пропорция постепенно менялась и после 2014 года количество прокатных удостоверений с указанием кассового сбора начало превышать количество прокатных удостоверений без сборов.

```
In [93]: # создадим сводную таблицу, сгруппировав данные по году и рассчитаем количество данных для прокатных удостоверений
# данных по кассовым сборам и без
pivot_percent = data.pivot_table(index = 'year_premier', values = ['box_office', 'puNumber'], aggfunc = 'count') \
    .rename(columns = {'box_office': 'Количество данных с box_office', 'puNumber': 'Общее количество'})
# добавим столбец с долей удостоверений с информацией box_office от общего количества
pivot_percent['Доля данных с указанием box_office от общего количества'] = \
    round((pivot_percent['Количество данных с box_office']/pivot_percent['Общее количество'])*100, 1)
```

```
In [94]: # построим график изменения доли указанных box_office по годам
pivot_percent['Доля данных с указанием box_office от общего количества'].plot(kind = 'bar', \
    grid = True, legend = False, title = "Доля прокатных удостоверений с указанием кассового сбора в %", \
    xlabel = 'год выдачи', ylabel = 'доля ', figsize = (13,3) );
display(pivot_percent)
```

year_premier	Количество данных с box_office	Общее количество	Доля данных с указанием box_office от общего количества
2010	101	980	10.30
2011	109	622	17.50
2012	123	588	20.90
2013	179	625	28.60
2014	273	800	34.10
2015	462	702	65.80
2016	524	816	64.20
2017	357	505	70.70
2018	475	891	53.30
2019	530	930	57.00



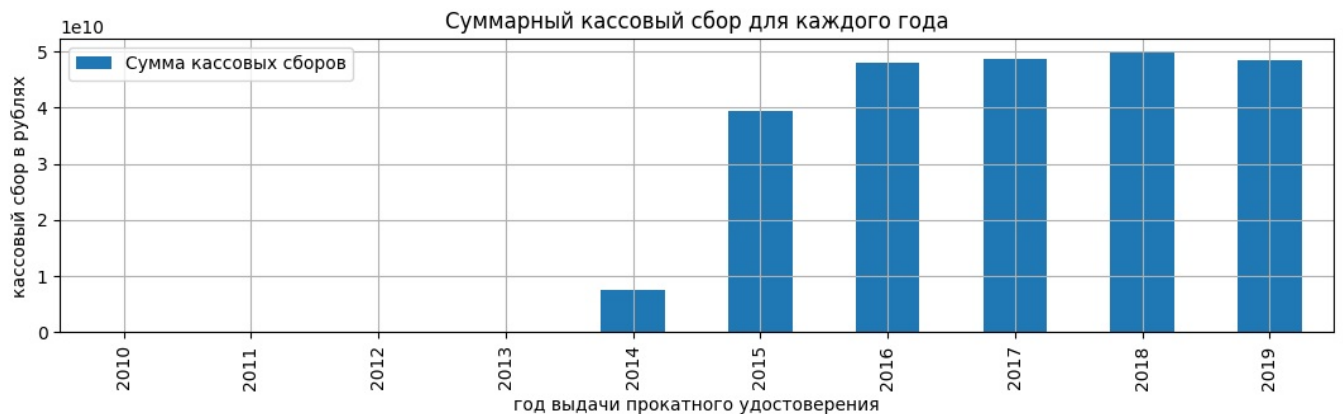
В процентном соотношении доля прокатных удостоверений с кассовыми сборами уверенно растет, наблюдается резкий скачок в 2015 году, максимум наблюдаем в 2017г. Если посмотреть на количественную представленность, то прокат без кассового сбора имеет в этом году минимум в количественном выражении. Можно сделать предположение о некоторых изменениях в выдаче прокатных удостоверений в целом в 2017 году, что привело к снижению их количества. Возможно изменение законодательства или наложение более строгих ограничений.

- Исследуем динамику продаж по годам:

```
In [95]: #создадим сводную таблицу и посчитаем суммарный кассовый сбор для каждого года
dinam_year = data.pivot_table(index = 'year_premier', values = 'box_office', aggfunc = 'sum')\
.rename(columns = {'box_office':'Сумма кассовых сборов'})
display(dinam_year)
```

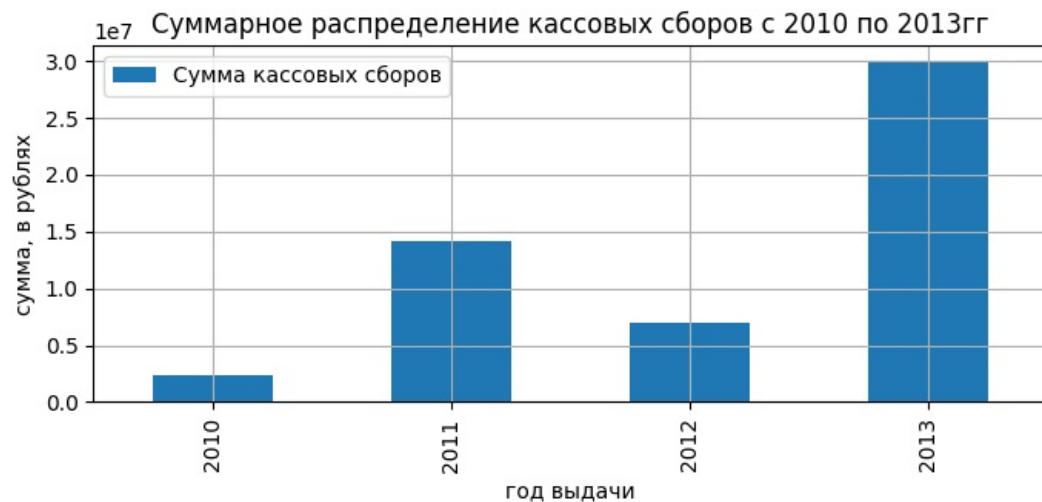
Сумма кассовых сборов	
year_premier	
2010	2,428,654.00
2011	14,102,765.00
2012	6,955,423.00
2013	29,799,706.20
2014	7,444,951,859.20
2015	39,497,365,196.40
2016	47,866,105,214.91
2017	48,563,707,217.51
2018	49,668,403,134.32
2019	48,425,708,779.59

```
In [96]: # построим график суммарного кассового сбора
dinam_year.plot(kind = 'bar',grid = True, title = 'Суммарный кассовый сбор для каждого года',\
xlabel = 'год выдачи прокатного удостоверения', ylabel = 'кассовый сбор в рублях', figsize = (1.
```



Как и предполагалось на графике видим провал до 2013 года.Из данных сводной таблицы можно увидеть, что до 2013 года включительно порядок исчисления иной. Попробуем проанализировать информацию в разрезе 2 периодов.

```
In [97]: # ввиду малых значений сборов в 2010-2013 построим дополнительную диаграмму
dinam_year.reset_index().query('year_premier < 2014').plot(kind = 'bar', x = 'year_premier',\
y = 'Сумма кассовых сборов', grid = True, title = 'Суммарное распределение кассовых сборов с 2010 по 2013гг',\
xlabel = 'год выдачи', ylabel = 'сумма, в рублях', figsize = (8,3));
```



В разрезе данных до 2013 года включительно, максимальный сбор был в 2013 году,этот год показывает и максимальный рост, минимальный сумма сборов была в 2010г

```
In [98]: # найдем год с максимальной суммой кассового сбора
print("Год с минимальным суммарным значение кассового сбора: ",dinam_year['Сумма кассовых сборов'].idxmin(),\
```

```

"Сумма сборов: ", dinam_year['Сумма кассовых сборов'].min(), sep = '\n' )
print()
# найдем год с минимальной суммой кассового сбора
print("Год с максимальным суммарным значение кассового сбора: ", dinam_year['Сумма кассовых сборов'].idxmax(), \
      "Сумма сборов: ", dinam_year['Сумма кассовых сборов'].max(), sep = '\n')

```

Год с минимальным суммарным значение кассового сбора:  
 2010  
 Сумма сборов:  
 2428654.0

Год с максимальным суммарным значение кассового сбора:  
 2018  
 Сумма сборов:  
 49668403134.31999

Суммарный кассовый сбор начиная с 2015г. показывает примерно один уровень, максимум приходится на 2018 год. В 2010 - 2013 гг суммарный кассовый сбор был ничтожно мал по сравнению с периодом после 2014г и рассматривается отдельно, ввиду разности в методе исчисления. Минимальная сумма получена в 2010г, максимум в 2013.

- Посчитаем среднюю и медианную стоимость проката по годам

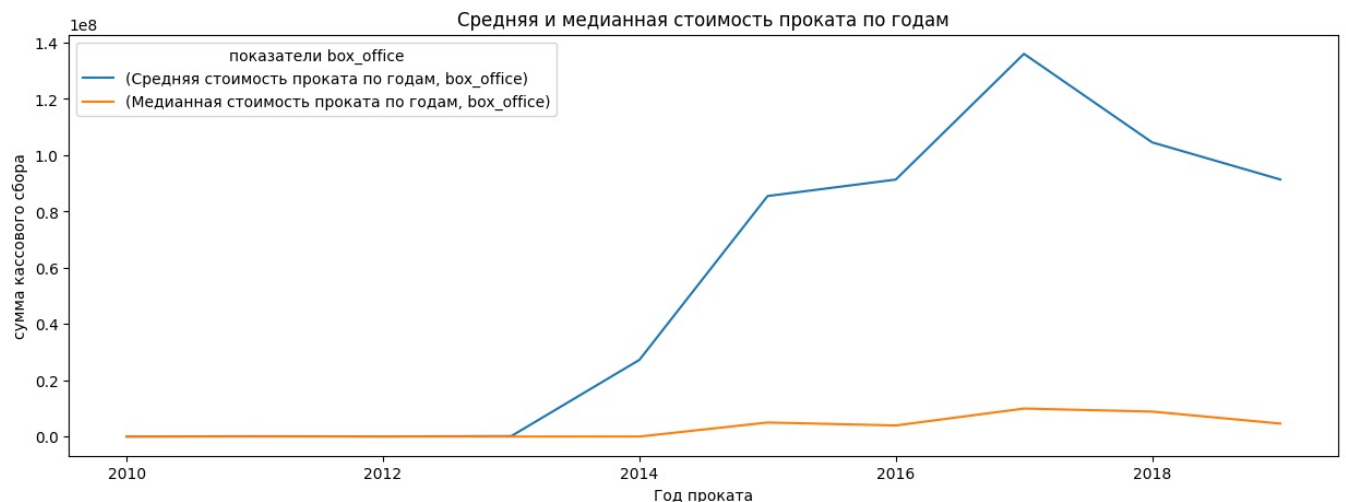
Рассмотрим среднюю и медианную стоимость проката по годам, стоит обратить внимание на разность представленных данных и учитывать порядок подсчета для периода до 2014 и после.

```

In [99]: # создадим сводную таблицу сгруппировав данные по году выпуска и посчитаем среднюю и медианную стоимость кассовых
data_med = data.query('box_office.isna() == False')\
.pivot_table(index = 'year_premier', values = 'box_office', aggfunc = ['mean', 'median']).reset_index()\
.rename(columns = {'year_premier': 'Год проката', 'mean': 'Средняя стоимость проката по годам', \
                  'median': 'Медианная стоимость проката по годам'})
display(data_med)
# построим график медианной и средней стоимости по годам
data_med.plot(x = 'Год проката', y = ['Средняя стоимость проката по годам', 'Медианная стоимость проката по годам'], \
              title = 'Средняя и медианная стоимость проката по годам', figsize = (15, 5));
plt.legend(title = 'показатели box_office')
plt.ylabel('сумма кассового сбора');

```

	Год проката	Средняя стоимость проката по годам	Медианная стоимость проката по годам
		box_office	box_office
0	2010	24,046.08	1,710.00
1	2011	129,383.17	3,000.00
2	2012	56,548.15	6,220.00
3	2013	166,478.81	3,580.00
4	2014	27,270,885.93	20,400.00
5	2015	85,492,132.46	5,003,450.15
6	2016	91,347,529.04	3,957,926.01
7	2017	136,032,793.33	9,968,340.00
8	2018	104,565,059.23	8,891,102.21
9	2019	91,369,261.85	4,627,798.34



Ожидаемо видим провал в данных до 2014г, обратившись к таблице можно увидеть числовые значения и определить, что в 2010 году медианная стоимость была минимальной, как и средняя. Максимальная медианная была в 2012, средняя же в 2013 году. Разница между медианой и средним арифметическим достаточно большая, что говорит о наличии сверх больших кассовых сборов. После 2014 года максимальная средняя и медианная стоимость приходится на 2017 год, минимум на 2014 год.



