

Цель исследования:

Необходимо выявить определяющие популярность марки вина закономерности и попытаться выяснить, что можно предложить покупателям вина при выборе вина. Это позволит сделать ставку на потенциально популярный продукт и спланировать например рекламную кампанию для интернет-магазинов, осуществляющих продажи вина.

Выполнение задачи предполагает:

1. Предобработку данных
2. Исследовательский анализ данных
3. Составление портрета пользователя.
4. Исследование статистических показателей.
5. Проверку гипотез.
6. Выводы

Цель этого проекта — выявить, какие признаки больше всего влияют на рейтинг вина. Для анализа используется набор данных из Kaggle, крупнейшего в мире сообщества специалистов по данным и машинному обучению. Набор данных состоит из 13 признаков (2 числовых признака и 11 категориальных признаков).

Столбцы данных

- Страна - страна происхождения вина.
- Описание — описание вкусового профиля вина.
- Обозначение - виноградник, откуда берется виноград для вина.
- Баллы - количество баллов на которое критик журнала Wine Enthusiast оценил вино по шкале от 1 до 100.
- Цена - стоимость одной бутылки вина.
- Провинция — провинция или штат, из которого произведено вино.
- Регион 1 — зона виноделия в провинции или штате (например, долина Напа в Калифорнии).
- Регион 2 — (не обязательно) более конкретный регион в винодельческой области (например, Резерфорд в долине Напа).
- Разновидность — сорт винограда, из которого делают вино (например, Пино Нуар).
- Винодельня — винодельня, производящая вино.

Шаг 1. Открытие файла с данными и изучение общей информации

Шаг 2. Подготовка данных

- Заменить названия столбцов (привести к нижнему регистру).
- Преобразовать данные в нужные типы. Описать, в каких столбцах заменили тип данных и почему.
- Обработать пропуски при необходимости.
- Объяснить, почему заполнили пропуски определенным образом или почему не стали это делать.
- Описать причины, которые могли привести к пропускам.
- Посчитать средние цены для каждой страны.
- Внести новый столбец "Континенты" country_to_continent = {
'Italy':'Europe',

```
'Portugal':'Europe',
'US':'North America',
'Spain':'Europe',
'France':'Europe',
'Germany':'Europe',
'Argentina':'Latin America',
'Chile':'Latin America',
'Australia': 'Oceania',
'Austria': 'Europe',
'South Africa': 'Africa',
'New Zealand': 'Oceania',
'Israel': 'Asia',
'Hungary':'Europe',
'Greece':'Europe',
'Romania':'Europe',
'Mexico':'Latin America',
'Canada':'North America',
'Turkey': 'Asia',
'Czech Republic': 'Europe',
'Slovenia': 'Europe',
'Luxembourg': 'Europe',
'Croatia': 'Europe',
'Georgia':'Europe',
'Uruguay': 'Latin America',
'England': 'Europe',
'Lebanon': 'Asia',
'Serbia': 'Europe',
'Brazil': 'Latin America',
'Moldova': 'Europe',
'Morocco':'Africa',
'Peru':'Latin America',
'India':'Asia',
'Bulgaria':'Europe',
'Cyprus': 'Europe',
'Armenia':'Asia',
'Switzerland':'Europe',
'Bosnia and Herzegovina':'Europe',
'Ukraine':'Europe',
'Slovakia':'Europe',
'Macedonia':'Europe',
'China':'Asia',
'Egypt':'Africa'
}
```

Шаг 3. Провести исследовательский анализ данных

- Определить, какие сорта лидируют по рейтингам. Найти популярные сорта по региону.
- Выбрать сорта с наибольшими ценами. Для каждого региона найдите среднюю цену вина.
- Определить, популярные сорта вина в бюджетном сегменте.
- Определить, какие сорта вина лидируют по рейтингам.
- Построить график «ящик с усами» по рейтингам в разбивке по странам, по сортам вина.

- Выявить закономерность влияния на цену цвета и рейтинга. Построить диаграмму рассеяния и посчитать корреляцию.

Шаг 4. Составить портрет потребителя каждого региона

Определить для пользователя каждого континента :

- Самые популярные сорта (топ-5).
- Влияет ли рейтинг на цены по регионам?

Шаг 5. Провести исследование статистических показателей

- Выполнить подсчитать среднего количества, дисперсии и стандартного отклонения для цен на продукт различных регионов. Построить гистограммы. Описать распределения.
- Построить линейную регрессию зависимости между ценой продукта и его рейтингом.

Шаг 6. Проверка гипотез

- H0: Средние пользовательские рейтинги красного и белого вина одинаковые.
- H1: Средние пользовательские рейтинги красного и белого вина разные.
- H0: Средние пользовательские рейтинги двух популярных сортов вина одинаковые.
- H1: Средние пользовательские рейтинги двух популярных сортов вина разные.
- H0: Средние пользовательские рейтинги двух континентов одинаковые.
- H1: Средние пользовательские рейтинги двух континентов разные.

Задать самостоятельно пороговое значение alpha.

Вывод

1. Предобработка данных

Основные задачи этапа:

1. Импортировать библиотеки;
2. Загрузить данные;
3. Провести замену названий столбцов;
4. Преобразовать данные в соответствующие типы;
5. Обработать пропуски (при необходимости);
6. Подсчитать суммарные продажи во всех регионах и вынести эти данные в отдельный столбец.

Импортируем необходимые библиотеки

In [1]:

```
import pandas as pd          # вызов библиотеки pandas
import numpy as np           # импорт библиотеки numpy
import matplotlib.pyplot as plt # импорт библиотеки matplotlib
import seaborn as sns         # импорт библиотеки seaborn
from scipy import stats as st # импорт библиотеки scipy
import scipy
import warnings              # импорт библиотеки warnings
```

Загрузка данных

In [2]:

```
#frame = pd.read_csv('/datasets/games.csv') # прописать адрес csv файла для корректной загрузки
games=pd.read_csv('games.csv')
```

In [4]:

```
# вывод на экран датасета
display(games)
```

	Name	Platform	Year_of_Release	Genre	NA_sales	EU_sales	JP_sales	Other_sales
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	11.27	8.89	10.22	
...
16710	Samurai Warriors: Sanada Maru	PS3	2016.0	Action	0.00	0.00	0.01	
16711	LMA Manager 2007	X360	2006.0	Sports	0.00	0.01	0.00	
16712	Haitaka no Psychedelica	PSV	2016.0	Adventure	0.00	0.00	0.01	
16713	Spirits & Spells	GBA	2003.0	Platform	0.01	0.00	0.00	
16714	Winning Post 8 2016	PSV	2016.0	Simulation	0.00	0.00	0.01	

16715 rows × 11 columns

Подготовка данных

In [6]:

```
# переименование столбцов, перевод названий в нижний регистр
games.columns = games.columns.str.lower()
```

In [7]:

```
display(games)
```

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_s
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	11.27	8.89	10.22	
...
16710	Samurai Warriors: Sanada Maru	PS3	2016.0	Action	0.00	0.00	0.01	
16711	LMA Manager 2007	X360	2006.0	Sports	0.00	0.01	0.00	
16712	Haitaka no Psychedelica	PSV	2016.0	Adventure	0.00	0.00	0.01	
16713	Spirits & Spells	GBA	2003.0	Platform	0.01	0.00	0.00	
16714	Winning Post 8 2016	PSV	2016.0	Simulation	0.00	0.00	0.01	

16715 rows × 11 columns

In [8]:

```
# просмотр уникальных значений столбца пользовательского рейтинга
print(games['user_score'].unique())
```

```
['8' nan '8.3' '8.5' '6.6' '8.4' '8.6' '7.7' '6.3' '7.4' '8.2' '9' '7.9'
 '8.1' '8.7' '7.1' '3.4' '5.3' '4.8' '3.2' '8.9' '6.4' '7.8' '7.5' '2.6'
 '7.2' '9.2' '7' '7.3' '4.3' '7.6' '5.7' '5' '9.1' '6.5' 'tbd' '8.8' '6.9'
 '9.4' '6.8' '6.1' '6.7' '5.4' '4' '4.9' '4.5' '9.3' '6.2' '4.2' '6' '3.7'
 '4.1' '5.8' '5.6' '5.5' '4.4' '4.6' '5.9' '3.9' '3.1' '2.9' '5.2' '3.3'
 '4.7' '5.1' '3.5' '2.5' '1.9' '3' '2.7' '2.2' '2' '9.5' '2.1' '3.6' '2.8'
 '1.8' '3.8' '0' '1.6' '9.6' '2.4' '1.7' '1.1' '0.3' '1.5' '0.7' '1.2'
 '2.3' '0.5' '1.3' '0.2' '0.6' '1.4' '0.9' '1' '9.7']
```

In [9]:

```
# просмотрены уникальные значения для столбца рейтинга, чтобы увидеть значение tbd из задан
# заменяем формат столбца user_score на числовой, а столбец year_of_release на дату
games['user_score']=pd.to_numeric(games['user_score'], errors='coerce')
```

In [10]:

```
# заменим пропущенные значения в столбце rating - на пустую строку
games['rating'] = games['rating'].fillna(value='')
# заменяем столбец year_of_release на дату
games['year_of_release'] = games['year_of_release'].fillna(2045)
```

In [11]:

```
#2045 год - год действий в фильме "Первому игроку приготовиться"
games['year_of_release'] = games['year_of_release'].astype('int')
games['year_of_release'] = games['year_of_release'].astype('str')
games['sample_date'] = '-01-01'
games['year_of_release'] = games['year_of_release'] + games['sample_date']
games['year_of_release'] = pd.to_datetime(games['year_of_release'], format='%Y-%m-%d').dt.y
del games['sample_date']
games.head()
```

Out[11]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales
0	Wii Sports	Wii	2006	Sports	41.36	28.96	3.77	8.45
1	Super Mario Bros.	NES	1985	Platform	29.08	3.58	6.81	0.77
2	Mario Kart Wii	Wii	2008	Racing	15.68	12.76	3.79	3.29
3	Wii Sports Resort	Wii	2009	Sports	15.61	10.93	3.28	2.95
4	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	11.27	8.89	10.22	1.00

In [12]:

```
# удаляем дубликаты  
games.drop_duplicates().reset_index(drop=True)
```

Out[12]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales
0	Wii Sports	Wii	2006	Sports	41.36	28.96	3.77	
1	Super Mario Bros.	NES	1985	Platform	29.08	3.58	6.81	
2	Mario Kart Wii	Wii	2008	Racing	15.68	12.76	3.79	
3	Wii Sports Resort	Wii	2009	Sports	15.61	10.93	3.28	
4	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	11.27	8.89	10.22	
...
16710	Samurai Warriors: Sanada Maru	PS3	2016	Action	0.00	0.00	0.01	
16711	LMA Manager 2007	X360	2006	Sports	0.00	0.01	0.00	
16712	Haitaka no Psychedelica	PSV	2016	Adventure	0.00	0.00	0.01	
16713	Spirits & Spells	GBA	2003	Platform	0.01	0.00	0.00	
16714	Winning Post 8 2016	PSV	2016	Simulation	0.00	0.00	0.01	

16715 rows × 11 columns

In [13]:

```
# удаление строк с пропущенными значениями не было выполнено, т.к. пропущены значения только в одном столбце  
# подсчет суммарных продаж во всех регионах, формирование отдельного столбца  
games['sum_sales']=games['na_sales']+games['eu_sales']+games['jp_sales']+games['other_sales']
```

In [14]:

```
# Вывод на экран преобразованного дата сета  
display(games)
```

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_s
0	Wii Sports	Wii	2006	Sports	41.36	28.96	3.77	
1	Super Mario Bros.	NES	1985	Platform	29.08	3.58	6.81	
2	Mario Kart Wii	Wii	2008	Racing	15.68	12.76	3.79	
3	Wii Sports Resort	Wii	2009	Sports	15.61	10.93	3.28	
4	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	11.27	8.89	10.22	
...
16710	Samurai Warriors: Sanada Maru	PS3	2016	Action	0.00	0.00	0.01	
16711	LMA Manager 2007	X360	2006	Sports	0.00	0.01	0.00	
16712	Haitaka no Psychedelica	PSV	2016	Adventure	0.00	0.00	0.01	
16713	Spirits & Spells	GBA	2003	Platform	0.01	0.00	0.00	
16714	Winning Post 8 2016	PSV	2016	Simulation	0.00	0.00	0.01	

16715 rows × 12 columns

2. Исследовательский анализ данных

Основные задачи этапа:

1. Определить, сколько игр выпускалось в разные годы.
2. Определить, как менялись продажи по платформам. Определить платформы с наибольшими суммарными продажами и построить распределение по годам. За какой характерный срок появляются новые и исчезают старые платформы?
3. Рассмотреть данные за соответствующий актуальный период. Актуальный период определить самостоятельно в результате исследования предыдущих вопросов. Основной фактор — эти данные помогут построить прогноз на 2017 год.
4. Определить, какие платформы лидируют по продажам, растут или падают. Выберите несколько потенциально прибыльных платформ.
5. Построить график «ящик с усами» по глобальным продажам игр в разбивке по платформам. Опишите результат.
6. Определить, как влияют на продажи внутри одной популярной платформы отзывы пользователей и критиков. Построить диаграмму рассеяния и посчитайте корреляцию между отзывами и продажами. Сформулировать выводы. Соотнесите выводы с продажами игр на других платформах.

7. Посмотреть на общее распределение игр по жанрам. Что можно сказать о самых прибыльных жанрах? Выделяются ли жанры с высокими и низкими продажами?

In [15]:

```
#Подсчет количества выпускаемых игр в разные годы
games.groupby('year_of_release')[['name']].count()
```

Out[15]:

```
year_of_release
1980      9
1981     46
1982     36
1983     17
1984     14
1985     14
1986     21
1987     16
1988     15
1989     17
1990     16
1991     41
1992     43
1993     60
1994    121
1995    219
1996    263
1997    289
1998    379
1999    338
2000    350
2001    482
2002    829
2003    775
2004    762
2005   939
2006  1006
2007 1197
2008 1427
2009 1426
2010 1255
2011 1136
2012   653
2013   544
2014   581
2015   606
2016   502
2045   269
Name: name, dtype: int64
```

In [16]:

```
(games.groupby('year_of_release').name.count().sort_values(ascending=False)
    .plot(grid=True, kind='bar', figsize=(10, 5), color="orange")
)
plt.title('Количество игр'); # присвоение заголовка гистограмме
plt.ylabel("Количество");
plt.xlabel("Года \n\n Рисунок 2.1 Количество выпускаемых игр в разные годы");
```

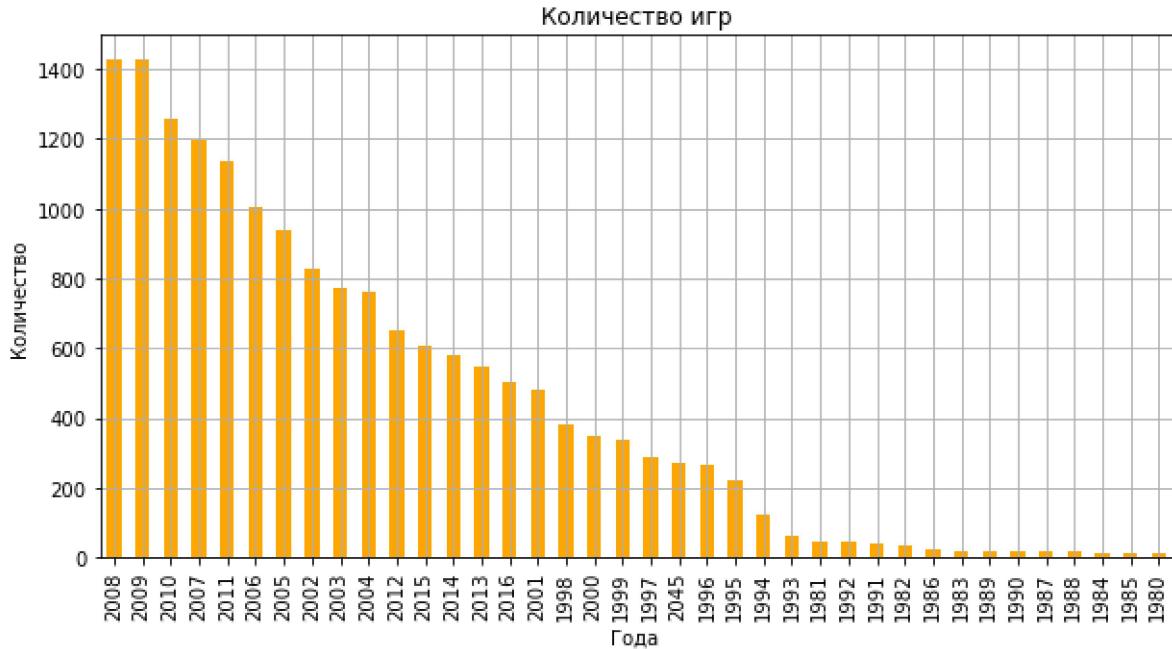


Рисунок 2.1 Количество выпускаемых игр в разные годы

Вывод:

Количество выпускаемых игр с 1980(9 игр) по 2008(1427 игр) увеличивалось, с 2009(1426 игр) - пошло на спад(в 2016 - 502 игры).

