

Загрузка и изученные данных

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
from scipy import stats

import warnings
warnings.simplefilter('ignore')
```

Добрый день, уважаемые коллеги! Рада вас приветствовать на презентации хода исследования и его результатов.

Описание проекта:

В современном мире быстро и высоко развивается цифровизация, развлечения и досуг все глубже уходят в виртуальную среду. Технологии совершенствуются, запросы потребителей повышаются. Магазинам, продающим развлекательный контент, в частности компьютерные игры, необходимо учитывать актуальный запрос потребителя и понимать успешность той или иной игры, чтобы планировать рекламные кампании. Для выявления закономерностей успешности игр необходимо проанализировать данные о прошлом. Важно учитывать такие факторы, как жанр игры, платформа, возрастная категория и оценка пользователей. Исследования показывают, что игры в жанре экшен и RPG часто занимают верхние позиции в рейтингах продаж. Например, проекты с глубоким сюжетом и открытым миром, как "The Witcher" или "Red Dead Redemption", привлекают значительное внимание пользователей.

Цель проекта:

Провести исследование, чтобы определить, на каких играх стоит сосредоточиться для проведения рекламной кампании. Проверить некоторые гипотезы, которые помогут спланировать рекламную кампанию интернет-магазина.

План исследования:

1. Загрузка данных. Первичное знакомство с представленной информацией.
2. Предобработка данных, стандартизация данных.
 - 2.1 Обработка пропусков
 - 2.2 Обработка дубликатов
 - 2.3 Добавление новых столбцов
3. Исследовательский анализ

- 3.1 Анализ по годам релиза игр
 - 3.2 Изучение продаж игр, сгруппированных по платформам
 - 3.3 Определение актуального периода
 - 3.4 Взаимосвязь отзывов критиков и пользователей с продажами
 - 3.5 Анализ игр, сгруппированных по жанрам
- 4. Составление портрета пользователя в каждом регионе
 - 4.1 Распределение долей продаж в регионах в зависимости от платформы релиза
 - 4.2 Распределение долей продаж в регионах в зависимости от жанра игры
 - 4.3 Распределение долей продаж в регионах в зависимости от типа возрастного рейтинга

Общий вывод:

Резюмирование полученных результатов, формулировка ключевых выводов и рекомендаций.

```
In [3]: games_data = pd.read_csv('/datasets/games.csv')
```

```
In [4]: games_data.columns
```

```
Out[4]: Index(['Name', 'Platform', 'Year_of_Release', 'Genre', 'NA_sales', 'EU_sales',
              'JP_sales', 'Other_sales', 'Critic_Score', 'User_Score', 'Rating'],
              dtype='object')
```

```
In [5]: games_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16715 entries, 0 to 16714
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Name                  16713 non-null  object
1   Platform              16715 non-null  object
2   Year_of_Release       16446 non-null  float64
3   Genre                 16713 non-null  object
4   NA_sales              16715 non-null  float64
5   EU_sales              16715 non-null  float64
6   JP_sales              16715 non-null  float64
7   Other_sales           16715 non-null  float64
8   Critic_Score          8137 non-null   float64
9   User_Score            10014 non-null  object
10  Rating                9949 non-null   object
dtypes: float64(6), object(5)
memory usage: 1.4+ MB
```

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

Out[6]:

| | 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 | |
| 5 | Tetris | GB | 1989.0 | Puzzle | 23.20 | 2.26 | 4.22 | |
| 6 | New Super Mario Bros. | DS | 2006.0 | Platform | 11.28 | 9.14 | 6.50 | |
| 7 | Wii Play | Wii | 2006.0 | Misc | 13.96 | 9.18 | 2.93 | |
| 8 | New Super Mario Bros. Wii | Wii | 2009.0 | Platform | 14.44 | 6.94 | 4.70 | |
| 9 | Duck Hunt | NES | 1984.0 | Shooter | 26.93 | 0.63 | 0.28 | |



In [7]:

```
games_data.tail(10)
```

Out[7]:

| | Name | Platform | Year_of_Release | Genre | NA_sales | EU_sales | JP_sales |
|--------------|--|----------|-----------------|------------|----------|----------|----------|
| 16705 | 15 Days | PC | 2009.0 | Adventure | 0.00 | 0.01 | 0.00 |
| 16706 | Men in Black II: Alien Escape | GC | 2003.0 | Shooter | 0.01 | 0.00 | 0.00 |
| 16707 | Aiyoku no Eustia | PSV | 2014.0 | Misc | 0.00 | 0.00 | 0.01 |
| 16708 | Woody Woodpecker in Crazy Castle 5 | GBA | 2002.0 | Platform | 0.01 | 0.00 | 0.00 |
| 16709 | SCORE International Baja 1000: The Official Game | PS2 | 2008.0 | Racing | 0.00 | 0.00 | 0.00 |
| 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 |



In [8]: `games_data.isna().sum()`

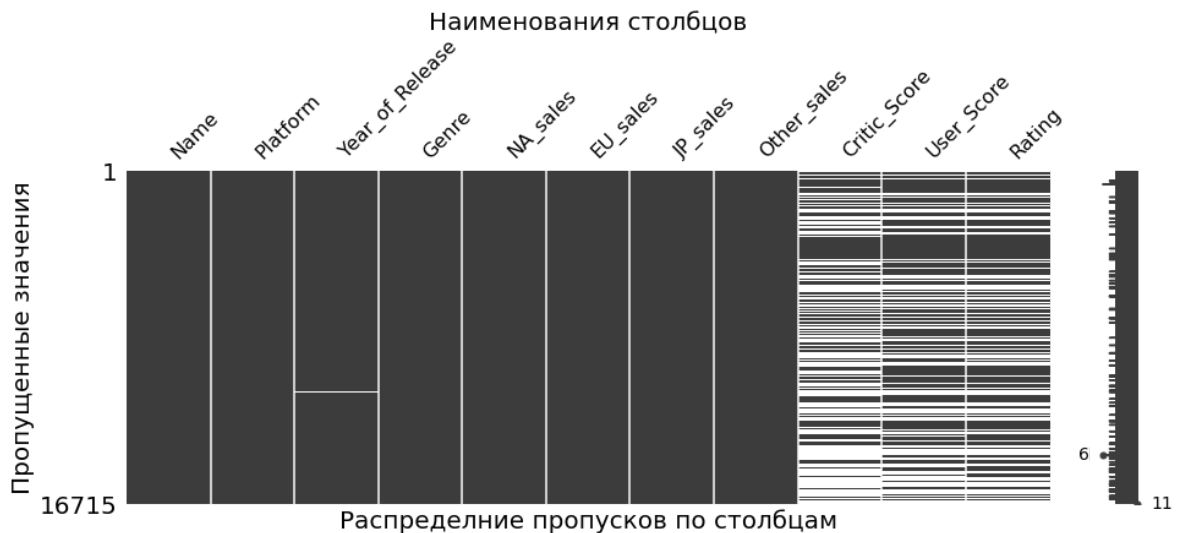
Out[8]:

| | |
|-----------------|-------|
| Name | 2 |
| Platform | 0 |
| Year_of_Release | 269 |
| Genre | 2 |
| NA_sales | 0 |
| EU_sales | 0 |
| JP_sales | 0 |
| Other_sales | 0 |
| Critic_Score | 8578 |
| User_Score | 6701 |
| Rating | 6766 |
| dtype: | int64 |

In [9]: `msno.matrix(games_data, figsize=(15,5))`

```
plt.xlabel('Распределение пропусков по столбцам', fontsize=20)
plt.ylabel('Пропущенные значения', fontsize=20)
```

