None

Исследование данных о российском кинопрокате

Содержание:

In [2]:
```
# импортируем библиотеки
import missingno as msno
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

к содержанию

Откроем файлы с данными и объедините их в один датафрейм.

In [4]:
```
# посмотрим на основную информацию о данных в movies
display(df_movies.head())
print('**************************************************','\n')
display(df_movies.info())
print(len(df_movies))
```

	title	puNumber	show_start_date	type	film_studio	production_country	
0	Открытый простор	221048915	2015-11-27T12:00:00.000Z	Художественный	Тачстоун Пикчерз, Кобальт Пикчерз, Бикон Пикче...	США	
1	Особо важное задание	111013716	2016-09-13T12:00:00.000Z	Художественный	Киностудия "Мосфильм"	СССР	
2	Особо опасен	221038416	2016-10-10T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Бе
3	Особо опасен	221026916	2016-06-10T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Бе
4	Особо опасен	221030815	2015-07-29T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Бе

```
******************************************************

<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
None
7486
```

In [5]:
```
# посмотрим на основную информацию о данных в shows
display(df_shows.head())
```

```
print('**********************************************************','\n')
print(df_shows.info())
print(len(df_shows))
```

	puNumber	box_office
0	111000113	2.450000e+03
1	111000115	6.104000e+04
2	111000116	1.530300e+08
3	111000117	1.226096e+07
4	111000118	1.636841e+08

```
**********************************************************

<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
None
3158
```

In [6]:
```
# необходимо привести puNumber в обеих таблицах к одному типу чтоб объедени
# проверим какие значения кроме числовых содержит puNumber
s = []
for i in df_movies['puNumber'].unique():
    try:
        int(i)
        s.append('ok')
    except:
        s.append(i)
print(set(s))
```

```
{'ok', 'нет'}
```

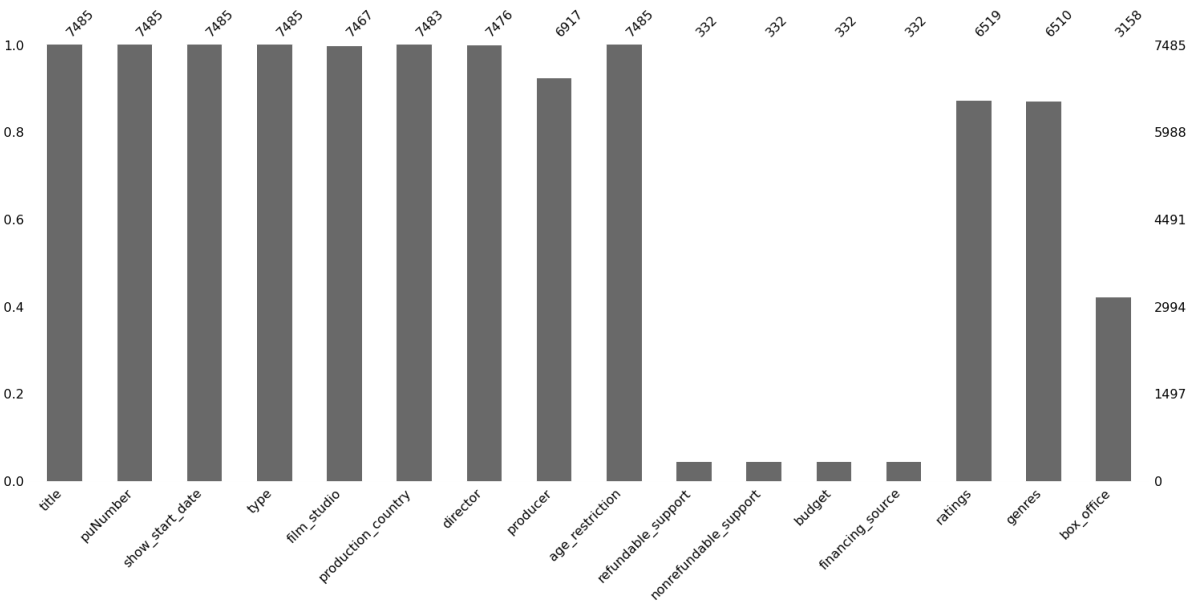
In [7]:
```
# посмотрим на данные строки
display(df_movies.loc[df_movies['puNumber']=='нет'])

# всего одна строка с пропусками по ключевым колонкам, уберем и приведем к 
df_movies = df_movies.loc[df_movies['puNumber']!='нет']

# приведем к int
df_movies['puNumber'] = df_movies['puNumber'].astype('int')
```

	title	puNumber	show_start_date	type	film_studio	production_country
1797	Курбан-роман. (История с жертвой)	нет	2014-05-15T12:00:00.000Z	Художественный	ФОНД "ИННОВАЦИЯ"	Россия

In [8]:
```
# объеденим таблицы по puNumber
data = pd.merge(df_movies,df_shows,on='puNumber',how='outer')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7485 entries, 0 to 7484
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   title                  7485 non-null   object 
 1   puNumber               7485 non-null   int64  
 2   show_start_date        7485 non-null   object 
 3   type                   7485 non-null   object 
 4   film_studio            7467 non-null   object 
 5   production_country     7483 non-null   object 
 6   director               7476 non-null   object 
 7   producer               6917 non-null   object 
 8   age_restriction        7485 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(4), int64(1), object(11)
memory usage: 994.1+ KB
```

In [9]:
```
# построим диаграмму количества пропусков
msno.bar(data)
plt.show()
```



- ***после предварительного ознакомления с общими данными видно большое количество пропусков в некоторых колонках, так же можно отметить проблемы с принадлежностью типов некоторых из них***

Предобработка и анализ данных по колонкам

к содержанию

Проведем анализ колонок в формате object, выявим неявные дубликаты, ошибки и поменяем на нужный формат там где это необходимо

In [10]:
```
# отобразим основную информацию о наших данных
print(data.info())
print('*****************************************************','\n')
display(data.head())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7485 entries, 0 to 7484
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   title                  7485 non-null   object 
 1   puNumber               7485 non-null   int64  
 2   show_start_date        7485 non-null   object 
 3   type                   7485 non-null   object 
 4   film_studio            7467 non-null   object 
 5   production_country     7483 non-null   object 
 6   director               7476 non-null   object 
 7   producer               6917 non-null   object 
 8   age_restriction        7485 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(4), int64(1), object(11)
memory usage: 994.1+ KB
None
*****************************************************
```

	title	puNumber	show_start_date	type	film_studio	production_country	
0	Открытый простор	221048915	2015-11-27T12:00:00.000Z	Художественный	Тачстоун Пикчерз, Кобальт Пикчерз, Бикон Пикче...	США	
1	Особо важное задание	111013716	2016-09-13T12:00:00.000Z	Художественный	Киностудия "Мосфильм"	СССР	
2	Особо опасен	221038416	2016-10-10T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Бе
3	Особо опасен	221026916	2016-06-10T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Бе
4	Особо опасен	221030815	2015-07-29T12:00:00.000Z	Художественный	Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Пл...	США	Бе

Проверим наличие дупликатов по прокатному номеру

In [11]:
```
# проверим наличие дубликатов по прокатному номеру, так как каждый прокат до
print('Количество дубликатов в puNumber равно:',data.duplicated(subset='puNu
```

```
Количество дубликатов в puNumber равно: 2
```

In [12]:
```
# посмотрим на них
display(data.loc[data.duplicated(subset='puNumber')==True])
```

	title	puNumber	show_start_date	type	film_studio	production_country
4638	Иоанна - женщина на папском престоле / По роман...	221154310	2010-12-17T12:00:00.000Z	Художественный	Константин Фильм, А Эр Ди Дегето Фильм, Дюне ...	Германия Великобритания Италия - Испания
5067	Анализируй то!	221054410	2010-05-25T12:00:00.000Z	Художественный	Уорнер Бразерс, Виллидж Роадшоу Пикчерз, Эн-Пи...	США

In [13]:
```
display((data.loc[(data['puNumber']==221154310)|(data['puNumber']==22105441(
```

	title	puNumber	show_start_date	type	film_studio	production_countr
4637	Как жениться и остаться холостым	221154310	2010-12-17T12:00:00.000Z	Художественный	Ше Вам, Скрипт Ассосье, Тэ Фэ 1 Фильм Продюксь...	Франци
4638	Иоанна - женщина на папском престоле / По роман...	221154310	2010-12-17T12:00:00.000Z	Художественный	Константин Фильм, А Эр Ди Дегето Фильм, Дюне ...	Германия Великобритания Италия - Испани
5066	Анализируй это!	221054410	2010-05-25T12:00:00.000Z	Художественный	Уорнер Браерс, Вилладж Роудшоу Филмз ЛТД	США-Австрали
5067	Анализируй то!	221054410	2010-05-25T12:00:00.000Z	Художественный	Уорнер Браерс, Виллидж Роадшоу Пикчерз, Эн-Пи...	СШ

Вся важная информация на месте, оставим эти строки

Переведем дату к верному формату

In [14]:
```
# переведем дату к верному формату
data['show_start_date'] = pd.to_datetime(data['show_start_date'],format='%Y-

# округлим до месяцев, результат перезапишем в новый столбец
data['show_start_date_by_month'] = pd.to_datetime(data['show_start_date']) -

# добавим столбец с годом
data['year_start'] = data['show_start_date'].dt.year
```

In [15]:
```
# простроим временную диограмму с количеством фильмов по датам
data_period = data.groupby(by='show_start_date_by_month')['puNumber'].count(
data_period.plot(figsize=(15,7),grid=True,linewidth=3)
plt.xlabel('дата')
plt.ylabel('количество фьльмов')
plt.title('Количество фьлбмов в прокате с 2010 по 2020 года')
plt.yticks(np.arange(0, 250, step=25))
plt.show()
print('************************************************************','\n'
display(data_period.sort_values(ascending=False).head(7))
```

****

```
show_start_date_by_month
2010-12-01 12:00:00+00:00    244
2014-11-01 12:00:00+00:00    126
2014-12-01 12:00:00+00:00    123
2015-04-01 12:00:00+00:00    123
2016-02-01 12:00:00+00:00    113
2016-06-01 12:00:00+00:00    113
2018-08-01 12:00:00+00:00    112
Name: puNumber, dtype: int64
```

- ***Данные предоставлены в период с 2010 по конец 2019 годов. На графике видна закономерность увеличения количества фильмов в прокате в зависимости от начала летнего или зимнего сезонов отпусков, что может объяснять расчет индустрии на сезонный приток посетителей.***

Проверим столбец type

In [16]:
```
# проверим столбец type на неявные дубликаты
data['type'].unique()
```

Out[16]:
```
array(['Художественный', 'Анимационный', 'Прочие', 'Документальный',
       'Научно-популярный', ' Художественный', ' Анимационный',
       'Музыкально-развлекательный'], dtype=object)
```

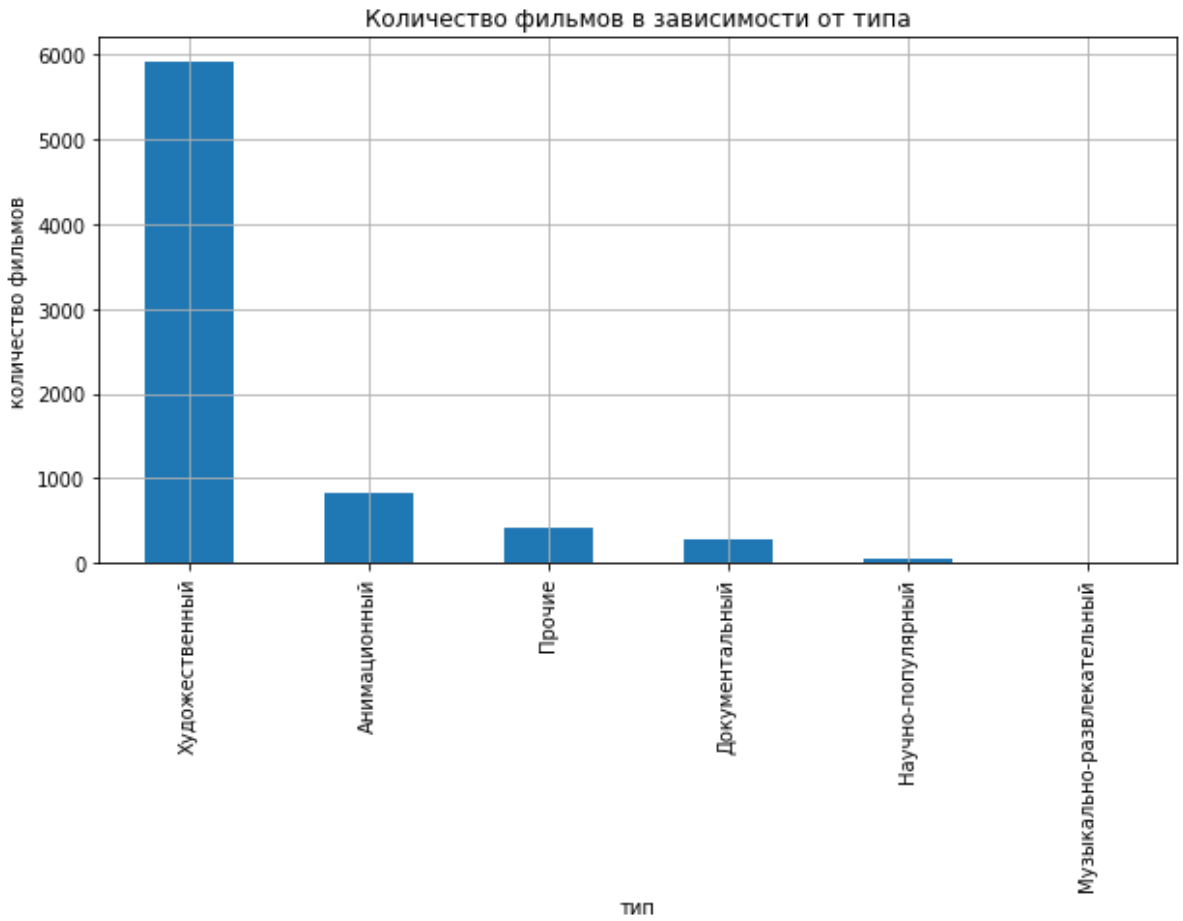
In [17]:
```
# зименим их и проверим
data['type'] = data['type'].replace({' Художественный':'Художественный',' Ан
data['type'].unique()
```

Out[17]:
```
array(['Художественный', 'Анимационный', 'Прочие', 'Документальный',
       'Научно-популярный', 'Музыкально-развлекательный'], dtype=object)
```

In [18]:
```
# приведем к категориальному типу
data['type'] = data['type'].astype('category')
```

In [19]:
```
# отобразим на графике
type_count = (data.pivot_table(index='type',values='puNumber',aggfunc='count
                  sort_values(by='puNumber',ascending=False))
type_count.columns = ['count']
type_count.plot(kind='bar',legend=False,grid=True,figsize=(10,5),title='Коли
plt.xlabel('тип')
plt.ylabel('количество фильмов')
```

```
plt.show()
print('**************************************')
display(type_count)
```



****

type	count
Художественный	5908
Анимационный	829
Прочие	406
Документальный	288
Научно-популярный	53
Музыкально-развлекательный	1

- ***Подавляющее большинство фильмов в прокате художественные***

Проверим столбец production_country

In [20]:
```
# проверим столбец production_country на неявные дебликаты
display(data['production_country'].unique()[:50])
print('**************************************************','\n')
print('Количество уникальных значений в production_country равно:',len(data[
```

```
array(['США', 'СССР', 'Франция', 'СССР, Венгрия',
       'Германия-Великобритания', 'Великобритания - Италия',
       'Чехословакия', 'США - Франция - Турция', 'Новая Зеландия',
       'Канада - Франция - Испания', 'США-Германия',
       'США - Великобритания', 'Великобритания', 'США - Германия',
       'Франция - Мексика - США', 'Россия, Казахстан, США',
       'СССР, Швеция', 'СССР, Франция, Англия, Куба, ГДР', 'Германия',
       'Великобритания-США-Германия-КНР',
       'СССР, ЧССР, Западный Берлин, ПНР', 'СССР, Италия', 'Гонконг, КНР',
       'США - Франция', 'США - Япония - Франция - Великобритания',
       'Гонконг - Сингапур - Таиланд - Великобритания', 'США-Канада',
       'Франция - Италия - Великобритания - США', 'Франция - США',
       'Ирландия-Великобритания-Германия', 'Чехия', 'США-Австралия',
       'СССР, Финляндия', 'США-Франция-Великобритания-Австрия',
       'США - Бельгия', 'США - Ирландия - Великобритания',
       'Великобритания - США',
       'Люксембург - Нидерланды - Испания - Великобритания - США - Италия',
       'Великобритания - Франция - США', 'Новая Зеландия - США',
       'США - Великобритания - Чехия',
       'Канада - Франция - Великобритания', 'Ирландия',
       'Великобритания - Германия - США',
       'США - Франция - Великобритания', 'Япония', 'СССР, Польша',
       'Франция - Испания', 'Канада-Франция', 'Германия - Италия - США'],
      dtype=object)
**********************************************************

Количество уникальных значений в production_country равно: 951
```

In [21]:

```python
# ужас перфекциониста) напишем функцию для корректировки
# функция уберает пробелы заменяет запятые на тире после возвращает пробелы
def ad_spaser(s):
    ss = s[0]
    for i in range(1,len(s)):
        if s[i]=='-':
            ss+=' '+s[i]
        elif (s[i].isupper)and(s[i-1].isupper()):
            ss+=s[i]
        elif s[i].isupper():
            ss+=' '+s[i]
        else:
            ss+=s[i]
    return ss

def clear(s):
    try:
        s = ''.join(s.split())
        s = '-'.join(s.split(','))
        return(ad_spaser(s))
    except:
        return s

print(clear('sdfsdfDsdfaDDDdfDs-Sfadfdas'))
```

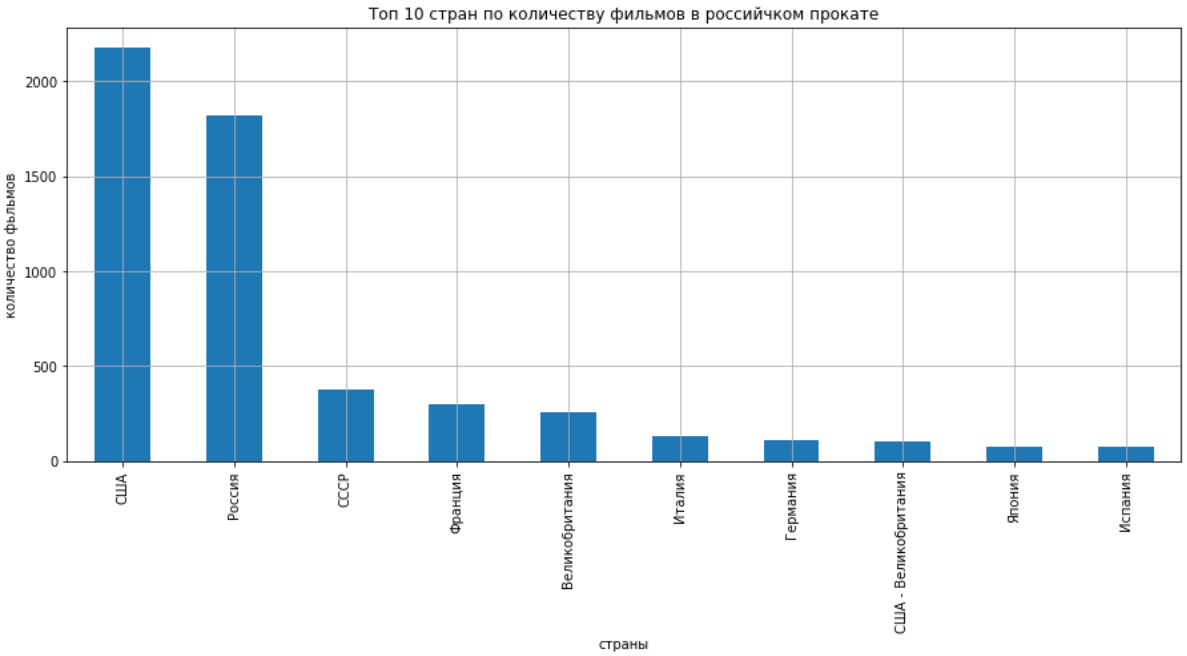
```
sdfsdf Dsdfa DDDdf Ds - Sfadfdas
```

In [22]:

```python
# применем функцию
data['production_country'] = data['production_country'].apply(clear)

# проверим результат
print('Количество уникальных значений в production_country равно:',len(data[
```

```
Количество уникальных значений в production_country равно: 813
```

In [23]:
```
# отобразим топ по количеству фильмов
top_county_fil = (data.pivot_table(index='production_country',values='puNumb
                  sort_values(by='puNumber',ascending=False).head(10))
top_county_fil.columns = ['count']
top_county_fil.plot(kind='bar',legend=False,grid=True,figsize=(15,6),title='
plt.xlabel('страны')
plt.ylabel('количество фильмов')
plt.show()
print('**************************************')
display(top_county_fil)
```




```
**************************************
```

production_country	count
США	2175
Россия	1820
СССР	377
Франция	302
Великобритания	259
Италия	131
Германия	110
США - Великобритания	106
Япония	77
Испания	74

- ***Топ 3 страны производителя по количеству фильмов в прокате — США, Россия и СССР. На лицо проблемы с формой записи данных, неявные дубликаты убраны***

Проверим столбец title

In [24]:
```
# проверим столбец title
print('Количество уникальных значений в title равно:',len(data['title'].unic
```