- Исследование влияния возрастного ограничения на кассовые сборы после 2014г

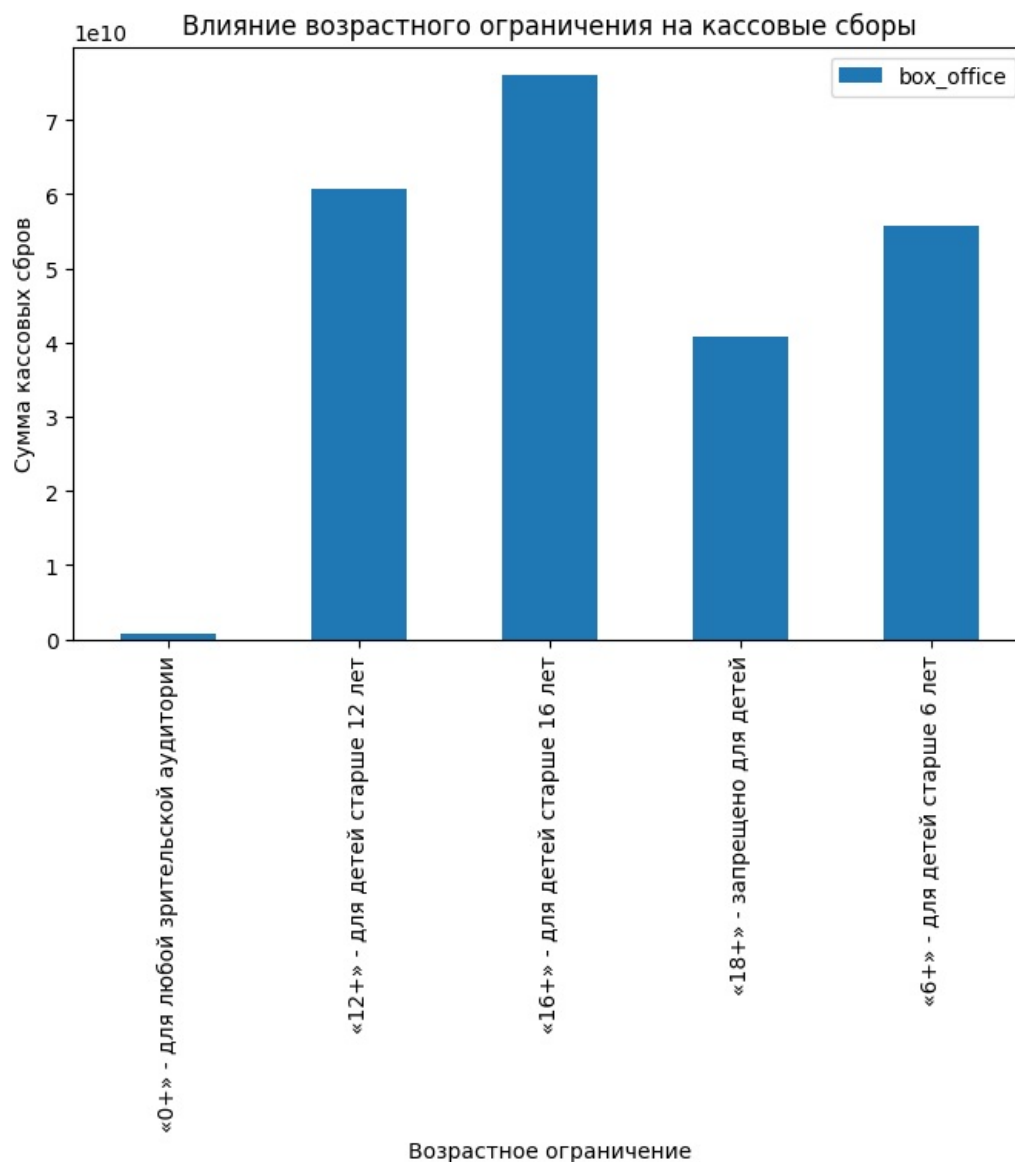
```
In [100]: # сгруппировав данные по возрастному ограничению, посчитаем суммарный кассовый сбор, медианная стоимость и общее количество
data_age = data.query('year_premier > 2014')\
.pivot_table(index = 'age_restriction', values = 'box_office', aggfunc = ['sum', 'median', 'count']).reset_index()
data_age.rename(columns = {'age_restriction': 'Возрастное ограничение', 'sum': 'Суммарный кассовый сбор', \
                           'median': 'Медианная стоимость кассовых сборов', 'count': 'Общее количество'})
data_age.sort_values(by = ('Общее количество', 'box_office'))
```

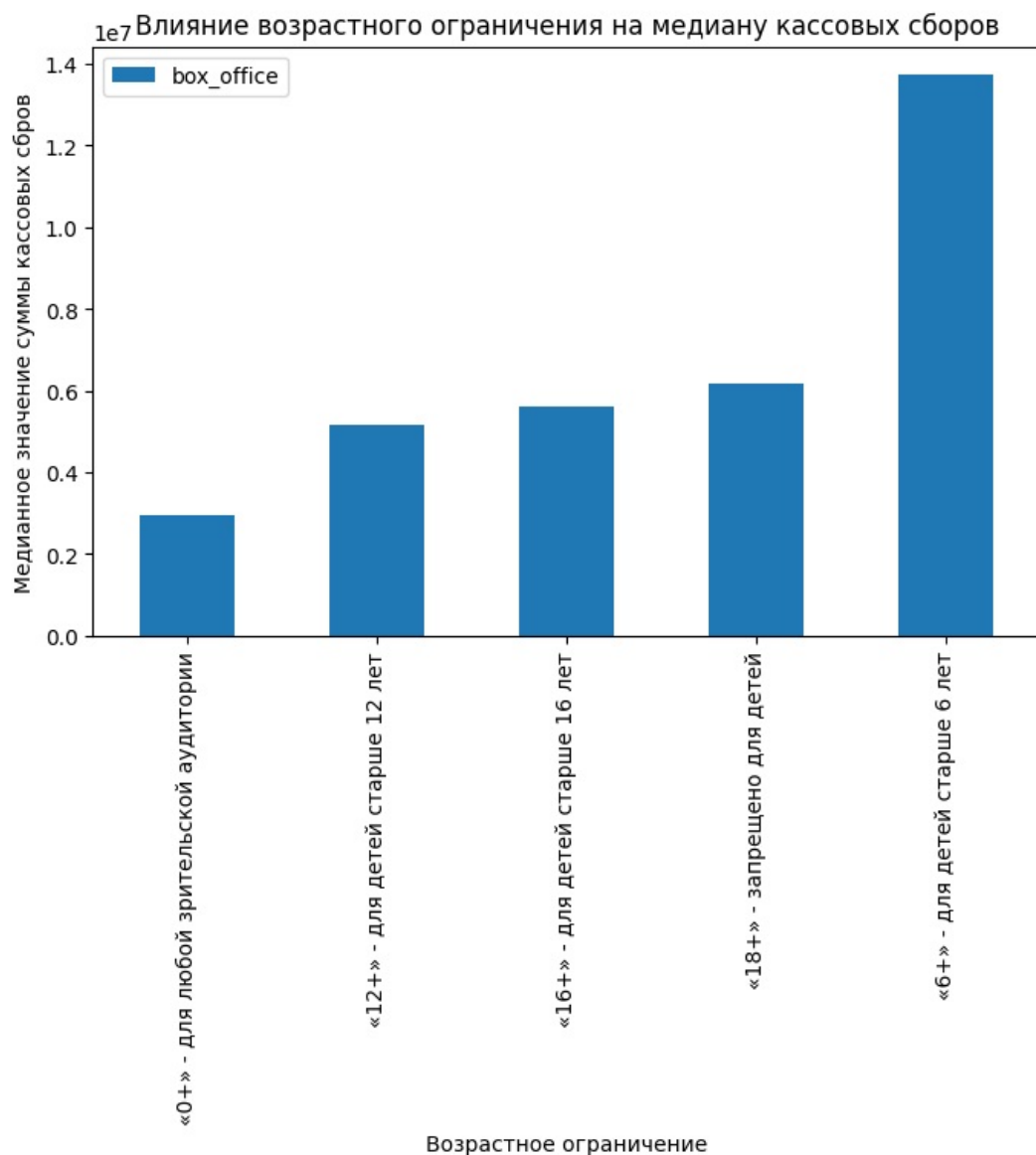
Out[100]:

	Возрастное ограничение	Суммарный кассовый сбор	Медианная стоимость кассовых сборов	Общее количество
		box_office	box_office	box_office
0	«0+» - для любой зрительской аудитории	808,882,898.97	2,955,934.60	89
4	«6+» - для детей старше 6 лет	55,798,610,799.68	13,736,207.98	317
1	«12+» - для детей старше 12 лет	60,619,446,628.33	5,175,580.50	347
3	«18+» - запрещено для детей	40,759,615,572.12	6,154,439.17	792
2	«16+» - для детей старше 16 лет	76,034,733,643.63	5,604,702.00	803

```
In [101]: data_age.plot(kind = 'bar', x = 'Возрастное ограничение', y = 'Суммарный кассовый сбор', figsize = (8,5))
plt.title("Влияние возрастного ограничения на кассовые сборы")
plt.ylabel('Сумма кассовых сборов')

data_age.plot(kind = 'bar', x = 'Возрастное ограничение', y = "Медианная стоимость кассовых сборов", figsize = (8,5))
plt.title("Влияние возрастного ограничения на медиану кассовых сборов")
plt.ylabel('Медианное значение суммы кассовых сборов');
```





Фильмы для любой зрительской аудитории имеют самые низкие сборы, как по суммарному, так и медианному значениям, количество также является наименьшим. Самый высокий суммарный кассовый сбор у возрастной группы 16+, учитывая средний уровень медианной стоимости для этой группы, можно сказать о влиянии на показатель количества фильмов с таким возрастным ограничением. Наибольшее медианное значение у фильмов 6+ при среднем количестве и суммарной годовой стоимости очевидно говорит о высокой посещаемости (либо стоимости входных билетов). Количество фильмов 18+ хотя и показывает большое значение, но суммарный кассовый сбор один из самых низких.