Определим платформы с наибольшими суммарными продажами и построим графики распределения суммарных продаж для каждой из этих платформ по годам.

In [17]:

```
# сортировка столбца платформ по количеству суммарных продаж
games.groupby(by='platform').agg({'sum_sales':'sum'}).sort_values(by='sum_sales', ascending=False)
```

Out[17]:

platform	sum_sales
PS2	1255.77
X360	971.42
PS3	939.65
Wii	907.51
DS	806.12
PS	730.86
GBA	317.85
PS4	314.14
PSP	294.05

In [18]:

```
sns.countplot(games[games['platform'].isin(games['platform'].value_counts().head().index)])
plt.title('Популярные платформы')
plt.xlabel('Наименование платформ')
Рис. 2.2 Популярные платформы исходя из объема продаж
```

Out[18]:

Text(0.5, 0, 'Наименование платформ \n\n Рис. 2.2 Популярные платформы исходя из объема продаж')

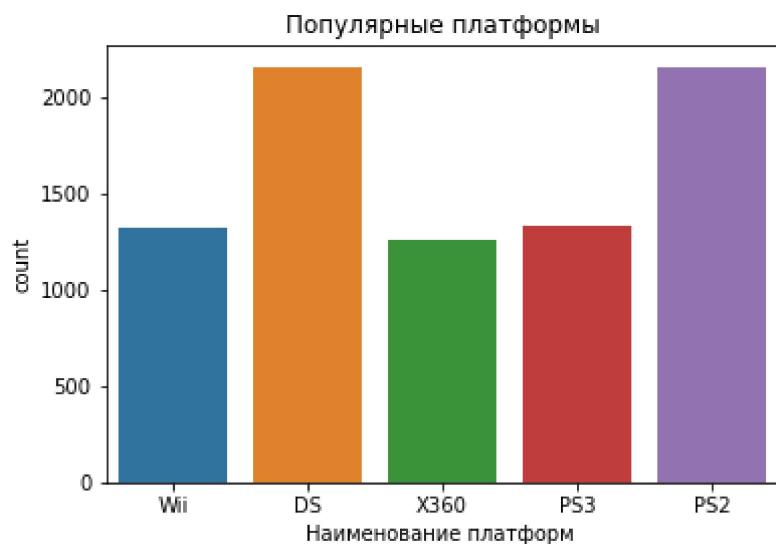


Рис. 2.2 Популярные платформы исходя из объема продаж

Вывод:

Наиболее популярны платформы: PS2, X360, PS3, Wii, DS.

In [19]:

```
# создание гистограмм суммарных продаж по годам для 5 самых популярных платформ.
(games
    .query('platform=="PS2"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('PS2'); # присвоение заголовка гистограмме
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.3 Суммарные продажи по годам для платформы PS2");

(games
    .query('platform=="X360"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('X360')
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.4 Суммарные продажи по годам для платформы X360");
(games
    .query('platform=="PS3"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('PS3')
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.5 Суммарные продажи по годам для платформы PS3");
(games
    .query('platform=="DS"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('DS')
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.6 Суммарные продажи по годам для платформы DS");
(games
    .query('platform=="Wii"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('Wii')
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.7 Суммарные продажи по годам для платформы Wii");
```

PS2



In [20]:

```
# Определим среднее время жизни самых популярных 5 платформ
#1
ps2=games.query('platform=="PS2"and year_of_release!=2045')
ps2_min=ps2['year_of_release'].min()
ps2_max=ps2['year_of_release'].max()
ps2_diff=ps2_max-ps2_min
#2
x360=games.query('platform=="X360"and year_of_release!=2045')
x360_min=x360['year_of_release'].min()
x360_max=x360['year_of_release'].max()
x360_diff=x360_max-x360_min
#3
ps3=games.query('platform=="PS3"and year_of_release!=2045')
ps3_min=ps3['year_of_release'].min()
ps3_max=ps3['year_of_release'].max()
ps3_diff=ps3_max-ps3_min
#4
ds=games.query('platform=="DS"and year_of_release!=2045')
ds_min=ds['year_of_release'].min()
ds_max=ds['year_of_release'].max()
ds_diff=ds_max-ds_min
#5
wii=games.query('platform=="Wii"and year_of_release!=2045')
wii_min=wii['year_of_release'].min()
wii_max=wii['year_of_release'].max()
wii_diff=wii_max-wii_min
mean=(ps2_diff+x360_diff+ps3_diff+ds_diff+wii_diff)/5
print('Среднее время жизни платформы: ',mean, 'лет.')
```

Среднее время жизни платформы: 14.0 лет.

Вывод:

Пятерка самых популярных платформ набрала свою популярность с начала 2000-х. Период от первых продаж до исчезнования продаж в среднем занимает 14 лет.

In [21]:

```
# посмотрим самые популярные платформы с 1980 до 2000 года
(games.query('year_of_release<2000').pivot_table(index='platform', values='sum_sales', aggfunc='sum')
plt.title('Популярные платформы')
plt.ylabel("Суммарные продажи");
plt.xlabel("Наименование платформ")
```



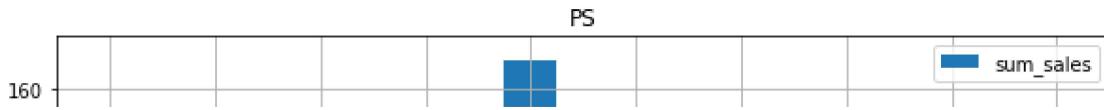
Рисунок 2.8 Самые популярные платформы с 1980 до 2000 года

Вывод:

Самые популярные платформы в период с 1980 до 2000: PS, NES, GB, SNES, N64.

In [22]:

```
# Выведем графики по самым популярным платформам в период с 1980 до 2000
(games
    .query('platform=="PS"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('PS')
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.9 Суммарные продажи по годам для платформы PS");
(games
    .query('platform=="NES"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('NES')
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.10 Суммарные продажи по годам для платформы NES");
(games
    .query('platform=="GB"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('GB')
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.11 Суммарные продажи по годам для платформы GB");
(games
    .query('platform=="SNES" and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('SNES')
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.12 Суммарные продажи по годам для платформы SNES");
(games
    .query('platform=="N64" and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('N64')
plt.ylabel("Суммарные продажи");
plt.xlabel("Года \n\n Рисунок 2.13 Суммарные продажи по годам для платформы N64");
```



In [23]:

```
#1
ps=games.query('platform=="PS"and year_of_release!=2045')
ps_min=ps['year_of_release'].min()
ps_max=ps['year_of_release'].max()
ps_diff=ps_max-ps_min
#2
nes=games.query('platform=="NES"and year_of_release!=2045')
nes_min=nes['year_of_release'].min()
nes_max=nes['year_of_release'].max()
nes_diff=nes_max-nes_min
#3
gb=games.query('platform=="GB"and year_of_release!=2045')
gb_min=gb['year_of_release'].min()
gb_max=gb['year_of_release'].max()
gb_diff=gb_max-gb_min
#4
snes=games.query('platform=="SNES"and year_of_release!=2045')
snes_min=snes['year_of_release'].min()
snes_max=snes['year_of_release'].max()
snes_diff=snes_max-snes_min
#5
n64=games.query('platform=="N64"and year_of_release!=2045')
n64_min=n64['year_of_release'].min()
n64_max=n64['year_of_release'].max()
n64_diff=wii_max-wii_min
mean=(ps_diff+nes_diff+gb_diff+snes_diff+n64_diff)/5
print('Среднее время жизни платформы: ',mean, ' лет.')
```

Среднее время жизни платформы: 10.4 лет.

Вывод: период для появления платформы и выхода из нее с 1980 до 2000 меньше, чем для платформ с 2000 до 2016, и составляет 10,4 года. Платформа PS трансформировалась в PS2 и PS3. Платформа NES уступила свое место более совершенным SNES с более совершенной приставкой Sega Mega Drive, также как до этого N64 уступила свое место NES. SNES 95 году обещали выпустить новую приставку, несколько раз переносили презентацию, в итоге для новой приставки выпустили 22 игры, все японского производства и платформа перестала существовать. GB были линейкой портативных устройств от компании Nintendo (N64, NES, SNES), с уходом Nintendo с рынка исчезли и GB.

Определить, какие платформы лидируют по продажам, растут или падают. Выберите несколько потенциально прибыльных платформ

Определение распределения данных

In [24]:

```
(games.groupby('year_of_release')['sum_sales'].sum().plot(grid=True, kind='bar', figsize=(10, 6))
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.14 Суммарные продажи по годам выпуска платформ");
```

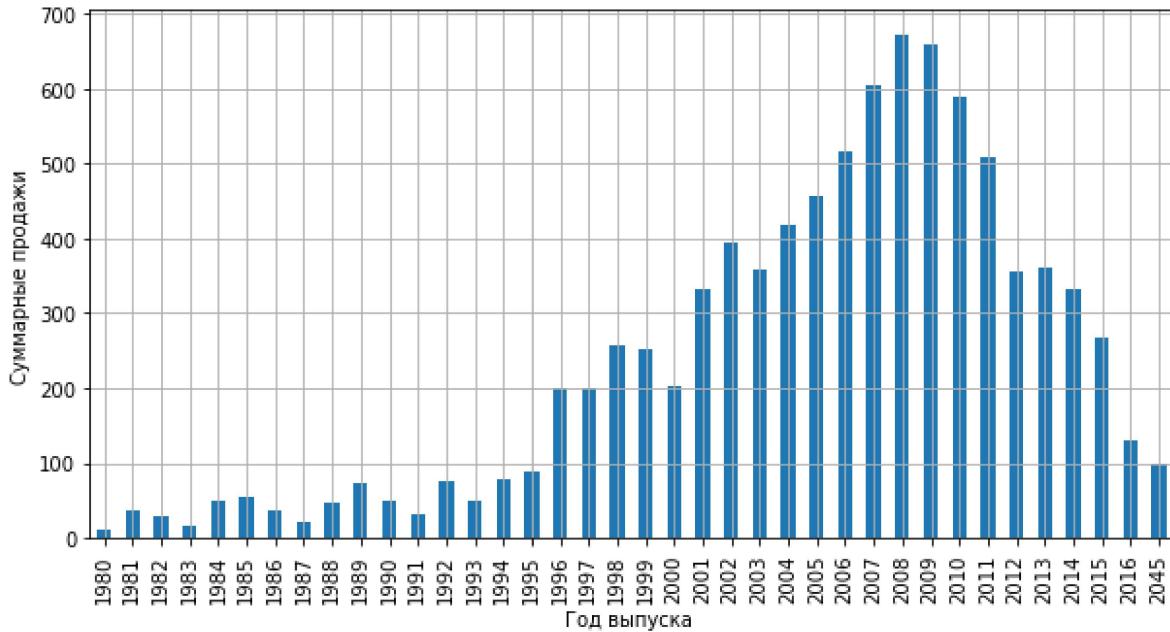


Рисунок 2.14 Суммарные продажи по годам выпуска платформ

Вывод:

Количество продаж резко падает в 2016, возможно, это связано с тем, что данные за 2016 не заполнены до конца. Нормальным выглядит распределение продаж с 2003 по 2014, нужно избавиться от выбросов, чтобы убедиться в этом. А также выберем платформы с ненулевыми продажами за 2016 год, чтобы работать с ними в дальнейшем.

Построим график «ящик с усами» по глобальным продажам игр в разбивке по платформам.

In [25]:

```
# Избавимся от выбросов.  
(games.boxplot(column=['sum_sales']))  
plt.title('График выбросов')  
plt.xlabel("Суммарные продажи \n\n Рисунок 2.15 Количество суммарных продаж");
```

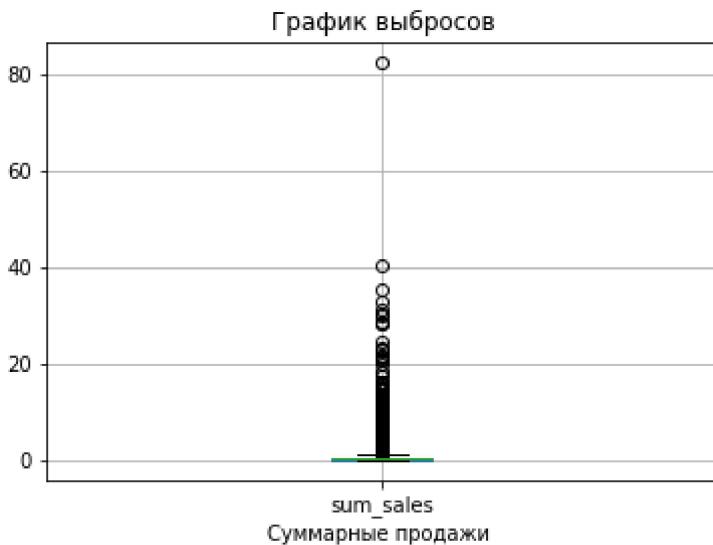


Рисунок 2.15 Количество суммарных продаж

Количество суммарных продаж более 80 является выбросом.

In [26]:

```
games_2016=games[(games.year_of_release==2016)&(games.sum_sales>0)] # Выберем платформы с н  
games_2016['platform'].unique()
```

Out[26]:

```
array(['PS4', '3DS', 'XOne', 'WiiU', 'PS3', 'PC', 'X360', 'PSV', 'Wii'],  
      dtype=object)
```

В 2016 существуют платформы: PS4, 3DS, XOne, WiiU, PS3, PC, X360, PSV, Wii.

In [27]:

```
# отбрасываем выбросы
games=games[(games.sum_sales < 80)]
# по гистограмме выбираем столбцы с нормальным распределением
games=games[(games.year_of_release > 2002) & (games.year_of_release < 2016)]
```

In [28]:

```
games.describe()
```

Out[28]:

	year_of_release	na_sales	eu_sales	jp_sales	other_sales	critic_score	
count	12306.000000	12306.000000	12306.000000	12306.000000	12306.000000	6558.000000	€
mean	2008.605965	0.240423	0.142046	0.054311	0.052296	68.550168	
std	3.275813	0.638966	0.468200	0.248666	0.196970	13.829596	
min	2003.000000	0.000000	0.000000	0.000000	0.000000	13.000000	
25%	2006.000000	0.000000	0.000000	0.000000	0.000000	60.000000	
50%	2009.000000	0.080000	0.020000	0.000000	0.010000	70.000000	
75%	2011.000000	0.230000	0.100000	0.030000	0.040000	79.000000	
max	2015.000000	15.680000	12.760000	6.500000	10.570000	98.000000	

Дальше работаем с данными только за период с 2003 по 2015.

Выход:

Нормально распределены данные за период с 2003 по 2015 год, его используем в дальнейшей работе. Выборку платформ из 2016 сделали.

Предсказываем популярность платформ.