```
print('******************************************************','\n')
display(data['title'].unique()[:50])
```

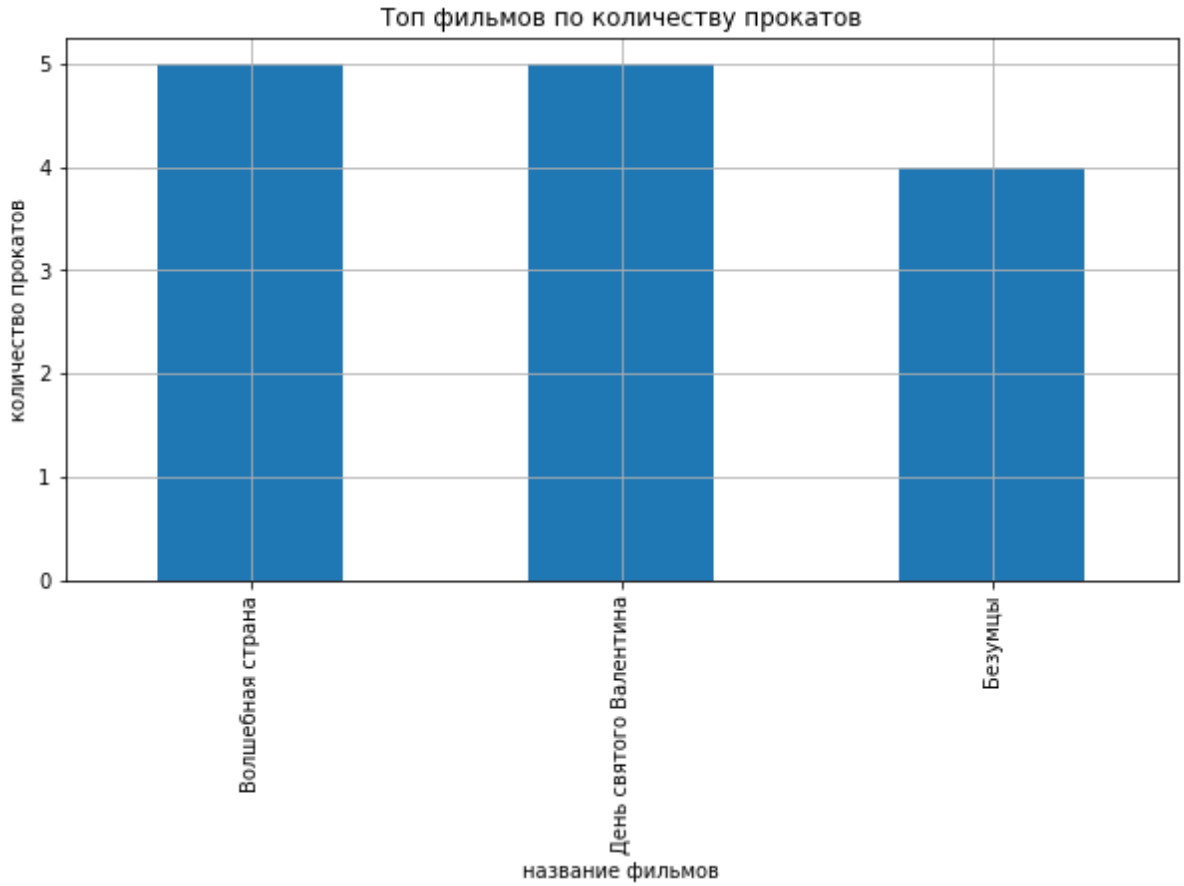
```
Количество уникальных значений в title равно: 6771
******************************************************

array(['Открытый простор', 'Особо важное задание', 'Особо опасен',
       'Остановился поезд', 'Любовь и голуби', 'Любовь и сигареты',
       'Отпетые мошенники.', 'Отпуск за свой счет',
       'Превосходство Борна /По одноименной новелле Роберта Ладлэма/',
       'Ответный ход',
       'Малышка на миллион /По мотивам рассказов Ф.Х.Тула из сборника "Клей
ма от канатов"/',
       'Преданный садовник', 'Отель /По мотивам пьесы Джона Уэбстера/',
       'Председатель', 'Осенний марафон', 'Осень', 'Неподдающиеся',
       'Неподсуден', 'Незабываемый 1919-й год', 'Незаконченная жизнь',
       'Операция "Ы" и другие приключения Шурика',
       'Неизвестные страницы из жизни разведчика', 'Неисправимый лгун',
       'Призрак замка Моррисвиль', 'Оружейный барон',
       'Отставной козы барабанщик', 'Паршивая овца',
       'Плюмбум, или Опасная игра', 'Первое свидание', 'Охота на лис.',
       'Пиноккио 3000', 'Перелом', 'Мисс Поттер',
       'Миссис Хендерсон представляет',
       'Планета КА-ПЭКС /По мотивам романа Джин Бруэр/',
       'Молчи в тряпочку', 'Мужики!..',
       'Автомобиль, скрипка и собака Клякса', 'Алекс и Эмма',
       'Мой лучший любовник', 'Мемуары гейши (по роману Артура Голдена)',
       'Адъютант его превосходительства', 'Без свидетелей', 'Без солнца',
       'Андрей Рублев.', 'Азартные игры', '36, Набережная Орфевр', 'Асса',
       'Бандитки.', 'Айболит - 66'], dtype=object)
```

In [25]:
```
# 6771 уникальный фильм был в прокате проверим на неявные дубликаты
# напишем функцию для очистки от паразитных знаков
punct = '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{}~'
def clear_x(s):
    try:
        for p in punct:
            s = ''.join(s.split(p))
        return s
    except:
        return s
```

In [26]:
```
data['title'] = data['title'].apply(clear_x)
print('Количество уникальных значений в title равно:',len(data['title'].unic
print('******************************************************','\n')
```

```
Количество уникальных значений в title равно: 6690
******************************************************
```

In [27]:
```
# некоторые фильмы показывались не один раз, покажем топ по названиям, неявн
# это не так важно при наличии уникального прокатного номера
top_title = (data.pivot_table(index='title',values='puNumber',aggfunc='count
                  sort_values(by='puNumber',ascending=False).head(3))
top_title.columns = ['count']
top_title.plot(kind='bar',legend=False,grid=True,figsize=(10,5),title='Топ ф
plt.xlabel('название фильмов')
plt.ylabel('количество прокатов')
plt.show()
print('************************************')
display(top_title)
```




```
**************************************
```

title	count
Волшебная страна	5
День святого Валентина	5
Безумцы	4

- ***Некоторые фильмы выходили в прокат несколько раз под разными прокатными номерами. Фильм 'Волшебная страна' выходил в прокат 5 раз. В столбце присутствовали проблемы с формой записи***

Разберем столбец age_restriction

In [28]:
```
# разберем столбец age_restriction
display(data['age_restriction'].unique())
print('**************************************************','\n')
print('Количество уникальных значений в production_country равно:',len(data[
```

```
array(['«18+» - запрещено для детей', '«6+» - для детей старше 6 лет',
       '«12+» - для детей старше 12 лет',
       '«16+» - для детей старше 16 лет',
       '«0+» - для любой зрительской аудитории'], dtype=object)
**************************************************

Количество уникальных значений в production_country равно: 5
```

In [29]:
```
# сложное название категорий похоже на масло масленое, заменим на более прос
# напишем функцию
def checker(s):
```

```
    try:
        ss=s[0]
        for letter in s[1:]:
            ss+=letter
            if letter=='»':
                return ss
    except:
        return 'err'
checker('«0+» - для любой зрительской аудитории')
```

Out[29]:
```
'«0+»'
```

In [30]:
```
# запишем в новый столбец
data['age_restriction_good'] = data['age_restriction'].apply(checker)

# приведем к категориальному типу
data['age_restriction_good'] = data['age_restriction_good'].astype('category
```

In [31]:
```
# отобразим количество фильмов по категориям возрастных ограничений
top_age_restriction = (data.pivot_table(index='age_restriction_good',values=
                  sort_values(by='puNumber',ascending=False))
top_age_restriction.columns = ['count']
top_age_restriction.plot(kind='bar',legend=False,grid=True,figsize=(10,5),\
                         title='Количество прокатов фильмов в зависимости от
plt.xlabel('категория возрастного ограничения')
plt.ylabel('количество прокатов')
plt.show()
print('****************************************')
display(top_age_restriction)
```


```
****************************************
```

	count
age_restriction_good	
«16+»	2851
«18+»	1605
«12+»	1592
«0+»	811
«6+»	626

- ***В топе по количеству прокатов фильмы 16+, ограничения 12+ и 18+ имеют примерно равное количество показав. Внизу рейтинга 0+ и 6+ соответственно. Все логично, количество показов соответствует размерам групп потребителей данного контента. Пропуски в данном столбце оставим так как они не критичны и данный ценз присваивает специальный орган на основании действующего закона***

Разберем столбец film_studio

In [32]:
```
#  разберем столбец film_studio
print(data['film_studio'].unique()[:20])
print('**************************************************','\n')
print('Количество уникальных значений в production_country равно:',len(data[
```

```
['Тачстоун Пикчерз, Кобальт Пикчерз, Бикон Пикчерз, Тиг Продакшнз'
 'Киностудия "Мосфильм"'
 'Юниверсал Пикчерз, Кикстарт Продакшнз, Марк Платт Продакшнз, Рилейтивити
Медиа, Спайгласс Интертейнмент, Стилкин Филмз, Топ Кау Продакшнз'
 'Юнайтед Артистс, Грин Стрит Филмз, Айкон Интертейнмент Интернэшнл'
 'Пульсар Продюксьон, ТФ1 Фильм ' 'Киностудия "Мосфильм", Телевидение ВНР'
 'Кеннеди/Маршал Компани, Юниверсал Пикчерз, Гипнотик, Калима Продакшнз, Лу
длум Интертейнмент'
 'Уорнер Браберс, Лейкшор Интертейнмент, Малпасо Продакшнз, Альберт С.Рудди
Продакшнз'
 'Потбойлер Продакшнз, Эпсилон Моушн Пикчерз, Скайон Филмз Лимитед, ЮК Филм
Каунсил'
 'Кэтлей, Отель Продакшнс, Мунстоун Интертейнмент, Рэд Маллет Продакшнс'
 'Инишиэл Интертейнмент Групп, Мирамакс Филмз, Персистент Интертейнмент, Ре
волюшн Студиос, Зе Лэдд Компани'
 'Фильмове Студио Баррандов'
 'Вэ И Пэ Медиенфондс 3, Асендант Пикчерз, Сатурн Филмз, Райзинг Стар, Эндг
ейм Интертейнмент, Интертейнмент Мэньюфэкчуринг Компани, Рилайз Филм'
 'Лайв Сток Филмз, Нью Зиланд Филм Комишн'
 'Синегруп, Анимакидс-Франс 2 Синема/Кастелао Продюксьон, Филмакс'
 'Уорнер Браберс, Нью Лайн Синема, Касл Рок Интертейнмент'
 'Феникс Пикчерз, Дэвид Киршнер Продакшнз, Айл оф Мэн Филм Коммишн, Ю Кей Ф
илм Каунсил, Ванштейн Компани, Метро Голдвин Майер'
 'БиБиСи Филмз, Фьючер Филмз, Хейман-Хоскинс Продакшнз, Микро Фьюжн, Патэ П
икчерз Интернэшнл'
 'Интермедиа Филмз, Юниверсал Пикчерз, Лоуренс Гордон Продакшнз'
 'Саммит Интертейнмент, Айл оф Мэн Филм, Эйзур Филмз, Таск Продакшнз']
**************************************************

Количество уникальных значений в production_country равно: 5491
```

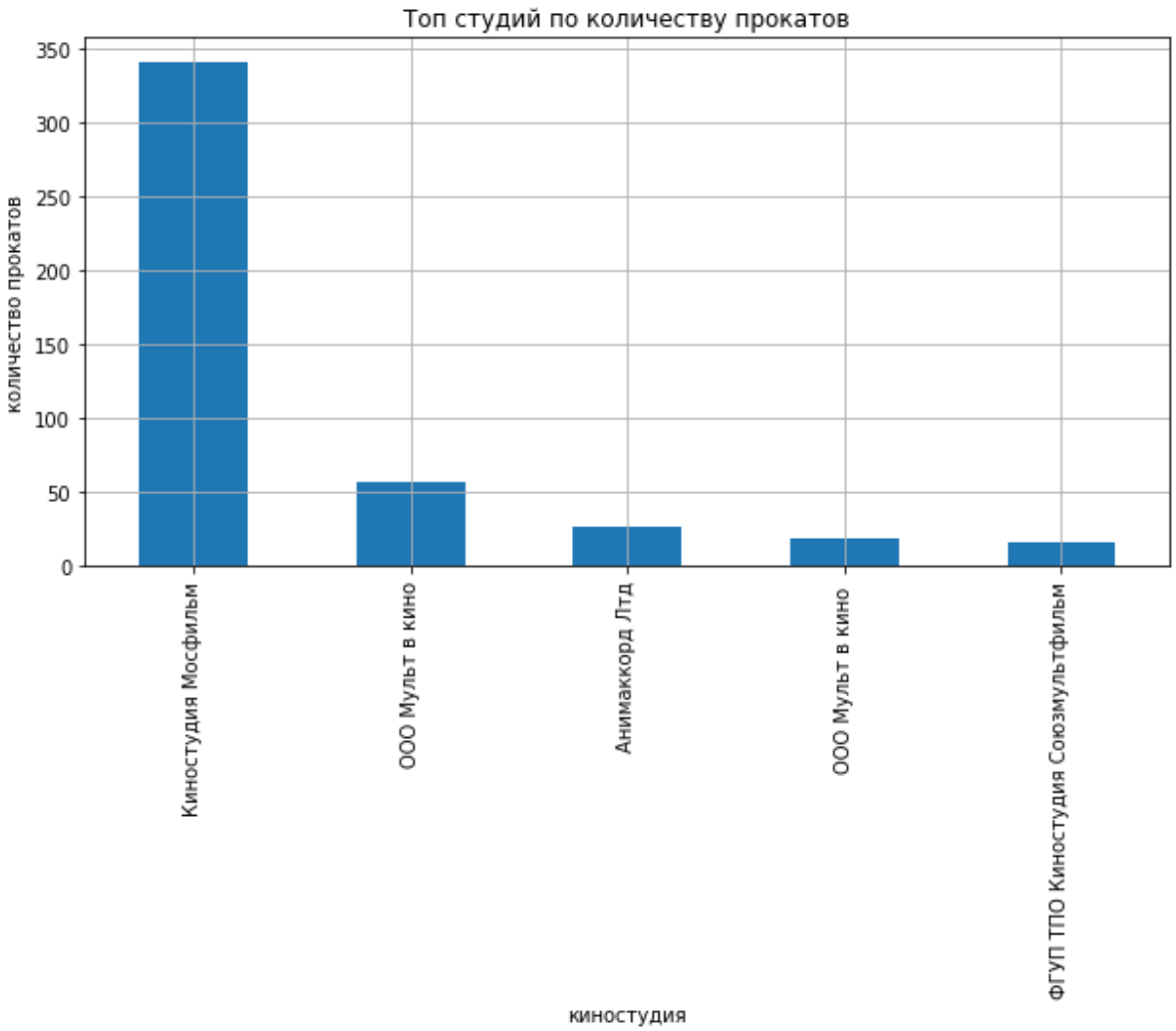
In [33]:
```
# применем функцию
punct = '!"#$%&\'()*+-./:;<=>?@[\\]^_`{}~'
```

```
data['film_studio'] = data['film_studio'].apply(clear_x)
```

In [34]:
```
print('*******************************************************','\n')
print('Количество уникальных значений в film_studio равно:',len(data['film_s
```

```
*******************************************************

Количество уникальных значений в film_studio равно: 5471
```

In [35]:
```
# отобразим топ студий по количеству прокатов
top_film_studio = (data.pivot_table(index='film_studio',values='puNumber',ag
                   sort_values(by='puNumber',ascending=False).head(5))
top_film_studio.plot(kind='bar',legend=False,grid=True,figsize=(10,5),\
                         title='Топ студий по количеству прокатов')
plt.xlabel('киностудия')
plt.ylabel('количество прокатов')
plt.show()
print('***************************************')
display(top_film_studio)
```




```
***************************************
```

film_studio	puNumber
Киностудия Мосфильм	341
ООО Мульт в кино	57
Анимаккорд Лтд	27
ООО Мульт в кино	18
ФГУП ТПО Киностудия Союзмультфильм	16

- ***Самое большое количество фильмов вышло в прокат от киностудии Мосфильм. В столбце присутствовали проблемы с формой записи***

Разберем столбец director

In [36]:
```
# разбрем director
display(data['director'].unique()[:50])
print('**************************************************','\n')
print('Количество уникальных значений в director равно:',len(data['director'
```

```
array(['Кевин Костнер', 'Е.Матвеев', 'Тимур Бекмамбетов', 'В.Абдрашитов',
       'В.Меньшов', 'Джон Туртурро', 'Эрик Беснард', 'В.Титов',
       'Пол Грингросс', 'М.Туманишвили', 'Клинт Иствуд',
       'Фернанду Мейрелльеш', 'Майк Фиггис', 'А.Салтыков', 'Г.Данелия',
       'А.Смирнов', 'Ю.Чулюкин', 'В.Краснопольский, В.Усков',
       'М.Чиаурели', 'Лассе Халлстрем', 'Л.Гайдай', 'В.Чеботарев',
       'В.Азаров', 'Боривой Земан', 'Эндрю Никкол', 'Г.Мыльников',
       'Джонатан Кинг', 'И.Бабич', 'Даниэль Робишо', 'Грегори Хоблит',
       'Крис Нунан', 'Стивен Фрирз', 'Йэн Софтли', 'Найл Джонсон',
       'Р.Быков', 'Роб Райнер', 'Бен Янгер', 'Роб Маршалл', 'Е.Ташков',
       'Н.Михалков', 'Ю.Карасик', 'А.Тарковский', 'Джон Франкенхаймер',
       'Оливье Маршал', 'С.Соловьев', 'Иоахим Реннинг, Эспен Сандберг',
       'Ролан Быков', 'Семен Туманов', 'Питер Сигал', 'М.Ромм'],
      dtype=object)
**************************************************

Количество уникальных значений в director равно: 4812
```

In [37]:
```
# Явных ошибок не видно, но прогоним через нашу функцию для очистки мусора и
# в некоторых ячейках видны несколько фамилий поэтому напишем функцию для вы

def first_director(s):
    try:
        ss=s[0]
        for letter in s[1:]:
            ss+=letter
            if letter==',':
                return ss[:-1]
        return ss
    except:
        return s



print(first_director('Стивен Содерберг, Дэвид Аубурн'))
```

```
Стивен Содерберг
```

In [38]:
```
punct = '!"#$%&\'()*+-/:;<=>?@[\\]^_`{}~'
# прогоним через нашу функцию для очистки мусора из строки
data['director'] = data['director'].apply(clear_x)

# выделим первую фамилию
data['director_first'] = data['director'].apply(first_director)
```

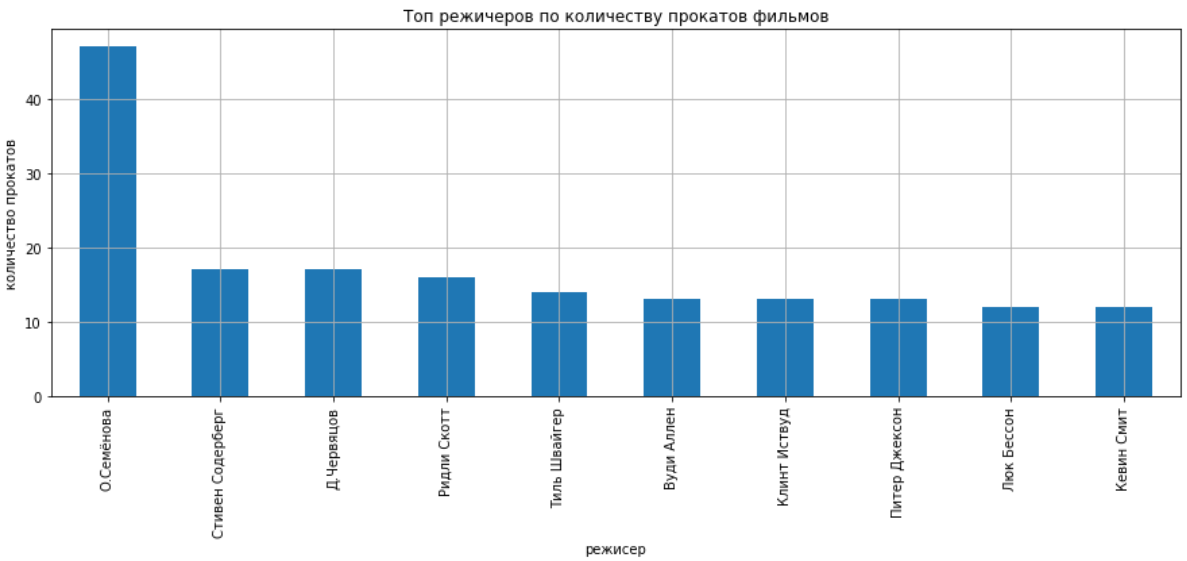
In [39]:
```
display(data['director_first'].unique()[:50])
print('**********************************************************','\n')
print('Количество уникальных значений в director равно:',len(data['director_
```

```
array(['Кевин Костнер', 'Е.Матвеев', 'Тимур Бекмамбетов', 'В.Абдрашитов',
       'В.Меньшов', 'Джон Туртурро', 'Эрик Беснард', 'В.Титов',
       'Пол Грингросс', 'М.Туманишвили', 'Клинт Иствуд',
       'Фернанду Мейрелльеш', 'Майк Фиггис', 'А.Салтыков', 'Г.Данелия',
       'А.Смирнов', 'Ю.Чулюкин', 'В.Краснопольский', 'М.Чиаурели',
       'Лассе Халлстрем', 'Л.Гайдай', 'В.Чеботарев', 'В.Азаров',
       'Боривой Земан', 'Эндрю Никкол', 'Г.Мыльников', 'Джонатан Кинг',
       'И.Бабич', 'Даниэль Робишо', 'Грегори Хоблит', 'Крис Нунан',
       'Стивен Фрирз', 'Йэн Софтли', 'Найл Джонсон', 'Р.Быков',
       'Роб Райнер', 'Бен Янгер', 'Роб Маршалл', 'Е.Ташков', 'Н.Михалков',
       'Ю.Карасик', 'А.Тарковский', 'Джон Франкенхаймер', 'Оливье Маршал',
       'С.Соловьев', 'Иоахим Реннинг', 'Ролан Быков', 'Семен Туманов',
       'Питер Сигал', 'М.Ромм'], dtype=object)
**********************************************************

Количество уникальных значений в director равно: 4613
```

In [40]:
```
# почти 200 не явных дубликатов выявлено, посмотрим на строки с пропусками в
print('Количество пропусков в director_first:',data.loc[data['director_first
```

```
Количество пропусков в director_first: 9
```

In [41]:
```
# отобразим топ 10 режисеров по количеству фильмов в прокате
top_director_first = (data.pivot_table(index='director_first',values='puNumb
                  sort_values(by='puNumber',ascending=False).head(10))
top_director_first.columns = ['count']
top_director_first.plot(kind='bar',legend=False,grid=True,figsize=(15,5),\
                        title='Топ режичеров по количеству прокатов фильмов
plt.xlabel('режисер')
plt.ylabel('количество прокатов')
plt.show()
print('****************************************')
display(top_director_first)
```




```
****************************************
```

director_first	count
О.Семёнова	47
Стивен Содерберг	17
Д.Червяцов	17
Ридли Скотт	16
Тиль Швайгер	14
Вуди Аллен	13
Клинт Иствуд	13
Питер Джексон	13
Люк Бессон	12
Кевин Смит	12

- ***О.Семёнова и Стивен Содерберг в топе по количеству фильмов. В столбце присутствовали проблемы с формой записи***

Разберем столбец genres

In [42]:
```
# разберем столбец genres
display(data['genres'].unique()[:50])
print('**************************************************','\n')
print('Количество уникальных значений в genres равно:',len(data['genres'].ur
```

```
array(['боевик,драма,мелодрама', 'драма,военный',
       'фантастика,боевик,триллер', 'драма', 'мелодрама,комедия',
       'мюзикл,мелодрама,комедия', 'комедия,криминал',
       'боевик,триллер,детектив', 'боевик,драма,приключения',
       'драма,спорт', 'триллер,драма,мелодрама', 'комедия,мелодрама',
       'драма,мелодрама,комедия', 'драма,мелодрама', 'драма,история',
       'драма,мелодрама,семейный', 'комедия,мелодрама,криминал',
       'комедия', 'боевик,драма,криминал', 'драма,комедия',
       'ужасы,фантастика,комедия', 'мультфильм,короткометражка,мелодрама',
       'драма,криминал', 'мультфильм,фантастика,фэнтези',
       'триллер,драма,криминал', 'драма,мелодрама,биография',
       'драма,комедия,военный', 'фантастика,драма,детектив',
       'мюзикл,семейный', nan, 'военный,приключения,драма',
       'документальный,драма', 'драма,биография,история',
       'боевик,триллер,драма', 'фэнтези,боевик',
       'боевик,комедия,криминал', 'мюзикл,комедия,детский',
       'комедия,мелодрама,драма', 'мультфильм,фэнтези,комедия',
       'комедия,история', 'мелодрама', 'драма,биография,музыка',
       'фэнтези,драма,мелодрама', 'триллер,военный',
       'драма,мелодрама,военный', 'мюзикл,драма,мелодрама',
       'мюзикл,комедия', 'мультфильм,приключения,семейный',
       'ужасы,триллер', 'боевик,драма,военный'], dtype=object)
**************************************************