## Исследовательский анализ данных фильмов с государственной поддержкой

Создадим новую переменную, где оставим только фильмы с государственной поддержкой

```
In [102.. gov_data = data.loc[data['financing_source'] != 'Иное финансирование',:]
```

```
In [103.. gov_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 332 entries, 1281 to 7478
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   title                  332 non-null   object
1   puNumber               332 non-null   float64
2   show_start_date        332 non-null   datetime64[ns, UTC]
3   type                   332 non-null   object
4   film_studio            332 non-null   object
5   production_country      332 non-null   object
6   director               332 non-null   object
7   producer               330 non-null   object
8   age_restriction         332 non-null   object
9   refundable_support     332 non-null   float64
10  nonrefundable_support  332 non-null   float64
11  budget                 332 non-null   float64
12  financing_source        332 non-null   object
13  ratings                 316 non-null   float64
14  genres                  332 non-null   object
15  box_office              318 non-null   float64
16  year_premier            332 non-null   int64
17  main_director           332 non-null   object
18  main_genres              332 non-null   object
19  gov_part                332 non-null   float64
dtypes: datetime64[ns, UTC](1), float64(7), int64(1), object(11)
memory usage: 54.5+ KB
```

```
In [104.. # удалим строки с пропусками в 'box_office'
gov_data = gov_data.dropna(subset = ['box_office'])
```

```
In [105.. print('Процент потерь:', round((14/332)*100,1), '%')
```

Процент потерь: 4.2 %

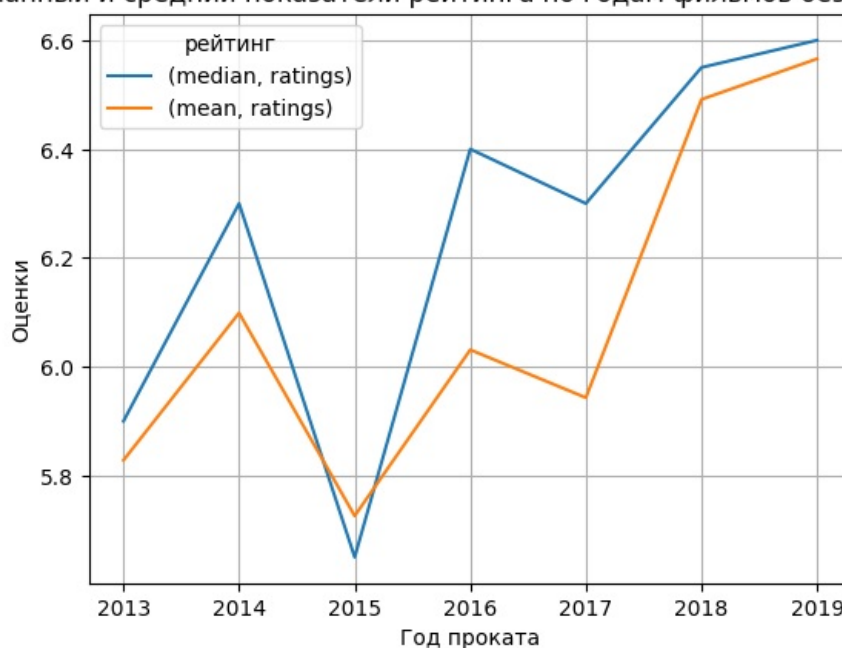
Сделаем некоторую визуальную оценку фильмам без господдержки, произведенным в России после 2012г

```
In [106.. # рассчитаем количество фильмов России без господдержки после 2013 года
data.loc[(data['production_country'] == 'Россия') & (data['budget'].isna()) & (data['year_premier'] > 2012)][ 'box_o'
.count()
```

Out[106]: 345

```
In [107.. # построим график зависимости рейтинга от года проката фильмов без господдержки после 2012
data.loc[(data['production_country'] == 'Россия') & (data['budget'].isna()) & (data['year_premier'] > 2012)]\
.pivot_table(index = 'year_premier', values = 'ratings', aggfunc = ['median', 'mean'])\
.plot(grid = True, legend = True, title = 'Медианный и средний показатели рейтинга по годам фильмов без поддержки',
xlabel = 'Год проката', ylabel = "Оценки")
plt.legend(title = 'рейтинг');
```

Медианный и средний показатели рейтинга по годам фильмов без поддержки



```
In [108.. # создадим таблицу зависимости показателей кассового сбора от года проката фильмов России без господдержки после
non_gov = data.loc[(data['production_country'] == 'Россия') & (data['budget'].isna()) & (data['year_premier'] > 2012)]\
.pivot_table(index = "year_premier", values = 'box_office', aggfunc = ['mean', 'median', 'sum'])\
.reset_index()\
.rename(columns = {"year_premier": 'год проката', 'mean': 'Средний показатель кассового сбора', \
```

```
'median':"Медианный показатель кассового сбора",'sum':"Суммарный вес кассовых сборов"})
```

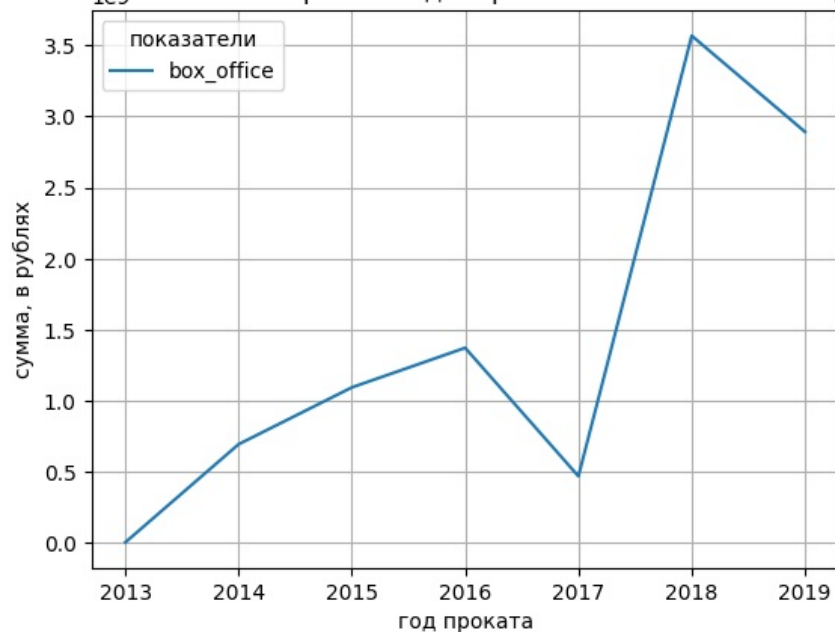
Out[108]:

год проката	Средний показатель кассового сбора	Медианный показатель кассового сбора	Суммарный вес кассовых сборов
		box_office	box_office
0	2013	58,180.31	2,734,474.60
1	2014	12,376,166.54	693,065,326.30
2	2015	23,764,094.56	1,093,148,349.85
3	2016	18,785,497.14	1,371,341,291.58
4	2017	36,011,324.88	468,147,223.46
5	2018	59,461,518.99	3,567,691,139.38
6	2019	57,835,955.50	2,891,797,774.92

In [109...

```
# построим график зависимости показателей кассового сбора от года проката фильмов России без господдержки после 2012г
non_gov.plot( x = "год проката", y = "Суммарный вес кассовых сборов",\
              grid = True,\
              ylabel = 'сумма, в рублях',\
              title = 'Суммарный показатель кассовых сборов по годам фильмов России без господдержки после 2012г')
plt.legend(title = 'показатели');
```

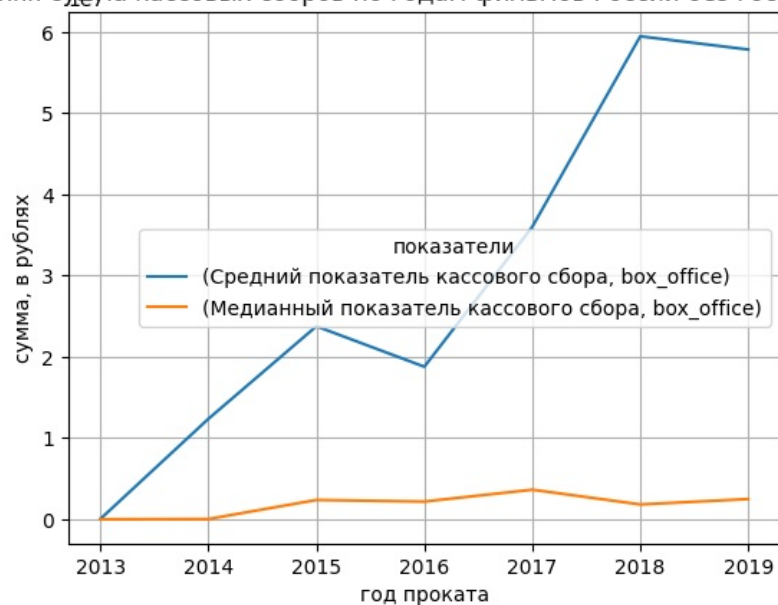
Суммарный показатель кассовых сборов по годам фильмов России без господдержки после 2012г



In [110...

```
# построим график зависимости показателей кассового сбора от года проката фильмов России без господдержки после 2012 года
non_gov.plot( x = "год проката",y = ['Средний показатель кассового сбора','Медианный показатель кассового сбора'],
              grid = True,\
              ylabel = 'сумма, в рублях',\
              title = 'Медианная и средняя сумма кассовых сборов по годам фильмов России без господдержки после 2012 года',\
              plt.legend(title = 'показатели'));
```

## Медианная и средняя сумма кассовых сборов по годам фильмов России без господдержки после 2012г



Данные российского кино без господдержки сравним с фильмами, финансирующимися госструктурами для общего вывода.

- Оценим количество фильмов с господдержкой относительно года выдачи

```
In [111]: #построим диаграмму зависимости года и количества прокатных удостоверений, рейтинга фильмов
gov_data.pivot_table(index = 'year_premier', values = 'puNumber', aggfunc = 'count')\
.rename(columns = {'puNumber': 'Количество выданных удостоверений'})\
.plot(kind = 'bar', grid = True, legend = True, title = 'Количество выданных прокатных удостоверений'\
, xlabel = 'Год проката', ylabel = "Количество единиц");

gov_data.pivot_table(index = 'year_premier', values = 'ratings', aggfunc = ['median', 'mean'])\
.rename(columns = {'median': 'Медианный рейтинг фильмов', 'mean': 'Средний рейтинг фильмов'}, level = 0)\
.plot(grid = True, legend = True, title = 'Медианный и средний показатели рейтинга по годам'\
, xlabel = 'Год проката', ylabel = "Оценки")
plt.legend(title = 'показатели рейтинга');
```





```
In [112]: gov_data['ratings'].describe()
```

```
Out[112]: count    303.00
mean         6.02
std          1.16
min           1.00
25%          5.35
50%          6.20
75%          6.70
max           9.40
Name: ratings, dtype: float64
```

Наибольшее количество фильмов с господдержкой вышло в 2015 году, более 80 штук, наименьшее в 2013. При этом фильмы в 2013 получили в среднем и медианном значениями наиболее высокую оценку зрителей, что скорее всего связано со сверхмалой выборкой. 2016 год показывает самые низкие баллы рейтинга.

```
In [113]: # создадим переменную, в которую запишем сводную таблицу, сгруппировав данные по году, где значения это
# средний, медианный и суммарный показатели
gov_box = gov_data.pivot_table(index = "year_premier", values = 'box_office', aggfunc = ['mean', 'median', 'sum'])\
.reset_index()\
.rename(columns = {"year_premier": "год проката", "mean": "Средний показатель кассового сбора", \
                    "median": "Медианный показатель кассового сбора", "sum": "Суммарный вес кассовых сборов"})
gov_box
```

```
Out[113]:
```

	год проката	Средний показатель кассового сбора	Медианный показатель кассового сбора	Суммарный вес кассовых сборов
		box_office	box_office	box_office
0	2013	10,161,192.55	10,161,192.55	20,322,385.10
1	2014	63,588,885.41	5,504,517.85	1,017,422,166.60
2	2015	68,062,181.39	9,200,824.50	5,785,285,418.14
3	2016	101,361,797.32	17,921,115.61	6,081,707,839.10
4	2017	278,591,674.38	50,451,949.00	10,865,075,300.96
5	2018	177,394,089.47	27,023,390.31	9,934,069,010.25
6	2019	140,160,457.58	25,977,291.30	8,409,627,454.63

```
In [114]: # построим график зависимости суммарного кассового сбора года проката
gov_box.plot( x = "год проката", y = "Суммарный вес кассовых сборов", grid = True, \
              ylabel = 'сумма, в рублях', title = 'Общая сумма кассовых сборов');
```

```
# построим график зависимости медианного и среднего показателя от года проката
gov_box.plot( x = "год проката", y = ['Средний показатель кассового сбора', "Медианный показатель кассового сбора"],
              grid = True, \
              ylabel = 'сумма, в рублях', title = 'Медианная и средняя сумма кассовых сборов по годам',)
plt.legend(title = 'показатели');
```



```
In [115]: print("Самый высокий суммарный кассовый сбор:", int(gov_box['Суммарный вес кассовых сборов'].max()), \
            'в', int(gov_box['год проката'][gov_box['Суммарный вес кассовых сборов'].idxmax()], 'году')
print()
print("Самый высокий медианный показатель кассового сбора:", int(gov_box['Медианный показатель кассового сбора'].max()), \
            'в', int(gov_box['год проката'][gov_box['Медианный показатель кассового сбора'].idxmax()], 'году')
print()
print("Самый высокий средний показатель кассового сбора:", int(gov_box['Средний показатель кассового сбора'].max()), \
            'в', int(gov_box['год проката'][gov_box['Средний показатель кассового сбора'].idxmax()], 'году')
```

Самый высокий суммарный кассовый сбор: 10865075300 в 2017 году

Самый высокий медианный показатель кассового сбора: 50451949 в 2017 году

Самый высокий средний показатель кассового сбора: 278591674 в 2017 году

Самый высокий суммарный кассовый сбор, как и медианное и среднее значение кассового сбора приходятся на 2017 год, в величину прокатных удостоверений в 40 единиц. Количественный пик приходящийся на 2016 год показывает средний уровень показателей кассового сбора. Минимум у 2013 года, что очевидно связано с количеством вышедших в прокат фильмов.