In [29]:

```
games['platform'].unique()
```

Out[29]:

```
array(['Wii', 'DS', 'X360', 'PS3', 'PS2', 'PS4', '3DS', 'GBA', 'XB', 'PC',
       'PSP', 'XOne', 'WiiU', 'GC', 'PSV', 'PS', 'DC'], dtype=object)
```

В выбраном нами периоде с нормальным распределением существуют платформы: Wii, DS, X360, PS3, PS2, PS4, 3DS, GBA, XB, PC, PSP, XOne, WiiU, GC, PSV, PS, DC. В 2016 существуют платформы: PS4, 3DS, XOne, WiiU, PS3, PC, X360, PSV, Wii. (вывод из предыдущего блока)

Выход:

Из этого следует, что ни одну платформу при анализе мы не пропускаем, при отсечении данных за 2016 год.

In [30]:

```
# Исключим те платформы, которые уже были изучены: PS, PS2, PS3, X360, DS, Wii.  
prognоз=games[(games.platform != 'PS') & (games.platform != 'PS2') & (games.platform != 'PS3') & (g  
прогноз.head()
```

Out[30]:

		name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales
31		Call of Duty: Black Ops 3	PS4	2015	Shooter	6.03	5.86	0.36	2.38
33		Pokemon X/Pokemon Y	3DS	2013	Role- Playing	5.28	4.19	4.35	0.78
40		Mario Kart 7	3DS	2011	Racing	5.03	4.02	2.69	0.91
42		Grand Theft Auto V	PS4	2014	Action	3.96	6.31	0.38	1.97
47		Pokemon Omega Ruby/Pokemon Alpha Sapphire	3DS	2014	Role- Playing	4.35	3.49	3.10	0.74

Построим графики для платформ PS4, 3DS, XOne, WiiU, PC, PSV, XB. Учтем, что в 2016 году существует платформа X360, но в выбранных нами данных ее нет.

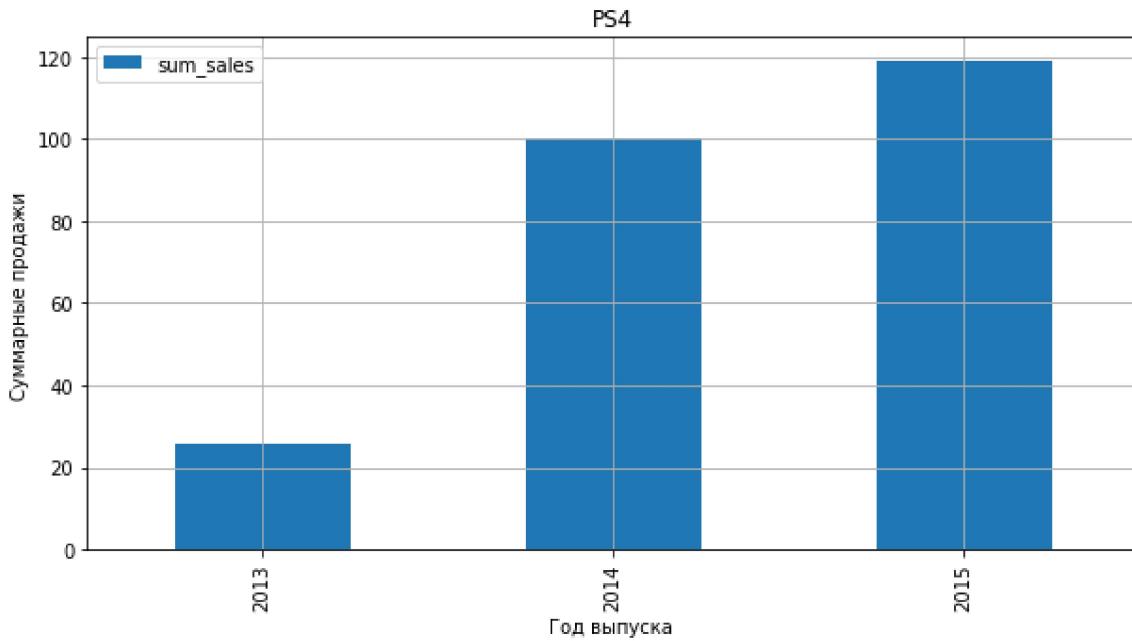
In [31]:

```
(prognoz
    .query('platform=="PS4"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('PS4')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.16 Суммарные продажи для платформы PS4 ");
(prognoz
    .query('platform=="3DS"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('3DS')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.17 Суммарные продажи для платформы 3DS ");
(prognoz
    .query('platform=="XOne" and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('XOne')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.18 Суммарные продажи для платформы XOne");
(prognoz
    .query('platform=="WiiU"and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('WiiU')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.19 Суммарные продажи для платформы WiiU");
(prognoz
    .query('platform=="PC" and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('PC')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.20 Суммарные продажи для платформы PC ");
(prognoz
    .query('platform=="PSV" and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('PSV')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.21 Суммарные продажи для платформы PSV");
(prognoz
    .query('platform=="XB" and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('XB')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.22 Суммарные продажи для платформы XB");
(prognoz
    .query('platform=="GC" and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
```

```

    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('GC')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.23 Суммарные продажи для платформы GC");
(prognoz
    .query('platform=="GBA" and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('GBA')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.24 Суммарные продажи для платформы GBA");
(prognoz
    .query('platform=="DC" and year_of_release!=2045')
    .pivot_table(index='year_of_release', values='sum_sales', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('DC')
plt.ylabel("Суммарные продажи");
plt.xlabel("Год выпуска \n\n Рисунок 2.25 Суммарные продажи для платформы DC");

```



Выход:

Потенциально прибыльны платформы PS4, X360 и XOne, вот в них и будем инвестировать;)

Построим график «ящик с усами» по глобальным продажам игр в разбивке по платформам.

In [32]:

```
plt.figure(figsize=(10,10))
sns.boxplot(x = 'platform', y = 'sum_sales', data = prognoz, palette='rainbow')
plt.ylabel("Суммарные продажи");
plt.xlabel("Наименование платформ \n\n Рисунок 2.26 Суммарные продажи для платформ");
```

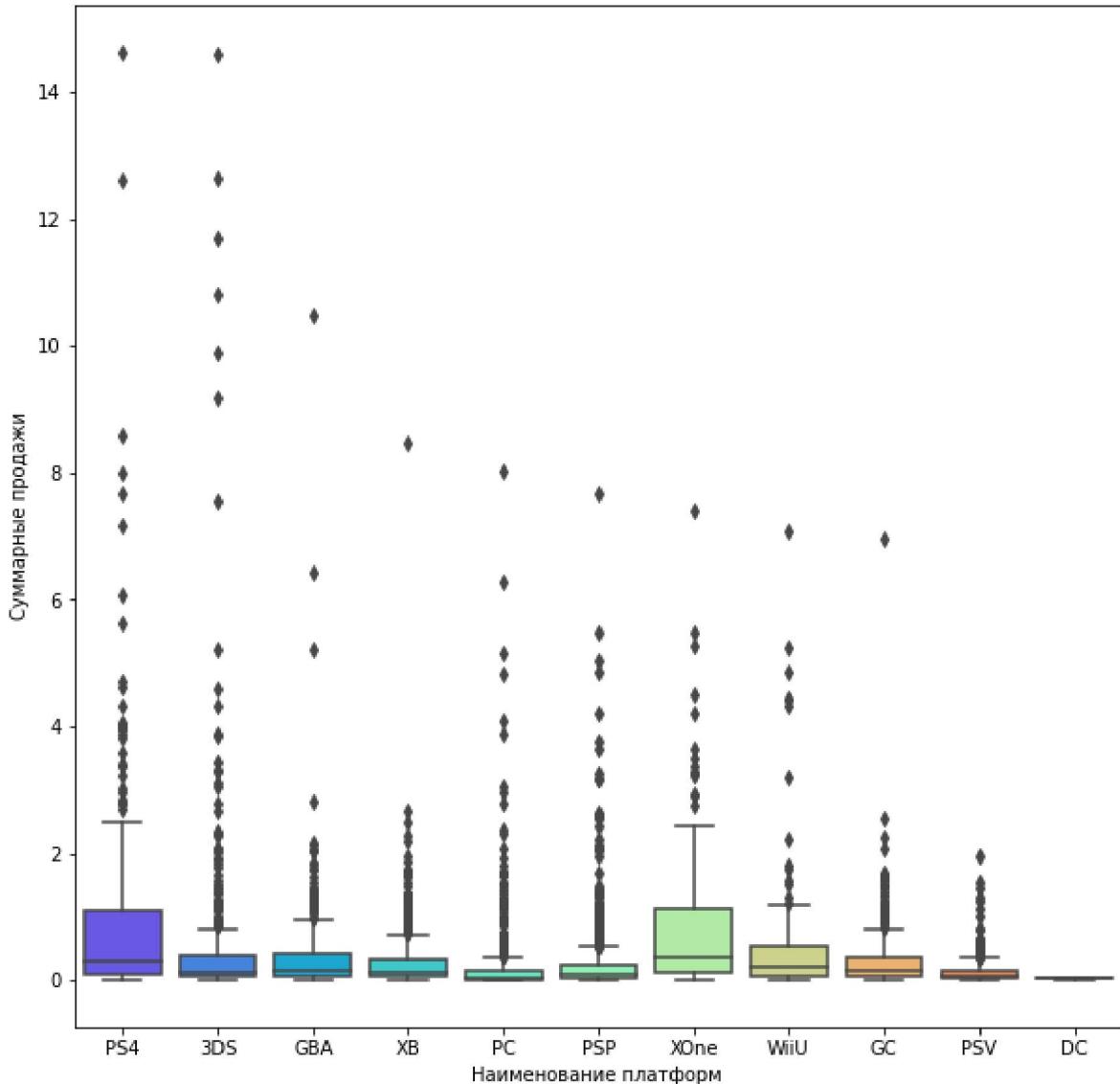


Рисунок 2.26 Суммарные продажи для платформ

Вывод:

Продажи XOne и PS4 сильно выделяются на фоне остальных, что неудивительно, потому что это лидеры рынка.

Посчитаем корреляцию между отзывами и продажами.

In [33]:

```
corr = games[['critic_score', 'user_score', 'na_sales', 'eu_sales', 'jp_sales']].corr()
corr.style.background_gradient(cmap='coolwarm')
```

Out[33]:

	critic_score	user_score	na_sales	eu_sales	jp_sales
critic_score	1.000000	0.577140	0.266193	0.249969	0.133017
user_score	0.577140	1.000000	0.078695	0.051019	0.110179
na_sales	0.266193	0.078695	1.000000	0.793403	0.417326
eu_sales	0.249969	0.051019	0.793403	1.000000	0.467837
jp_sales	0.133017	0.110179	0.417326	0.467837	1.000000

Вывод:

Линейная зависимость между оценками критиков и пользователей минимальна. Посмотрим что на графиках.

Построим диаграмму рассеяния зависимости продаж от оценок

In [34]:

```
plt.figure(figsize=(30,10))
sns.stripplot(x = 'user_score', y = 'sum_sales', data = prognoz, jitter = True)
plt.ylabel("Оценка пользователей");
plt.xlabel("Суммарные продажи \n\n Рис. 2.27 Линейная зависимость продаж от оценки пользователей")
plt.figure(figsize=(30,10))
sns.stripplot(x = 'critic_score', y = 'sum_sales', data = prognoz, jitter = True)
plt.ylabel("Оценка критиков");
plt.xlabel("Суммарные продажи \n\n Рис. 2.28 Линейная зависимость продаж от оценки критиков")
```

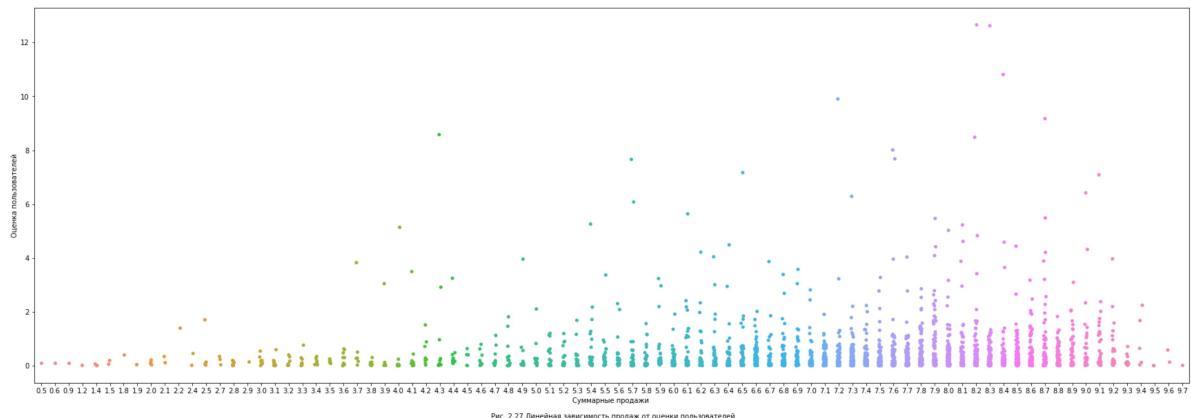


Рис. 2.27 Линейная зависимость продаж от оценки пользователей

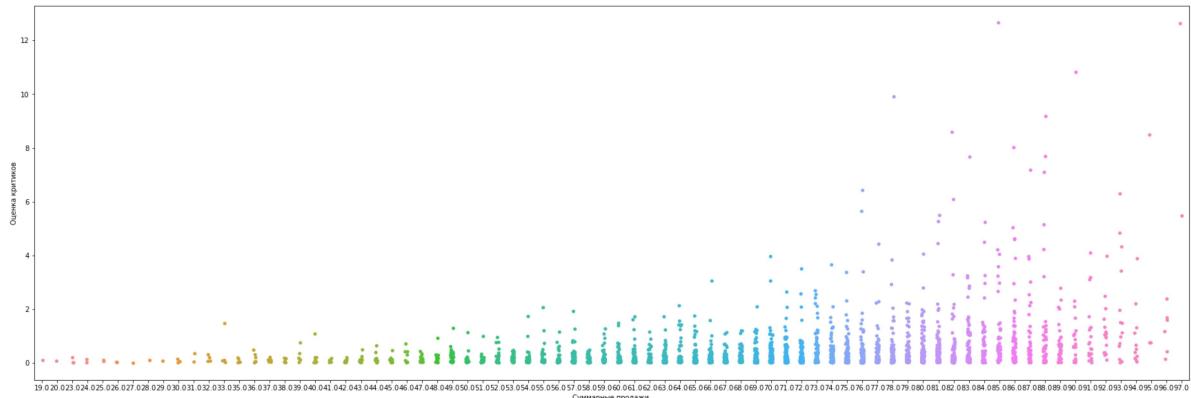


Рис. 2.28 Линейная зависимость продаж от оценки критиков

Вывод:

Чем выше оценка, тем больше продаж, и не важно оценил игру пользователь или критик. А также самые дорогие продажи приходятся на самые высокие оценки критиков.

In [35]:

```
sns.stripplot(x = 'platform', y = 'sum_sales', data = prognoz, jitter = True)
plt.title('Распределение продаж для различных платформ')
plt.ylabel("Суммарные продажи");
plt.xlabel('Наименование платформы \n\n Рис. 2.29 Распределение продаж для различных платформ')
```

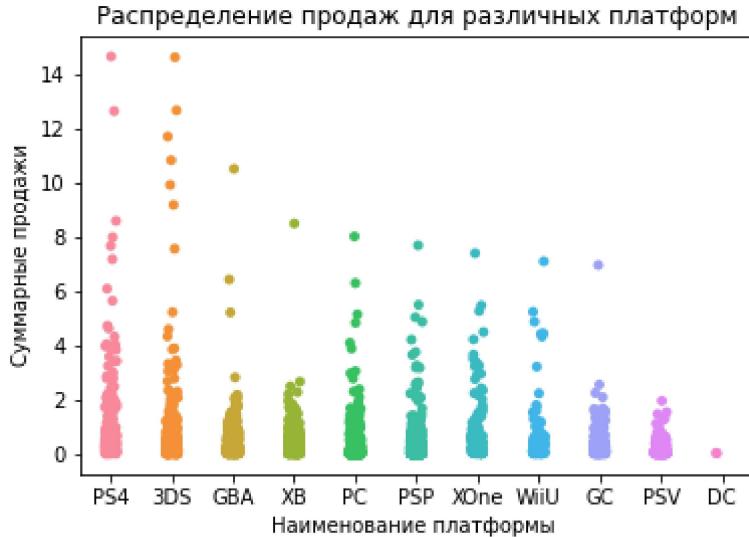


Рис. 2.29 Распределение продаж для различных платформ

Вывод:

В основном общие продажи игр распределены в диапазоне до 5. Это означает, что зарабатывают на играх для платформ в основном создатели платформ, а не разработчики игр для них.