Количество уникальных значений в genres равно: 743
```

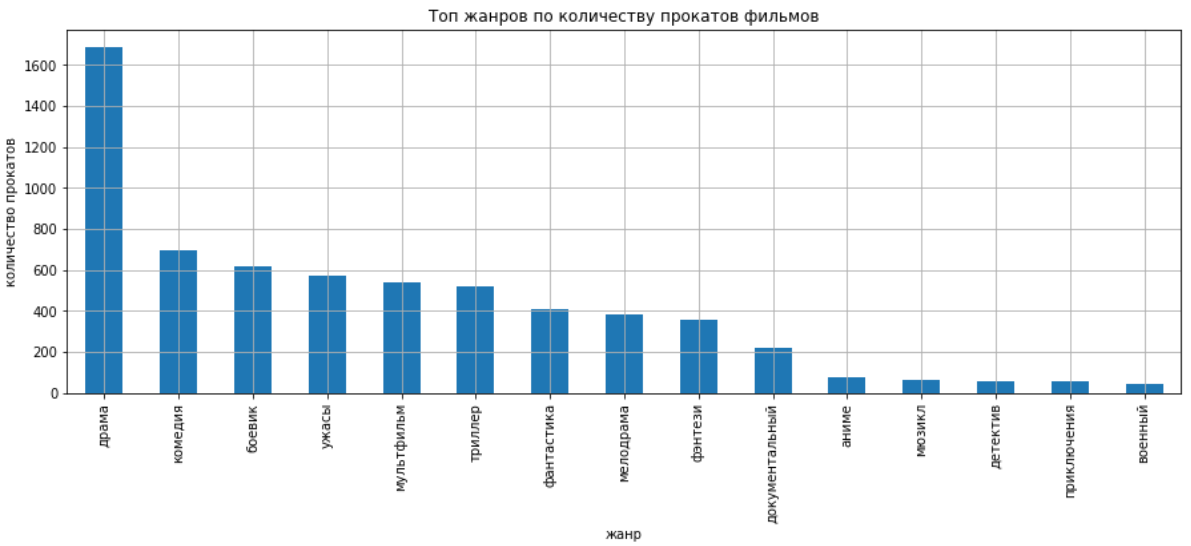
In [43]:
```
# Явных ошибок не видно, но прогоним через нашу функцию для очистки мусора и
punct = '!"#$%&\'()*+-/:;<=>?@[\\]^_`{}~'
```

```
data['genres'] = data['genres'].apply(clear_x)
data['genres_main'] = data['genres'].apply(first_director)
```

In [44]:
```
# проверим
display(data['genres_main'].unique()[:50])
print('**************************************************','\n')
print('Количество уникальных значений в director равно:',len(data['genres_ma
```

```
array(['боевик', 'драма', 'фантастика', 'мелодрама', 'мюзикл', 'комедия',
       'триллер', 'ужасы', 'мультфильм', nan, 'военный', 'документальный',
       'фэнтези', 'криминал', 'приключения', 'аниме', 'детектив',
       'для взрослых', 'семейный', 'концерт', 'история',
       'короткометражка', 'детский', 'спорт', 'биография', 'вестерн',
       'музыка', 'фильмнуар', 'реальное ТВ'], dtype=object)
**************************************************

Количество уникальных значений в director равно: 29
```

In [45]:
```
# отобразим топ 10 жанров по количеству фильмов в прокате
top_genres_main = (data.pivot_table(index='genres_main',values='puNumber',ag
                   sort_values(by='puNumber',ascending=False).head(15))
top_genres_main.columns = ['count']
top_genres_main.plot(kind='bar',legend=False,grid=True,figsize=(15,5),\
                         title='Топ жанров по количеству прокатов фильмов')
plt.xlabel('жанр')
plt.ylabel('количество прокатов')
plt.show()
print('****************************************')
display(top_genres_main)
```




```
****************************************
```

genres_main	count
драма	1688
комедия	697
боевик	617
ужасы	573
мультфильм	538
триллер	521
фантастика	410
мелодрама	383
фэнтези	358
документальный	219
аниме	74
мюзикл	64
детектив	56
приключения	55
военный	46

- ***самое большое количество фильмов в прокате были в жанре драма, далее идут комедии и боевики. Основной жанр выделен. Пропуски в значениях оставим***

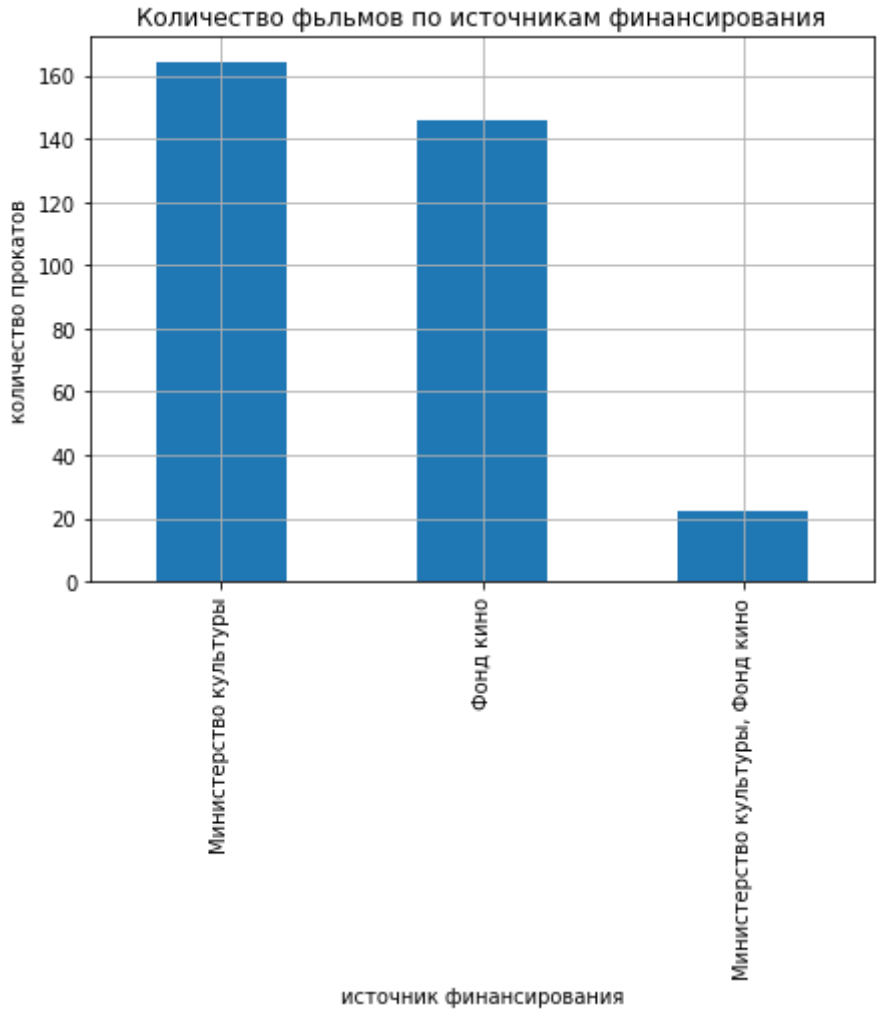
разберем financing_source

In [46]:
```
# разберем financing_source
display(data['financing_source'].unique())
print('**************************************************','\n')
print('Количество уникальных значений в financing_source равно:',len(data['f
```

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

Количество уникальных значений в financing_source равно: 4
```

In [47]:
```
# приведем в категориальный формат
data['financing_source'] = data['financing_source'].astype('category')
```

In [48]:
```
financing_source = (data.pivot_table(index='financing_source',values='puNumb
                    sort_values(by='puNumber',ascending=False))
financing_source.columns = ['count']
financing_source.plot(kind='bar',legend=False,grid=True,figsize=(7,5),\
                          title='Количество фильмов по источникам финансирова
plt.xlabel('источник финансирования')
plt.ylabel('количество прокатов')
plt.show()
print('****************************************')
display(financing_source)
```



****

financing_source	count
Министерство культуры	164
Фонд кино	146
Министерство культуры, Фонд кино	22

- ***Министерство культуры и Фонд кино поддержали по 164 и 146 фильм соответственно, 22 фильма получили их совместную поддержку. Проблем в столбце не выявлено, тип сменен***

к содержанию

Работа с пропусками и числовыми столбцами

разберем ratings

In [49]:
```
# разберем ratings
# проверим
display(data['ratings'].unique()[:50])
print('**************************************************','\n')
print('Количество уникальных значений в ratings равно:',len(data['ratings'].
```

```
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'], dtype=object)
**************************************************

Количество уникальных значений в ratings равно: 95
```

In [50]:
```
# очевидно что ratings это числовое значение и оно должно отражать среднюю о
# напишем функцию для решения этой проблемы
def to_d_type(s):
    try:
        if s[2]=='%':
            i=float(s[:2])/10
            round(i,1)
            return i
        else:
            s=float(s)
            return s
    except:
        return s
to_d_type('90%')
```

Out[50]:
```
9.0
```

In [51]:
```
# применем функцию и проверим результат
data['ratings'] = data['ratings'].apply(to_d_type)
display(data['ratings'].unique()[:50])
print('**************************************************','\n')
print('Количество уникальных значений в ratings равно:',len(data['ratings'].
```

```
array([7.2, 6.6, 6.8, 7.7, 8.3, 8. , 7.8, 8.1, 7.1, 6. , 7.4, 5.8, 8.7,
       6.3, 6.9, 5. , 4.3, 7.3, 7. , 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,
       9.9, 4.4, 4.5, 5.3, 4.1, 8.4, 2.6, 3.8, 4.6, 4.8, 4. ])
**************************************************

Количество уникальных значений в ratings равно: 85
```

In [52]:
```
# отобразим распределение оценок на гистаграме чтобы лучше понимать их распр


data['ratings'].plot(bins=40,kind='hist',grid=True,figsize=(10,5),\
                         title='Распределение рейтингов фильмов')
plt.xlabel('рейтинг')
plt.ylabel('количество оценок')
plt.xticks(range(0,10),)
plt.show()
print('**************************************')
print('Описательная статистика ratings')
display(data['ratings'].describe())
```


```
**************************************
Описательная статистика ratings
count    6519.000000
mean        6.488173
std         1.114638
min         1.000000
25%         5.900000
50%         6.600000
75%         7.200000
max         9.900000
Name: ratings, dtype: float64
```

- ***средняя медианная оценка равна 6.6 баллам, основная их часть расположилась между 5.9 и 7.2 баллами. В столбце присутствовали проблемы с формой записи***

разберем box_office

In [53]:
```
# разберем box_office, здесь мы видим большое количество пропусков оставим и
# возможно информация о кассовых сборах не известна либо фильм не прокатывал
# отобразим описательную статистику
display(data['box_office'].describe())
print('**************************************************','\n')
print('Количество пропусков в box_office равно:',data.loc[data['box_office']
```

```
count    3.158000e+03
mean     7.647870e+07
std      2.403531e+08
min      0.000000e+00
25%      8.623900e+04
50%      2.327988e+06
75%      2.397967e+07
max      3.073569e+09
Name: box_office, dtype: float64
**************************************************

Количество пропусков в box_office равно: 4327
```

In [54]:
```
# посмотрим на подозрительные нулевые значения в box_office
display(data.loc[data['box_office']==0].head())
print('**************************************************','\n')
print('Количество 0 в box_office равно:',data.loc[data['box_office']==0]['pu
```

	title	puNumber	show_start_date	type	film_studio	production_cou
66	Анна Павлова	111011013	2013-12-19 12:00:00+00:00	Художественный	совместное производство Киностудия Мосфильм, К...	СССР - Франц Англия - К
237	Подранки	111007613	2013-10-18 12:00:00+00:00	Художественный	Киностудия Мосфильм	С
596	Запах вереска	111003012	2012-05-23 12:00:00+00:00	Художественный	ООО Студия РИМ	Рос
914	В тумане По одноименной повести Василя Быкова	121027712	2012-11-07 12:00:00+00:00	Художественный	Ма Йа Де Фикшн, Лемминг Филм, Беларусьфильм, Д...	Герман Нидерлан Беларусь - Рос -
932	Письмо для Момо	124002912	2012-10-25 12:00:00+00:00	Анимационный	Кадокава Пикчерз, Продакшнз И Джи, Токио Броад...	Япо

5 rows × 21 columns

```
**********************************************************

Количество 0 в box_office равно: 24
```

In [55]:
```
# 24 строки с 0 значением, возможно фильмы не были в прокате кинотеатра, зам
# при условии наличия пропуска в информации о гос поддержке
data.loc[(data['box_office']==0)&(data['refundable_support'].isna()),'box_of
```

In [56]:
```
data['box_office'].describe()
```

Out[56]:
```
count    3.134000e+03
mean     7.706437e+07
std      2.411784e+08
min      4.000000e+01
25%      1.010288e+05
50%      2.409099e+06
75%      2.456979e+07
max      3.073569e+09
Name: box_office, dtype: float64
```

In [57]:
```
## посмотрим на подозрительные значения в box_office в хвостах
display(data.loc[(data['box_office']<3333)].sort_values(by='box_office',asce
print('**********************************************************','\n')
print('Количество значений ниже 3333 в box_office равно:',data.loc[(data['bo
```

	title	puNumber	show_start_date	type	film_studio	production_cou
151	Жестокий романс	111006013	2013-10-18 12:00:00+00:00	Художественный	Киностудия Мосфильм	С
2273	Каменный цветок	111016714	2014-12-01 12:00:00+00:00	Художественный	Киностудия Мосфильм	С
3916	22 пули Бессмертный	121006410	2010-04-01 12:00:00+00:00	Художественный	Еуроп Корпорейшн	Фра
1180	Астерикс и Обеликс в Британии 3D	121025012	2012-10-05 12:00:00+00:00	Художественный	Уайлд Банч, Фиделите Фильм, Филм Кайрос, Синет...	Франция - Ита - Испа Вен
164	За спичками	111006113	2013-10-18 12:00:00+00:00	Художественный	Киностудия Мосфильм, СУОМИФИЛЬМ	СССР - Финля

5 rows × 21 columns

```
**************************************************

Количество значений ниже 3333 в box_office равно: 374
```

In [58]:
```
# посмотрим на подозрительные нулевые значения в box_office
display(data.loc[(data['box_office']<1000)&(data['production_country']=='Ро
print('**************************************************','\n')
print('Количество значений ниже 1000 в box_office при условии "production_co
```

	title	puNumber	show_start_date	type	film_studio	production_
4062	Ловец ветра	111009310	2010-10-21 12:00:00+00:00	Художественный	ГУП РБ Киностудия Башкортостан	
4541	Фобос	111001510	2010-02-05 12:00:00+00:00	Художественный	ООО Арт Пикчерс Студия	
4528	Ёлки	111010710	2010-12-08 12:00:00+00:00	Художественный	ООО ТаББаК	
2359	Детский юмористический киножурнал Ералаш выпус...	111014014	2014-10-10 12:00:00+00:00	Художественный	ООО Продюсерский центр ЕРАЛАШ	
4660	Без мужчин	111011310	2010-12-15 12:00:00+00:00	Художественный	ООО Кинокомпания ВВЫСЬ, ООО ВВП Альянс	

5 rows × 21 columns

```
**************************************************

Количество значений ниже 1000 в box_office при условии "production_country=
Россия" равно: 39
```

In [59]:
```
# посмотрим на подозрительные нулевые значения в box_office_m
display(data.loc[(data['box_office']<1000)&(data['production_country']=='США
print('*****************************************************','\n')
print('Количество значений ниже 1000 в box_office при условии "production_co
      data.loc[(data['box_office']<1000)&(data['production_country']=='США')
```

	title	puNumber	show_start_date	type	film_studio	production_countr
5195	Форсаж 5	121006311	2011-04-19 12:00:00+00:00	Художественный	Дарк Сайд Продакшнз, Ориджинал Филм	США
3920	Принц Персии Пески времени	121007010	2010-05-25 12:00:00+00:00	Художественный	Уолт Дисней Пикчерз, Джерри Брукхаймер Филмз	США
793	21 и больше	121004113	2013-02-21 12:00:00+00:00	Художественный	Мендевилль Филмз, Релативити Медиа, Скайленд И...	США
2859	Зверополис	224002216	2016-06-07 12:00:00+00:00	Анимационный	Уолт Дисней Анимейшн Студиос, Уолт Дисней Пикчерз	США
1071	Черный дрозд	121028812	2012-11-29 12:00:00+00:00	Художественный	Магнолиа Пикчерз, СтудиоКанал, 2929 Продакшнз,...	США

5 rows × 21 columns

```
*****************************************************

Количество значений ниже 1000 в box_office при условии "production_country=
США" равно: 68
```

In [60]:
```
# посмотрим на подозрительные значения в box_office
display(data.loc[(data['box_office']>1000)&(data['production_country']=='Рос
print('*****************************************************','\n')
print('Количество значений выше 1000 в box_office при условии "production_co
      data.loc[(data['box_office']>1000)&(data['production_country']=='Росси
```

	title	puNumber	show_start_date	type	film_studio	production_cou
7455	Холоп	111021719	2019-12-19 12:00:00+00:00	Художественный	ООО МЕММЕДИА по заказу АО ВБД Груп	Рос
5652	Движение вверх	111011817	2017-12-21 12:00:00+00:00	Художественный	ООО Студия ТРИТЭ Никиты Михалкова	Рос
6548	Т34	111024918	2018-12-21 12:00:00+00:00	Художественный	ООО Кинокомпания МАРСфильм по заказу ООО ММЕ, ...	Рос
6469	Полицейский с рублевки Новогодний беспредел	111023318	2018-12-20 12:00:00+00:00	Художественный	ООО ЛЕГИО ФЕЛИКС, ООО Ника ТВ	Рос
5504	Последний богатырь	111007017	2017-10-19 12:00:00+00:00	Художественный	ООО Киностудия Слово по заказу ООО Уолт Дисней...	Рос
5707	Лёд	111000518	2018-02-01 12:00:00+00:00	Художественный	ООО Водород 2011, ООО Арт Пикчерс Студия, Госу...	Рос
2919	Экипаж	111005416	2016-03-21 12:00:00+00:00	Художественный	ООО Студия ТРИТЭ Никиты Михалкова	Рос
7387	Полицейский с Рублевки Новогодний Беспредел 2	111019519	2019-12-12 12:00:00+00:00	Художественный	АО ТНТТелесеть, ООО ЛЕГИО ФЕЛИКС, ООО 123 Прод...	Рос
3564	Притяжение 2016	111018116	2016-12-16 12:00:00+00:00	Художественный	ООО Водород 2011, ООО Арт Пикчерс Студия	Рос
5640	Ёлки Новые	111011617	2017-12-21 12:00:00+00:00	Художественный	ООО ТаББаК	Рос

10 rows × 21 columns

```
**************************************************

Количество значений выше 1000 в box_office при условии "production_country=
Россия" равно: 726
```

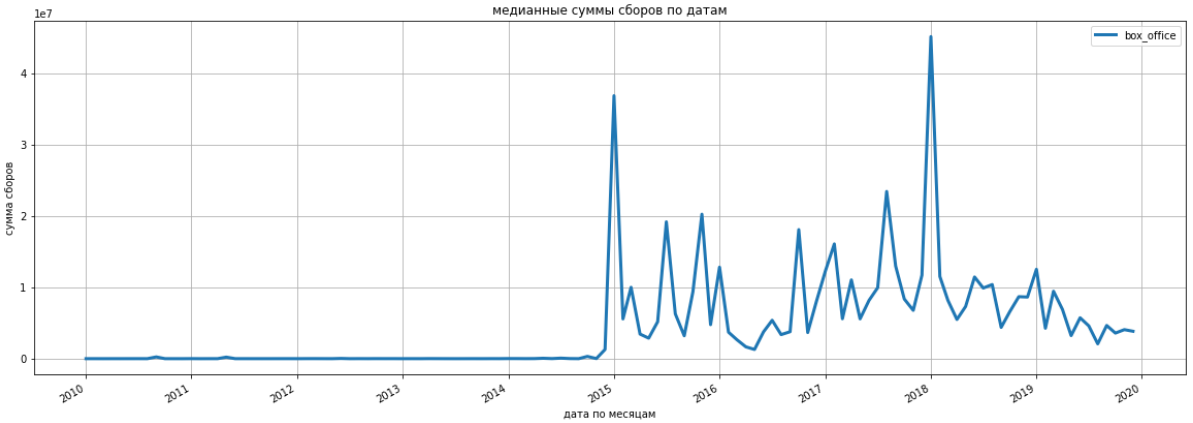
In [61]:
```
display(data['box_office'].sort_values(ascending=True).unique()[:25])
```

```
array([ 40.,  50.,  75.,  80., 100., 115., 120., 125., 130., 135., 140.,
       150., 165., 170., 180., 190., 200., 210., 225., 235., 240., 250.,
       260., 295., 300.])
```

Очевидно в данных с кассовыми сборами какие-то проблемы в формате сумм, форсаж 5 явно собрал 975 млн, а запись идет обычным числом. Не совсем ясно с чем

это связано, но прослеживается связь с датами, в нижнем хвосте по сборам - даты начала отсчета, в верхнем - конца

In [62]:
```
# попробуем графически отобразить медианные суммы сборов по датам нашего да
data.pivot_table(index='show_start_date_by_month',values='box_office',aggfu
plot(figsize=(20,7),grid=True,linewidth=3)
plt.title('медианные суммы сборов по датам')
plt.xlabel('дата по месяцам')
plt.ylabel('сумма сборов')
plt.show()
```



In [63]:
```
# четко видны кратные изменения сумм после 2014-11-01 что может говорить о 
# необходимо для корректного отображения разделить данные на 2 части и опис

# запишем часть после '2014-11-01 12:00:00'
after_data = data.loc[(data['show_start_date_by_month']>'2014-9-01 12:00:00
pd.options.display.float_format = '{:,.2f}'.format
display(after_data['box_office'].describe())
print('****************************************')
print('Количество фильмов после "2014-9-01 12:00:00" с указанными кассовыми
```