- Анализ фильмов с государственной поддержкой в разрезе возрастного ограничения

```
In [116]: # создадим сводную таблицу, сгруппировав данные по возрастному ограничению. значениями выберем кассовый сбор и i
```

```

# создадим среднюю сумму, для группировки данных по возрастному ограничению, для оценки среднего кассового сбора
# количество, медианное значение и сумму, для рейтинга - среднее арифметическое
gov_age = gov_data.pivot_table(index = 'age_restriction', values = ['box_office', 'ratings'], \
aggfunc = {'box_office': ['sum', 'count', 'median'], 'ratings': 'mean'}).reset_index()

gov_age['ratings'] = round(gov_age['ratings'], 1)
gov_age

```

Out[116]:

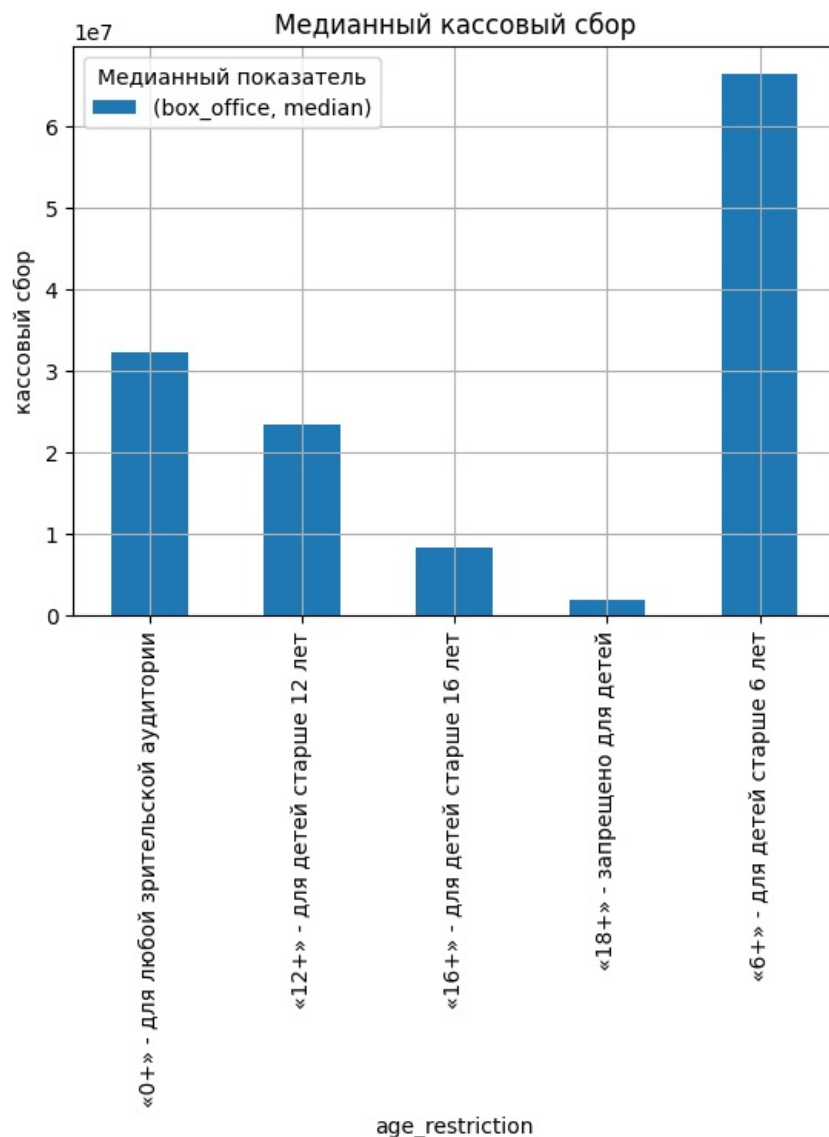
	age_restriction			box_office	ratings
		count	median	sum	mean
0	«0+» - для любой зрительской аудитории	5	32,244,810.00	552,932,752.50	5.70
1	«12+» - для детей старше 12 лет	100	23,365,689.49	19,957,963,755.94	6.10
2	«16+» - для детей старше 16 лет	120	8,195,307.25	7,658,462,636.93	6.00
3	«18+» - запрещено для детей	28	1,828,557.75	560,158,730.15	6.30
4	«6+» - для детей старше 6 лет	65	66,504,721.61	13,383,991,699.26	6.00

In [117...]

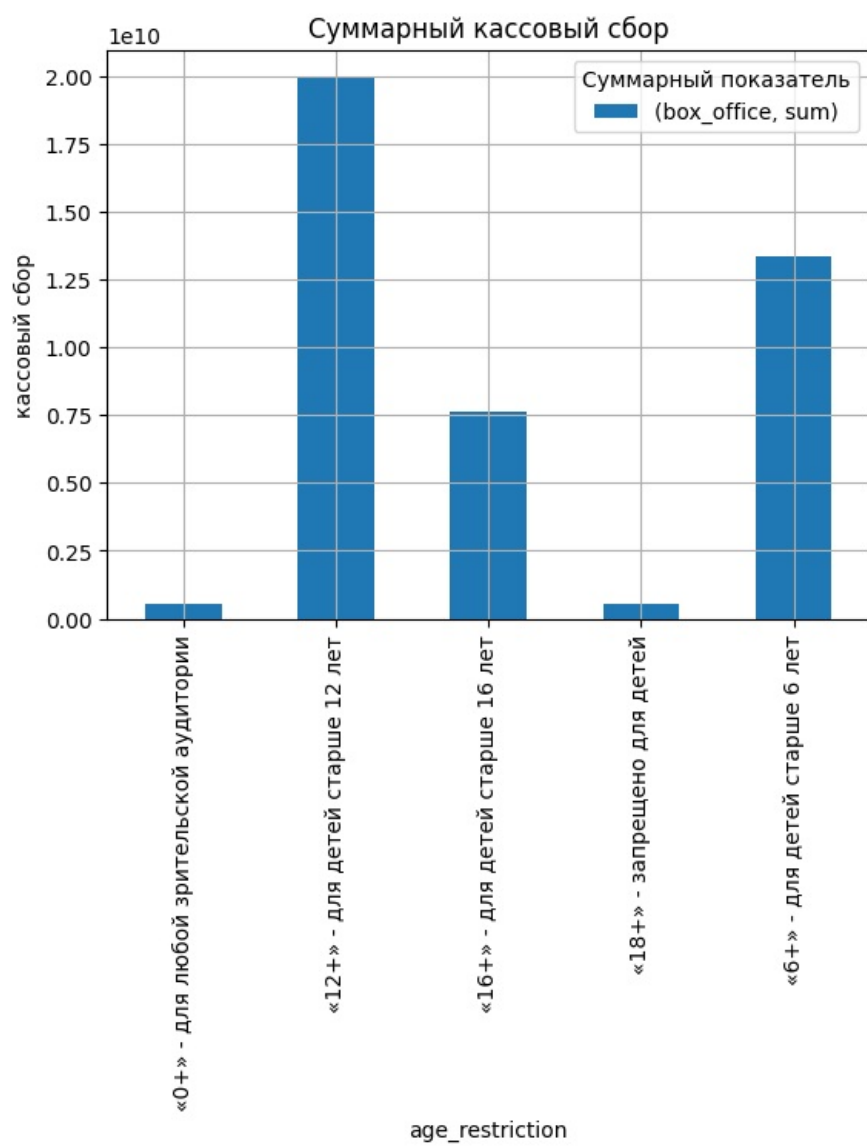
```

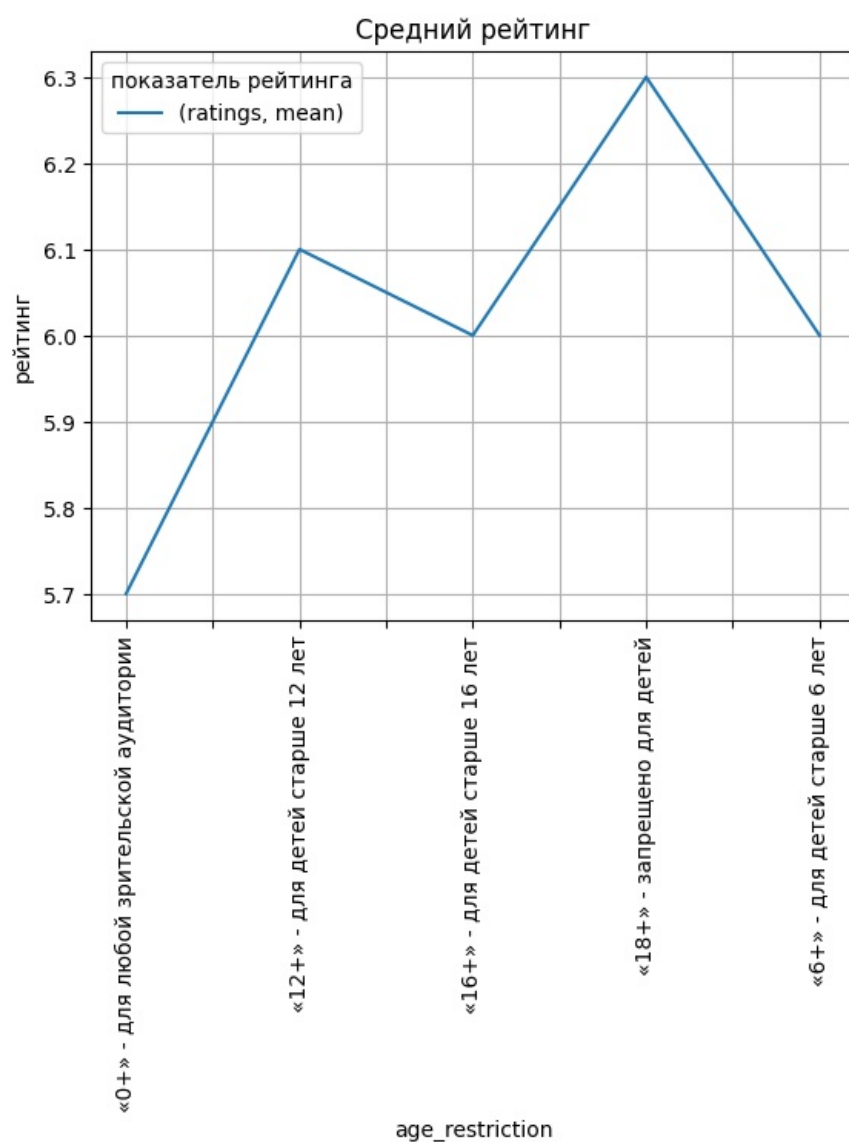
# построим график зависимости медианного показателя от возрастного ограничения
gov_age.plot(kind = 'bar', x = 'age_restriction', y = ('box_office', 'median'), grid = True, \
title = 'Медианный кассовый сбор')
plt.ylabel('кассовый сбор')
plt.legend(title = 'Медианный показатель');
# построим график зависимости суммарного кассового сбора и возрастного ограничения
gov_age.plot(kind = 'bar', x = 'age_restriction', y = ('box_office', 'sum'), grid = True, title = 'Суммарный кассовый сбор')
plt.ylabel('кассовый сбор')
plt.legend(title = 'Суммарный показатель');
# построим график зависимости рейтинга и возрастного ограничения
gov_age.plot(x = 'age_restriction', y = ('ratings', 'mean'), title = 'Средний рейтинг', rot = 90, grid = True)
plt.ylabel('рейтинг')
plt.legend(title = 'показатель рейтинга');