Посмотрим на общее распределение игр по жанрам.

In [36]:

```
print(games.groupby(by='genre').agg({'sum_sales':'sum'}).sort_values(by='sum_sales', ascending=False))
print(games.groupby(by='genre').agg({'sum_sales':'sum'}).sort_values(by='sum_sales', ascending=True))

sum_sales
genre
Action           1316.04
Sports            866.88
Shooter           794.37
Misc              668.18
Role-Playing      616.84

sum_sales
genre
Simulation       299.79
Fighting          244.15
Adventure         156.73
Puzzle             121.09
Strategy           92.36
```

Вывод:

Наиболее популярны такие жанры игр как: экшн, спорт, шутеры, аркады и ролевые игры, те игры, в которых получаешь эмоцию от происходящего. Наименее популярны игры, в которых нужно думать или совершать однотипные действия: симуляции, бои, приключения, пазлы и стратегии. Связано это еще и с ценой на игры, производство игр в жанрах Action или Shooter обходится сильно дороже, чем производство Puzzle-игр.

3. Составить портрет пользователя каждого региона

Основные задачи этапа:

1. Составить портрет пользователя для каждого региона. Определить для пользователя каждого региона (NA, EU, JP):
2. Самые популярные платформы (топ-5). Описать различия в долях продаж.
3. Самые популярные жанры (топ-5). Объяснить разницу.
4. Самые популярные игры (топ-5). Рассказать про одну игру.
5. Определить влияние рейтинга ESRB на продажи в отдельном регионе.

Построим графики самых популярных платформ для пользователя каждого региона.

In [37]:

```
games.groupby(by='platform').agg({'eu_sales':'sum'}).sort_values(by='eu_sales', ascending=False)
plt.title('Самые популярные платформы')
plt.ylabel("Наименование платформы");
plt.xlabel('Продажи в Европе (миллионы долларов) \n\n Рис. 3.1 Самые популярные платформы в Европе')
games.groupby(by='platform').agg({'na_sales':'sum'}).sort_values(by='na_sales', ascending=False)
plt.title('Самые популярные платформы')
plt.ylabel("Наименование платформы");
plt.xlabel('Продажи в Северной Америке (миллионы долларов) \n\n Рис. 3.2 Самые популярные платформы в Северной Америке')
games.groupby(by='platform').agg({'jp_sales':'sum'}).sort_values(by='jp_sales', ascending=False)
plt.title('Самые популярные платформы')
plt.ylabel("Наименование платформы");
plt.xlabel('Продажи в Японии (миллионы долларов) \n\n Рис. 3.3 Самые популярные платформы в Японии')
```

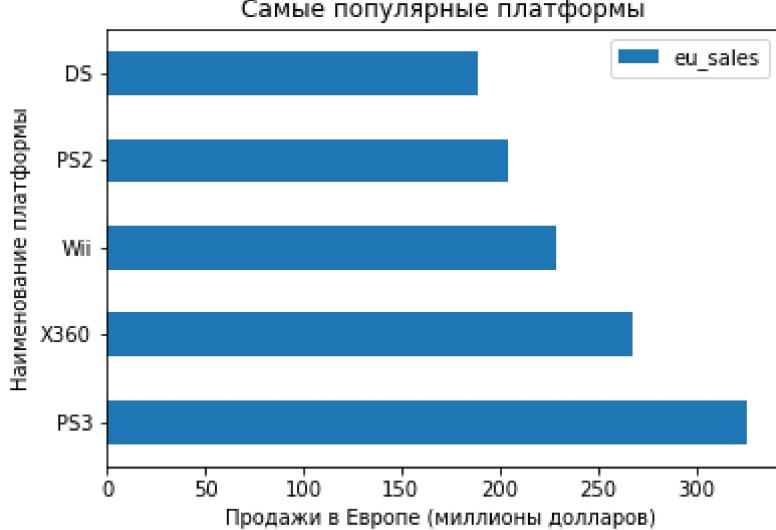


Рис. 3.1 Самые популярные платформы в Европе

Построим графики самых популярных жанров игр для пользователя каждого региона.

In [38]:

```
games.groupby(by='genre').agg({'eu_sales':'sum'}).sort_values(by='eu_sales', ascending=False)
plt.title('Самые популярные жанры игры')
plt.ylabel("Жанры игры");
plt.xlabel('Продажи в Европе (миллионы долларов) \n\n Рис. 3.4 Самые популярные жанры игры')
games.groupby(by='genre').agg({'na_sales':'sum'}).sort_values(by='na_sales', ascending=False)
plt.title('Самые популярные жанры игры')
plt.ylabel("Жанры игры");
plt.xlabel('Продажи в Северной Америке (миллионы долларов) \n\n Рис. 3.5 Самые популярные жанры игры')
games.groupby(by='genre').agg({'jp_sales':'sum'}).sort_values(by='jp_sales', ascending=False)
plt.title('Самые популярные жанры игры')
plt.ylabel("Жанры игры");
plt.xlabel('Продажи в Японии (миллионы долларов) \n\n Рис. 3.6 Самые популярные жанры игры')
```

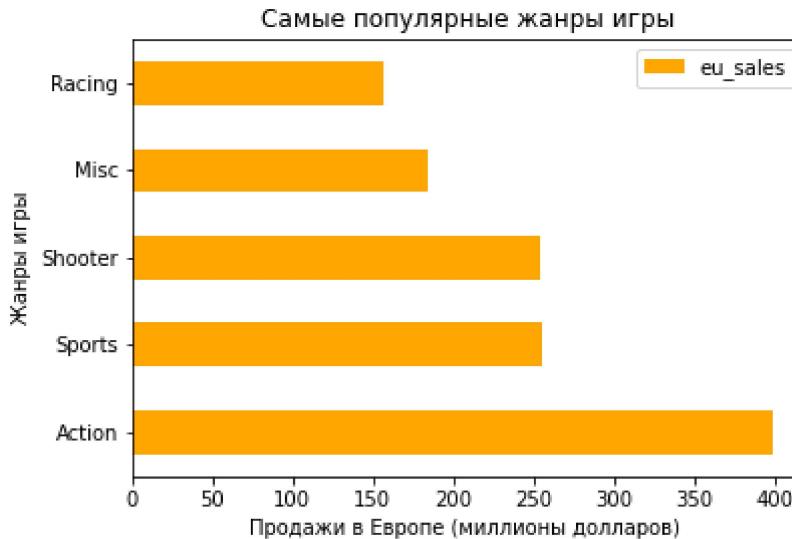


Рис. 3.4 Самые популярные жанры игры в Европе

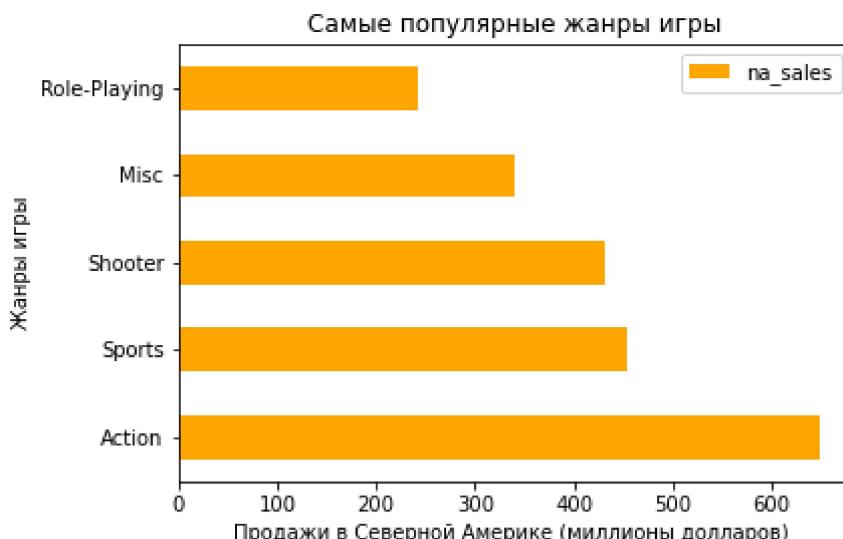
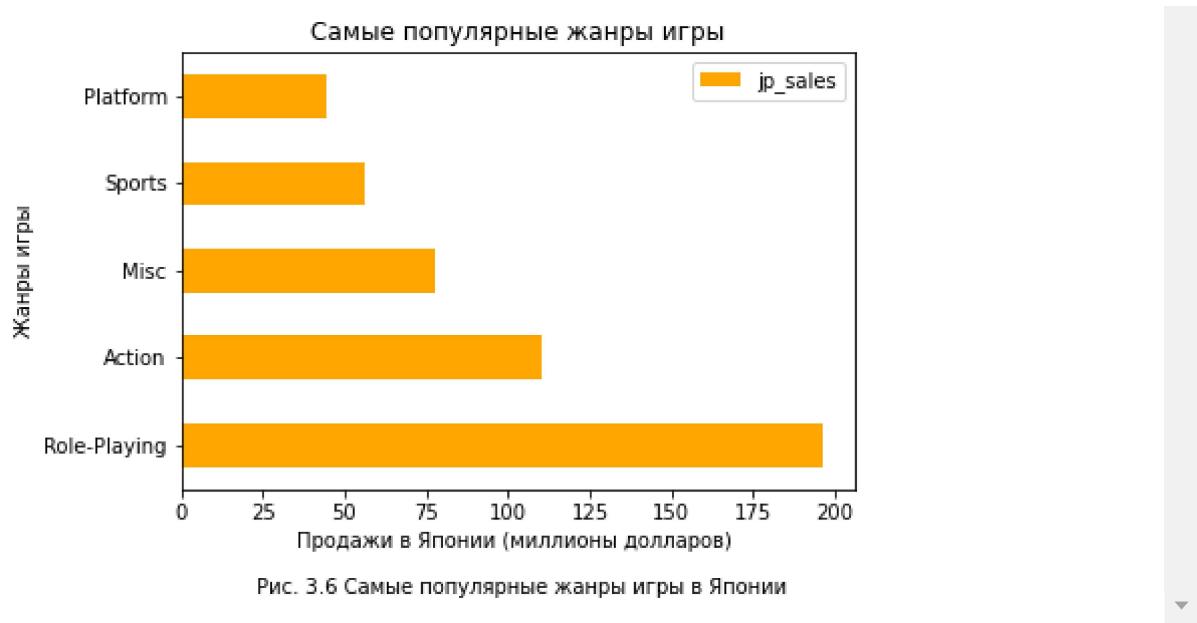


Рис. 3.5 Самые популярные жанры игры в Северной Америке



Вывод:

Жанры для игр в Северной Америке, Европе и Японии популярны одни и те же, а вот игровые платформы у японцев значительно отличаются. В разы различаются и продажи в этих регионах: больше всего продаж в Северной Америке, меньше всего в Японии. Предполагаю, что связано это с разным менталитетом этих регионов.

Построим графики самых популярных игр для пользователя каждого региона.

In [39]:

```
games.groupby(by='name').agg({'eu_sales':'sum'}).sort_values(by='eu_sales', ascending=False)
plt.title('Самые популярные игры')
plt.ylabel("Наименование игры");
plt.xlabel('Продажи в Европе (миллионы долларов) \n\n Рис. 3.7 Самые популярные игры в Европе')
games.groupby(by='name').agg({'na_sales':'sum'}).sort_values(by='na_sales', ascending=False)
plt.title('Самые популярные игры')
plt.ylabel("Наименование игры");
plt.xlabel('Продажи в Северной америке (миллионы долларов) \n\n Рис. 3.8 Самые популярные игры в Северной Америке')
games.groupby(by='name').agg({'jp_sales':'sum'}).sort_values(by='jp_sales', ascending=False)
plt.title('Самые популярные игры')
plt.ylabel("Наименование игры");
plt.xlabel('Продажи в Японии (миллионы долларов) \n\n Рис. 3.9 Самые популярные игры в Японии')
```



Рис. 3.7 Самые популярные игры в Европе

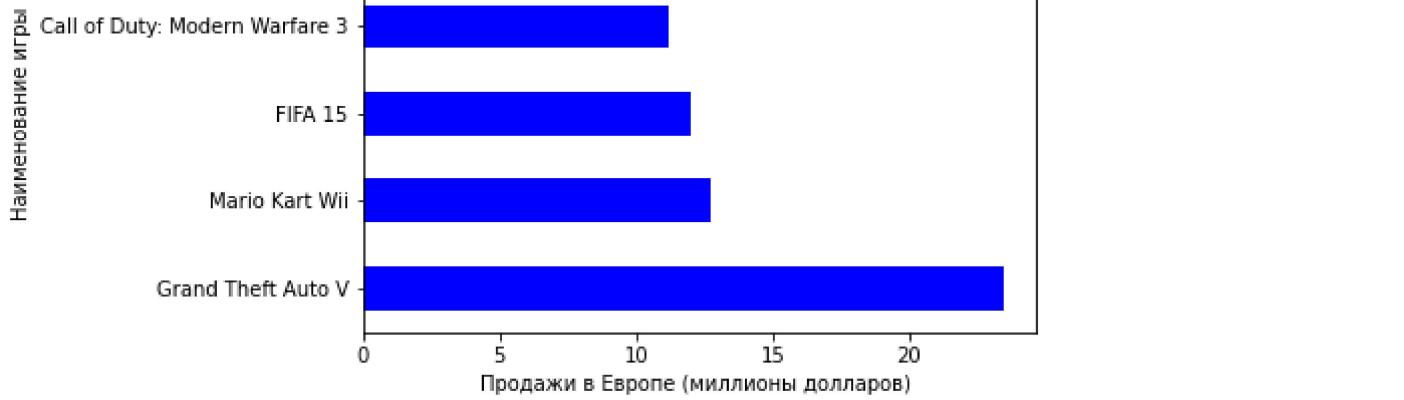


Рис. 3.8 Самые популярные игры в Северной Америке



Рис. 3.9 Самые популярные игры в Японии

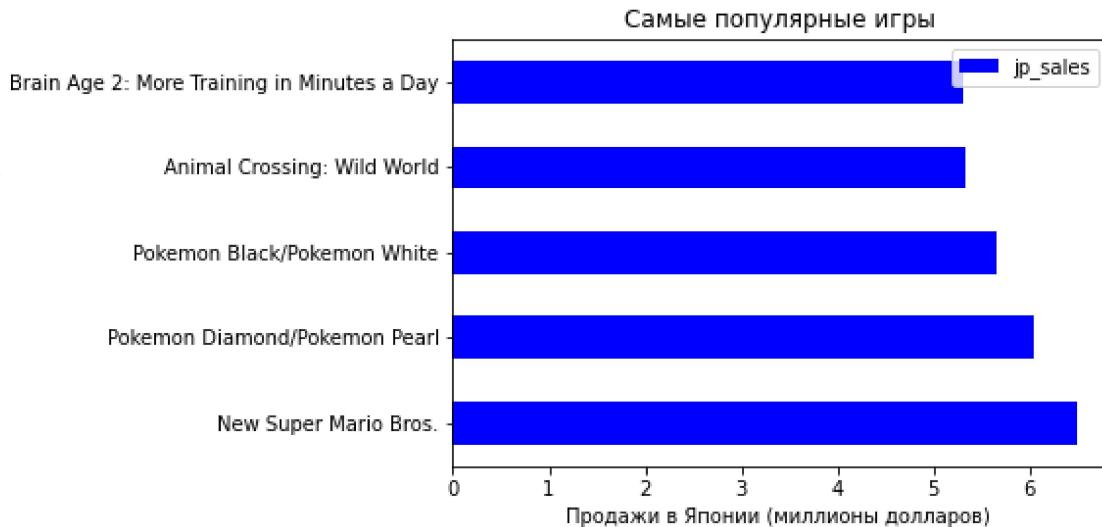


Рис. 3.9 Самые популярные игры в Японии

Вывод:

Самая популярная игра в Европе и Северной Америке Grand Theft Auto V, а в Японии New Super Mario Bros, о ней сейчас и поговорим подробнее.