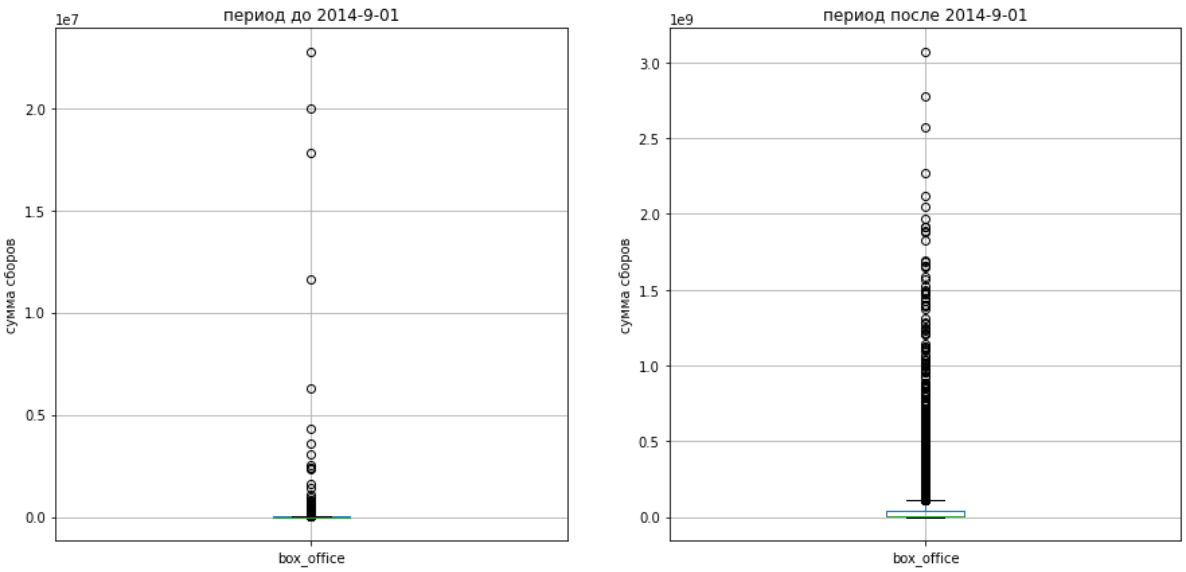
```
count           2,466.00
mean       97,849,693.13
std       268,138,855.07
min                50.00
25%         1,097,463.75
50%         5,510,940.42
75%        44,534,753.29
max     3,073,568,690.79
Name: box_office, dtype: float64
****************************************
Количество фильмов после "2014-9-01 12:00:00" с указанными кассовыми сборам
и равно: 2466
```

In [64]:
```
# запишем часть до '2014-11-01 12:00:00'
before_data = data.loc[data['show_start_date_by_month']<'2014-9-01 12:00:00
display(before_data['box_office'].describe())
print('****************************************')
print('Количество фильмов до "2014-9-01 12:00:00" с указанными кассовыми сбо
```

```
count          638.00
mean       197,025.28
std      1,514,434.73
min             40.00
25%            932.50
50%          3,975.00
75%         18,457.50
max     22,769,680.00
Name: box_office, dtype: float64
```

```
****************************************
Количество фильмов до "2014-9-01 12:00:00" с указанными кассовыми сборами р
авно: 638
```

In [65]:
```
# посмотрим на выбросы этих периодов
fig = plt.figure()
fig.set_figheight(7)
fig.set_figwidth(15)

ax1 = fig.add_subplot(121)
before_data['box_office'].plot(kind='box',ax=ax1,grid=True,title='период до
plt.ylabel('сумма сборов')
plt.ylim()

ax2 = fig.add_subplot(122)
after_data['box_office'].plot(kind='box',grid=True,title='период после 2014-
plt.ylim()
plt.ylabel('сумма сборов')
plt.show()
```



In [66]:
```
## посмотрим на подозрительные значения в box_office после "2014-9-01 12:00
display(after_data.loc[(after_data['box_office']<100000)].sort_values(by='bo
print('**********************************************************','\n')
print('Количество значений ниже 100000 после "2014-9-01 12:00:00" в box_offi
```

	title	puNumber	show_start_date	type	film_studio	production
2273	Каменный цветок	111016714	2014-12-01 12:00:00+00:00	Художественный	Киностудия Мосфильм	
351	Волшебное приключение	124000905	2015-01-18 12:00:00+00:00	Анимационный	Экшион Филмз, Патэ Синема, Болексбразерс	Великобр Ф
211	Одиноким предоставляется общежитие	111018614	2014-12-01 12:00:00+00:00	Художественный	Киностудия Мосфильм	
1034	Монстры на острове	124002515	2015-06-25 12:00:00+00:00	Анимационный	Аби Шуи, Хакайдо Медиа Партнерс, Джи Дрим, Роб...	Япония -
212	Обыкновенный фашизм	112000215	2015-04-23 12:00:00+00:00	Документальный	Киностудия Мосфильм	
2359	Детский юмористический киножурнал Ералаш выпус...	111014014	2014-10-10 12:00:00+00:00	Художественный	ООО Продюсерский центр ЕРАЛАШ	
1968	Срок давности	111021214	2014-12-01 12:00:00+00:00	Художественный	Киностудия Мосфильм	
2651	Песнь моря	124001316	2016-04-19 12:00:00+00:00	Анимационный	Биг Фарм, Картун Салун, Диджитал Графикс, Ириш...	Ирландия - Б Люкс Ф
3193	Мама	111002216	2016-02-12 12:00:00+00:00	Художественный	Киностудия Мосфильм, Киностудия Букурешти, Кин...	СССР - Ру Ф
3076	Снупи и мелочь пузатая в кино	224001016	2016-02-24 12:00:00+00:00	Анимационный	Блю Скай Студиос, XX век Фокс Анимейшн	

10 rows × 21 columns

```
**************************************************

Количество значений ниже 100000 после "2014-9-01 12:00:00" в box_office рав
но: 197
```

In [67]:
```
## посмотрим на подозрительные значения в box_office после "2014-9-01 12:00
display(after_data.loc[(after_data['box_office']>100000)].sort_values(by='bo
print('**************************************************','\n')
```

	title	puNumber	show_start_date	type	film_studio	production_c
7455	Холоп	111021719	2019-12-19 12:00:00+00:00	Художественный	ООО МЕММЕДИА по заказу АО ВБД Груп	Р
5652	Движение вверх	111011817	2017-12-21 12:00:00+00:00	Художественный	ООО Студия ТРИТЭ Никиты Михалкова	Р
6819	Мстители Финал	121005519	2019-04-29 12:00:00+00:00	Художественный	Марвел Студиос	
6548	Т34	111024918	2018-12-21 12:00:00+00:00	Художественный	ООО Кинокомпания МАРСфильм по заказу ООО ММЕ, ...	Р
3487	Пираты Карибского моря Мертвецы не рассказываю...	121009217	2017-05-17 12:00:00+00:00	Художественный	Джерри Брукхаймер Филмз, Уолт Дисней Пикчерз, ...	
2858	Зверополис	124000316	2016-02-15 12:00:00+00:00	Анимационный	Уолт Дисней Анимейшн Студиос, Уолт Дисней Пикчерз	
3754	Тайная жизнь домашних животных Миньоны против ...	124002816	2016-07-05 12:00:00+00:00	Анимационный	Иллюминейшн Интертейнмент, Юниверсал Пикчерз	
6273	Веном	121022018	2018-10-04 12:00:00+00:00	Художественный	Коламбиа Пикчерз, Марвел Интертейнмент, Паскал...	
7215	Малефисента Владычица тьмы	121026219	2019-10-04 12:00:00+00:00	Художественный	Рот Филмз, Уолт Дисней Пикчерз	
7257	Джокер	121027519	2019-10-03 12:00:00+00:00	Художественный	Брон Студиос, Ди Си Комикс, Джоинт Эффорт, Вил...	США - К

10 rows × 21 columns

```
**********************************************************
```

In [68]:

```
## посмотрим на подозрительные значения в box_office до "2014-9-01 12:00:00
display(before_data.loc[(before_data['box_office']<100000)].sort_values(by='
print('**********************************************************','\n')
print('Количество значений ниже 100000 до "2014-9-01 12:00:00" в box_office
```

	title	puNumber	show_start_date	type	film_studio	production_count
1737	Как поймать перо ЖарПтицы	114000513	2013-10-11 12:00:00+00:00	Художественный	ООО Визарт Фильм, ООО Кинокомпания СТВ	Росси
5224	Кунгфу Панда 2	124000911	2011-04-28 12:00:00+00:00	Анимационный	ДримУоркс Анимэйшн	СШ
5039	Иван Царевич и Серый Волк	114000911	2011-12-12 12:00:00+00:00	Анимационный	ООО Студия анимационного кино Мельница	Росси
1025	Большая Ржака	111003512	2012-07-23 12:00:00+00:00	Художественный	ООО Авеста филмс	Росси
691	Дочь	111004712	2012-09-25 12:00:00+00:00	Художественный	ОАО ТПО Киностудия им МГорького, ООО Валдай	Росси

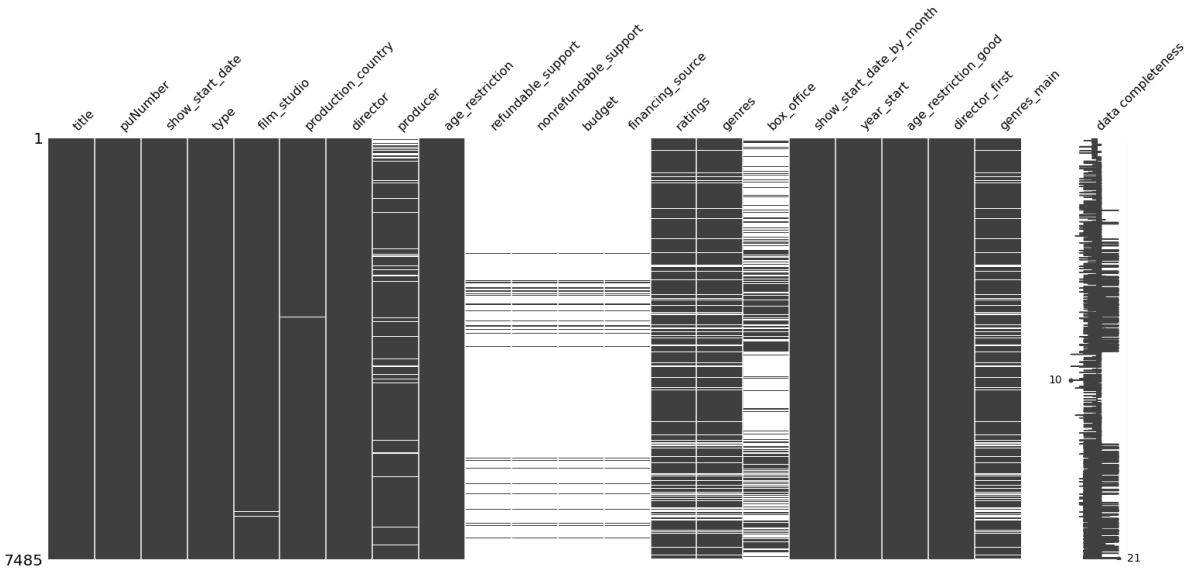
5 rows × 21 columns

```
**************************************************

Количество значений ниже 100000 до "2014-9-01 12:00:00" в box_office равно: 565
```

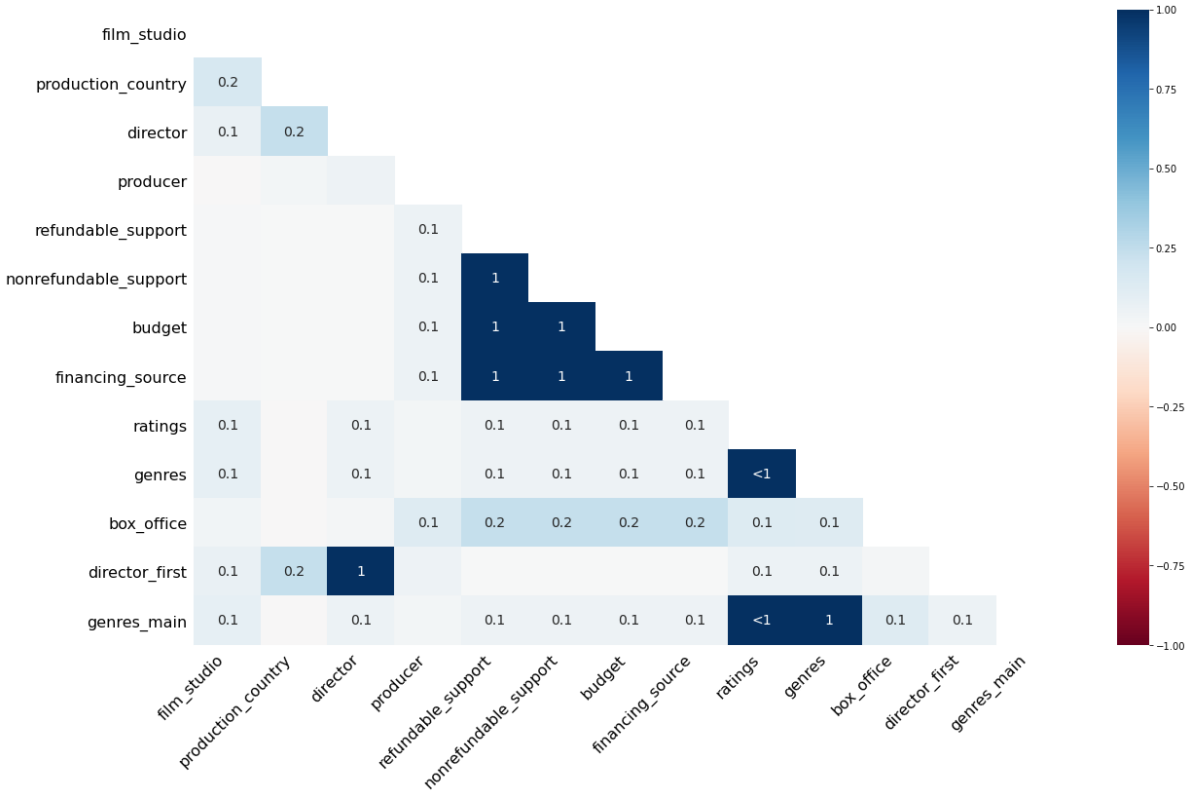
- ***Разобрав значения кассовых сборов можно сделать вывод, что формат записи сумм сменился после 2014-9-01 плюс наложилась возможная валютная переоценка на суммы в рублях, так до указанной даты количество подозрительных значений сумм у фильмов с неверным форматом, либо с ошибками в нем равно 75% тогда как после всего 20%. Выбросы в большую сторону это очень успешные фильмы с высокими кассовыми сборами, они несут ценную информацию. Средняя медианная сумма сборов в этих группах различается на 3 порядка. Более репрезентативная выборка будет после до "2014-9-01 12:00:00". Пропуски оставим, значения править не будем***

Проведём визуальное отображение пропусков и их корреляций

In [69]:
```
# построим матрицу заполняемости данных
msno.matrix(data, labels=True)
plt.show()
```



In [70]:
```
# построим таблицу корреляций пропусков в данных
msno.heatmap(data)
plt.show()
```



- ***после графического отображения наполнения данных видно два уплотнения в столбцах с финансовыми показателями, по видимому это периоды до конца 2014 года и после, что может объяснять разный формат кассовых сборов в зависимости от даты. Верхняя группа имеет кратное количественное превосходство и с учетом более корректной записи сборов будет более информативна для анализа финансовых показателей. Также на гистограмме корреляций пропусков можно увидеть что фильмы получившие гос. поддержку имеют сто процентный показатель взаимосвязи, поэтому эту группу имеет смысл разобрать отдельно. Еще одна пара столбцов с высокой степенью связи пропусков - жанры и рейтинги, что может говорить и дополнении этими столбцами наших данных из отдельного источника. С учетом наличия большого***

количества ошибок и способов записи информации в колонках можно предположить что данные были собраны из разных источников

Разберем значения в столбцах с количественными значениями фильмов получивших финансовую поддержку

In [71]:
```
print('Количество фильмов, получивших государственную поддержку:',data.loc[
```

```
Количество фильмов, получивших государственную поддержку: 332
```

Очевидно что столбцы с данными по фин поддержке и бюджет взаимосвязаны, бюджет складывается из возвратных и невозвратных средств гос поддержки плюс дополнительные привлеченные средства, если они есть. Поэтому сумма первых двух не может превышать бюджет

In [72]:
```
# посмотрим на эти строки
pd.set_option('display.max_columns', None)
display(data.loc[~data['refundable_support'].isna()].sort_values(by='nonrefu
```

	title	puNumber	show_start_date	type	film_studio	production_cou
6471	Три богатыря и наследница престола	114008818	2018-12-15 12:00:00+00:00	Анимационный	ООО Студия анимационного кино Мельница	Рос
2682	Дабл трабл	111009215	2015-05-18 12:00:00+00:00	Художественный	ООО ТаББаК, ООО Весёлая Компания	Рос
6626	Рассвет	111000419	2019-01-31 12:00:00+00:00	Художественный	ООО Форс Медиа	Рос
2531	Бармен	111009615	2015-05-26 12:00:00+00:00	Художественный	АО ВайТ Медиа, ООО Арт Пикчерс Студия	Рос
2732	Неуловимые последний герой	111017415	2015-09-30 12:00:00+00:00	Художественный	ООО Энджой мувиз, ООО Ультра стори	Рос
5658	Три богатыря и принцесса Египта	114003317	2017-12-21 12:00:00+00:00	Анимационный	ООО Студия анимационного кино Мельница	Рос
3041	Крякнутые каникулы	114003615	2015-12-22 12:00:00+00:00	Анимационный	ООО Анимационная студия РИМ	Рос
3223	Кухня Последняя битва	111001517	2017-03-22 12:00:00+00:00	Художественный	ООО Кинокомпания Аврора продакшнс по заказу ОО...	Рос
7465	Иван Царевич и Серый Волк 4	114005019	2019-12-20 12:00:00+00:00	Анимационный	ООО Студия анимационного кино Мельница	Рос
7179	Байкал Сердце мира 3D	112004619	2019-11-01 12:00:00+00:00	Документальный	ООО Продюсерский центр Новое Время	Рос

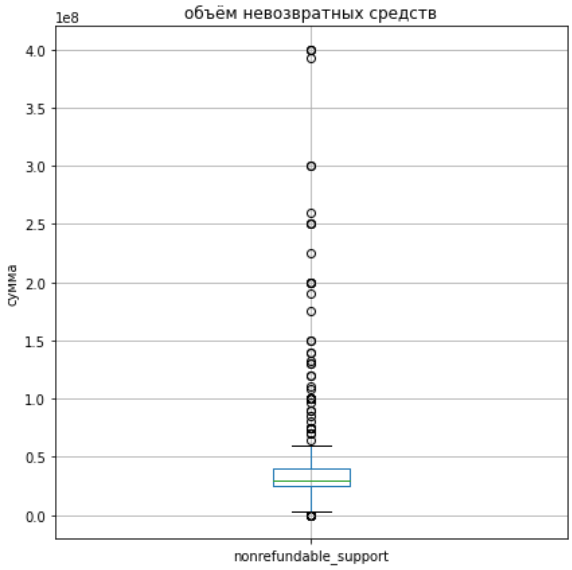
In [73]:
```
# выведим описательную статистику для каждого
# Описательная статистика для nonrefundable_support
print('Описательная статистика для refundable_support:\n**********
display(data.loc[~data['refundable_support'].isna(),'refundable_support'].de
print('**********************************************\n')

# Описательная статистика для nonrefundable_support
print('Описательная статистика для nonrefundable_support:\n*******
display(data.loc[~data['nonrefundable_support'].isna(),'nonrefundable_suppor
print('**********************************************\n')

# Описательная статистика для budget
print('Описательная статистика для budget:\n**********************
display(data.loc[~data['budget'].isna(),'budget'].describe())
print('**********************************************\n')
```

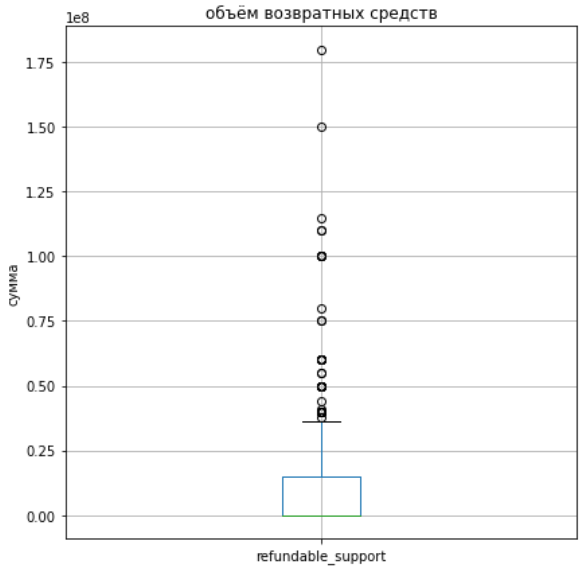
```
Описательная статистика для refundable_support:
**********************************************
```

```
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
**********************************************

Описательная статистика для nonrefundable_support:
**********************************************
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
**********************************************

Описательная статистика для budget:
**********************************************
count             332.00
mean      127,229,716.68
std       188,588,333.12
min                 0.00
25%        42,000,000.00
50%        68,649,916.00
75%       141,985,319.50
max     2,305,074,303.00
Name: budget, dtype: float64
**********************************************
```

In [74]:

```
# посмотрим на выбросы
fig = plt.figure()
fig.set_figheight(7)
fig.set_figwidth(15)

ax1 = fig.add_subplot(121)

data.loc[~data['nonrefundable_support'].isna(),'nonrefundable_support'].plot
plt.ylabel('сумма')
plt.ylim()

ax2 = fig.add_subplot(122)
data.loc[~data['refundable_support'].isna(),'refundable_support'].plot(kind=
plt.ylim()
plt.ylabel('сумма')
plt.show()
```





- ***основная часть средств выделяется на безвозвратной основе, средняя мидианная сумма поддержки около 30 млн.***

In [75]:
```
# основная часть средств выделяется на безвозвратной основе
# проверим по условиям на ошибки в нанных

# отобразим строки где все 3 значения равны 0
display(data.loc[(data['nonrefundable_support']==0)&(data['refundable_suppor
```

title	puNumber	show_start_date	type	film_studio	production_country	director	producer	age_re

In [76]:
```
# проверим отсуцтвые поддержки
display(data.loc[(data['nonrefundable_support']==0)&(data['refundable_suppor
```

title	puNumber	show_start_date	type	film_studio	production_country	director	producer	age_re

In [77]:
```
# если бюджет равет 0, а помощь нет то это явная ошибка. Заменим такие значе
data.loc[data['budget']==0,'budget'] = data.loc[data['budget']==0,'nonrefunc
```

In [78]:
```
# проверим превишение поддержки над бюджетом
display(data.loc[(data['nonrefundable_support']+data['refundable_support'])>
```

title	puNumber	show_start_date	type	film_studio	production_country	director	producer	age_re

- ***Проблем с суммами гос поддержки не выявлено, кроме нулевых значений бюджета при ее наличии. Бросается в глаза разный порядок в формате сумм бюджета и кассовых сборов в строках с датой проката до 2015 года.***
  - **На основе выявленных особенностей можно прийти к выводу, что для дальнейшего анализа лучше использовать выборку после указанной даты.**

к содержанию

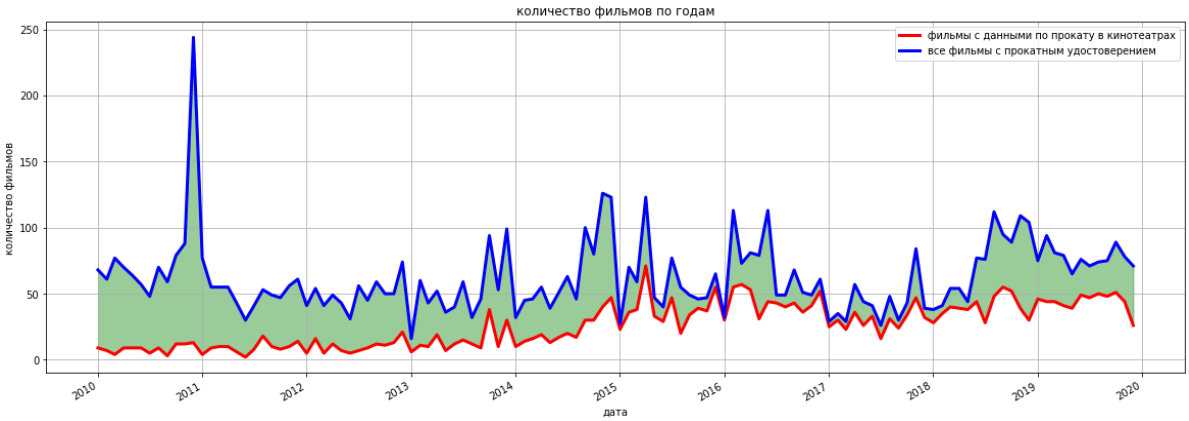
Исследование взаимосвязей по общей выборке