```





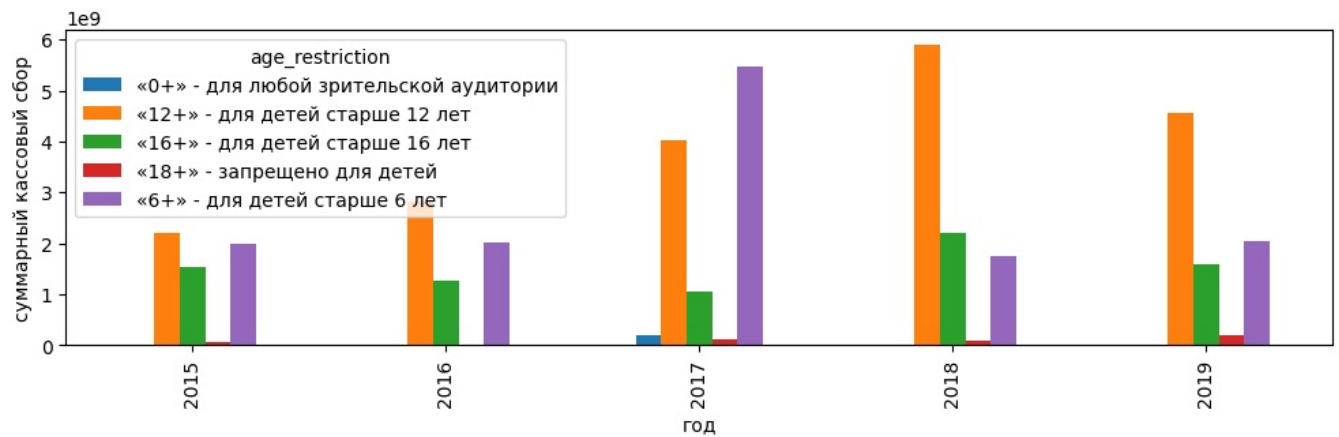




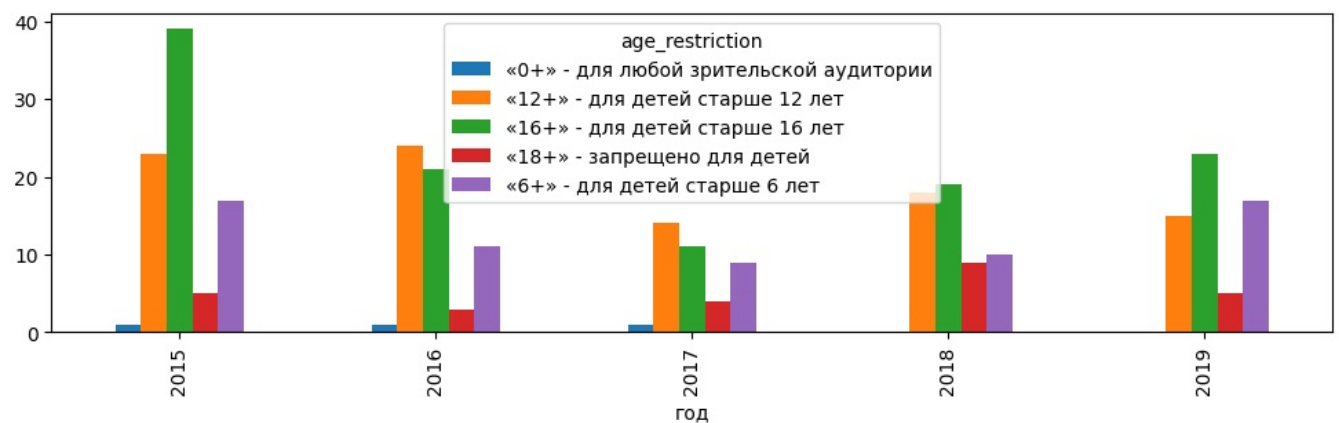
Рассмотрим динамику изменения суммарного показателя кассового сбора по годам в разрезе возрастного ограничения

In [118... #создадим сводную таблицу, сгруппировав данные по году, субкатегориями будут возрастные ограничения, данными - с  
# кассовый сбор, построим диаграмму  
gov\_data.query('year\_premier > 2014')\

```
.pivot_table(index = 'year_premier', columns = 'age_restriction', values = 'box_office', aggfunc = 'sum') \
.plot(kind = 'bar', rot = 90, figsize = (12,3), ylabel = 'суммарный кассовый сбор', xlabel = 'год' );
```



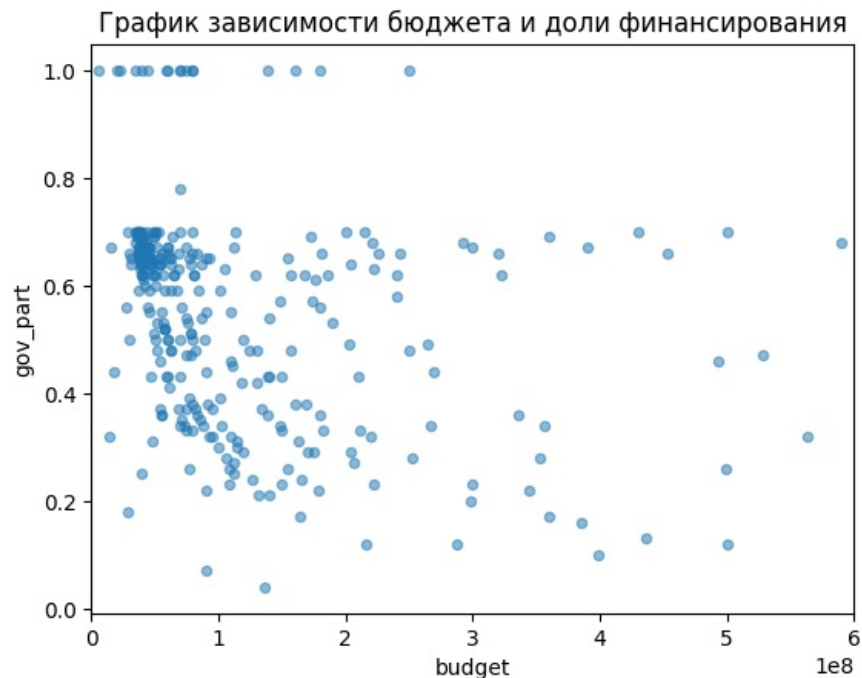
```
In [119... # сводная таблица сгруппированных по году данных в разрезе возрастного ограничения, где значения это количество
gov_data.query('year_premier > 2014') \
.pivot_table(index = 'year_premier', columns = 'age_restriction', values = 'puNumber', aggfunc = 'count') \
.plot(kind = 'bar', rot = 90, figsize = (12,3), xlabel = 'год');
```



Больше всего фильмов снимается с категорией 12 и 16+. Суммарный кассовый сбор ожидаемо у категории 12+. При этом самый высокий медианный показатель у категории 6+, при количестве фильмов в 65 единиц и довольно низким рейтингом. Меньше всего фильмов с возрастной категорией - для любого возраста, отсюда самый низкий суммарный показатель, как и у фильмов категории 18+, которые в свою очередь показывают самый низкий медианный кассовый сбор при количестве выпущенных фильмов в 28 штук и рейтингом выше среднего. В 2017 г. произошел рост суммарного кассового сбора для категории 6+ при самом невысоком количестве выпущенных фильмов. Фильмы 12+ от года к году показывают хорошие показатели суммарного кассового сбора, пик приходится на 2018 год. Фильмы 0+ и 18+ имеют низкие кассовые сборы от года к году, самый высокий показатель в категории 0+ - 2017г, а в 2019г - 18+. До 2017 года большой суммарный сбор фильмов 6+ категории, после 2017 резкое изменение в сторону фильмов 12+.

- Рассмотрим бюджет фильмов с государственной поддержкой, его возвратную и невозвратную части

```
In [120... # построим график зависимости бюджета и доли финансирования
gov_data.plot(kind = 'scatter', x = 'budget', y = 'gov_part', xlim = (0,600000000), \
title = 'График зависимости бюджета и доли финансирования', alpha = 0.5);
```



Можно заметить, что большинство фильмов имеют господдержку от 0,2 до 0,7 доли от общего бюджета

```
In [121]: gov_data['gov_part'].describe()
```

```
Out[121]: count    318.00
          mean      0.54
          std       0.20
          min       0.04
          25%       0.38
          50%       0.61
          75%       0.66
          max       1.00
          Name: gov_part, dtype: float64
```

Медианное значение доли госфинансирования составляет около 62%, максимальное - 100%

```
In [122]: # построим график зависимости кассового сбора и доли финансирования госструктурами
gov_data.plot(kind = 'scatter', x = 'gov_part', y = 'box_office', \
              title = 'График зависимости кассового сбора и доли госфинансирования', alpha = 0.5);
```



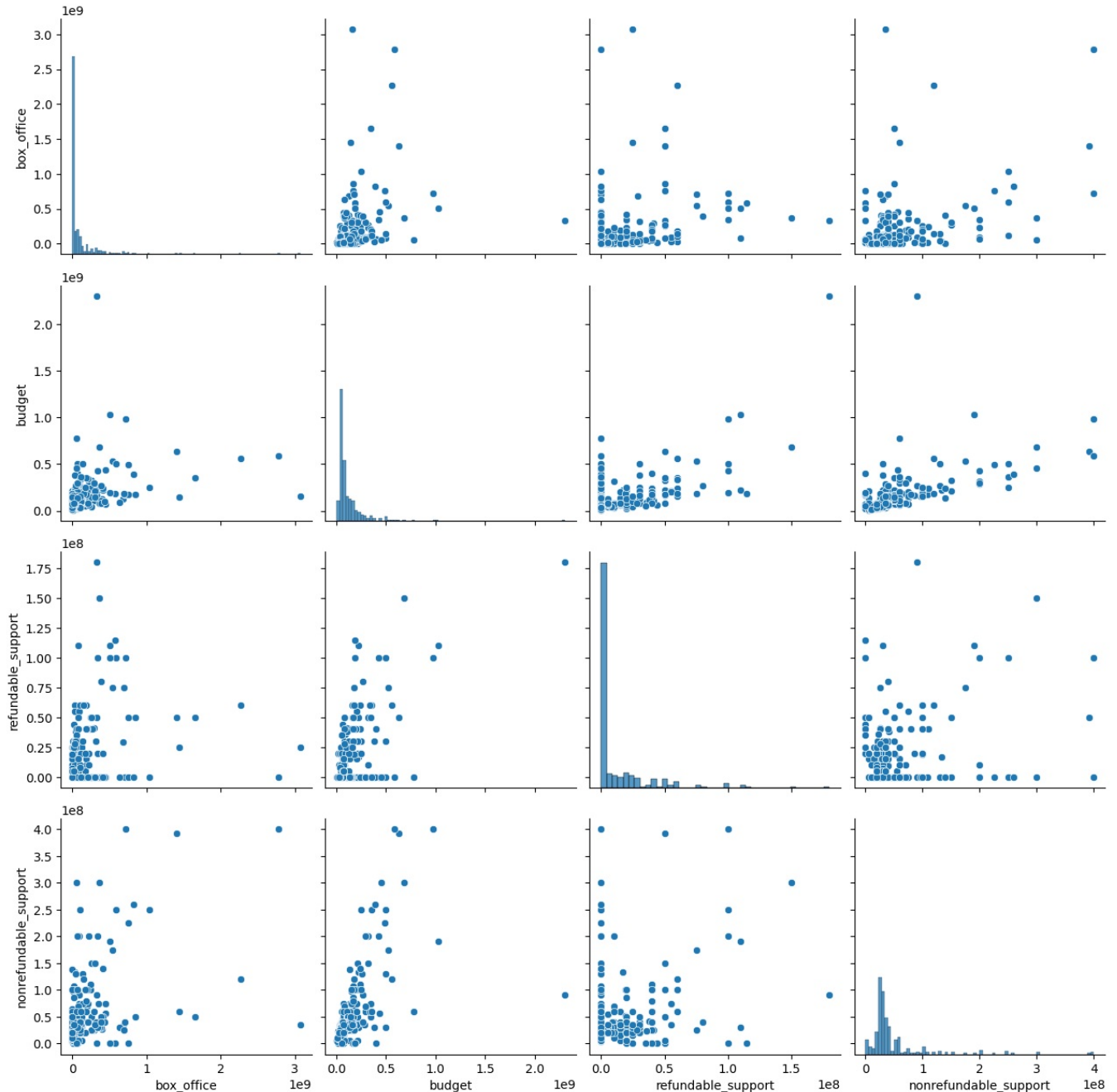
```
In [123]: print('Коэффициент корреляции кассового сбора и общего бюджета', round(gov_data['budget'].corr(gov_data['box_of
print('Коэффициент корреляции кассового сбора и доли финансирования', round(gov_data['gov_part']\
          .corr(gov_data['box_office']),2))
print('Коэффициент корреляции невозвратной части бюджета и общего бюджета', round(gov_data['budget']\
          .corr(gov_data['nonrefundable_support']
print('Коэффициент корреляции возвратной части бюджета и общего бюджета', round(gov_data['budget']\
          .corr(gov_data['refundable_support']),2
```