New Super Mario Bros— видеоигра в жанре платформер, разработанная и выпущенная компанией Nintendo для портативной консоли Nintendo DS. Игра была выпущена в Северной Америке и Японии в мае 2006 года, а также в Австралии и Европе в июне того же года. Игра сайд-скроллер.

Сюжет игры схож с другими скроллинговыми играми о Марио. В New Super Mario Bros. главный персонаж, соответственно, проходит длительный путь и спасает принцессу Пич от Боузера. В течение игры Марио находит несколько полезных предметов — Супергриб, Огненный цветок, а также Суперзвезду, которые позволяют ему обретать уникальные способности и помогают в пути. Во время путешествия через 80 уровней-зон из восьми миров Марио должен победить Боузера-младшего и старшего Боузера, и спасти принцессу.

Рецензии на игру были, в основном, весьма положительными. Многие критики отметили усовершенствование главного персонажа, общую простоту игры и сходство с предыдущими играми. В Японии игра побила абсолютный рекорд по популярности в качестве дебютной игры для Nintendo DS. В целом, было продано 26 миллионов копий игры по всему миру, что сделало её самой продаваемой видеоигрой для Nintendo DS (англ.)рус.[2].

Продолжение игры New Super Mario Bros. Wii, которое предоставляет возможность играть сразу четырём пользователям, было выпущено в определённых регионах 12 ноября 2009 года, а 3 декабря — по всему миру.

Отзывы

Игра получила высокие оценки и многочисленные отзывы, в которых было отмечено, что New Super Mario Bros. стала одной из лучших игр для Nintendo DS. «GameZone» отметил её «огромный потенциал» и переосмысление игровой платформы жанра[25]. Том Бромвелл, представляющий сайт Eurogamer, заявил: «Играть в неё будет интересно и через тысячи дней. Я всё ещё люблю её»[21], полагая, что опытным игрокам потребуется совсем немного времени, чтобы пройти её[27]. Сайт GamesRadar отметил, что она «полностью утвердила себя в качестве сольной игры — простой, но захватывающей»[26].

Ряд обозревателей сделал несколько сравнений между New Super Mario Bros. и другими играми о Марио. И хотя некоторые обнаружили, что многие другие игры с Марио были лучше, большинство обозревателей по-прежнему остались довольны общим впечатлением от игры. Крэйг Харрис с сайта IGN пришёл в восторг, утверждая, что это его любимая игра для новой платформы, которая «обогнала» его предыдущую любимую игру — Super Mario World 2: Yoshi's Island[30]. Несмотря на то, что Мистер Марблес из журнала GamePro считал Super Mario World и Super Mario Bros. З лучшими двухмерными играми о Марио, он решил добавить к ним и New Super Mario Bros., как свою третью любимую игру о Марио[23]. Официальный журнал компании Official Nintendo Magazine поставил игре рейтинг в 96 % от и присудил «Золотую премию».

Награды

Игра получила множество наград и похвал, в том числе от ежемесячных журналов Game Informer[44] и Electronic Gaming Monthly[45], а также награды от IGN[46] и GameSpot[27]. В 2006 году игра была признана лучшей по мнению Spike Video Game Awards[47], лучшей игрой для Nintendo DS по мнению GameSpot[48], а также стала лучшим платформером по мнению X-Play[49] и Nintendo Power[50]. В том же году игра получила премию Golden Joystick Award[51] и была помещена в 100 величайших игр Nintendo всех времён в журнале ONM.

Построим графики влияние рейтинга ESRB (англ. Entertainment Software Rating Board) на продажи в отдельном регионе. Эта ассоциация определяет рейтинг компьютерных игр и присваивает им подходящую возрастную категорию.

In [40]:

```
games.groupby(by='rating').agg({'eu_sales':'sum'}).sort_values(by='eu_sales', ascending=False)
plt.title('Влияние рейтинга от организации ESRB на продажи')
plt.ylabel("Рейтинг");
plt.xlabel('Продажи в Европе (миллионы долларов) \n\n Рис. 1. Влияние рейтинга от организац
games.groupby(by='rating').agg({'na_sales':'sum'}).sort_values(by='na_sales', ascending=False)
plt.title('Влияние рейтинга от организации ESRB на продажи')
plt.ylabel("Рейтинг");
plt.xlabel('Продажи в Северной Америке (миллионы долларов) \n\n Рис. 1. Влияние рейтинга от
games.groupby(by='rating').agg({'jp_sales':'sum'}).sort_values(by='jp_sales', ascending=False)
plt.title('Влияние рейтинга на от организации ESRB продажи')
plt.ylabel("Рейтинг");
plt.xlabel('Продажи в Японии (миллионы долларов) \n\n Рис. 1. Влияние рейтинга от организац
```

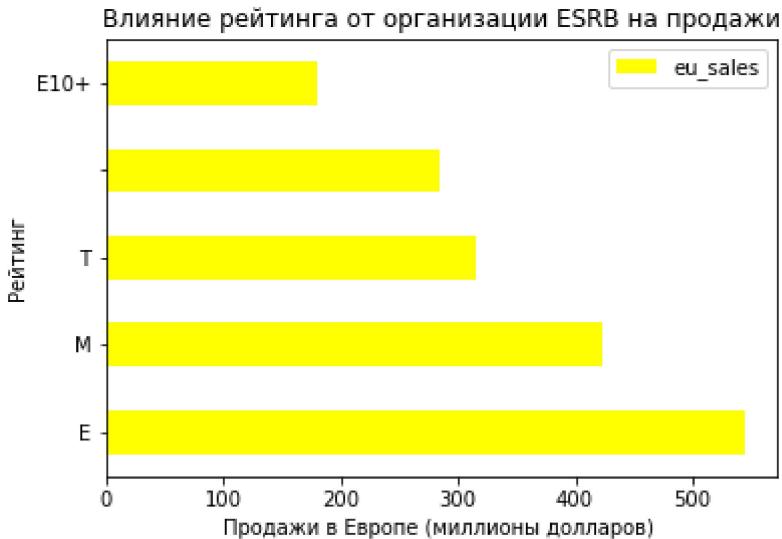


Рис. 1. Влияние рейтинга от организации ESRB на продажи в Европе

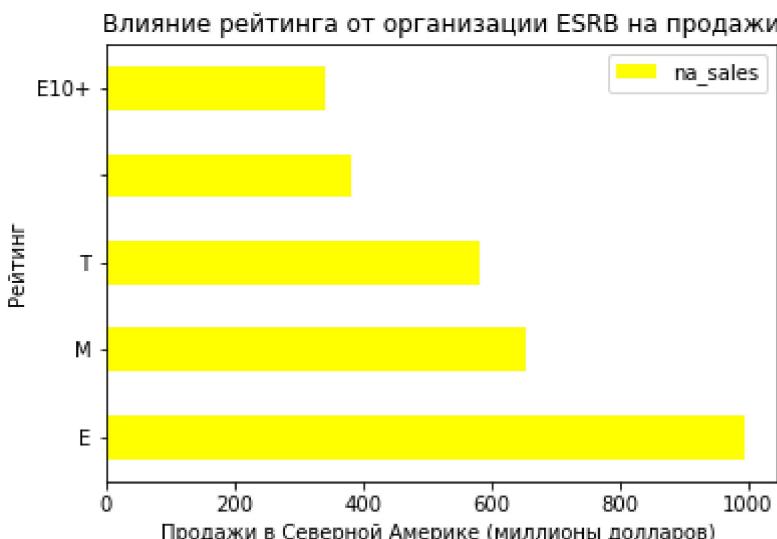


Рис. 1. Влияние рейтинга от организации ESRB на продажи в Северной Америке

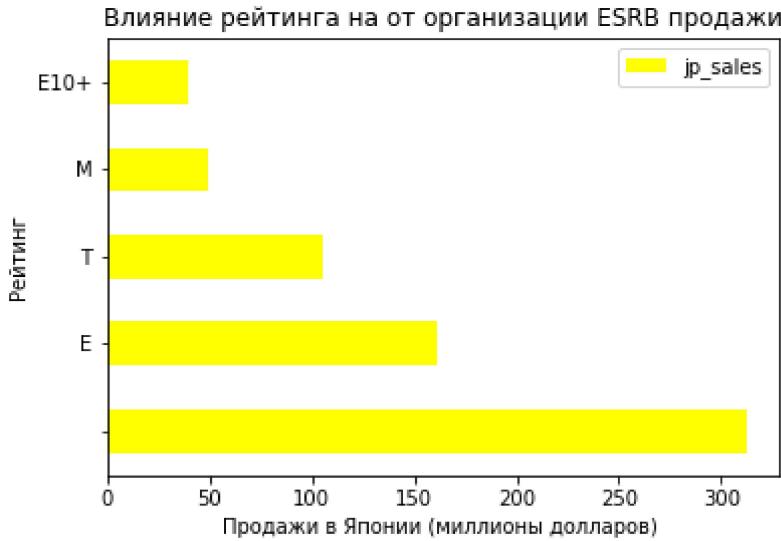


Рис. 1. Влияние рейтинга от организации ESRB на продажи в Японии

Выход:

Зависимость от рейтинга отслеживается во всех трех регионах. Наиболее популярны игры с рейтингом "для всех", "подросткам", "детям старше 10 лет" и "для взрослых" (в Европе и Северной Америке спрос на них выше, японцы больше предпочитают игры без определенного рейтинга). Ситуация в Японии связана с тем, что получить возрастной рейтинг на японском рынке сложно и дорого. Начинает это с того, что всё общение и вся документация — на японском языке. «Google Переводчиком» в данной ситуации пользоваться не стоит, лучше найти человека со знанием языка или студию локализации.

Есть 2 варианта для построения отношений:

- разовый платёж. Вы хотите единожды сделать рейтинг для одной игры и на одну (или несколько) платформу — стоило это на 2018 год примерно 2 тысячи евро.
- членство. У вас есть как вступительный взнос, так и взнос за каждую игру, но цена за оценку игры ниже, чем разовая оценка в первом варианте. То есть в перспективе, если вы планируете делать 5-7 релизов и на 3-4 платформы, вам стоит рассмотреть и членство в данном агентстве.

4. Исследование статистических показателей

Посмотрим как изменяется пользовательский рейтинг и рейтинг критиков в различных жанрах

In [41]:

```
games.groupby('genre')['critic_score'].sum().plot(grid=True, kind='barh', figsize=(10, 5));
plt.title('Влияние жанра игры на оценку критика')
plt.ylabel("Жанр Игры");
plt.xlabel('Оценка критиков \n\n Рис. 1. Влияние жанра игры на оценку критика');
```

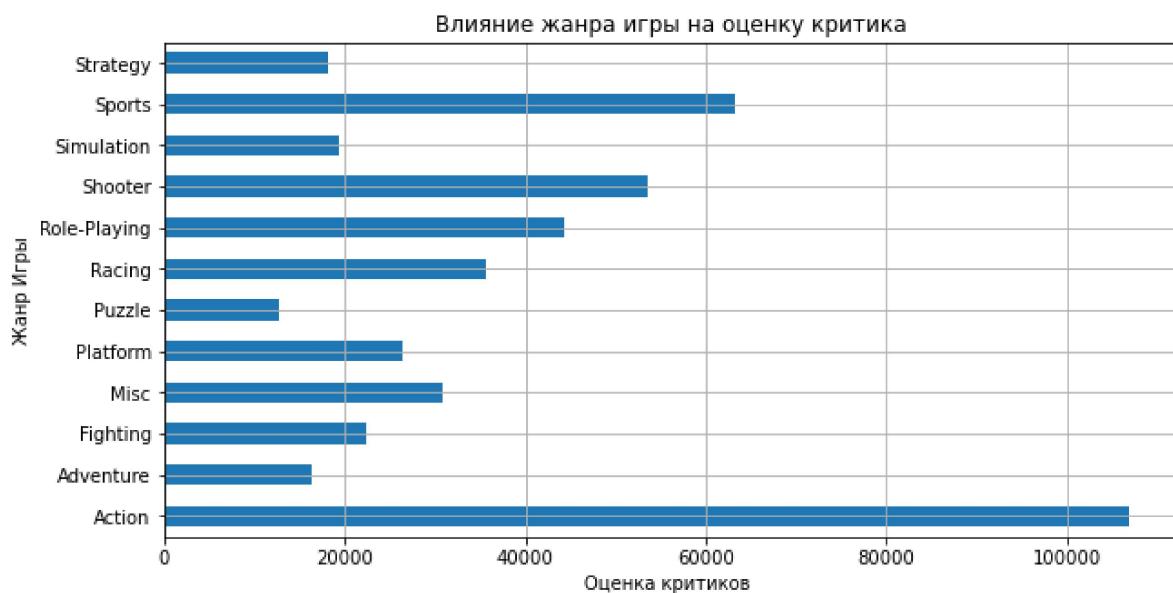


Рис. 1. Влияние жанра игры на оценку критика

In [42]:

```
games.groupby('genre')['user_score'].sum().plot(grid=True, kind='barh', figsize=(10, 5));
plt.title('Влияние жанра игры на оценку пользователя')
plt.ylabel("Жанр Игры");
plt.xlabel('Оценка пользователей \n\n Рис. 1. Влияние жанра игры на оценку пользователя');
```

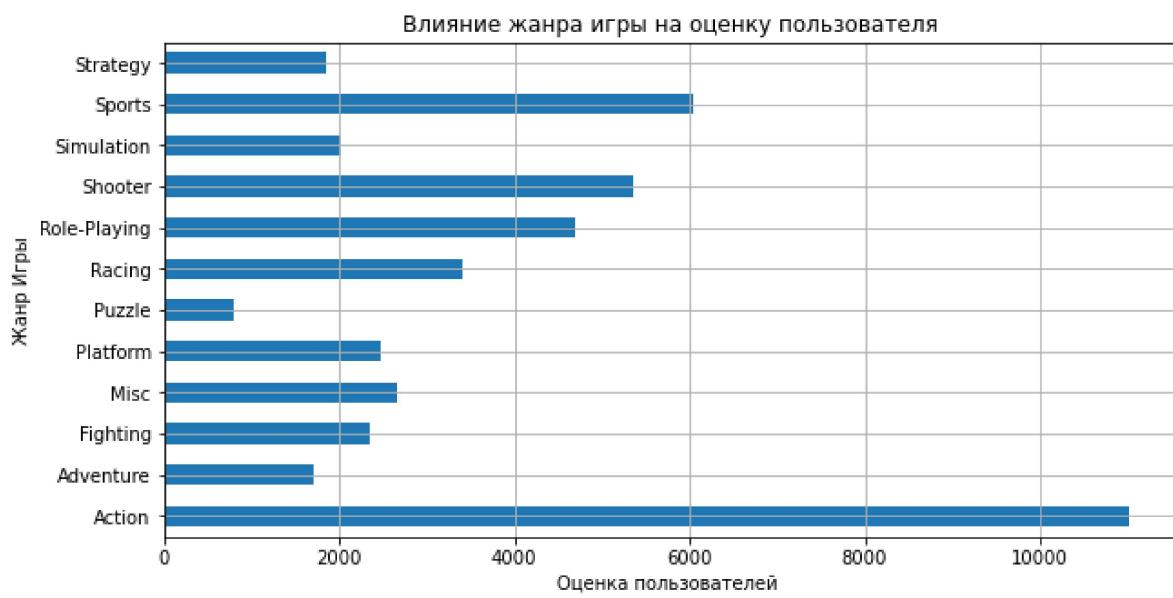


Рис. 1. Влияние жанра игры на оценку пользователя

Вывод:

Наибольшую оценку у пользователей и критиков получает жанр Action или боевик — жанр компьютерных игр, в котором делается упор на эксплуатацию физических возможностей игрока.