По общей выборке покажем взаимосвязи точность которых не будет страдать от проблем с данными до 2015 года

Проведем анализ изменения количества выходов фильмов в прокат в зависимости от даты

Создадим график изменения количества фильмов по годам для групп с наличием кассовых сборов в кинотеатре и без

In [79]:
```
# сделаем сводные таблица по количества фильмов по годам для разных групп
all_movies = data.pivot_table(index='show_start_date_by_month',values='puNum
movies_by_theatre = data.loc[~data['box_office'].isna()].pivot_table(index='
                                                                     values=

# Переименуем столбцы для легенды
all_movies.columns = ['все фильмы с прокатным удостоверением']
movies_by_theatre.columns = ['фильмы с данными по прокату в кинотеатрах']


# создадим график изменения количества фильмов по годам
ax = movies_by_theatre.plot(c='r',linewidth=3,title='количество фильмов по г
all_movies.plot(ax=ax,figsize=(20,7),grid=True,color='blue',linewidth=3)
plt.xlabel('дата')
plt.ylabel('количество фильмов')
plt.fill_between(all_movies.index,all_movies['все фильмы с прокатным удостов
                 movies_by_theatre['фильмы с данными по прокату в кинотеатрах
plt.show()
print('*************************************************************************

# выделим доли
share = (data.loc[~data['box_office'].isna()].groupby(by='year_start')['puNu
data.groupby(by='year_start')['puNumber'].count())*100
share = share.round(1)
share.index = share.index.to_list()

# построим второй график с долями по годам
share.plot(kind='bar',figsize=(20,5),grid=True,color='g',alpha=0.4)
plt.title('доли фильмов с известными данными по прокату от общего количества

i = range(0,len(share))
for i,k in zip(i,share):
    plt.annotate(k,xy=(i,k+1),horizontalalignment='center')

plt.xlabel('дата')
plt.ylabel('%')
plt.show()
```

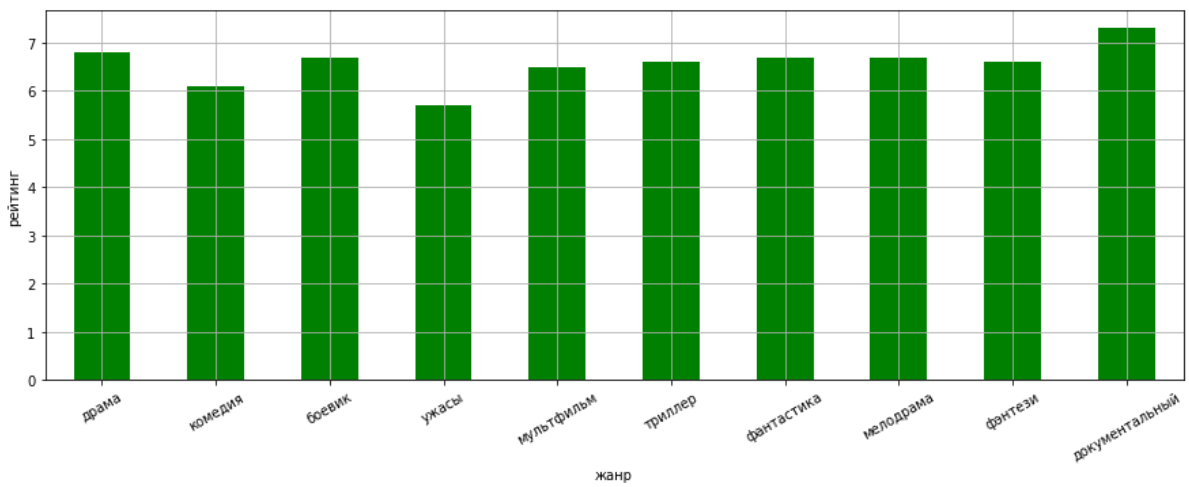


****
****

- ***Количество фильмов, получавших прокатные удостоверения относительно равномерно распределена на всем наблюдаемом периоде, тогда как доля фильмов с указанными сборами сильно меньше до конца 2014 года и держится в пределах 20-30 процентов от общей массы, после этой даты уходит в уровень существенно больший и находится между 50-70 процентами. Данное наблюдение еще раз подтверждает нашу склонность о недостаточной репрезентативности этого периода для анализа финансовых показателей***

отобразим распределение оценок по жанрам топе по количеству фильмов

In [80]:
```
# отобразим распределение оценок на графике

pivot_ganre = data.pivot_table(index='genres_main',values='ratings',aggfunc=
pivot_ganre.columns = ['оценка','количество фильмов']
pivot_ganre = pivot_ganre.sort_values(by='количество фильмов',ascending=Fals
pivot_ganre['оценка'].plot(kind='bar',figsize=(15,5),legend=False,grid=True,
plt.xlabel('жанр')
plt.ylabel('рейтинг')
plt.xticks(rotation=30)
plt.show()
print('**************************************')
print('Топ 10 жанров по медианной оценке с максимальным количеством фильмов'
display(pivot_ganre.sort_values(by='оценка',ascending=False).head(10))
```



****

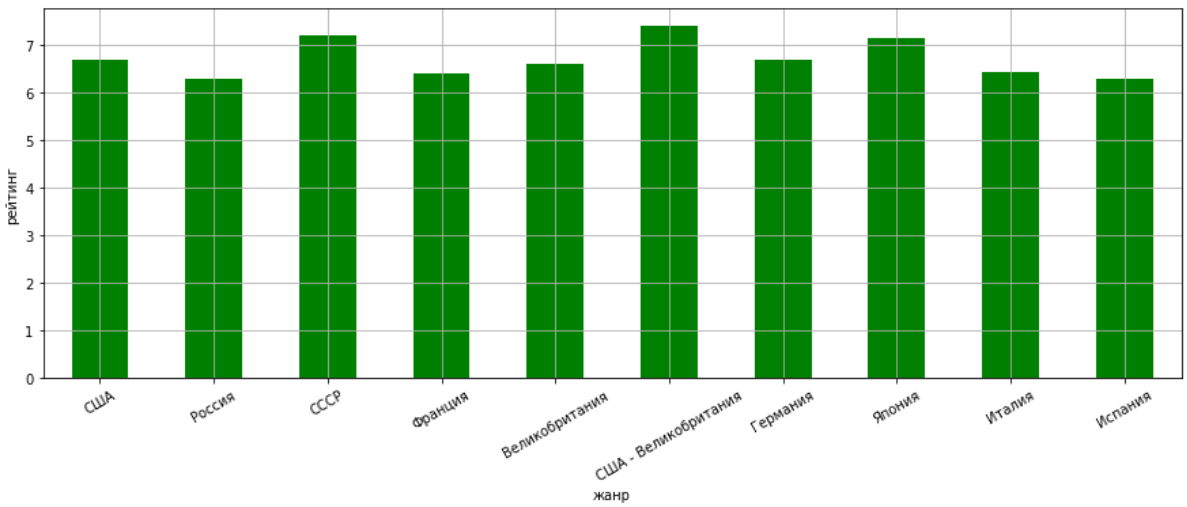
Топ 10 жанров по медианной оценке с максимальным количеством фильмов

genres_main	оценка	количество фильмов
документальный	7.30	219
драма	6.80	1688
боевик	6.70	617
фантастика	6.70	410
мелодрама	6.70	383
триллер	6.60	521
фэнтези	6.60	358
мультфильм	6.50	538
комедия	6.10	697
ужасы	5.70	573

- ***Самый высокий уровень медианной оценки в топ 10 по количеству фильмов имеют жанры документальный и драма.***

отобразим среднюю медианную оценку по странам производства фильма

In [81]:
```
# отобразим среднюю менианную оценку по странам
pivot_country = data.pivot_table(index='production_country',values='ratings'
pivot_country.columns = ['оценка','количество фильмов']
pivot_country = pivot_country.sort_values(by='количество фильмов',ascending=
pivot_country['оценка'].plot(kind='bar',figsize=(15,5),legend=False,grid=Tru
plt.xlabel('жанр')
plt.ylabel('рейтинг')
plt.xticks(rotation=30)
plt.show()
print('****************************************')
print('Топ 10 жанров по медианной оценке с максимальным количеством фильмов'
display(pivot_country.sort_values(by='оценка',ascending=False).head(10))
```



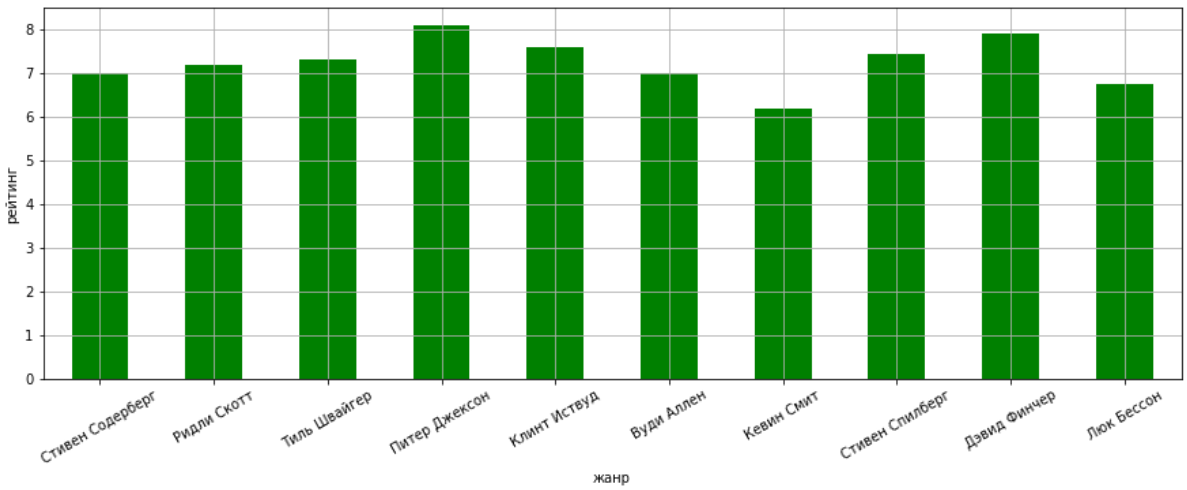

```
****************************************
Топ 10 жанров по медианной оценке с максимальным количеством фильмов
```

production_country	оценка	количество фильмов
США - Великобритания	7.40	105
СССР	7.20	361
Япония	7.15	74
США	6.70	2107
Германия	6.70	96
Великобритания	6.60	192
Италия	6.45	72
Франция	6.40	290
Россия	6.30	1296
Испания	6.30	71

- ***Самый высокий медианный рейтинг из топа по количеству фильмов в у картин снятых СССР и США - Великобритания***

отобразим среднюю медианную оценку по director_first

In [82]:
```
# отобразим среднюю менианную оценку по director_first
pivot_dire = data.pivot_table(index='director_first',values='ratings',aggfur
pivot_dire.columns = ['оценка','количество фильмов']
pivot_dire = pivot_dire.sort_values(by='количество фильмов',ascending=False)
pivot_dire['оценка'].plot(kind='bar',figsize=(15,5),legend=False,grid=True,c
plt.xlabel('жанр')
plt.ylabel('рейтинг')
plt.xticks(rotation=30)
plt.show()
print('****************************************')
print('Топ 10 жанров по медианной оценке с максимальным количеством фильмов'
display(pivot_dire.sort_values(by='оценка',ascending=False).head(10))
```



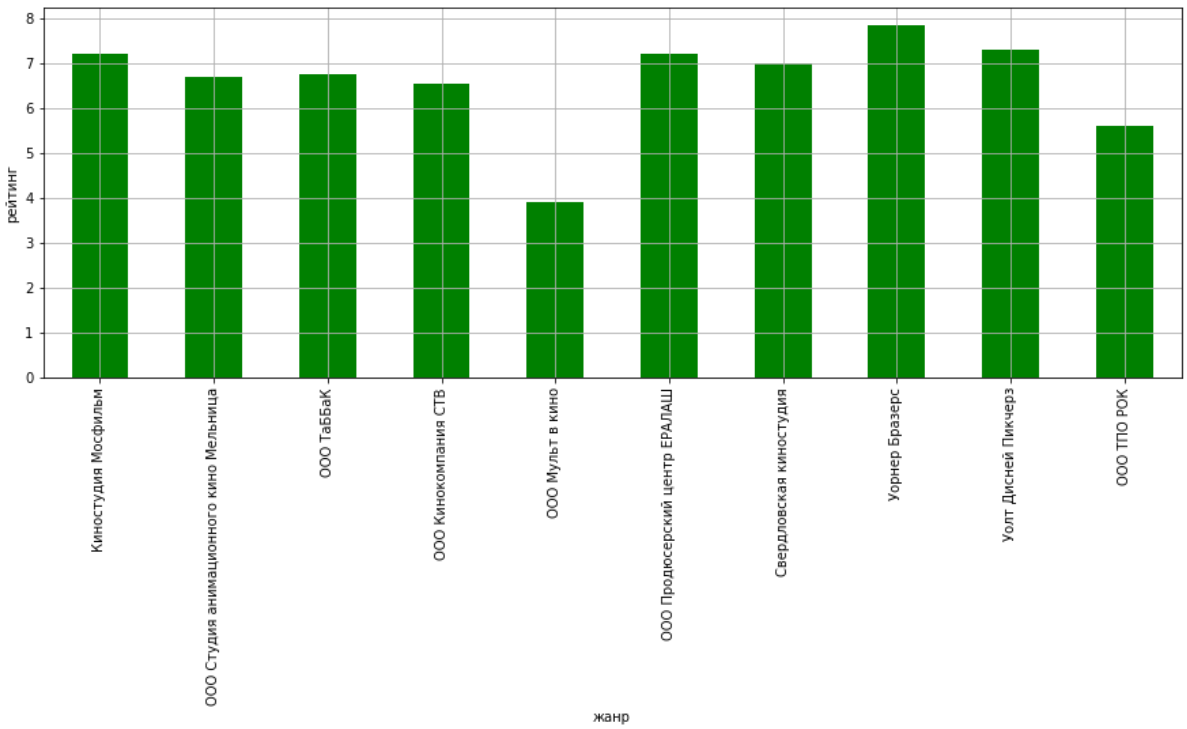

```
****************************************
Топ 10 жанров по медианной оценке с максимальным количеством фильмов
```

director_first	оценка	количество фильмов
Питер Джексон	8.10	13
Дэвид Финчер	7.90	12
Клинт Иствуд	7.60	13
Стивен Спилберг	7.45	12
Тиль Швайгер	7.30	14
Ридли Скотт	7.20	16
Стивен Содерберг	7.00	17
Вуди Аллен	7.00	13
Люк Бессон	6.75	12
Кевин Смит	6.20	12

- ***Самый высокий медианный рейтинг из топа по количеству фильмов у картин Питера Джексона и Дэвида Финчера***

отобразим среднюю медианную оценку по film_studio

In [83]:
```
# отобразим среднюю менианную оценку по film_studio
pivot_studio = data.pivot_table(index='film_studio',values='ratings',aggfunc
pivot_studio.columns = ['оценка','количество фильмов']
pivot_studio = pivot_studio.sort_values(by='количество фильмов',ascending=Fa
pivot_studio['оценка'].plot(kind='bar',figsize=(15,5),legend=False,grid=True
plt.xlabel('жанр')
plt.ylabel('рейтинг')
plt.show()
print('**************************************')
print('Топ 10 жанров по медианной оценке с максимальным количеством фильмов'
display(pivot_studio.sort_values(by='оценка',ascending=False).head(10))
```



```
************************************
Топ 10 жанров по медианной оценке с максимальным количеством фильмов
```

film_studio	оценка	количество фильмов
Уорнер Бразерс	7.85	10
Уолт Дисней Пикчерз	7.30	10
Киностудия Мосфильм	7.20	331
ООО Продюсерский центр ЕРАЛАШ	7.20	10
Свердловская киностудия	7.00	10
ООО ТаББаК	6.75	12
ООО Студия анимационного кино Мельница	6.70	13
ООО Кинокомпания СТВ	6.55	12
ООО ТПО РОК	5.60	9
ООО Мульт в кино	3.90	11

- ***Самый высокий медианный рейтинг из топа по количеству фильмов у картин студий: Уорнер Бразерс, Уолт Дисней Пикчерз и Киностудия Мосфильм***

к содержанию

Анализ финансовых показателей и особенностей по выборке "2014-2019" годов

Для анализа финансовых показателей будем использовать более репрезентативную выборку после "2014-9-01" ввиду большей плотности данных в этом периоде и меньшего количества фильмов с подозрительными показателями в кассовых сборах

In [84]:
```
# выделим данные
data_f = data.loc[(data['show_start_date_by_month']>'2014-09-01 12:00:00+00
```

Отобразим динамику изменения средней и медианной суммы сборов по годам

In [85]:
```
# сделаем график со средними и медианными суммами сборов
pivot_boxes = data_f.pivot_table(index='show_start_date_by_month',values='bo
pivot_boxes.columns = ['Средняя сумма сборов','Медианная сумма сборов']


# построим график изменения этих величин по годам
ax = pivot_boxes['Средняя сумма сборов'].plot(legend=('Средняя сумма сборов'
pivot_boxes['Медианная сумма сборов'].plot(figsize=(20,6),ax=ax,linewidth=4,
plt.fill_between(pivot_boxes.index,pivot_boxes['Средняя сумма сборов'],\
                 pivot_boxes['Медианная сумма сборов'],color='y',alpha=0.1)
plt.title('Изменения средней арифметической и медианной суммы сборов по года
plt.xlabel('дата')
plt.ylabel('сумма')
plt.show()
```

```
# сделаем сводную таблицу со значениями для каждого года
pivot_by_year = data_f.pivot_table(index='year_start',values='box_office',ag
pivot_by_year = pivot_by_year.reset_index()
pivot_by_year.columns = ['год','Средняя сумма сборов','Медианная сумма сборо
print('*****************************************************************\n'
print('       средняя и медианная сумма по годам')
display(pivot_by_year.sort_values(by='Медианная сумма сборов',ascending=Fals
```


```
*****************************************************************
```

средняя и медианная сумма по годам

	год	Средняя сумма сборов	Медианная сумма сборов
3	2017	136,032,793.33	9,968,340.00
4	2018	104,565,059.23	8,891,102.21
1	2015	85,492,132.46	5,003,450.15
5	2019	91,369,261.85	4,627,798.34
2	2016	91,173,904.27	3,915,041.02
0	2014	62,186,830.77	261,210.00

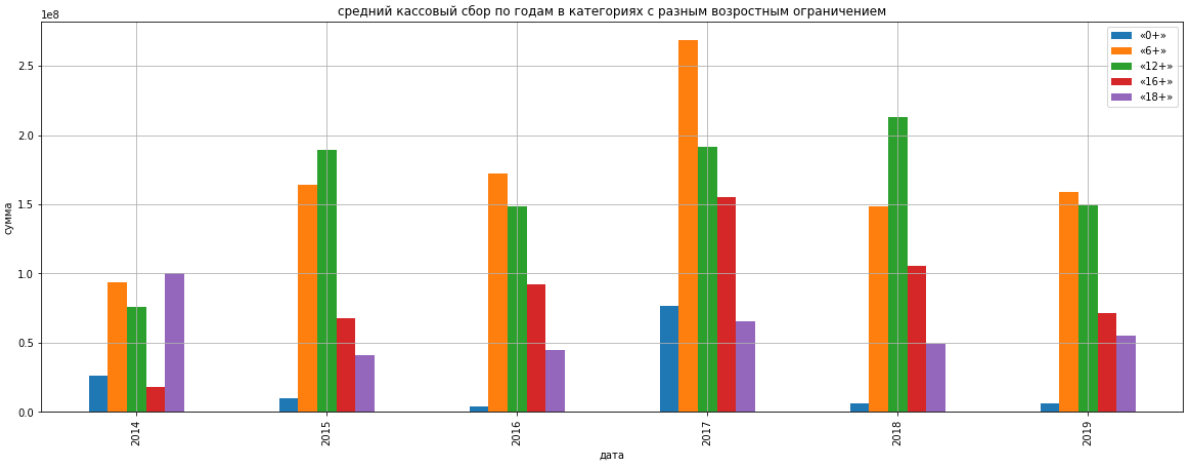
- ***На графике мы отчетливо видим расхождение между средними и медианными значениями сборов на 1-2 порядка в пользу среднего показателя. Это связанно с кратным превышением сумм сборов в высоко бюджетных или просто очень удачных картинах, которые встречаются не так часто в индустрии. Медианная средняя в нашем случае показывает просто тенденцию и не описывает провальные и очень успешные фильмы, тогда как средняя хорошо реагирует на них и учитывает количество фильмов. Более информативный показатель в нашем случае будет среднеарифметическая средняя, она опишет и наличие крайних значений и учтет число фильмов. Опираясь на этот вывод далее финансовые показатели по сборам будем рассчитывать на основе среднеарифметической***

Проанализируем зависимость кассовых сборов от категории возрастных ограничений

Построим линейный график отражающий изменение среднего значения сборов по годам, толщина линии зависит от количества фильмов за весь период для каждой группы. Дополним столбчатой диаграммой для наглядности

In [86]:
```
# зададим переменные
pivot_restr = pd.DataFrame()
pivot_count = pd.DataFrame()
```

```
age_dif = []
ages = ['«0+»','«6+»','«12+»','«16+»','«18+»']
#  создадим таблицу значения по возростам
for age in ages:
    pivot_restr[age] = data_f.loc[data_f['age_restriction_good']==age].\
    pivot_table(index='year_start',values='box_office',aggfunc='mean')['box_

for age in ages:
    pivot_count[age] = data_f.loc[data_f['age_restriction_good']==age].\
    pivot_table(index='year_start',values='box_office',aggfunc='count')['box

# создадим список коэффициентов на основе количества фильмов
for age in ages:
    age_dif.append((len(data_f.loc[data_f['age_restriction_good']==age])/100

# создадим график изменения сумм по годам в зависимости от возростной группы
for a,d in zip(ages,age_dif):
    pivot_restr[a].plot(figsize=(20,4),legend=a,linewidth=d,marker='o',grid=
plt.title('изменение среднего кассового сбора по годам в категориях с разным
plt.xlabel('дата')
plt.ylabel('сумма')

# создадим второй график со средним сборои по годам
pivot_restr.plot(kind='bar',figsize=(20,7),grid=True)
plt.title('средний кассовый сбор по годам в категориях с разным возростным о
plt.xlabel('дата')
plt.ylabel('сумма')
plt.show()
print('**************************************************')
print('средний кассовый сбор по годам в категориях с разным возростным огран
display(pivot_restr)
print('**************************************************')
print('количество фильмов в категориях с разным возростным ограничением')
display(pivot_count)
```




```
**************************************************
средний кассовый сбор по годам в категориях с разным возростным ограничение
м
```