Коэффициент корреляции кассового сбора и общего бюджета 0.38  
 Коэффициент корреляции кассового сбора и доли финансирования -0.08  
 Коэффициент корреляции невозвратной части бюджета и общего бюджета 0.59  
 Коэффициент корреляции возвратной части бюджета и общего бюджета 0.64

Наблюдается слабая положительная корреляция между бюджетом фильма и кассовыми сборами. Очевидно, что возвратная и невозвратная части бюджета имеют положительную корреляцию с общим бюджетом. Коэффициент финансирования и кассовые сборы имеют нулевую корреляцию, а значит, успешность фильма и доля финансирования никак не зависят друг от друга.

Рассмотрим графики корреляции для бюджета, его частей и кассового сбора

```
In [124]: sns.pairplot(gov_data[['box_office', 'budget', 'refundable_support', 'nonrefundable_support']])
plt.gcf().set_size_inches(13,13);
```



Чаще фильмы с господдержкой не имеют возвратной части бюджета, невозвратная часть чаще составляет сумму до 50 млн. рублей и зависит от бюджета фильма. Возвратная часть бюджета связана с бюджетом, на графике видим положительную связь. Кассовые сборы фильмов и бюджет имеют слабую положительную корреляцию, как и возвратная и невозвратная части бюджета.

- Анализ успешности фильмов финансируемых государством

Посчитаем количество фильмов провалившихся в прокате, т.е. тех, чей бюджет превышает кассовый сбор

```
In [125]: down = gov_data.loc[gov_data['box_office'] < gov_data['budget']]['puNumber'].count()
total = gov_data['budget'].count()
print('Доля провалившихся в прокате фильмов с государственной поддержкой:', round((down/total*100),0), '%')
```

Доля провалившихся в прокате фильмов с государственной поддержкой: 75.0 %

Внушительная часть фильмов провалилась в прокате, лишь четверть фильмов окупили свой бюджет

```
In [126... # добавим столбец с информацией о разнице кассового сбора и бюджета
gov_data['difference'] = gov_data['box_office'] - gov_data['budget']
# рассчитаем коэффициент результативности
gov_data['perc_income'] = round(gov_data['box_office']/gov_data['budget'],2)
```

```
In [127... # посмотрим на 5 самых провальных фильмов
gov_data.sort_values(by = 'difference').head(5)
```

Out[127]:	title	puNumber	show_start_date	type	film_studio	production_country	director	producer	age
7114	Тайна Печати Дракона	111,012,819.00	2019-09-12 12:00:00+00:00	Художественный	ООО "Кинокомпания "СТВ"; ООО "РФГ Корп"; ООО "...	Китай - Россия	О.Степченко	С.Сельянов, А.Петрухин	д
7171	Грех	111,014,419.00	2019-08-31 12:00:00+00:00	Художественный	НО Фонд Андрея Кончаловского по поддержке кино...	Италия - Россия	А.Михалков (псевдоним А.Кончаловский)	НО Фонд Андрея Кончаловского по поддержке кино...	д
5437	Матильда	111,005,717.00	2017-08-17 12:00:00+00:00	Художественный	ООО "ТПО "РОК", ООО "Матильда"	Россия	А.Учитель	А.Учитель, А.Достман, В.Винокур	д
2142	Территория	111,000,515.00	2015-01-27 12:00:00+00:00	Художественный	ООО "Творческое объединение "Кинокомпания "Анд...	Россия	А.Мельник	А.Мельник	д
6887	Девятая	111,007,019.00	2019-09-19 12:00:00+00:00	Художественный	ООО "Нон-Стоп Продакшн"	Россия	Н.Хомерики	С.Мелькумов, А.Роднянский	д

Антирейтинг фильмов по успешности проката возглавляет фильм "Тайна печати Дракона", с суммой потерь в 1971477000 рублей.После идут 'Грех' и 'Матильда'

```
In [128... # проверим финансирование провалившихся в прокате фильмов
gov_data.query('difference < 0')['financing_source'].value_counts()
```

```
Out[128]: Министерство культуры      135
Фонд кино                          83
Министерство культуры, Фонд кино   21
Name: financing_source, dtype: int64
```

```
In [129... # посчитаем суммарную потерю фильмов, для фильмов провалившихся в прокате сгруппированных по источнику финанси
gov_data.query('difference < 0').pivot_table(index = 'financing_source', values = 'difference', aggfunc = 'sum')
```

```
Out[129]:
```

difference	
financing_source	
Министерство культуры	-8,101,763,006.71
Министерство культуры, Фонд кино	-1,859,346,236.44
Фонд кино	-11,538,418,041.22

Большинство фильмов, провалившихся в прокате, финансировались Министерством культуры. Посчитав суммарную разницу между бюджетом и кассовым сбором фильмов, провалившихся в прокате, можно увидеть, что Фонд кино лидирует по финансированию убыточных фильмов.

```
In [130... # посмотрим на 3 самых прибыльных фильма
gov_data.sort_values(by = 'difference', ascending = False).head(3)
```

Out[130]:	title	puNumber	show_start_date	type	film_studio	production_country	director	producer	age
7456	Холоп	111,021,719.00	2019-12-19 12:00:00+00:00	Художественный	ООО "МЕМ-МЕДИА" по заказу АО "ВБД Груп"	Россия	К.Шипенко	Э.Илюян, Д.Жалинский, В.Дусмухаметов, Т.Бадзие...	де
5653	Движение вверх	111,011,817.00	2017-12-21 12:00:00+00:00	Художественный	ООО "Студия "ТРИТЭ" Никиты Михалкова"	Россия	А.Мегердичев	Л.Верещагин, А.Златопольский, Н.Михалков, ООО ...	де
6549	Т-34	111,024,918.00	2018-12-21 12:00:00+00:00	Художественный	ООО "Кинокомпания МАРС-фильм" по заказу ООО "М...	Россия	А.Сидоров	А.Златопольский, Р.Дишдишян, Л.Блаватник, Н.Яр...	де

Самаым успешным фильмом с профитом в 2913569000 рублей является фильм "Холоп" 2019 г.

```
In [131]: # проверим источник финансирования успешных фильмов
gov_data.query('difference >= 0')['financing_source'].value_counts()
```

```
Out[131]: Фонд кино                                61
Министерство культуры                             17
Министерство культуры, Фонд кино                   1
Name: financing_source, dtype: int64
```

```
In [132]: # посчитаем прибыльность фильмов, имевших успех в прокате, сгруппировав данные по источнику финансирования
gov_data.query('difference >= 0').pivot_table(index = 'financing_source', values = 'difference', aggfunc = 'sum')
```

```
Out[132]:
```

financing_source		difference
Министерство культуры		2,508,527,855.64
Министерство культуры, Фонд кино		15,375,045.70
Фонд кино		19,110,996,334.81

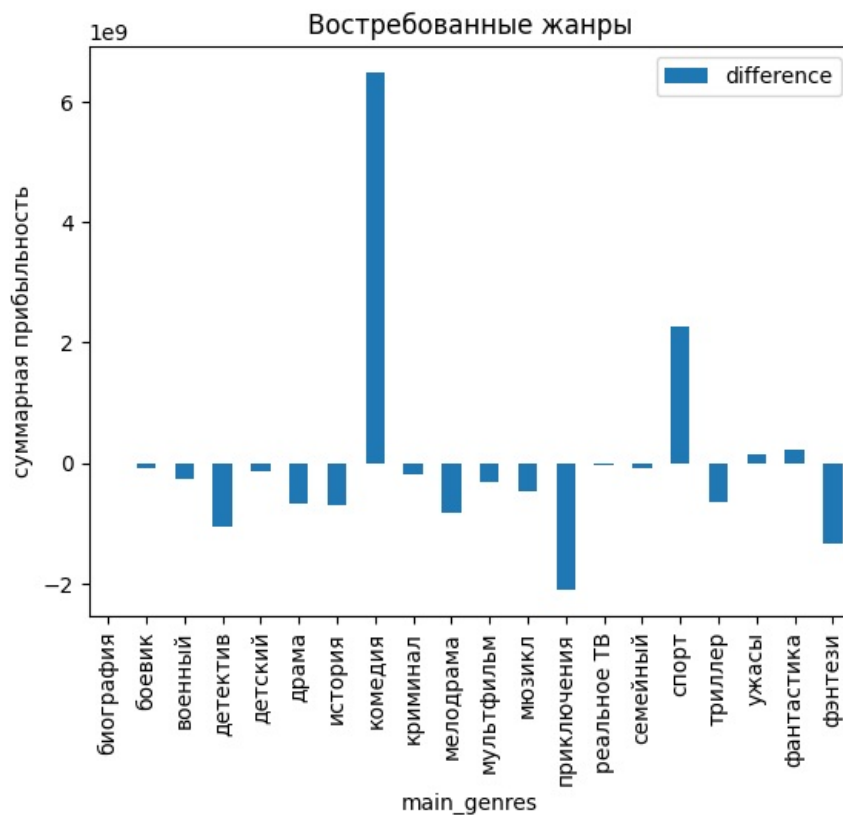
Успешные фильмы финансировались чаще 'Фонд культуры'

```
In [133]: # проверим в каком жанре чаще всего снимаются фильмы
gov_data['main_genres'].value_counts().head(5)
```

```
Out[133]: драма                                100
комедия                                         73
мультфильм                                    31
триллер                                        15
мелодрама                                     14
Name: main_genres, dtype: int64
```