In [43]:

```
(games
    .query('genre=="Action"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Action по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Action по годам')
(games
    .query('genre=="Action"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Action по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Action по годам')
```



Рис. 1. Распределение оценок критиков игр в жанре Action по годам



Рис. 1. Распределение оценок пользователей игр в жанре Action по годам

In [44]:

```
(games
    .query('genre=="Adventure"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Adventure по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Adventure по годам')
(games
    .query('genre=="Adventure"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Adventure по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Adventure по годам')
```

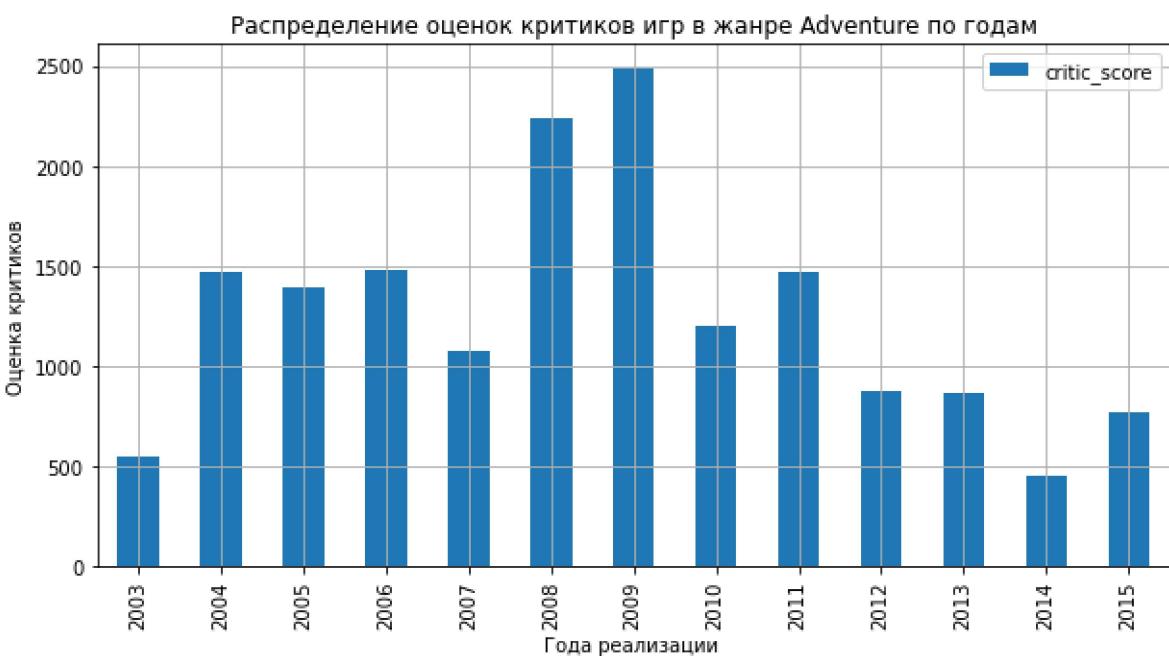
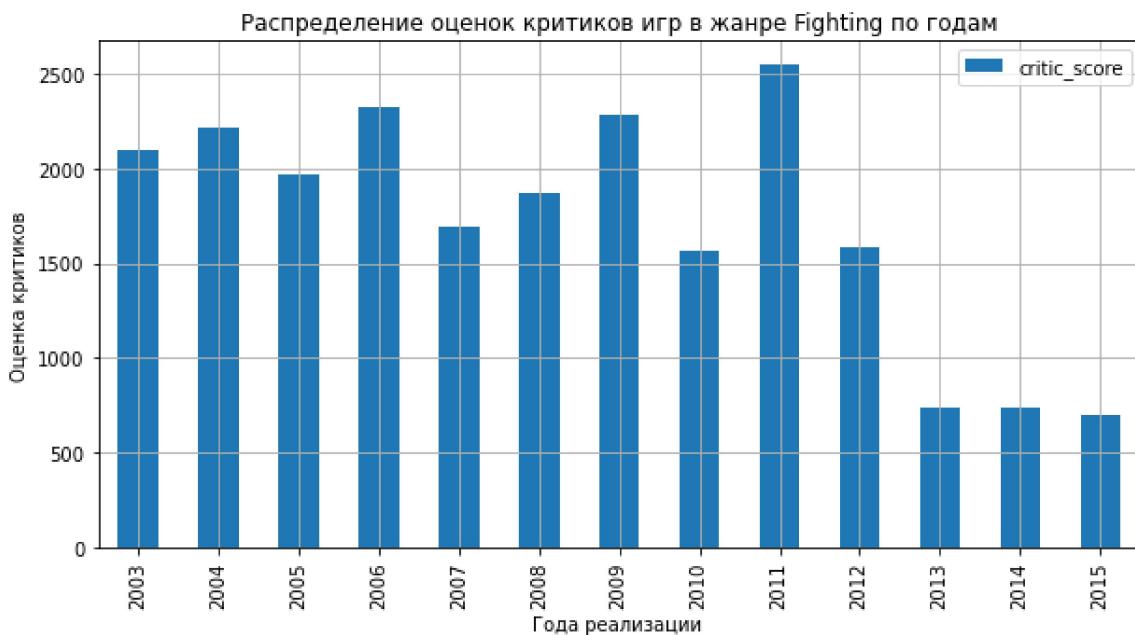


Рис. 1. Распределение оценок критиков игр в жанре Adventure по годам

In [45]:

```
(games
    .query('genre=="Fighting"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Fighting по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Fighting
(games
    .query('genre=="Fighting"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Fighting по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Fig
```



In [46]:

```
(games
    .query('genre=="Misc"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Misc по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Misc по годам')
(games
    .query('genre=="Misc"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('Распределение оценок пользователей игр в жанре Misc по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Misc по годам')
```



Рис. 1. Распределение оценок критиков игр в жанре Misc по годам

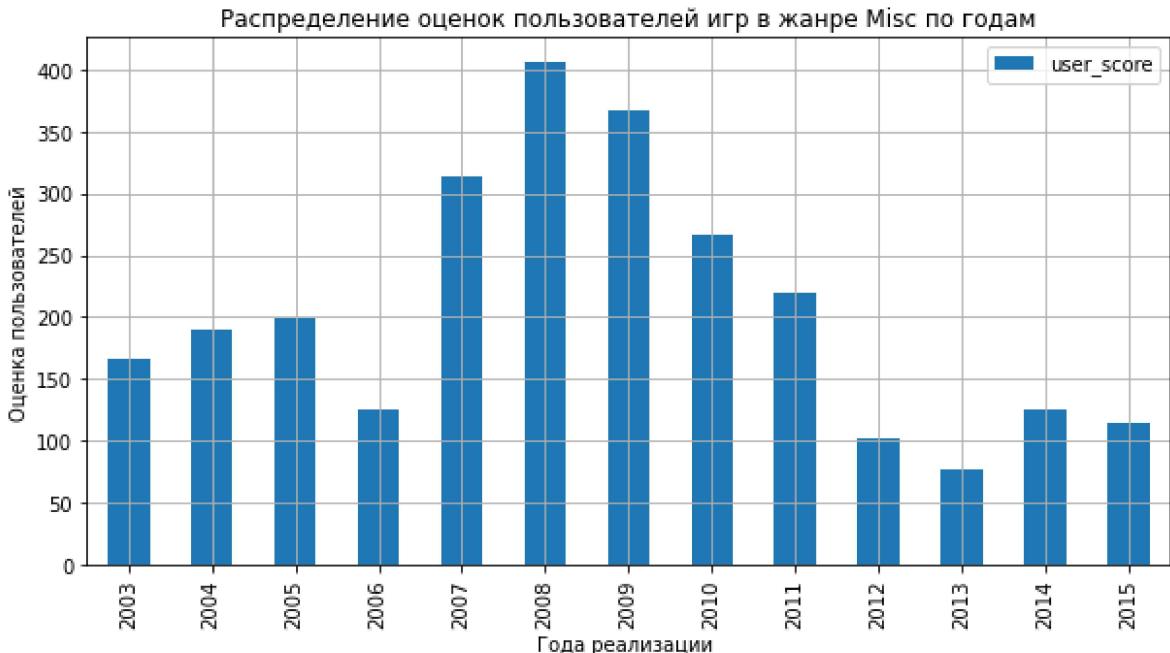


Рис. 1. Распределение оценок пользователей игр в жанре Misc по годам

In [47]:

```
(games
    .query('genre=="Platform"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Platform по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Platform
(games
    .query('genre=="Platform"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Platform по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Platform')
```

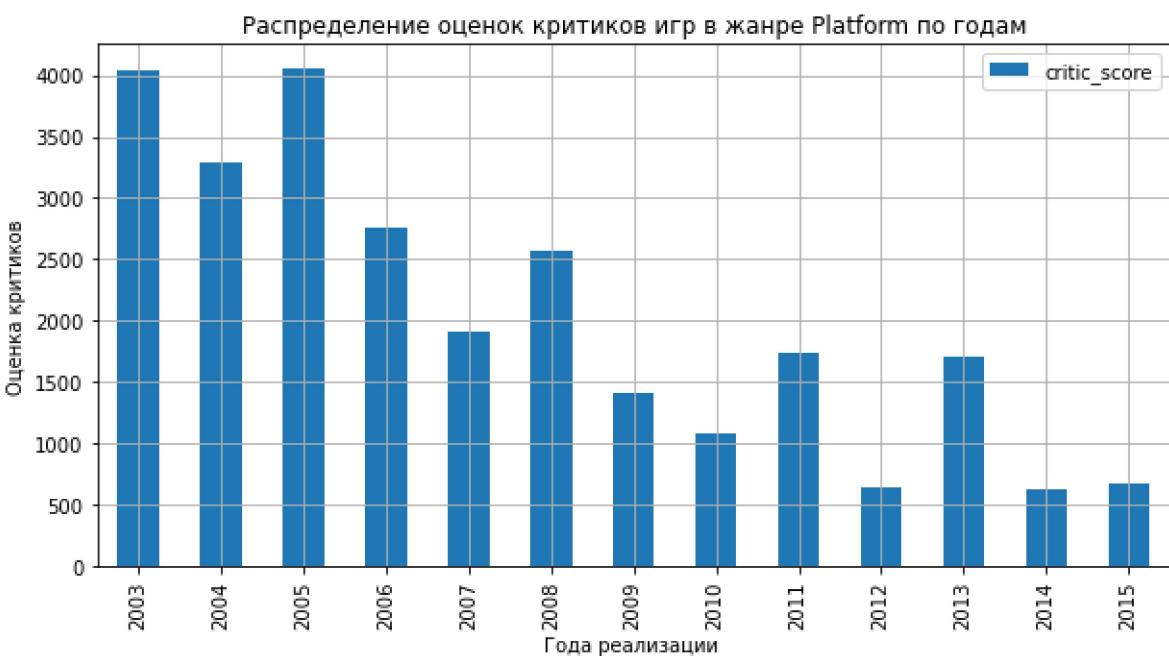


Рис. 1. Распределение оценок критиков игр в жанре Platform по годам



Рис. 1. Распределение оценок пользователей игр в жанре Platform по годам

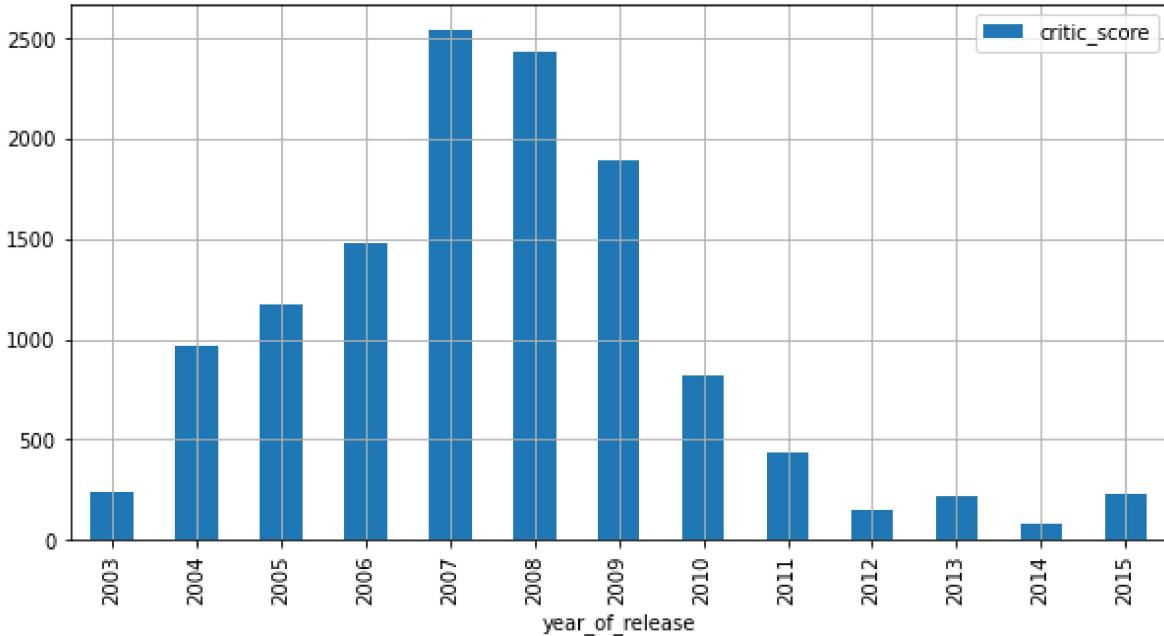
In [48]:

```
(games
    .query('genre=="Puzzle"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('Распределение оценок критиков игр в жанре Puzzle по годам')
(games
    .query('genre=="Puzzle"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
)
plt.title('Распределение оценок пользователей игр в жанре Puzzle по годам')
```

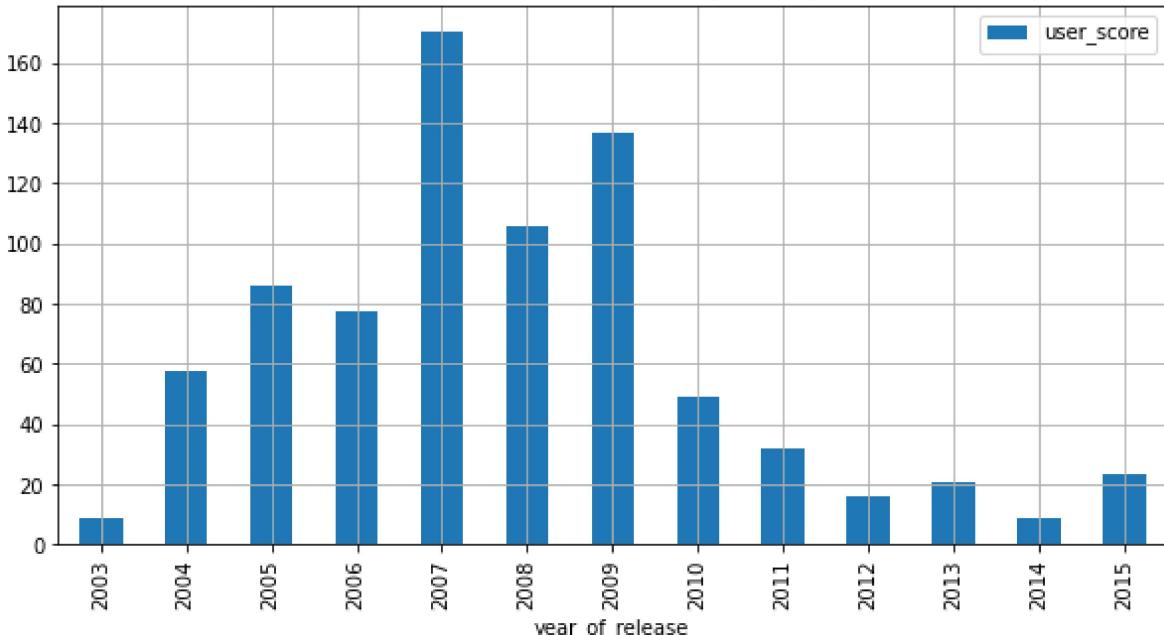
Out[48]:

Text(0.5, 1.0, 'Распределение оценок пользователей игр в жанре Puzzle по годам')