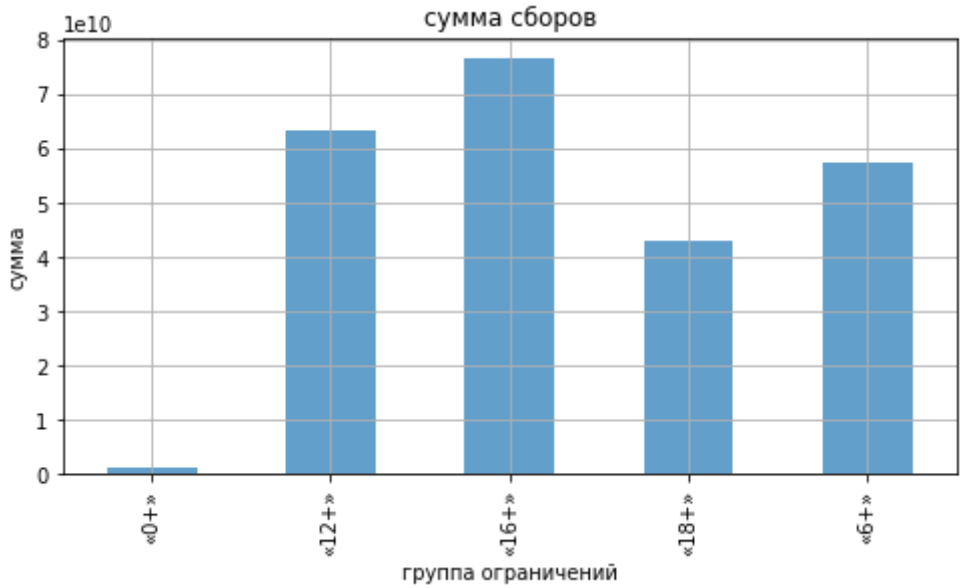
year_start	«0+»	«6+»	«12+»	«16+»	«18+»
2014	26,435,369.94	93,756,158.80	75,565,751.71	17,832,035.10	100,664,597.66
2015	9,975,120.48	164,184,893.80	189,112,250.12	68,072,580.06	41,153,851.27
2016	3,664,118.26	172,187,800.12	148,834,713.71	92,069,328.96	44,992,912.71
2017	76,532,976.67	268,580,936.57	191,498,235.63	154,917,709.92	65,656,432.52
2018	6,489,800.42	148,602,567.99	212,944,651.65	105,792,322.74	48,939,025.15
2019	5,915,355.61	158,673,440.13	149,178,809.10	71,648,421.26	55,310,362.31

****
количество фильмов в категориях с разным возростным ограничением

year_start	«0+»	«6+»	«12+»	«16+»	«18+»
2014	14	18	34	30	21
2015	38	53	72	167	132
2016	41	70	82	181	151
2017	3	45	41	121	147
2018	5	68	67	156	179
2019	3	81	85	178	183

- **покажем суммы сборов за весь период**

In [87]:
```
# сделаем сводную таблицу
pivot_restr_sum = data_f.pivot_table(index='age_restriction_good',values='bo

# построим график
pivot_restr_sum.plot(kind='bar',figsize=(8,4),grid=True,alpha=0.7)
plt.title('сумма сборов')
plt.xlabel('группа ограничений')
plt.ylabel('сумма')
plt.show()
print('**********************************************************')
display(pivot_restr_sum)
```

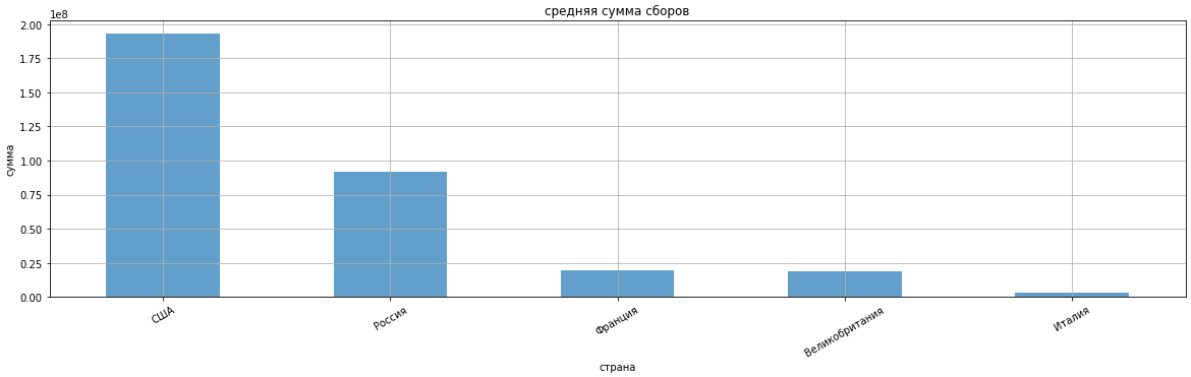


```
**********************************************************
age_restriction_good
«0+»      1,179,172,605.17
«12+»    63,188,682,186.63
«16+»    76,569,694,696.63
«18+»    42,873,572,122.92
«6+»     57,486,221,658.08
Name: box_office, dtype: float64
```

- ***Самый высокий средний кассовый сбор в группах с 6+ и 12+ далее 16+ так как они рассчитаны на основного потребителя и в них самое большое количество высоко бюджетных удачных фильмов. Самую большую общую сумму сборов за этот период имеют фильмы с ограничением 16+ ввиду сочетание количества успешных картин с общим их числом. Тенденция сохраняется на всем периоде наблюдений, незначительно реагируя на общее количество фильмов и наличие высокодоходных экземпляров***

дополнительный расчет максимальной средней доходности в разных группах

In [88]:
```
# отобразим топ по странам производства
pivot_restr_sum = data_f.pivot_table(index='production_country',values='box_
pivot_restr_sum.columns = ['сумма','количество фильмов']
pivot_restr_sum = pivot_restr_sum.sort_values(by='количество фильмов',ascend

pivot_restr_sum['сумма'].plot(kind='bar',figsize=(20,5),grid=True,alpha=0.7)
plt.title('средняя сумма сборов')
plt.xlabel('страна')
plt.ylabel('сумма')
plt.xticks(rotation=30)
plt.show()
print('**********************************************************')
display(pivot_restr_sum)
```

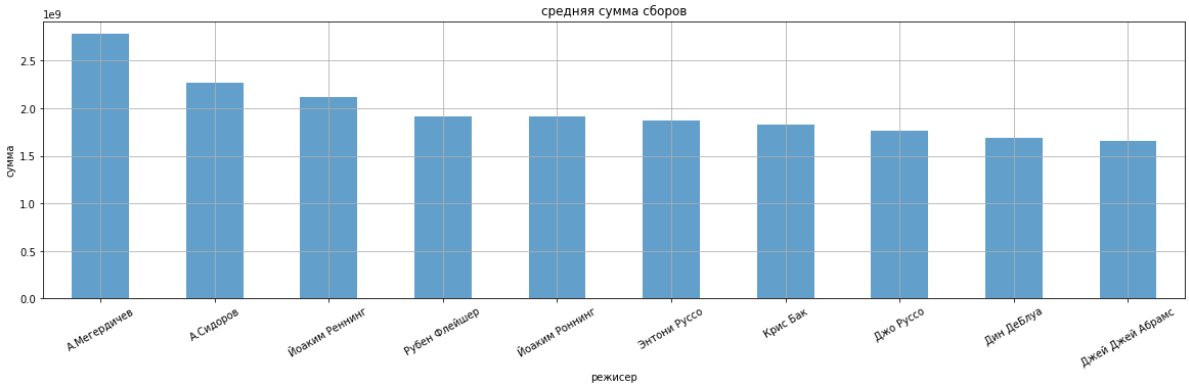



```
**********************************************************
```

production_country	сумма	количество фильмов
США	192,821,965.77	677
Россия	92,013,161.57	553
Франция	19,685,479.52	131
Великобритания	18,603,385.36	69
Италия	3,287,080.89	59

- ***Самые высокие среднии сборы у фильмов из США и России***

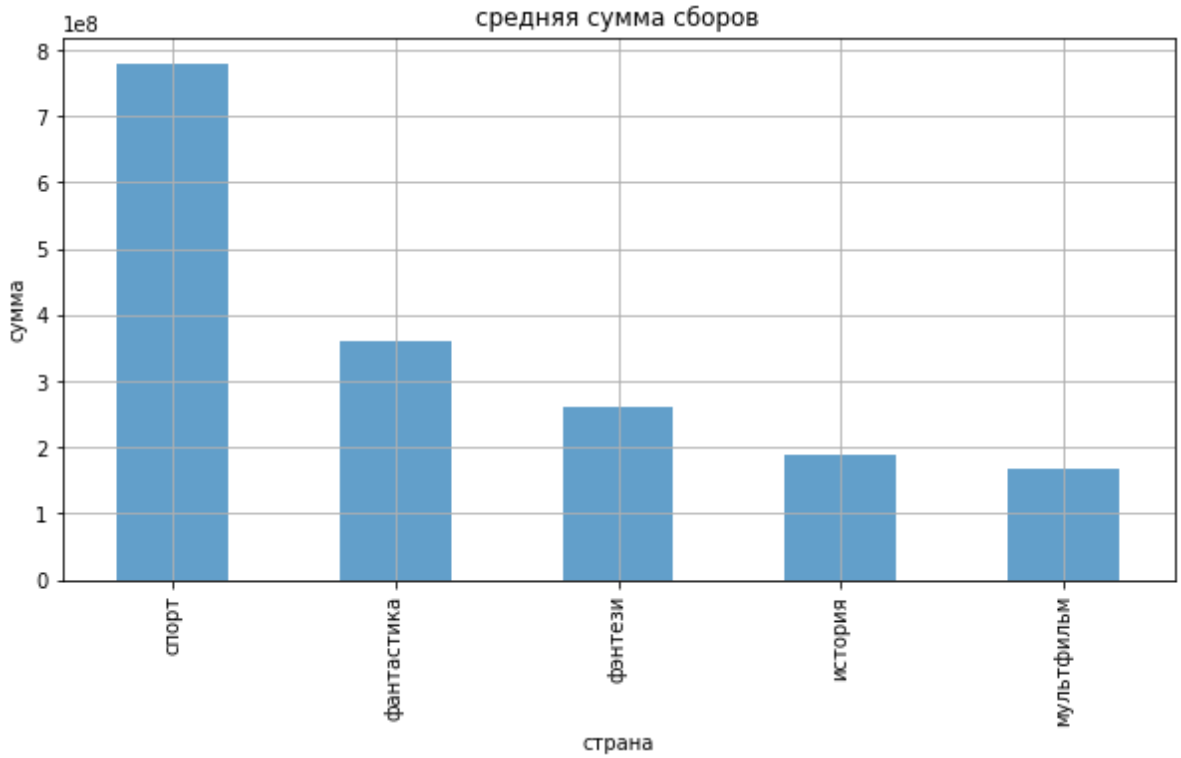
In [89]:
```
# отобразим топ по режисерам
pivot_mean = (data_f.pivot_table(index='director_first',values='box_office',
                  ['box_office'].sort_values(ascending=False).head(10))

pivot_mean.plot(kind='bar',figsize=(20,5),grid=True,alpha=0.7)
plt.title('средняя сумма сборов')
plt.xlabel('режисер')
plt.ylabel('сумма')
plt.xticks(rotation=30)
plt.show()
print('************************************************************')
display(pivot_mean)
```




```
************************************************************
director_first
А.Мегердичев       2,779,686,144.00
А.Сидоров          2,271,754,004.52
Йоаким Реннинг     2,118,396,119.00
Рубен Флейшер      1,913,257,923.06
Йоаким Роннинг     1,911,944,865.95
Энтони Руссо       1,875,989,712.42
Крис Бак           1,827,244,672.81
Джо Руссо          1,766,060,595.91
Дин ДеБлуа         1,689,540,830.24
Джей Джей Абрамс   1,658,861,425.50
Name: box_office, dtype: float64
```

- ***самый высокий средний сбор имеют А.Мегердичев и А.Сидоров***

In [90]:
```
pivot_mean = (data_f.pivot_table(index='genres_main',values='box_office',agg
                  ['box_office'].sort_values(ascending=False).head(5))

pivot_mean.plot(kind='bar',figsize=(10,5),grid=True,alpha=0.7)
plt.title('средняя сумма сборов')
plt.xlabel('страна')
plt.ylabel('сумма')
plt.show()
print('************************************************************')
display(pivot_mean)
```




```
**********************************************************
genres_main
спорт         780,136,863.86
фантастика    359,037,424.15
фэнтези       259,877,656.60
история       187,927,400.20
мультфильм    166,442,631.97
Name: box_office, dtype: float64
```

- ***самый высокий средний сбор имеют фильмы в жанрах: спорт, фантастика и фэнтези***

к содержанию

Исследование фильмов с государственной поддержкой "2014-2019" годов

выделим данные для изучения в отдельный датасет, добавим в них колонку необходимые для анализа колонки

In [91]:
```
# выделим данные
data_s = data_f.loc[~data_f['financing_source'].isna()].copy()

# добавим новые колонки
data_s['sum_support'] = data_s['refundable_support'] + data_s['nonrefundable
data_s['share'] = round(data_s['sum_support']/data_s['budget']*100,1)
data_s['income'] = data['box_office'] - data['budget']
data_s['paid_off'] = data_s['income']>=0
```

In [92]:
```
# проверим
data_s.sort_values(by='box_office').head(5)
```

Out[92]:

	title	puNumber	show_start_date	type	film_studio	production_co
3148	ЯУчитель	111019715	2015-12-02 12:00:00+00:00	Художественный	НП Киностудия детских и юношеских фильмов Илья...	Ро
2526	Вдвоем на льдине	111011015	2015-06-25 12:00:00+00:00	Художественный	ООО Первое творческое объединение	Ро
2802	Битва с экстрасенсами	111011315	2015-07-20 12:00:00+00:00	Художественный	ООО КИНОДАНЦ, ООО КИНОБАЙТ	Ро
3503	День До	111010916	2016-07-06 12:00:00+00:00	Художественный	ООО Кинобюро по заказу ООО Среда и Ко	Ро
2149	РЕВЕРБЕРАЦИЯ	111003515	2015-03-30 12:00:00+00:00	Художественный	ООО Артлайт	Ро

Отобразим динамику изменения количества фильмов с гос поддержкой и изменения сумм выделенных на поддержку в указанный период

In [ ]:

In [93]:
```
# сделаем  сводную саблицу
pivot_gos = data_s.pivot_table(index='year_start',values='box_office',aggfur
pivot_gos.columns = ['количество фильмов','сумма поддержки']

pivot_gos['сумма поддержки'].plot(kind='bar',figsize=(10,4),alpha=0.7,grid=T
plt.title('сумма государственной поддержки в "2014-2020" годах')
plt.xlabel('дата')
plt.ylabel('сумма')
plt.show()
print('*****************************************')
pivot_gos['количество фильмов'].plot(marker='o',figsize=(10,4),alpha=0.7,lir
plt.title('количество фильмов с поддерджкой в "2014-2020" годах')
plt.xlabel('дата')
plt.ylabel('количество фильмов')
plt.show()


print('*****************************************\n')
display(pivot_gos)
```

****

****

year_start	количество фильмов	сумма поддержки
2014	10	922,153,920.80
2015	85	5,785,285,418.14
2016	60	6,081,707,839.10
2017	39	10,865,075,300.96
2018	56	9,934,069,010.25
2019	60	8,409,627,454.63

- ***сумма поддержки растет с 2015 по 2017 далее плавно снижается, количество фильмов ведёт себя противоположным образом***

Исследуем распределение количества фильмов и объем средств выделенных по источникам в указанном периоде

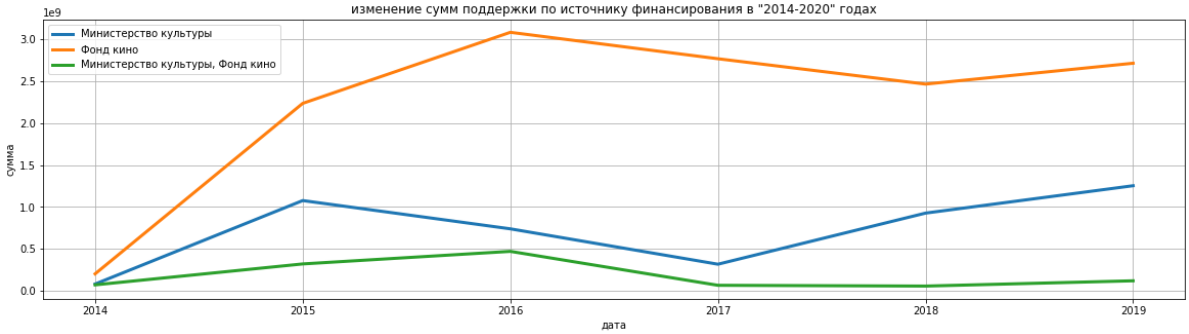
In [94]:
```
data_s.head()
```

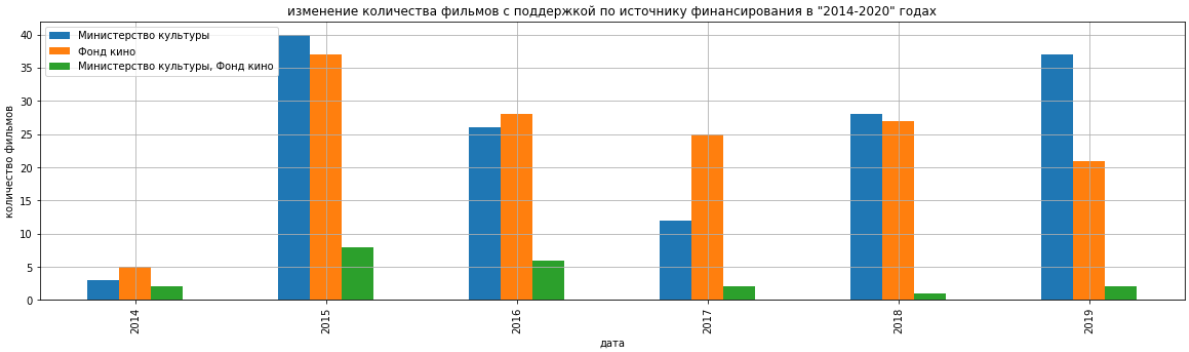
Out[94]:

	title	puNumber	show_start_date	type	film_studio	production_count
1853	Тайна Сухаревой башни Чародей равновесия	114000115	2015-02-18 12:00:00+00:00	Анимационный	ООО Студия МастерФильм	Росс
1869	А зори здесь тихие	111002915	2015-03-16 12:00:00+00:00	Художественный	ООО Компания РеалДакота	Росс
1870	Две женщины	111013714	2014-10-02 12:00:00+00:00	Художественный	ООО Продюсерский Центр Хорошо Продакшн, Rezo P...	Россия - Франц - Латв
1902	Призрак	111001815	2015-03-02 12:00:00+00:00	Художественный	ООО Водород 2011	Росс
1911	Ведьма	111002215	2015-03-05 12:00:00+00:00	Художественный	ООО Кинокомпания Ракурс	Росс

In [95]:

```
# сделаем  сводные таблицы с суммами поддержки и количеством фильмов
pivot_sum = pd.DataFrame()
pivot_count = pd.DataFrame()

for sourse in data_s['financing_source'].unique():
    pivot_sum[sourse] = data_s.loc[data_s['financing_source']==sourse].pivot

for sourse in data_s['financing_source'].unique():
    pivot_count[sourse] = data_s.loc[data_s['financing_source']==sourse].piv


# отобразим  изменение сумм поддержки по источнику финансирования в "2014-2(
pivot_sum.plot(figsize=(20,5),grid=True,linewidth=3)
plt.title('изменение сумм поддержки по источнику финансирования в "2014-202(
plt.xlabel('дата')
plt.ylabel('сумма')

# отобразим изменение количества фильмов с поддержкой по источнику финансиро
pivot_count.plot(kind='bar',figsize=(20,5),grid=True)
plt.title('изменение количества фильмов с поддержкой по источнику финансиров
plt.xlabel('дата')
plt.ylabel('количество фильмов')
plt.show()

print('*****************************************\n')
print('суммы поддержки')
display(pivot_sum)
print('*****************************************\n')
print('количество фильмов')
display(pivot_count)
```





****

суммы поддержки

year_start	Министерство культуры	Фонд кино	Министерство культуры, Фонд кино
2014	79,500,000.00	201,000,000.00	69,502,299.00
2015	1,075,810,000.00	2,236,049,285.00	319,382,174.00
2016	738,331,000.00	3,084,104,482.00	469,200,000.00
2017	316,000,000.00	2,768,624,781.00	64,346,881.00
2018	926,000,000.00	2,465,969,465.00	55,000,000.00
2019	1,253,000,000.00	2,715,000,000.00	118,000,000.00

****

количество фильмов

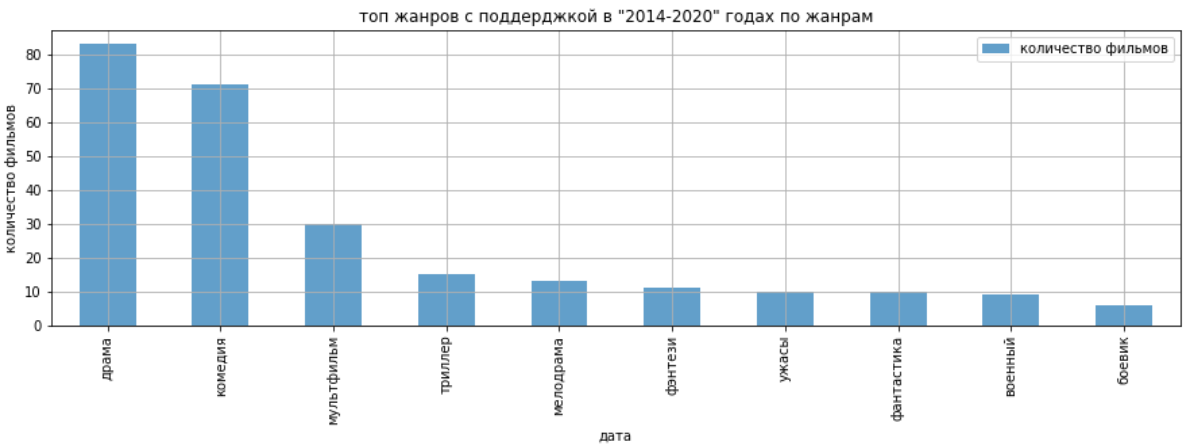
year_start	Министерство культуры	Фонд кино	Министерство культуры, Фонд кино
2014	3	5	2
2015	40	37	8
2016	26	28	6
2017	12	25	2
2018	28	27	1
2019	37	21	2

- ***самую большую поддержку получают фильмы от Фонда кино, причем ее сумма имеет небольшую тенденцию к росту, тогда как количество фильмов падает до 2017 года и затем стабилизируется. Поддержка от Министерства культуры имеет гораздо меньший объем и снижается до 2017 года с последующим плавным ростом, количество фильмов повторяет тенденцию объема финансирования.***

Объемы совместной помощи незначительны. Отсюда можно сделать вывод, что сумма поддержки фонда кино на каждый фильмы выросла тогда как Министерство культуры нет

отобразим основные жанры поддержки

In [96]:
```
ganre = data_s.pivot_table(index='genres_main',values='puNumber',aggfunc='co
sort_values(by='puNumber',ascending=False).head(10)
ganre.columns = ['количество фильмов']

ganre.plot(kind='bar',figsize=(15,4),alpha=0.7,grid=True)
plt.title('топ жанров с поддерджкой в "2014-2020" годах по жанрам')
plt.xlabel('дата')
plt.ylabel('количество фильмов')
plt.show()


print('****************************************\n')
display(ganre)
```