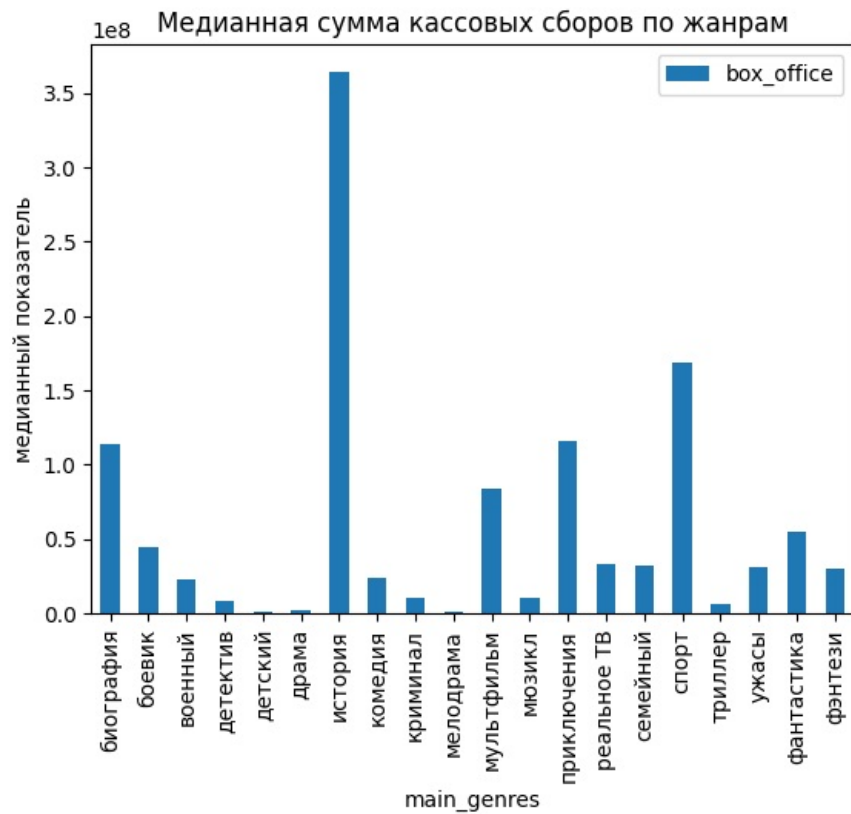
Большинство фильмов снимается в жанре **драма** - 107 единиц и **комедия** - 74 единицы. Посмотрим какие фильмы прибыльны по жанру, для этого посчитаем суммарную разность столбцов **budget** и **box\_office**

```
In [134]: gov_data.pivot_table(index = 'main_genres', values = 'difference', aggfunc = 'sum')\
.plot(kind = 'bar', title = 'Востребованные жанры', ylabel = 'суммарная прибыльность');
```



Самые прибыльные фильмы за исследуемый период с учетом потерь это 'комедии' и жанр 'спорт', самые убыточные в категории 'приключения' и 'фэнтези'. Популярный для финансирования жанр 'драма' по итогу оказался убыточным. Стоит добавить график медианного кассового сбора по жанрам

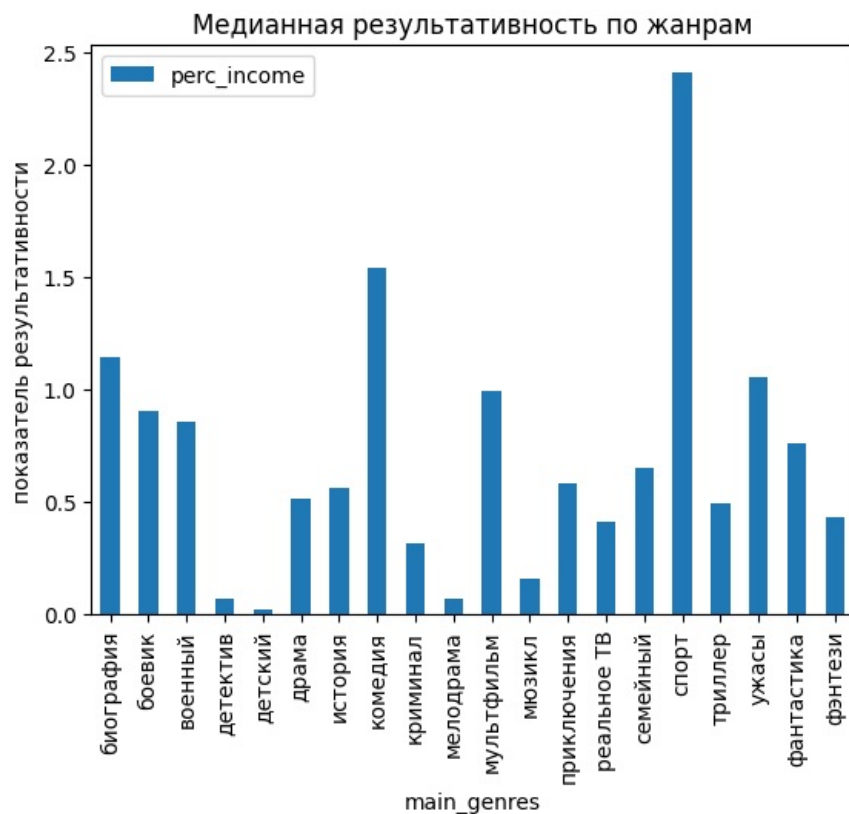
```
In [135]: # построим сводную таблицу, где посчитаем средний кассовый сбор по жанру
gov_data.pivot_table(index = 'main_genres', values = 'box_office', aggfunc = 'median')\
.plot(kind = 'bar', title = 'Медианная сумма кассовых сборов по жанрам', ylabel = 'медианный показатель');
```



По медианному значению кассового сбора лидируют жанр 'история' и 'спорт'. 'Детский', 'драма', 'мелодрама' имеют самое низкое значение кассового сбора

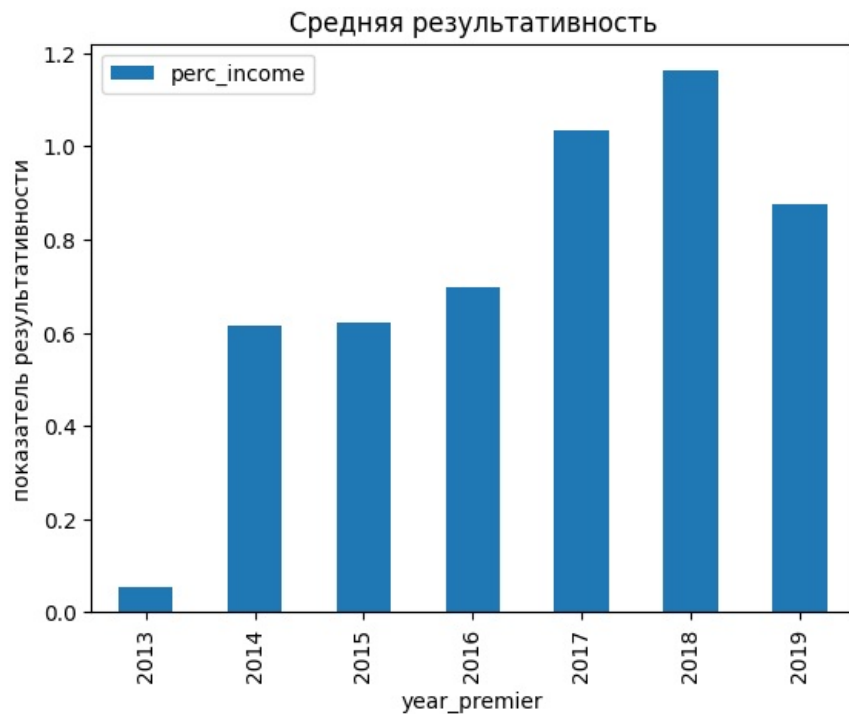
Проведем анализ прибыльности по коэффициенту результативности

```
In [136.. # построим график самых результативных жанров
gov_data.pivot_table(index = 'main_genres', values = 'perc_income', aggfunc = 'mean')\
.plot(kind = 'bar', title = 'Медианная результативность по жанрам', ylabel = 'показатель результативности');
```



```
In [137.. # построим график самых успешных с точки зрения результативности лет
gov_data.pivot_table(index = 'year_premier', values = 'perc_income', aggfunc = 'mean')\
.plot(kind = 'bar', title = 'Средняя результативность', ylabel = 'показатель результативности');
```





```
In [138]: # 5-ка лучших фильмов с точки зрения результативности
gov_data.sort_values(by = 'perc_income', ascending = False).head(5)
```

```
Out[138]:
```

	title	puNumber	show_start_date	type	film_studio	production_country	director	producer	age_res
7456	Холоп	111,021,719.00	2019-12-19 12:00:00+00:00	Художественный	ООО "МЕМ-МЕДИА" по заказу АО "ВБД Груп"	Россия	К.Шипенко	Э.Илюян, Д.Жалинский, В.Дусмухаметов, Т.Бадзие...	«12+ детей»
5708	Лёд	111,000,518.00	2018-02-01 12:00:00+00:00	Художественный	ООО "Водород 2011", ООО "Арт Пикчерс Студия", ...	Россия	О.Трофим	М.Врубель, А.Андрющенко, Ф.Бондарчук, Д.Рудовс...	«12+ детей»
5749	Я хуюеу	111,001,618.00	2018-03-07 12:00:00+00:00	Художественный	ООО "Друг Друга", ООО "Ди Пи Джи"	Россия	А.Черномазов (псевдоним Алексей Нужный)	С.Корнихин, Н.Куликов, А.Черномазов, А.Кремер,...	«16+ детей»
3229	Гуляй, Вася!	111,018,216.00	2016-12-20 12:00:00+00:00	Художественный	ООО Кинокомпания "Инвада Фильм", АО "ВБД Груп"	Россия	Р.Каримов	А.Новиков, А.Котелевский, Э.Илюян, Д.Жалинский...	«16+ детей»
2058	Духless 2	111,000,415.00	2015-01-26 12:00:00+00:00	Художественный	ООО "Киностудия "Слово", ООО "Арт Пикчерс Студия"	Россия	Р.Прыгунов	П.Ануров, Ф.Бондарчук, Д.Рудовский	«16+ детей»

В тройку самых результативных фильмов входят 'Холоп', 'Лёд' и 'Я хуюеу'. В среднем высокий коэффициент показывают фильмы в жанре 'спорт' и 'комедия'. Средний показатель результативности выше всего в 2017 - 2018 гг.

Проверим уникальные значения столбца `main_director` и избавимся от неявных дубликатов

```
In [139]: # создадим список уникальных значений столбца main_director
family = sorted(gov_data['main_director'].unique())
```

```
In [140]: # создадим словарь и посчитаем количество вхождений слов начинающихся с большой буквы (фамилий)
dict = {}
for i in family:
    for d in range(len(i)-1):
        if i[d].isupper() and i[d+1] != '.':
            dict[i[d:d+6]] = dict.get(i[d:d+6], 0) + 1
        else:
            continue
```

```
In [141]: # создадим результирующий словарь, где проверяем только ключи словаря dict со значениями больше 2, с возможными дубликатами
result = {}
for key in dict:
    if dict[key] >= 2:
        for j in sorted(family):
            if key in j:
```

```
result[key] = result.get(key, []) + [j]
```

```
In [142] result
```

```
Out[142]: {'Герман': ['А.Герман', 'Валерия Гай Германика'],
'Зайцев': ['А.Зайцев', 'И.Зайцев'],
'Ковале': ['А.Коваленко', 'К.Коваленко'],
'Кончал': ['А.Кончаловский', 'А.Михалков (псевдоним А.Кончаловский)'],
'Лунгин': ['А.Лунгин', 'П.Лунгин'],
'Миндад': ['А.Миндадзе', 'Е.Миндадзе (псевдоним Катя Шагалова)'],
'Михалк': ['А.Михалков',
'А.Михалков (псевдоним А.Кончаловский)',
'Н.Михалкова'],
'Нужный': ['А.Нужный', 'А.Черномазов (псевдоним Алексей Нужный)'],
'Сидоро': ['А.Сидоров', 'П.Сидоров'],
'Соколо': ['А.Соколов', 'К.Соколов', 'С.Соколов'],
'Учител': ['А.Учитель', 'И.Учитель'],
'Аленик': ['В.Алеников', 'Т.Аленикова'],
'Куценк': ['Г.Куценко (Ю.Куценко)'],
'Волков': ['И.Волкова', 'М.Волков'],
'Максим': ['И.Максимов', 'К.Максимов', 'Максим Свешников'],
'Белеви': ['К.Белевич', 'К.Белевич (псевдоним - Кирилл Астахов)'],
'Василь': ['К.Васильев', 'Р.Артемьев (псев.С.Васильев)', 'Ю.Васильев'],
'Оганес': ['К.Оганесян', 'Э.Оганесян'],
'Бальчю': ['М.Бальчюнас', 'М.Бальчюнас (псевдоним Марюс Вайсберг)'],
'Свешни': ['М.Свешников', 'Максим Свешников'],
'Бондар': ['Н.Бондарчук', 'Ф.Бондарчук'],
'Лопато': ['Н.Лопато', 'О.Лопато'],
'Артемь': ['Р.Артемьев', 'Р.Артемьев (псев.С.Васильев)'],
'Попов': ['О.Попова', 'С.Попов', 'Ф.Попов', 'Ю.Попович'],
'Игумен': ['Т.Игуменцева', 'Т.Игуменцева (псевдоним Татьяна Капитан)']}
```

```
In [143] # замена неявных дубликатов
```

```
gov_data['main_director'] = gov_data['main_director'].replace('Р.Артемьев (псев.С.Васильев)', 'Р.Артемьев')
gov_data['main_director'] = gov_data['main_director'].replace('М.Бальчюнас (псевдоним М.Вайсберг)', 'М.Бальчюнас')
gov_data['main_director'] = gov_data['main_director'].replace('М.Бальчюнас (псевдоним Марюс Вайсберг)', 'М.Бальчюнас')
gov_data['main_director'] = gov_data['main_director'].replace('К.Белевич (псевдоним - Кирилл Астахов)', 'К.Белевич')
gov_data['main_director'] = gov_data['main_director'].replace('Т.Игуменцева (псевдоним Татьяна Капитан)', 'Т.Игуменцева')
gov_data['main_director'] = gov_data['main_director'].replace('А.Михалков (псевдоним А.Кончаловский)', 'А.Михалков')
gov_data['main_director'] = gov_data['main_director'].replace('А.Черномазов (псевдоним Алексей Нужный)', 'А.Нужный')
gov_data['main_director'] = gov_data['main_director'].replace('Максим Свешников', 'М.Свешников')
```

```
In [144] # выведем 5 имен из список наиболее часто встречающихся главных режиссеров
```

```
gov_data['main_director'].value_counts().head(5)
```

```
Out[144]: Р.Давлетьяров      5
С.Андреасян      4
Д.Дьяченко      4
Д.Шмидт         3
А.Аксененко     3
Name: main_director, dtype: int64
```