Распределение оценок критиков игр в жанре Puzzle по годам



Распределение оценок пользователей игр в жанре Puzzle по годам



In [49]:

```
(games
    .query('genre=="Racing"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Racing по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Racing по годам')
(games
    .query('genre=="Racing"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Racing по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Racing по годам')
```



Рис. 1. Распределение оценок критиков игр в жанре Racing по годам

Распределение оценок пользователей игр в жанре Racing по годам

▼

In [50]:

```
(games
    .query('genre=="Role-Playing"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Role_Playing по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Role_Pl
(games
    .query('genre=="Role-Playing"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Role_Playing по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Rol
```



Рис. 1. Распределение оценок критиков игр в жанре Role_Playing по годам



Рис. 1. Распределение оценок пользователей игр в жанре Role_Playing по годам

In [51]:

```
(games
    .query('genre=="Shooter"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Shooter по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Shooter
(games
    .query('genre=="Shooter"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Shooter по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Sho
```



Рис. 1. Распределение оценок критиков игр в жанре Shooter по годам

Распределение оценок пользователей игр в жанре Shooter по годам

▼

In [52]:

```
(games
    .query('genre=="Simulation"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Simulation по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Simulation по годам')
(games
    .query('genre=="Simulation"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Simulation по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Simulation по годам')
```

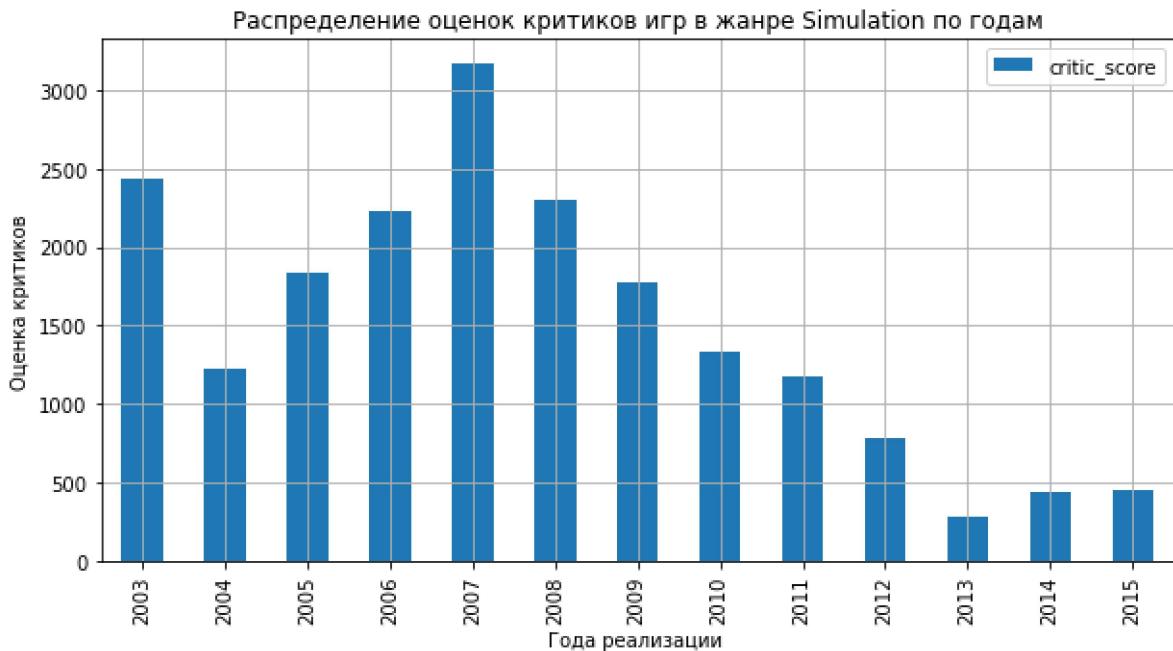


Рис. 1. Распределение оценок критиков игр в жанре Simulation по годам

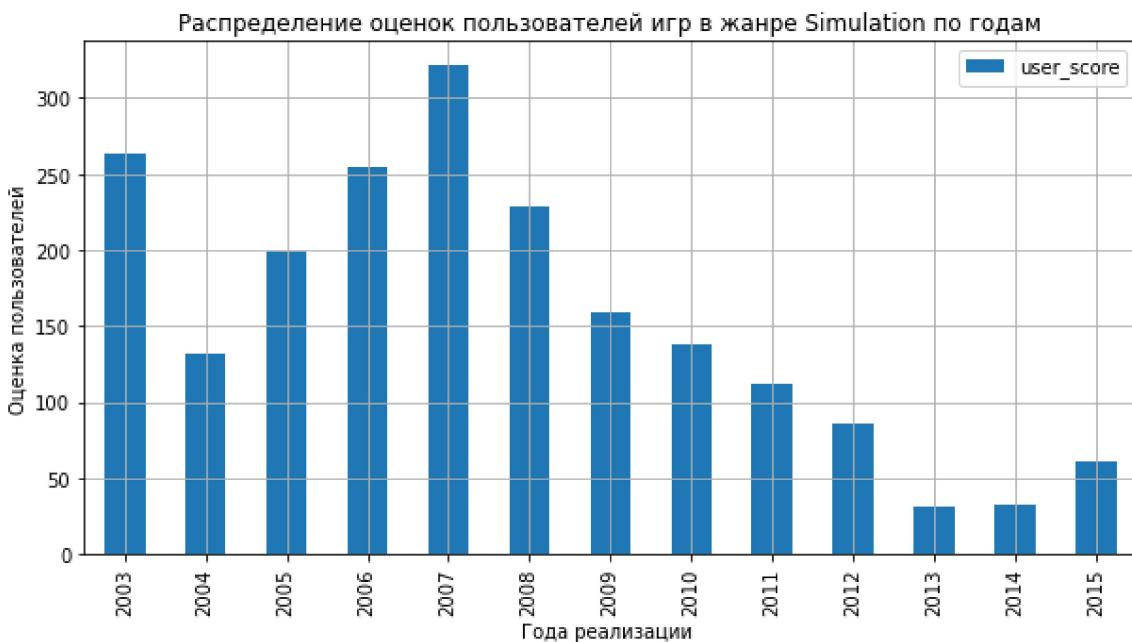


Рис. 1. Распределение оценок пользователей игр в жанре Simulation по годам

In [53]:

```
(games
    .query('genre=="Sports"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Sports по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Sports по годам')
(games
    .query('genre=="Sports"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Sports по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Sports по годам')
```



Рис. 1. Распределение оценок критиков игр в жанре Sports по годам

Распределение оценок пользователей игр в жанре Sports по годам

▼

In [54]:

```
(games
    .query('genre=="Strategy"')
    .pivot_table(index='year_of_release', values='critic_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок критиков игр в жанре Strategy по годам')
plt.ylabel("Оценка критиков");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок критиков игр в жанре Strategy')
(games
    .query('genre=="Strategy"')
    .pivot_table(index='year_of_release', values='user_score', aggfunc=sum)
    .plot(grid=True, kind='bar', figsize=(10, 5))
);
plt.title('Распределение оценок пользователей игр в жанре Strategy по годам')
plt.ylabel("Оценка пользователей");
plt.xlabel('Года реализации \n\n Рис. 1. Распределение оценок пользователей игр в жанре Strategy')
```



Рис. 1. Распределение оценок критиков игр в жанре Strategy по годам



Рис. 1. Распределение оценок пользователей игр в жанре Strategy по годам

Вывод:

В большинстве жанров оценок и критиков, и пользователей после 2007 года уменьшилось. Связано это с тем, что игры ушли на мобильные устройства и ПК.

In [55]:

```
corr = games[['critic_score', 'user_score']].corr()
corr.style.background_gradient(cmap='coolwarm')
```

Out[55]:

	critic_score	user_score
critic_score	1.000000	0.577140
user_score	0.577140	1.000000

Вывод:

Оценка критиков и пользователей зависят на 57,7%. Эта корреляция действительна и для жанров. Критики выше оценивают жанры, чем пользователи.

Посчитаем среднее количество, дисперсию и стандартное отклонение

In [56]:

```
action_user = games[(games['genre'] == 'Action') & (games['user_score'] > 0)]['user_score']
adventure_user = games[(games['genre'] == 'Adventure') & (games['user_score'] > 0)]['user_score']
fighting_user = games[(games['genre'] == 'Fighting') & (games['user_score'] > 0)]['user_score']
misc_user = games[(games['genre'] == 'Misc') & (games['user_score'] > 0)]['user_score']
platform_user = games[(games['genre'] == 'Platform') & (games['user_score'] > 0)]['user_score']
puzzle_user = games[(games['genre'] == 'Puzzle') & (games['user_score'] > 0)]['user_score']
role_playing_user = games[(games['genre'] == 'Role-Playing') & (games['user_score'] > 0)]['user_score']
shooter_user = games[(games['genre'] == 'Shooter') & (games['user_score'] > 0)]['user_score']
simulation_user = games[(games['genre'] == 'Simulation') & (games['user_score'] > 0)]['user_score']
sports_user = games[(games['genre'] == 'Sports') & (games['user_score'] > 0)]['user_score']
strategy_user = games[(games['genre'] == 'Strategy') & (games['user_score'] > 0)]['user_score']
```

In [57]:

```
action_critic = games[(games['genre'] == 'Action') & (games['critic_score'] > 0)]['critic_score']
adventure_critic = games[(games['genre'] == 'Adventure') & (games['critic_score'] > 0)]['critic_score']
fighting_critic = games[(games['genre'] == 'Fighting') & (games['critic_score'] > 0)]['critic_score']
misc_critic = games[(games['genre'] == 'Misc') & (games['critic_score'] > 0)]['critic_score']
platform_critic = games[(games['genre'] == 'Platform') & (games['critic_score'] > 0)]['critic_score']
puzzle_critic = games[(games['genre'] == 'Puzzle') & (games['critic_score'] > 0)]['critic_score']
role_playing_critic = games[(games['genre'] == 'Role-Playing') & (games['critic_score'] > 0)]['critic_score']
shooter_critic = games[(games['genre'] == 'Shooter') & (games['critic_score'] > 0)]['critic_score']
simulation_critic = games[(games['genre'] == 'Simulation') & (games['critic_score'] > 0)]['critic_score']
sports_critic = games[(games['genre'] == 'Sports') & (games['critic_score'] > 0)]['critic_score']
strategy_critic = games[(games['genre'] == 'Strategy') & (games['critic_score'] > 0)]['critic_score']
```

In [58]:

```
print('Дисперсия пользовательских оценок Action: {:.2f}'.format(np.var(action_user)))
print('Дисперсия оценок критиков Action: {:.2f}'.format(np.var(action_critic)))
print('Среднее отклонение оценок пользователей Action: {:.2f}'.format(np.std(action_user)))
print('Среднее отклонение оценок критиков Action: {:.2f}'.format(np.std(action_critic)))
```

Дисперсия пользовательских оценок Action: 1.94
Дисперсия оценок критиков Action: 196.24
Среднее отклонение оценок пользователей Action: 1.39
Среднее отклонение оценок критиков Action: 14.01

In [59]:

```
print('Дисперсия пользовательских оценок Adventure: {:.2f}'.format(np.var(adventure_user)))
print('Дисперсия оценок критиков Adventure: {:.2f}'.format(np.var(adventure_critic)))
print('Среднее отклонение оценок пользователей Adventure: {:.2f}'.format(np.std(adventure_user)))
print('Среднее отклонение оценок критиков Adventure: {:.2f}'.format(np.std(adventure_critic)))
```

Дисперсия пользовательских оценок Adventure: 2.76
Дисперсия оценок критиков Adventure: 195.65
Среднее отклонение оценок пользователей Adventure: 1.66
Среднее отклонение оценок критиков Adventure: 13.99

In [60]:

```
print('Дисперсия пользовательских оценок Fighting: {:.2f}'.format(np.var(fighting_user)))
print('Дисперсия оценок критиков Fighting: {:.2f}'.format(np.var(fighting_critic)))
print('Среднее отклонение оценок пользователей Fighting: {:.2f}'.format(np.std(fighting_use
print('Среднее отклонение оценок критиков Fighting: {:.2f}'.format(np.std(fighting_critic))
```

```
Дисперсия пользовательских оценок Fighting: 1.84
Дисперсия оценок критиков Fighting: 196.92
Среднее отклонение оценок пользователей Fighting: 1.36
Среднее отклонение оценок критиков Fighting: 14.03
```

In [61]:

```
print('Дисперсия пользовательских оценок Misc: {:.2f}'.format(np.var(misc_user)))
print('Дисперсия оценок критиков Misc: {:.2f}'.format(np.var(misc_critic)))
print('Среднее отклонение оценок пользователей Misc: {:.2f}'.format(np.std(misc_user)))
print('Среднее отклонение оценок критиков Misc: {:.2f}'.format(np.std(misc_critic)))
```

```
Дисперсия пользовательских оценок Misc: 2.45
Дисперсия оценок критиков Misc: 191.64
Среднее отклонение оценок пользователей Misc: 1.57
Среднее отклонение оценок критиков Misc: 13.84
```

In [62]:

```
print('Дисперсия пользовательских оценок Platform: {:.2f}'.format(np.var(platform_user)))
print('Дисперсия оценок критиков Platform: {:.2f}'.format(np.var(platform_critic)))
print('Среднее отклонение оценок пользователей Platform: {:.2f}'.format(np.std(platform_use
print('Среднее отклонение оценок критиков Platform: {:.2f}'.format(np.std(platform_critic))
```

```
Дисперсия пользовательских оценок Platform: 2.18
Дисперсия оценок критиков Platform: 167.93
Среднее отклонение оценок пользователей Platform: 1.48
Среднее отклонение оценок критиков Platform: 12.96
```

In [63]:

```
print('Дисперсия пользовательских оценок Puzzle: {:.2f}'.format(np.var(puzzle_user)))
print('Дисперсия оценок критиков Puzzle: {:.2f}'.format(np.var(puzzle_critic)))
print('Среднее отклонение оценок пользователей Puzzle: {:.2f}'.format(np.std(puzzle_user)))
print('Среднее отклонение оценок критиков Puzzle: {:.2f}'.format(np.std(puzzle_critic)))
```

```
Дисперсия пользовательских оценок Puzzle: 2.40
Дисперсия оценок критиков Puzzle: 156.50
Среднее отклонение оценок пользователей Puzzle: 1.55
Среднее отклонение оценок критиков Puzzle: 12.51
```

In [64]:

```
print('Дисперсия пользовательских оценок Role-Playing: {:.2f}'.format(np.var(role_playing_user)))
print('Дисперсия оценок критиков Role-Playing: {:.2f}'.format(np.var(role_playing_critic)))
print('Среднее отклонение оценок пользователей Role-Playing: {:.2f}'.format(np.std(role_playing_user)))
print('Среднее отклонение оценок критиков Role-Playing: {:.2f}'.format(np.std(role_playing_critic)))
```

```
Дисперсия пользовательских оценок Role-Playing: 1.17
Дисперсия оценок критиков Role-Playing: 139.18
Среднее отклонение оценок пользователей Role-Playing: 1.08
Среднее отклонение оценок критиков Role-Playing: 11.80
```

In [65]:

```
print('Дисперсия пользовательских оценок Shooter: {:.2f}'.format(np.var(shooter_user)))
print('Дисперсия оценок критиков Shooter: {:.2f}'.format(np.var(shooter_critic)))
print('Среднее отклонение оценок пользователей Shooter: {:.2f}'.format(np.std(shooter_user)))
print('Среднее отклонение оценок критиков Shooter: {:.2f}'.format(np.std(shooter_critic)))
```

```
Дисперсия пользовательских оценок Shooter: 2.32
Дисперсия оценок критиков Shooter: 218.65
Среднее отклонение оценок пользователей Shooter: 1.52
Среднее отклонение оценок критиков Shooter: 14.79
```

In [66]:

```
print('Дисперсия пользовательских оценок Simulation: {:.2f}'.format(np.var(simulation_user)))
print('Дисперсия оценок критиков Simulation: {:.2f}'.format(np.var(simulation_critic)))
print('Среднее отклонение оценок пользователей Simulation: {:.2f}'.format(np.std(simulation_user)))
print('Среднее отклонение оценок критиков Simulation: {:.2f}'.format(np.std(simulation_critic)))
```

```
Дисперсия пользовательских оценок Simulation: 2.58
Дисперсия оценок критиков Simulation: 153.48
Среднее отклонение оценок пользователей Simulation: 1.60
Среднее отклонение оценок критиков Simulation: 12.39
```

In [67]:

```
print('Дисперсия пользовательских оценок Sports: {:.2f}'.format(np.var(sports_user)))
print('Дисперсия оценок критиков Sports: {:.2f}'.format(np.var(sports_critic)))
print('Среднее отклонение оценок пользователей Sports: {:.2f}'.format(np.std(sports_user)))
print('Среднее отклонение оценок критиков Sports: {:.2f}'.format(np.std(sports_critic)))
```

```
Дисперсия пользовательских оценок Sports: 2.57
Дисперсия оценок критиков Sports: 189.54
Среднее отклонение оценок пользователей Sports: 1.60
Среднее отклонение оценок критиков Sports: 13.77
```

In [68]:

```
print('Дисперсия пользовательских оценок Strategy: {:.2f}'.format(np.var(strategy_user)))
print('Дисперсия оценок критиков Strategy: {:.2f}'.format(np.var(strategy_critic)))
print('Среднее отклонение оценок пользователей Strategy: {:.2f}'.format(np.std(strategy_use)
print('Среднее отклонение оценок критиков Strategy: {:.2f}'.format(np.std(strategy_critic))
```

```
Дисперсия пользовательских оценок Strategy: 2.87
Дисперсия оценок критиков Strategy: 137.15
Среднее отклонение оценок пользователей Strategy: 1.69
Среднее отклонение оценок критиков Strategy: 11.71
```

Вывод:

Оценки критиков и пользователей для большинства жанров разбросаны равномерно. Пользователи оценивают стратегии, спорт и симуляции выше критиков, а критики ценят пазлы больше пользователей.