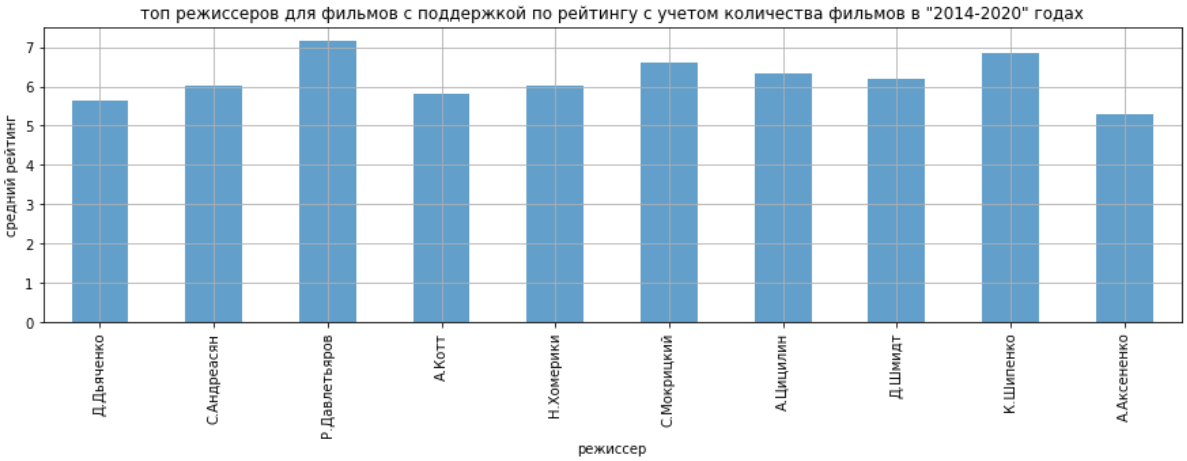
****

genres_main	количество фильмов
драма	83
комедия	71
мультфильм	30
триллер	15
мелодрама	13
фэнтези	11
ужасы	10
фантастика	10
военный	9
боевик	6

- ***поддержку в основном получают фильмы в жанрах: драма и комедия***

отобразим топ режиссеров для фильмов с поддержкой по рейтингу с учетом количества фильмов

In [97]:
```
# построим график
director_reit = data_s.pivot_table(index='director_first',values='ratings',a

director_reit.columns = ['средний рейтинг','количество фильмов']
director_reit = director_reit.sort_values(by='количество фильмов',ascending=

director_reit['средний рейтинг'].plot(kind='bar',figsize=(15,4),alpha=0.7,gr
plt.title('топ режиссеров для фильмов с поддержкой по рейтингу с учетом коли
plt.xlabel('режиссер')
plt.ylabel('средний рейтинг')
plt.show()


print('*****************************************\n')
display(round(director_reit.sort_values(by='средний рейтинг',ascending=False
```

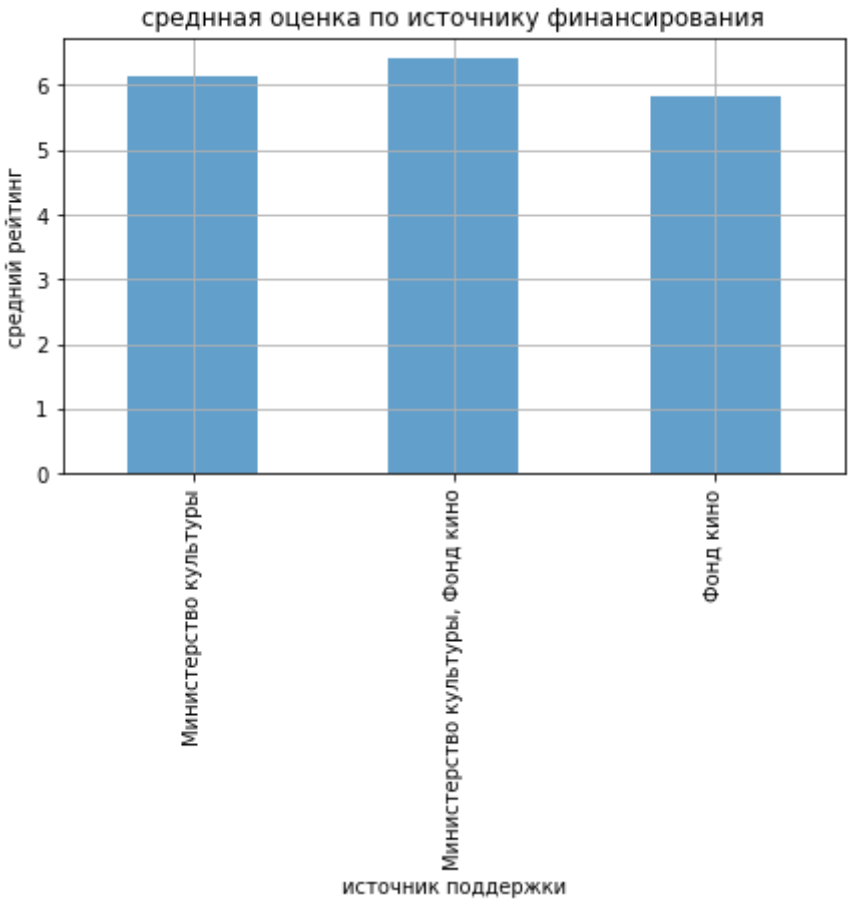


****

director_first	средний рейтинг	количество фильмов
Р.Давлетьяров	7.20	4
К.Шипенко	6.80	3
С.Мокрицкий	6.60	3
А.Цицилин	6.30	3
Д.Шмидт	6.20	3
Н.Хомерики	6.00	3
С.Андреасян	6.00	4
А.Котт	5.80	3
Д.Дьяченко	5.60	4
А.Аксененко	5.30	3

- ***самый высокий средний рейтинг с учетом их количества у фильмов Р.Давлетьяров и К.Шипенко***

In [98]:
```
sours_reit = data_s.pivot_table(index='financing_source',values='ratings',ag

sours_reit.columns = ['средний рейтинг']


sours_reit.plot(kind='bar',figsize=(7,4),alpha=0.7,grid=True,legend=False)
plt.title('среднная оценка по источнику финансирования')
plt.xlabel('источник поддержки')
plt.ylabel('средний рейтинг')
plt.show()


print('*****************************************\n')
display(round(sours_reit,1))
```



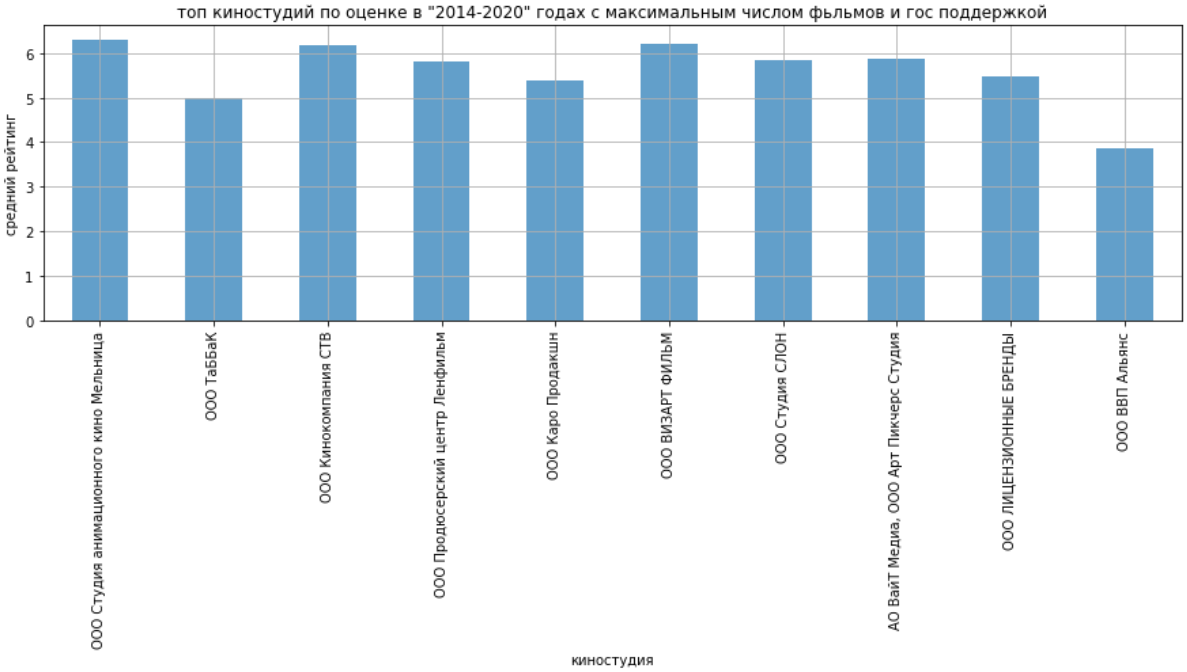
*****

financing_source	средний рейтинг
Министерство культуры	6.10
Министерство культуры, Фонд кино	6.40
Фонд кино	5.80

- ***Совместные фильмы Министерства культуры и Фонда кино в целом имеют средний рейтинг в 6.4 балла тогда как Фонд кино - 5.8***

In [99]:
```
film_studio_reit = data_s.pivot_table(index='film_studio',values='ratings',a

film_studio_reit.columns = ['средний рейтинг','количество фильмов']
film_studio_reit = film_studio_reit.sort_values(by='количество фильмов',asce
```

```
film_studio_reit['средний рейтинг'].plot(kind='bar',figsize=(15,4),alpha=0.7
plt.title('топ киностудий по оценке в "2014-2020" годах с максимальным числ
plt.xlabel('киностудия')
plt.ylabel('средний рейтинг')
plt.show()


print('****************************************\n')
display(round(film_studio_reit.sort_values(by='средний рейтинг',ascending=Fa
```




```
****************************************
```

film_studio	средний рейтинг	количество фильмов
ООО Студия анимационного кино Мельница	6.30	7
ООО ВИЗАРТ ФИЛЬМ	6.20	4
ООО Кинокомпания СТВ	6.20	6
АО ВайТ Медиа, ООО Арт Пикчерс Студия	5.90	3
ООО Студия СЛОН	5.80	3
ООО Продюсерский центр Ленфильм	5.80	4
ООО ЛИЦЕНЗИОННЫЕ БРЕНДЫ	5.50	3
ООО Каро Продакшн	5.40	4
ООО ТаББаК	5.00	6
ООО ВВП Альянс	3.90	3

- ***работы от "ООО Студия анимационного кино Мельница" с гос поддержкой имеют средний рейтинг 6.3 балла, а "ООО ВВП Альянс " напротив 3.9***

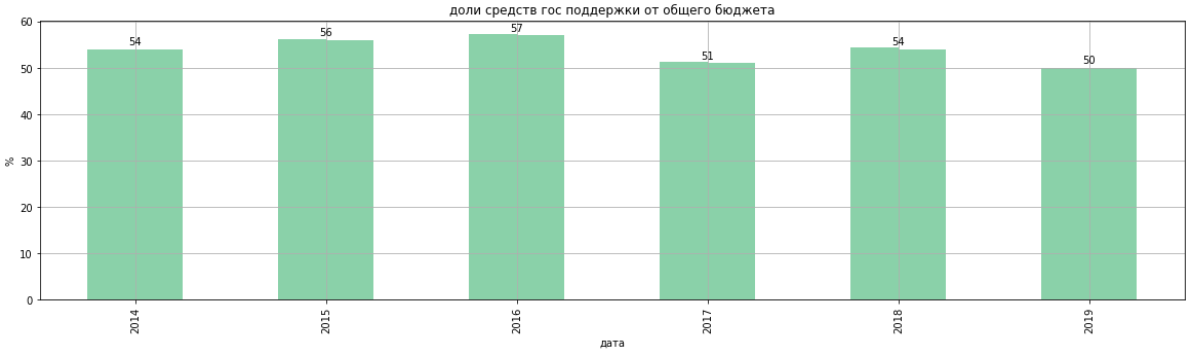
доходность фильмов с гос поддержкой

Построим график общей доходности фильмов с гос поддержкой

In [100…

```
# сделаем сводные таблица по количества фильмов по годам для разных групп
all_budget = data_s.pivot_table(index='show_start_date_by_month',values='buc
sum_support = data_s.pivot_table(index='show_start_date_by_month',values='su

# Переименуем столбцы для легенды
all_budget.columns = ['средняя сумма бюджета']
sum_support.columns = ['средняя сумма поддержки']


# создадим график изменения количества фильмов по годам
ax = all_budget.plot(c='r',linewidth=3,title='изменение средних значений сум
sum_support.plot(ax=ax,figsize=(20,7),grid=True,color='b',linewidth=3)
plt.xlabel('дата')
plt.ylabel('сумма')
plt.fill_between(all_budget.index,all_budget['средняя сумма бюджета'],sum_su
plt.show()
print('**********************************************************************


# построим второй график с долями по годам
share_support = data_s.pivot_table(index='year_start',values='share',aggfunc
share_support['доля'] = share_support.astype(int)
share_support.plot(kind='bar',figsize=(20,5),grid=True,color='mediumseagreen
plt.title('доли средств гос поддержки от общего бюджета')

i = range(0,len(share_support))

for i,k in zip(i,share_support['доля']):
    plt.annotate(k,xy=(i,k+1),horizontalalignment='center')

plt.xlabel('дата')
plt.ylabel('%')
plt.show()
```

****

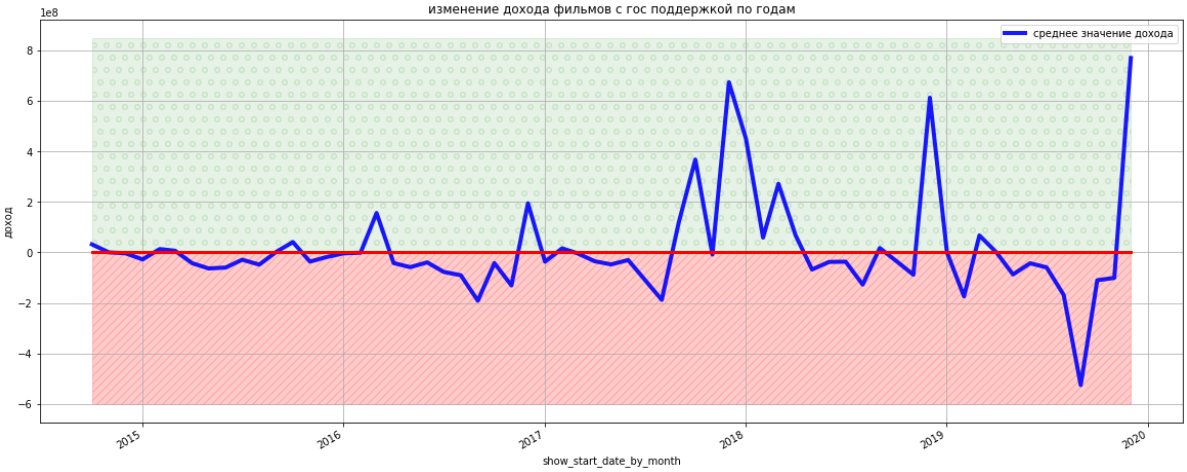


- ***средняя доля гос поддержки от общего бюджета держится между 50 и 57 процентами на протяжении всего периода анализа***

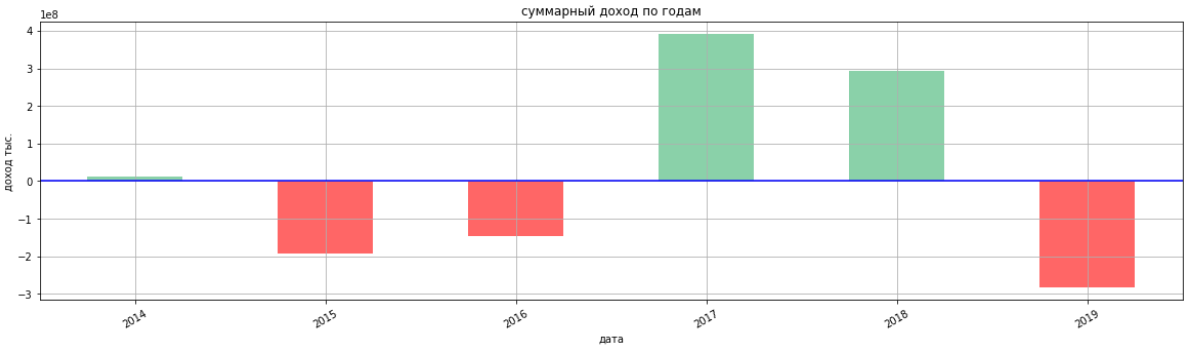
Исследуем доходны фильмов в указанный период

In [101…

```
# сделаем сводные таблица по количества фильмов по годам для разных групп
income_all = data_s.pivot_table(index='show_start_date_by_month',values='inc
# Переименуем столбцы для легенды

income_all.columns = ['среднее значение дохода']


income_all['line'] = income_all['среднее значение дохода']*0
ax = income_all['среднее значение дохода'].plot(alpha=0.9,grid=True,color='b
income_all['line'].plot(figsize=(20,8),grid=True,color='r',linewidth=3,lines


plt.fill_between(income_all.index,income_all['line'],850000000,\
                 hatch='o',color='g',alpha=0.1)
plt.fill_between(income_all.index,income_all['line'],-600000000,\
                 hatch='///',color='r',alpha=0.2)

plt.title('изменение дохода фильмов с гос поддержкой по годам')
plt.ylabel('дата')
plt.ylabel('доход')
plt.show()

print('*******************************************************')

share_paid_off = data_s.pivot_table(index='year_start',values='income',aggfu
share_paid_off.columns = ['суммарный доход тыс.','количество фильмов']
share_paid_off['суммарный доход тыс.'] = round(share_paid_off['суммарный дох


share_paid_off['суммарный доход тыс.'].plot(kind='bar',figsize=(20,5),grid=T
                           color=(share_paid_off['суммарный доход тыс.']>=0



plt.title('суммарный доход по годам')

plt.axhline(y=0, color='b')
plt.xticks(rotation=30)
plt.xlabel('дата')
plt.ylabel('доход тыс.')
plt.show()
print('***********************************************')
display(share_paid_off.sort_values(by='суммарный доход тыс.',ascending=False
```



****



****

year_start	суммарный доход тыс.	количество фильмов
2017	391,350,273.30	39
2018	293,340,491.10	56
2014	11,585,964.90	10
2016	-145,452,882.90	60
2015	-191,138,480.50	85
2019	-282,544,643.70	60

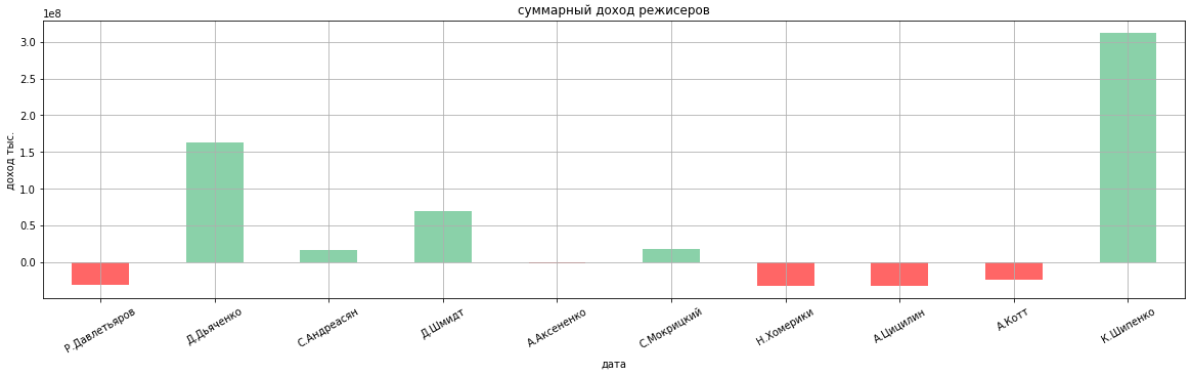
- ***только в 2017 и в 2018 годах из периода наблюдения фильмы с гос поддержкой имеют положительную среднюю сумму дохода. Самый убыточный был 2019 самый прибыльный 2017***

Отобразим доход для топа режиссеров по количеству фильмов с гос поддержкой

In [102…

```
share_paid_off = data_s.pivot_table(index='director_first',values='income',a
share_paid_off.columns = ['суммарный доход тыс.','количество фильмов']
share_paid_off['суммарный доход тыс.'] = round(share_paid_off['суммарный дох

share_paid_off = share_paid_off.sort_values(by='количество фильмов',ascendir

share_paid_off['суммарный доход тыс.'].plot(kind='bar',figsize=(20,5),grid=T
                            color=(share_paid_off['суммарный доход тыс.']>=0
```

```
plt.title('суммарный доход режисеров')


plt.xticks(rotation=30)
plt.xlabel('дата')
plt.ylabel('доход тыс.')
plt.show()
print('************************************************')
display(share_paid_off.sort_values(by='суммарный доход тыс.',ascending=False
```




```
************************************************
```

director_first	суммарный доход тыс.	количество фильмов
К.Шипенко	312,284,155.00	3
Д.Дьяченко	162,420,332.20	4
Д.Шмидт	69,831,226.10	3
С.Мокрицкий	18,389,703.70	3
С.Андреасян	15,769,416.90	4
А.Аксененко	-1,193,891.50	3
А.Котт	-24,820,973.10	3
Р.Давлетьяров	-31,510,084.60	4
А.Цицилин	-32,183,656.70	3
Н.Хомерики	-32,556,171.60	3

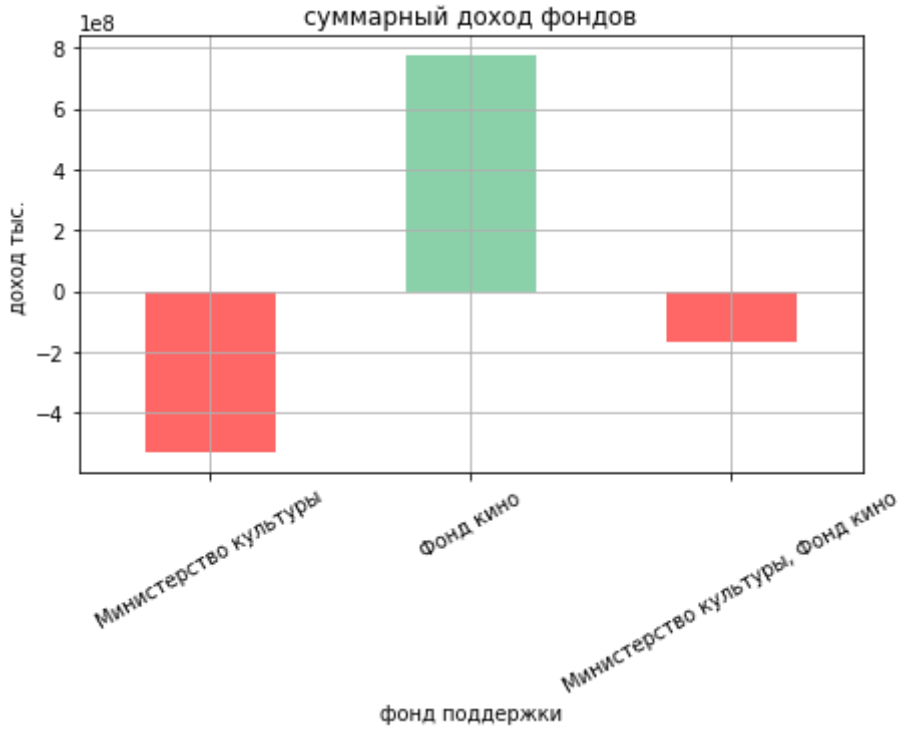
- ***у фильмов К.Шипенко и Д.Дьяченко самый высокий суммарный доход, тогда как фильмы А.Цицилин и Н.Хомерики понесли самые большие убытки из топ 10 по их количеству***