Проверим насколько успешны фильмы режиссеров, которые получают наибольшее число одобренных заявок на финансирование

```
In [145] gov_data.query('(main_director in ["Р.Давлетьяров", "Д.Дьяченко", "С.Андреасян", "Н.Хомерики", "А.Аксененко", "М.Балицкий", "Т.Игуменцева", "Т.Игуменцева (псевдоним Татьяна Капитан)", "А.Михалков", "А.Михалков (псевдоним А.Кончаловский)", "А.Черномазов", "А.Черномазов (псевдоним Алексей Нужный)", "Максим Свешников", "М.Свешников"] and difference > 0)).sort_values(by = 'main_director')
```

	title	puNumber	show_start_date	type	film_studio	production_country	director	producer
3135	Чемпионы: Быстрее. Выше. Сильнее	111,021,115.00	2015-12-22 12:00:00+00:00	Художественный	ООО "Энджой мувиз"	Россия	А.Аксененко	Г.Малков, В.Поляков
2767	Страна чудес	111,018,415.00	2015-10-16 12:00:00+00:00	Художественный	ООО "Ол Медиа Компани"	Россия	Д.Дьяченко	Г.Шабанов, Р.Татаринцев, Э.Илюян
2872	Супербобровы	111,004,716.00	2016-03-01 12:00:00+00:00	Художественный	ООО "КЕЙСТОУН ПРОДАКШН ГРУП" по заказу ООО "Фи...	Россия	Д.Дьяченко	Э.Илюян, В.Шляппо, А.Троцюк, Д.Жалинский, Д.Дь...
5505	Последний богатырь	111,007,017.00	2017-10-19 12:00:00+00:00	Художественный	ООО "Киностудия "Слово" по заказу ООО "Уолт Ди...	Россия	Д.Дьяченко	Э.Илюян, Д.Жалинский, М.Озкан, А.Троцюк, В.Шляппо
6320	СУПЕРБОБРОВЫ. НАРОДНЫЕ МСТИТЕЛИ	111,020,318.00	2018-10-20 12:00:00+00:00	Художественный	ООО "Кейстоун продакшн груп" по заказу АО "ВБД...	Россия	Д.Дьяченко	Э.Илюян, В.Шляппо, Д.Жалинский, А.Троцюк
2205	8 новых свиданий	111,014,914.00	2014-10-30 12:00:00+00:00	Художественный	ООО "Смарт ТВ Продакшн"	Россия	М.Бальчюнас	ООО "Грин Филмс", ООО "ПЦ "Горад"
6632	Бабушка лёгкого поведения 2	111,000,519.00	2019-01-23 12:00:00+00:00	Художественный	ООО "Вайс Филмс"	Россия	М.Бальчюнас (псевдоним Марюс Вайсберг)	А.Ревва, М.Бальчюнас (псевдоним Марюс Вайсберг)
5673	Селфи	111,012,117.00	2018-02-01 12:00:00+00:00	Художественный	ООО "Киностудия "Слово", ООО "Арт Пикчерс Студия"	Россия	Н.Хомерики	П.Ануров, Ф.Бондарчук, Д.Рудовский
1870	А зори здесь тихие..._	111,002,915.00	2015-03-16 12:00:00+00:00	Художественный	ООО "Компания "Реал-Дакота"	Россия	Р.Давлетьяров	NaN
6178	Непрощенный	111,016,018.00	2018-09-27 12:00:00+00:00	Художественный	ООО "Кинокомпания Большое кино"	Россия	С.Андреасян	С.Андреасян, Г.Андреасян, А.Ананикян

Главный режиссер Д.Дьяченко снимает фильмы которые окупаются в прокате, Р.Давлетьяров, который имеет наибольшее количество одобренной господдержки снял 1 фильм, чей кассовый сбор првышает бюджет. С.Андреасян снял 1 успешный фильм из 4. А.Аксененко, Н.Хомерики также по 1 из 3. М.Бальчюнас снял 2 успешных фильма из 3.

Вывод: Финансирование государственными фондами большей частью приходится на фильмы с низкой результативностью, больше деньги выделяются на драмы, но комедии и спорт чаще делают кассовый сбор. Самой кассовой комедией является фильм 'Холоп', в спортивном жанре 'Движение вверх', драма - фильм 'Т-34'. Как правило финансирование получают разные режиссеры, но есть и повторы финансирования одного режиссера, не всегда успешные. На поддержку госфондов приходится как правило около 60 % бюджета фильма, причем большая доля за невозвратной частью бюджета.

Вывод:

В имеющихся данных средний рейтинг фильмов равен 6.3 балла.Данные кассовых сборов имеют разный способ подсчета, поэтому описательный анализ для столбца разделен на 2 периода: до 2014 и после для корректности исследования.Фильмы с поддержкой государства чаще не имеют части возвратного бюджета, невозвратный же составляет от 0 до 400 млн. Общий бюджет таких фильмов составляет 71 млн в медианном значении. Количество прокатных удостоверений меняется от года к году , наибольшее количество, чуть менее 1000, выдано в 2010, наименьшее в 2017 с последующем максимальным ростом за исследуемый период. В имеющихся данных количество прокатных удостоверений с указанной суммой кассовых сборов - 3133, что составляет 42% от общего количества прокатных удостоверений. При визуальной оценке изменения количества указанных кассовых сборов от года к году видим постепенное изменение в сторону увеличения представления информации, если в 2010 году соотношение наличия/отсутствия кассовых сборов было примерно 1 к 8, то после 2014 года пропорция изменилась и составляла уже 1 к 2 в сторону представленности данных(наличия информации о кассовых сборах). В процентном соотношении с 2010 до 2015 года видим уверенный прирост доли информации о box\_office, в среднем на примерно 5 % в год. С 2015 года видим прирост почти в 50%, после уровень представленности остается на довольно высоком уровне,показывая незначительные снижения в 2017 -2018годах. Суммарный кассовый сбор следует рассматривать в 2 временных периодах: до 2014 и после .С 2015 года сумма box\_office показывает некоторую стабильность, макимальный рост с 2014 по 2015, максимальная сумма сбора в 2018 году и составляет около 45 млрд.Период до 2014 имеет низкие показатели box\_office, минимум в 2010г с суммой примерно в 2,5 млн, что возможно связано с дрким способ исчисления.

Анализ влияния возрастного ограничения на кассовые сборы показывает, что наибольший суммарный кассовый сбор у

возрастной группы 16+, наименьшее значение, как и медианный показатель кассового сбора у возрастной группы "для любой зрительской аудитории". Стоит отметить и довольно низкое количество фильмов, выпускаемых в указанной категории. Самое высокое медианное значение box\_office у фильмов с ограничением 6+.

Для анализа фильмов с государственным финансированием используются данные 332 прокатных удостоверений. Наибольшее количество фильмов выпущено в 2015 году, рейтинги в среднем были самые низкие в 2016 году. Наименьшее количество прокатных удостоверений выдано в 2013. Средний рейтинг фильмов около 6 баллов, что немного ниже, чем в целом по набору данных. 2017 год наиболее успешен в плане показателей кассовых сборов при относительно невысоком количестве фильмов и хорошем рейтинге. 2015 год при наибольшем количестве выпущенных фильмов имеет невысокие показатели кассового сбора. Больше всего фильмов за исследуемый период снималось с возрастным ограничением 16+. Абсолютное меньшинство у фильмов - для любой возрастной категории. Наименьший кассовый сбор у фильмов для любой зрительской аудитории и 18+, причем медианная стоимость у первой категории на хорошем уровне и имеет относительно высокий рейтинг. Самые низкие рейтинги у фильмов 12+ и 16+ и 6+ при высоких суммарных и медианных сборах. Если рассматривать каждый год отдельно, то категории 12+ и 6+ как правило лидируют по суммарному кассовому сбору. Сумма бюджета фильмов слабо связана с суммой государственной поддержки, скорее с долей от бюджета, которая в свою очередь обычно составляет около 62% от бюджета. Бюджет разделен на возвратную и невозвратную части, возвратная часть, как правило равна 0. Невозвратная имеет положительную корреляцию с бюджетом. Кассовые сборы имеют слабую положительную связь с суммой кассовых сборов, что является вполне логичным, так как высокий бюджет максимум подогревает ожидание, но при низких рейтингах и отзывах, посещаемость снижается.

Согласно исследуемым данным 75% фильмов финансируемых госфондами провалились в прокате, причем большинство из них одобрены Министерством культуры, наибольшие потери связаны с фильмом 'Печать Дракона' и суммой почти в 2 млрд. рублей. Тройка кассовых фильмов: "Холоп", "Движение вверх", "Т-34". Предпочтение в финансировании отдается жанру 'драма', когда как 'комедии' и 'спорт' показывают лучшие результаты по прибыльности с учетом потерь. Медианный показатель у жанра 'история' один из самых высоких, у 'драм', 'детский', 'мелодрама' - низкие показатели. Государственное финансирование оказывалось максимум 5 раз одному и тому же главному режиссеру, Р. Давлетьярову, из 5 фильмов только 1 собрал хороший кассовый сбор. Д. Дьяченко стал одним из самых успешных режиссеров, из 4 фильмов - 4 с кассовыми сборами превышающими бюджет.

Если считать миссией государственного софинансирования фильмов поддержку отечественного кинопроизводителя, повышение конкурентоспособности отечественного кино и поддержания социально-значимой повестки будет верным предположить, что безусловно выполняется финподдержка. Рейтинги российских фильмов без господдержки по среднему показателю выше, чем фильмы с господдержкой за исключением 2015г. Суммарный же и медианные показатели кассового сбора при примерно одинаковом количестве фильмов за наблюдаемый период сильно уступают фильмам с господдержкой. Возможно стоит пересмотреть критерии отбора для финансирования в сторону актуализации социально-значимой повестки, ужесточению сценарных критериев для соответствия ожиданиям потребителя.

In [ ]:

Loading [MathJax]/extensions/Safe.js