In [69]:

```
mean_value_user = games['user_score'].mean()
mean_value_critic = games['critic_score'].mean()
spacing_all_user = games['user_score'] - mean_value_user # для каждого элемента датасета находим расстояние до среднего
spacing_all_critic = games['critic_score'] - mean_value_critic
spacing_all_mean_user = spacing_all_user.mean()
spacing_all_mean_critic = spacing_all_critic.mean()
print('Среднее расстояние между пользовательскими оценками', spacing_all_mean_user)
print('Среднее расстояние между оценками критиков', spacing_all_mean_critic) # считаем среднее
```

```
Среднее расстояние между пользовательскими оценками -3.2269021142363005e-14
Среднее расстояние между оценками критиков -6.066591775035e-14
```

Откуда взялся нулевой разброс, если все значения в наборе данных разные? Это произошло из-за того, что я складывала расстояния от среднего до значений больше него с расстояниями от среднего до значений меньше него. Одни получились положительными, другие — отрицательными. Расстояния компенсировали друг друга, и в итоге получился ноль. Избавимся от знаков при подсчёте, сделав все значения неотрицательными — возведём значения расстояний в квадрат. Улучшенная метрика разброса — не просто среднее расстояние между значениями датасета и средним, а средний квадрат этого расстояния. Эта величина называется дисперсия (лат. dispersio, «рассеяние»).

In [70]:

```
variance_user = np.var(games['user_score'])
print('Дисперсия пользовательских оценок: {:.2f}'.format(variance_user))
variance_critic = np.var(games['critic_score'])
print('Дисперсия оценок критиков: {:.2f}'.format(variance_critic))
```

```
Дисперсия пользовательских оценок: 2.23
Дисперсия оценок критиков: 191.23
```

In [71]:

```
standart_dev_user=np.sqrt(variance_user)
print ('Стандартное отклонение пользовательских оценок: {:.2f}'.format(standart_dev_user))
standart_dev_critic=np.sqrt(variance_critic)
print ('Стандартное отклонение оценок критиков: {:.2f}'.format(standart_dev_critic))
```

```
Стандартное отклонение пользовательских оценок: 1.49
Стандартное отклонение оценок критиков: 13.83
```

Вывод:

Пользовательские оценки и оценки критиков разбросаны равномерно для большинства жанров. Критики чаще ставят низкие оценки и оценивают жанры, которые пользователи считают непопулярными. Пользователи оценивают стратегии, спорт и симуляции выше критиков, а критики ценят пазлы больше пользователей.

5.Проверка гипотез

- Средние пользовательские рейтинги платформ Xbox One и PC одинаковые;
- Средние пользовательские рейтинги жанров Action (англ. «действие», экшен-игры) и Sports (англ. «спортивные соревнования») разные.

Задать самостоятельно пороговое значение alpha.

Пояснить:

- Как вы сформулировали нулевую и альтернативную гипотезы;
- Какой критерий применили для проверки гипотез и почему.

Проверка гипотез:

1. H0 Средние пользовательские рейтинги платформ Xbox One и PC одинаковы.
2. H1 Средние пользовательские рейтинги платформ Xbox One и PC различны.

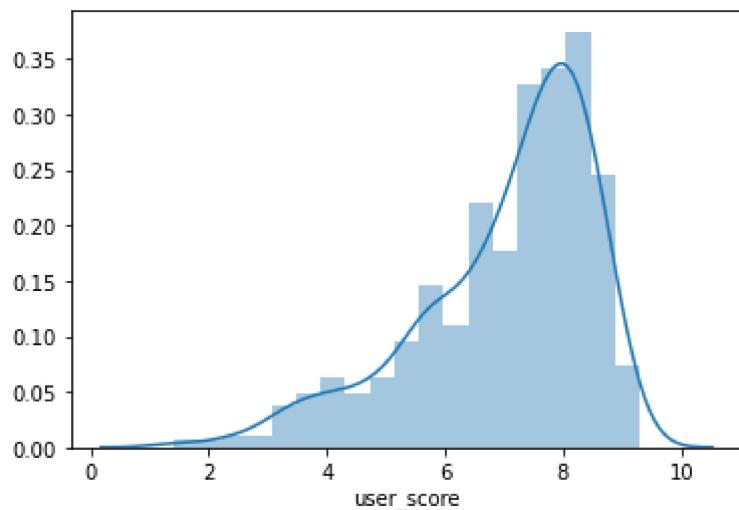
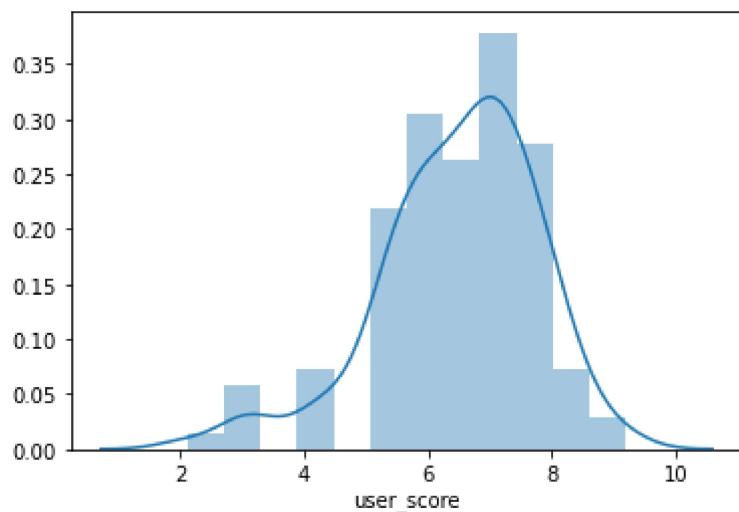
In [72]:

```
# Задать самостоятельно пороговое значение alpha.
alpha = 0.05
xbox = games[(games['platform'] == 'XOne') & (games['user_score'] > 0)]['user_score']
pc = games[(games['platform'] == 'PC') & (games['user_score'] > 0)]['user_score']
if np.var(xbox)==np.var(pc):
    print('True')
else:
    print('False')
```

```
False
```

In [73]:

```
sns.distplot(xbox)
plt.show()
sns.distplot(pc)
plt.show()
```



Распределение Гауссовское, используем Т-критерий Стьюдентса, дисперсия выборок не равна.,

In [74]:

```
results = scipy.stats.ttest_ind(xbox, pc, equal_var = False)
print('p-значение:', results.pvalue)

if (results.pvalue < alpha):
    print("Отвергаем нулевую гипотезу")
else:
    print("Не получилось отвергнуть нулевую гипотезу")
```

p-значение: 6.984889036841548e-05

Отвергаем нулевую гипотезу

Проверка гипотез:

1. H0 Средние пользовательские рейтинги жанров Action (англ. «действие») и Sports (англ. «виды спорта») одинаковы.
2. H1 Средние пользовательские рейтинги жанров Action (англ. «действие») и Sports (англ. «виды спорта») разные.

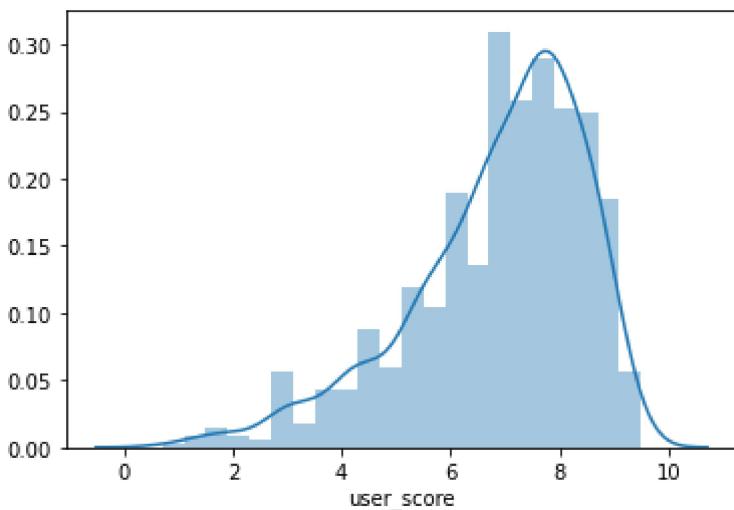
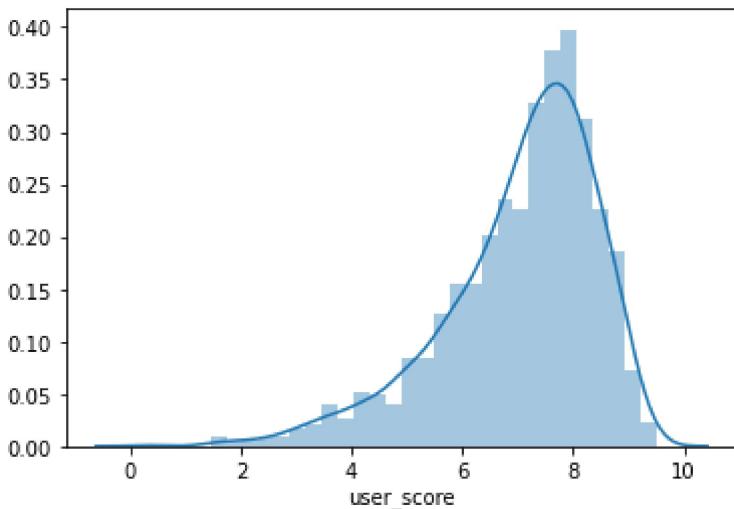
In [75]:

```
alpha = 0.05
action = games[(games['genre'] == 'Action') & (games['user_score'] > 0)]['user_score']
sports = games[(games['genre'] == 'Sports') & (games['user_score'] > 0)]['user_score']
if np.var(action)==np.var(sports):
    print('True')
else:
    print('False')
```

False

In [76]:

```
sns.distplot(action)
plt.show()
sns.distplot(sports)
plt.show()
```



Распределение Гауссовское, используем Т-критерий Стьюдентса, дисперсия выборок не равна.

In [77]:

```
results=scipy.stats.ttest_ind(action, sports, equal_var = False)
print('р-значение:', results.pvalue)

if (results.pvalue < alpha):
    print("Отвергаем нулевую гипотезу")
else:
    print("Не получилось отвергнуть нулевую гипотезу")
```

р-значение: 0.011878765616484769

Отвергаем нулевую гипотезу

Вывод:

Пользовательские рейтинги для платформ Xbox и PC различны. Средние пользовательские рейтинги жанров Action (англ. «действие») и Sports (англ. «виды спорта») тоже различны.

6. Выводы

Геймдэйв медленно и верно теряет обороты, игры ушли на мобильные устройства. Люди играть в приставки не перестали, поэтому не останавливаются и разработчики, хотя делают это уже более осмотрительно.

Пятерка самых популярных платформ набрала свою популярность с начала 2000-х. Период от первых продаж до исчезновения продаж в среднем занимает 14 лет. Период для появления платформы и выхода из нее с 1980 до 2000 такой же, как и для платформ с 2000 до 2016, от 9 до 11 лет. Платформа PS трансформировалась в PS2 и PS3. Платформа NES уступила свое место более совершенным SNES с более совершенной приставкой Sega Mega Drive, также как до этого N64 уступила свое место NES. SNES 95 году обещали выпустить новую приставку, несколько раз переносили презентацию, в итоге для новой приставки выпустили 22 игры, все японского производства и платформа перестала существовать. GB были линейкой портативных устройств от компании Nintendo (N64, NES, SNES), с уходом Nintendo с рынка исчезли и GB. Период для появления платформы и выхода из нее с 1980 до 2000 меньше, чем для платформ с 2000 до 2016, и составляет 10,4 года. Платформа PS трансформировалась в PS2 и PS3. Платформа NES уступила свое место более совершенным SNES с более совершенной приставкой Sega Mega Drive, также как до этого N64 уступила свое место NES. SNES 95 году обещали выпустить новую приставку, несколько раз переносили презентацию, в итоге для новой приставки выпустили 22 игры, все японского производства и платформа перестала существовать. GB были линейкой портативных устройств от компании Nintendo (N64, NES, SNES), с уходом Nintendo с рынка исчезли и GB. Нормально распределены данные за период с 2003 по 2015 год, их используем в дальнейшей работе. Потенциально прибыльны платформы PS4, X360 и XOne, что неудивительно, потому что это мастодонты рынка.

От уровня оценки прямо пропорционально зависят продажи, и не важно оценил игру пользователь или критик. А также самые дорогие продажи приходятся на самые высокие оценки критиков. Жанры для игр в Северной Америке, Европе и Японии популярны одни и те же, а вот игровые платформы у японцев значительно отличаются. В разы различаются и продажи в этих регионах: больше всего продаж в Северной Америке, меньше всего в Японии. Предполагаю, что связано это с разным менталитетом этих регионов. Зависимость от рейтинга отслеживается во всех трех регионах. Наиболее популярны игры с рейтингом "для всех", "подросткам", "детям старше 10 лет" и "для взрослых" (в Европе и Северной Америке спрос на них выше, японцы больше предпочитают игры без определенного рейтинга). Ситуация в Японии связана с тем, что получить возрастной рейтинг на японском рынке сложно и дорого. Начинает это с того, что всё общение и вся документация — на японском языке. «Google Переводчиком» в данной ситуации пользоваться не стоит, лучше найти человека со знанием языка или студию локализации.

Есть 2 варианта для построения отношений:

1. Разовый платёж. Вы хотите единожды сделать рейтинг для одной игры и на одну (или несколько) платформу — стоило это на 2018 год примерно 2 тысячи евро.
2. Членство. У вас есть как вступительный взнос, так и взнос за каждую игру, но цена за оценку игры ниже, чем разовая оценка в первом варианте. То есть в перспективе, если вы планируете делать 5-7 релизов и на 3-4 платформы, вам стоит рассмотреть и членство в данном агентстве.

В большинстве жанров оценок и критиков, и пользователей после 2007 года уменьшилось. Связано это с тем, что рынок игр в то время было принято хоронить, игры ушли на мобильные устройства и ПК, росли продажи только на стратегии и ролевые игры. Оценки критиков и пользователей для большинства жанров разбросаны равномерно. Пользователи оценивают стратегии, спорт и симуляции выше критиков, а критики ценят пазлы больше пользователей.