Отобразим доход фильмов с гос поддержкой по источникам финансирования

In [103…

```
share_paid_financing_source = data_s.pivot_table(index='financing_source',va
share_paid_financing_source.columns = ['суммарный доход тыс.','количество фи
share_paid_financing_source['суммарный доход тыс.'] = round(share_paid_finar

share_paid_financing_source = share_paid_financing_source.sort_values(by='ко

share_paid_financing_source['суммарный доход тыс.'].plot(kind='bar',figsize=
                            color=(share_paid_financing_source['суммарный до
```

```
plt.title('суммарный доход фондов')


plt.xticks(rotation=30)
plt.xlabel('фонд поддержки')
plt.ylabel('доход тыс.')
plt.show()
print('************************************************')
display(share_paid_financing_source.sort_values(by='суммарный доход тыс.',as
```




```
************************************************
```

financing_source	суммарный доход тыс.	количество фильмов
Фонд кино	772,864,475.20	143
Министерство культуры, Фонд кино	-166,480,234.00	21
Министерство культуры	-529,243,519.00	146

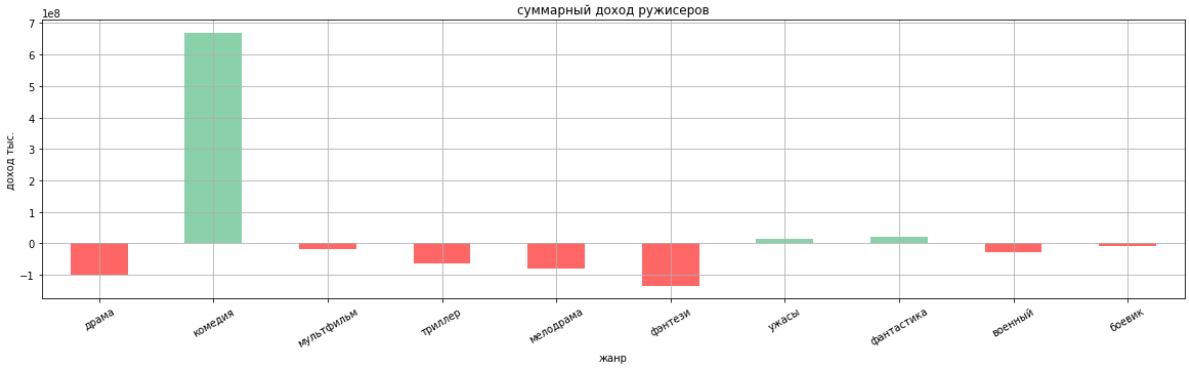
- ***только фильмы с поддержкой от Фонда кино имеют положительный суммарный доход за период наблюдения, тогда как фильмы с поддержкой Министерства культуры понесли убытки***

Проанализируем доход по жанрам для фильмов с поддержкой

In [104…

```
# сделаем сводную таблицу
share_paid_genres_main = data_s.pivot_table(index='genres_main',values='inc
share_paid_genres_main.columns = ['суммарный доход тыс.','количество фильмов
share_paid_genres_main['суммарный доход тыс.'] = round(share_paid_genres_mai

# запишем топ
share_paid_genres_main = share_paid_genres_main.sort_values(by='количество
```

```
# построим график
share_paid_genres_main['суммарный доход тыс.'].plot(kind='bar',figsize=(20,5
                                color=(share_paid_genres_main['суммарный доход т
plt.title('суммарный доход ружисеров')
plt.xticks(rotation=30)
plt.xlabel('жанр')
plt.ylabel('доход тыс.')
plt.show()
print('***********************************************')
display(share_paid_genres_main.sort_values(by='суммарный доход тыс.',ascendi
```




```
***********************************************
```

genres_main	суммарный доход тыс.	количество фильмов
комедия	670,423,673.90	71
фантастика	21,910,346.40	10
ужасы	14,624,556.40	10
боевик	-8,750,787.40	6
мультфильм	-17,477,925.20	30
военный	-26,823,635.30	9
триллер	-65,051,075.00	15
мелодрама	-78,951,111.90	13
драма	-100,049,179.60	83
фэнтези	-134,428,715.70	11

- ***в целом самый высокий доход и количество имеют фильмы в жанре комедия тогда как самые высокие убытки у фэнтези и драм из топ 10 по количеству***

Выделим доли окупившихся фильмов

In [105…

```
# сделаем сводную таблицу
share_paid_off = data_s.pivot_table(index='year_start',values='paid_off',agg
share_paid_off.columns = ['доля','количество фильмов']
share_paid_off['доля'] = round(share_paid_off['доля']*100,1)


# построим график
share_paid_off['доля'].plot(kind='bar',figsize=(20,5),grid=True,color='mediu
plt.title('доли окупившихся фильмов')

i = range(0,len(share_paid_off))
```

```
for i,k in zip(i,share_paid_off['доля']):
    plt.annotate(k,xy=(i,k+0.25),horizontalalignment='center')
plt.xticks(rotation=30)
plt.xlabel('дата')
plt.ylabel('%')
plt.show()
print('************************************************')
display(share_paid_off)
```


```
************************************************
```

year_start	доля	количество фильмов
2014	30.00	10
2015	21.20	85
2016	20.00	60
2017	38.50	39
2018	35.70	56
2019	18.30	60

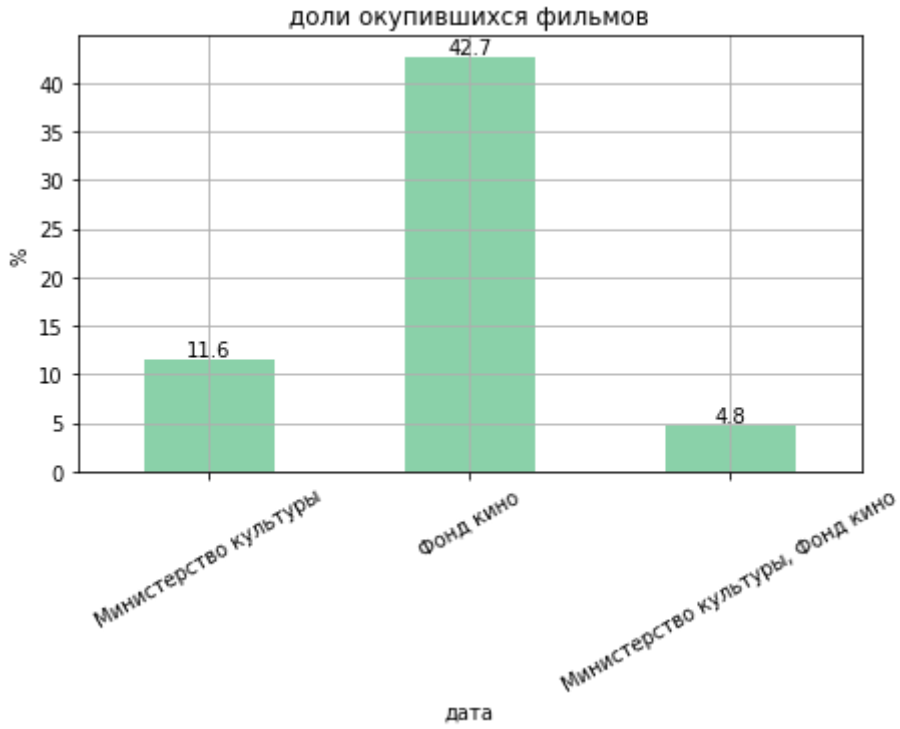
- ***доля окупившихся фильмов держится между 18% и 38%, 2017 и 2018 года самые удачные в этом плане и имеют 38.5% и 35.7% окупившихся фильмов тогда как 2019 всего 18.3%***

Посчитаем доли окупившихся фильмов для разных источников финансирования

In [106…

```
# сделаем сводную таблицу
share_paid_off = data_s.pivot_table(index='financing_source',values='paid_of
share_paid_off.columns = ['доля','количество фильмов']
share_paid_off['доля'] = round(share_paid_off['доля']*100,1)

share_paid_off = share_paid_off.sort_values(by='количество фильмов',ascendir

# построим график
share_paid_off['доля'].plot(kind='bar',figsize=(7,4),grid=True,color='mediu
plt.title('доли окупившихся фильмов')

i = range(0,len(share_paid_off))

for i,k in zip(i,share_paid_off['доля']):
    plt.annotate(k,xy=(i,k+0.25),horizontalalignment='center')
plt.xticks(rotation=30)
plt.xlabel('дата')
plt.ylabel('%')
```

```
plt.show()
print('************************************************')
display(share_paid_off)
```




```
************************************************
```

financing_source	доля	количество фильмов
Министерство культуры	11.60	146
Фонд кино	42.70	143
Министерство культуры, Фонд кино	4.80	21

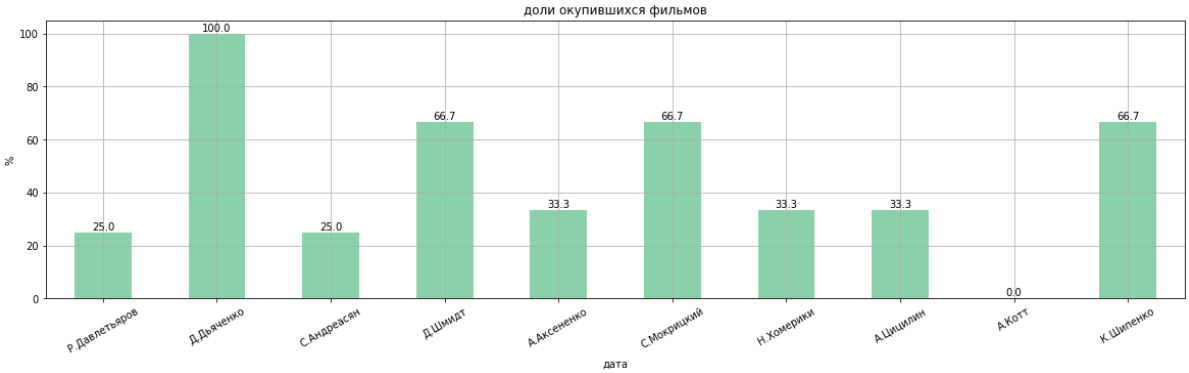
- ***фильмы с поддержкой Фонда кино имеют кратно выше долю окупившихся в сравнении с поддержанными Министерством культуры, она составляет 42.7% в отличии от 11.6% с поддержкой Министерства культуры***

Посчитаем доли окупившихся фильмов для режиссеров из топ 10 по количеству фильмов

In [107…

```
# сделаем сводную таблицу
share_paid_off = data_s.pivot_table(index='director_first',values='paid_off'
share_paid_off.columns = ['доля','количество фильмов']
share_paid_off['доля'] = round(share_paid_off['доля']*100,1)

share_paid_off = share_paid_off.sort_values(by='количество фильмов',ascendir

# построим график
share_paid_off['доля'].plot(kind='bar',figsize=(20,5),grid=True,color='mediu
plt.title('доли окупившихся фильмов')

i = range(0,len(share_paid_off))

for i,k in zip(i,share_paid_off['доля']):
    plt.annotate(k,xy=(i,k+1),horizontalalignment='center')
plt.xticks(rotation=30)
```

```
plt.xlabel('дата')
plt.ylabel('%')
plt.show()
print('************************************************')
display(share_paid_off)
```



****

director_first	доля	количество фильмов
Р.Давлетьяров	25.00	4
Д.Дьяченко	100.00	4
С.Андреасян	25.00	4
Д.Шмидт	66.70	3
А.Аксененко	33.30	3
С.Мокрицкий	66.70	3
Н.Хомерики	33.30	3
А.Цицилин	33.30	3
А.Котт	0.00	3
К.Шипенко	66.70	3

- ***все фильмы Д.Дьяченко окупились тогда как у А.Котт ни одного***

Посчитаем доли окупившихся фильмов для жанров из топ 10 по количеству фильмов

In [108…

```
# сделаем сводную таблицу
share_paid_off = data_s.pivot_table(index='genres_main',values='paid_off',ag
share_paid_off.columns = ['доля','количество фильмов']
share_paid_off['доля'] = round(share_paid_off['доля']*100,1)

share_paid_off = share_paid_off.sort_values(by='количество фильмов',ascendir

share_paid_off['доля'].plot(kind='bar',figsize=(20,5),grid=True,color='mediu
plt.title('доли окупившихся фильмов')

i = range(0,len(share_paid_off))

for i,k in zip(i,share_paid_off['доля']):
    plt.annotate(k,xy=(i,k+0.25),horizontalalignment='center')
plt.xticks(rotation=30)
plt.xlabel('дата')
plt.ylabel('%')
```

```
plt.show()
print('************************************************')
display(share_paid_off)
```

****

genres_main	доля	количество фильмов
драма	16.90	83
комедия	43.70	71
мультфильм	40.00	30
триллер	13.30	15
мелодрама	0.00	13
фэнтези	9.10	11
ужасы	30.00	10
фантастика	20.00	10
военный	33.30	9
боевик	33.30	6

- ***в среднем комедии и мультфильмы лучше окупаются чем прочие жанры, в жанре мелодрама не окупилось ни одного фильма***

Подведем итог, составим сводную таблицу с основными финансовыми годовыми параметрами для фильмов с гос поддержкой***

In [109…

```
# общая сводная таблица
pivot_all = data_s.pivot_table(index='year_start',values='paid_off',aggfunc=
pivot_all.columns = ['количество фильмов','доля окупившихся фильмов в %']
pivot_all['доля окупившихся фильмов в %'] = round(pivot_all['доля окупившихс
pivot_all['сумма сборов'] = data_s.pivot_table(index='year_start',values='bo
pivot_all['сумма бюджетов'] = data_s.pivot_table(index='year_start',values='
pivot_all['суммарный доход'] = data_s.pivot_table(index='year_start',values=
pivot_all['суммарная поддержка'] = data_s.pivot_table(index='year_start',val


display(pivot_all)
```

year_start	количество фильмов	доля окупившихся фильмов в %	сумма сборов	сумма бюджетов	суммарный доход	с п
2014	10	30.00	922,153,920.80	922,153,920.80	115,859,648.80	350,
2015	85	21.20	5,785,285,418.14	5,785,285,418.14	-1,911,384,804.86	3,631,
2016	60	20.00	6,081,707,839.10	6,081,707,839.10	-1,454,528,828.90	4,291,
2017	39	38.50	10,865,075,300.96	10,865,075,300.96	3,913,502,732.96	3,148,
2018	56	35.70	9,934,069,010.25	9,934,069,010.25	2,933,404,911.25	3,446,
2019	60	18.30	8,409,627,454.63	8,409,627,454.63	-2,825,446,437.37	4,086,

In [110…
```
print('Суммарный доход за 5 лет составил:',round(pivot_all['суммарный доход'
```

```
Суммарный доход за 5 лет составил: 771407221.9
```

к содержанию

Итог исследования

Выводы по итогам предобработки и анализу данных по столбцам

- ***данные предоставлены в период с 2010 по конец 2019 годов. Видна закономерность увеличения количества фильмов в прокате в зависимости от начала летнего или зимнего сезонов отпусков, что может объяснять расчет индустрии на сезонный приток посетителей.***
- ***Подавляющее большинство фильмов в прокате художественные***
- ***топ 3 страны производителя по количеству фильмов в прокате - США, Россия и СССР.***
- ***некоторые фильмы выходили в прокат несколько раз под разными прокатными номерами. Фильм 'Волшебная страна' выходил в прокат 5 раз.***
- ***в топе по количеству прокатов фильмы 16+, ограничения 12+ и 18+ имеют примерно равное количество показав. Внизу рейтинга 0+ и 6+ соответственно. Все логично, количество показав соответствует размерам групп потребителей данного контента.***
- ***самое большое количество фильмов вышло в прокат от киностудии Мосфильм.***
- ***О.Семёнова и Стивен Содерберг в топе по количеству фильмов в прокате.***
- ***самое большое количество фильмов в прокате были в жанре драмы, далее идут комедии и боевики.***
- ***Министерство культуры и Фонд кино поддержали по 164 и 146 фильма соответственно, 22 фильма получили их совместную поддержку.***
- ***средняя медианная оценка равна 6.6 баллам, основная их часть расположилась между 5.9 и 7.2 баллами.***
- ***формат записи сумм сборов сменился после 2014-9-01 плюс наложилась возможная валютная переоценка на суммы в рублях, так до указанной даты количество подозрительных значений сумм у фильмов с неверным форматом,***

либо с ошибками в нем равно 75% тогда как после всего 20%. Выбросы в большую сторону по суммам сборов это очень успешные фильмы с высокими кассовыми сборами, они несут ценную информацию. Средняя медианная сумма сборов до и после указанной даты различается на 3 порядка. Более репрезентативная выборка будет после до "2014-9-01 12:00:00".

- *после графического отображения наполнения данных видно два уплотнения в столбцах с финансовыми показателями, по-видимому это периоды до конца 2014 года и после, что может объяснять разный формат кассовых сборов в зависимости от даты. Верхняя группа имеет кратное количественное превосходство и с учетом более корректной записи сборов будет более информативна для анализа финансовых показателей. Также на гистограмме корреляций пропусков можно увидеть что фильмы получившие гос. поддержку имеют сто процентный показатель взаимосвязи, поэтому эту группу имеет смысл разобрать отдельно. Еще одна пара столбцов с высокой степенью связи пропусков - жанры и рейтинги, что может говорить и дополнении этими столбцами наших данных из отдельного источника. С учетом наличия большого количества ошибок и способов записи информации в колонках можно предположить что данные были собраны из разных источников*
- *столбцы с данными по фин поддержке и бюджет взаимосвязаны, бюджет складывается из возвратных и невозвратных средств гос поддержки плюс дополнительные привлеченные средства, если они есть. Поэтому сумма первых двух не может превышать бюджет*
- *основная часть средств выделяется на безвозвратной основе, средняя медианная сумма поддержки около 30 млн.*

Выводы по итогам исследование взаимосвязей по общей выборке:

- *Количество фильмов, получавших прокатные удостоверения относительно равномерно распределена на всем наблюдаемом периоде, тогда как доля фильмов с указанными сборами сильно меньше до конца 2014 года и держится в пределах 20-30 процентов от общей массы, после этой даты уходит в уровень существенно больший и находится между 50-70 процентами. Данное наблюдение еще раз подтверждает нашу склонность о недостаточной репрезентативности этого периода для анализа финансовых показателей*
- *Самый высокий уровень медианной оценки в топ 10 по количеству фильмов имеют жанры документальный и драма.*
- *Самый высокий медианный рейтинг из топа по количеству фильмов у картин снятых СССР и США - Великобритания*
- *Самый высокий медианный рейтинг из топа по количеству фильмов у картин Питера Джексона и Дэвида Финчера*
- *Самый высокий медианный рейтинг из топа по количеству фильмов у картин студий: Уорнер Бразерс, Уолт Дисней Пикчерз и Киностудия Мосфильм*

Выводы по итогам анализа финансовых показателей и особенностей по выборке "2014-2019" годов:

- *имеется расхождение между средними и медианными значениями сборов на 1-2 порядка в пользу среднего показателя. Это связанно с кратным превышением сумм сборов в высоко бюджетных или просто очень удачных картинах, которые встречаются не так часто в индустрии. Медианная средняя в нашем случае показывает просто тенденцию и не описывает провальные и очень успешные фильмы, тогда как средняя хорошо реагирует на них и учитывает количество фильмов*
- *Самый высокий средний кассовый сбор в группах с 6+ и 12+ далее 16+ так как они рассчитаны на основного потребителя и в них самое большое количество высоко бюджетных удачных фильмов. Самую большую общую сумму сборов за этот период имеют фильмы с ограничением 16+ ввиду сочетание количества успешных картин с общим их числом. Тенденция сохраняется на всем периоде наблюдений, незначительно реагируя на общее количество фильмов и наличие высокодоходных экземпляров*
- *Самые высокие средние сборы у фильмов из США и России*
- *самый высокий средний сбор имеют А.Мегердичев и А.Сидоров*
- *самый высокий средний сбор имеют фильмы в жанрах: спорт, фантастика и фэнтези*

Выводы по итогам исследования фильмов с государственной поддержкой "2014-2019" годов:

- *сумма поддержки ростет с 2015 по 2017 далее плавно снижается, количество фильмов ведёт себя противоположным образом*
- *самую большую поддержку получают фильмы от Фонда кино, причем ее сумма имеет небольшую тенденцию к росту, тогда как количество фильмов падает до 2017 года и затем стабилизируется. Поддержка от Министерства культуры имеет гораздо меньший объем и снижается до 2017 года с последующим плавным ростом, количество фильмов повторяет тенденцию объема финансирования. Объемы совместной помощи незначительны. Отсюда можно сделать вывод, что сумма поддержки фонда кино на каждый фильмы выросла тогда как Министерство культуры нет*
- *средняя доля гос поддержки от общего бюджета фильмов держится между 50 и 57 процентами на протяжении всего периода анализа*
- *только в 2017 и в 2018 годах из периода наблюдения фильмы с гос поддержкой имеют положительную среднюю сумму дохода. Самый убыточный был 2019 самый прибыльный 2017*
- *у фильмов К.Шипенко и Д.Дьяченко самый высокий суммарный доход, тогда как фильмы А.Цицилин и Н.Хомерики понесли самые большие убытки из топ 10 по их количеству*
- *только фильмы с поддержкой от Фонда кино имеют положительный суммарный доход за период наблюдения, тогда как фильмы с поддержкой Министерства культуры понесли убытки*
- *в целом самый высокий доход и количество имеют фильмы в жанре комедия тогда как самые высокие убытки у фэнтези и драмм из топ 10 по количеству*
- *Поддержку в основном получают фильмы в жанрах: драма и комедия*

- *доля окупившихся фильмов держится между 18% и 38%, 2017 и 2018 года самые удачные в этом плане и имеют 38.5% и 35.7% окупившихся фильмов тогда как 2019 всего 18.3%*
- *фильмы с поддержкой Фонда кино имеют кратно выше долю окупившихся в сравнении с поддержанными Министерством культуры, она составляет 42.7% в отличие от 11.6% с поддержкой Министерства культуры*
- *все фильмы Д.Дьяченко окупились тогда как у А.Котт ни одного*
- *в среднем комедии и мультфильмы лучше окупаются чем прочие жанры, в жанре мелодрама не окупилось ни одного фильма*
- *самый высокий средний рейтинг с учетом их количества у фильмов Р.Давлетьяров и К.Шипенко*
- *Совместные фильмы Министерства культуры и Фонда кино в целом имеют средний рейтинг в 6.4 балла тогда как Фонд кино - 5.8*
- *работы от "ООО Студия анимационного кино Мельница" с гос поддержкой имеют средний рейтинг 6.3 балла, а "ООО ВВП Альянс " напротив 3.9*
- *Суммарный доход за 5 лет составил: 771407221.9*