```
plt.title('Наименования столбцов', fontsize=20)
plt.show()
```



После вывода основной информации можно сделать некоторые выводы о состоянии датасета и сформировать дальнейший план исследования:

- * Название столбцов необходимо привести к "змеиному" типу;
- * Привести тип столбца, в котором указан год выпуска, к целочисленному типу;
- * Обработать пропуски в столбцах;
- * Добавить новые столбцы, необходимые для продолжения анализа;
- * Проверить дубликаты и в случае их наличия удалить повторяющиеся строки.

Далее приступаю к преобработке данных, так как уже имеется понимание, с какой информацией имеем дело и как дальше работать.

Преобработка данных

Стандартизация датасета (наименования столбцов, тип данных)

Стандартизируем названия столбцов, придавая им единообразие и ясность. Приведем столбцы к необходимым типам данных, чтобы избежать ошибок и конфликтов в дальнейшей работе с данными. Обработаем пропуски, поскольку они могут исказить итоги нашего исследования. Проведем тщательную работу по устранению явных и неявных дубликатов, которые также способны исказить результаты и усложнить процесс анализа. Каждое из этих действий — это наш шаг к созданию точного и надежного полотна данных, на котором мы сможем построить наши выводы. Исправив недостатки, мы откроем перед собой возможность для более четкого понимания и глубокого анализа, что, в конечном счете, приведет к обоснованным выводам. Внимание к деталям и качественная обработка информации в процессе подготовки данных — залог успеха нашего исследования, позволяющий избежать подводных камней и неясностей на пути к истине..

```
In [10]: for column in games_data.columns:
         if column not in ['NA_sales', 'EU_sales', 'JP_sales']:
             games_data.rename(columns={column: column.lower()}, inplace=True)
         else:
             continue
```

```
In [11]: games_data.columns
```

```
Out[11]: Index(['name', 'platform', 'year_of_release', 'genre', 'NA_sales', 'EU_sales',
               'JP_sales', 'other_sales', 'critic_score', 'user_score', 'rating'],
              dtype='object')
```

```
In [12]: games_data['year_of_release'] = games_data['year_of_release'].astype('Int64')
         games_data['critic_score'] = games_data['critic_score'].astype('Int64')
```

```
In [13]: print(games_data['year_of_release'].dtype)
         print(games_data['critic_score'].dtype)
```

Int64

Int64

Выводы по разделу:

В ячейках выше в цикле привожу названия столбцов к "змеиному" типу, кроме столбцов, имеющих в своём названии аббревиатуры названий стран, так как считаю важным для понимания сохранить сокращённое написание заглавными буквами

Произвела замену типа данных в столбце, в котором указан год релиза игры и оценка критика, так как обозначение года не может быть представлено вещественным числом, а в столбце с оценкой от критиков все значения в десятичной части имеют значение ноль.

Проверка Дубликатов

```
In [14]: len(games_data[games_data.duplicated()])
```

```
Out[14]: 0
```

```
In [15]: duplicates = games_data[games_data.duplicated(subset=['name', 'platform', 'year_


         if duplicates.empty:
             print("Неявные дубликаты не найдены.")
         else:
             print("Найдены неявные дубликаты:")

         duplicates
```

Найдены неявные дубликаты:

Out[15]:

| | name | platform | year_of_release | genre | NA_sales | EU_sales | JP_sales | other_s |
|-------|---------------|----------|-----------------|--------|----------|----------|----------|---------|
| 604 | Madden NFL 13 | PS3 | 2012 | Sports | 2.11 | 0.22 | 0.00 | |
| 659 | NaN | GEN | 1993 | NaN | 1.78 | 0.53 | 0.00 | |
| 14244 | NaN | GEN | 1993 | NaN | 0.00 | 0.00 | 0.03 | |
| 16230 | Madden NFL 13 | PS3 | 2012 | Sports | 0.00 | 0.01 | 0.00 | |



In [16]: `games_data.loc[659, 'JP_sales'] = games_data.loc[14244, 'JP_sales']`

In [17]: `games_data = games_data.drop(index=[14244, 16230])`

In [18]: `games_data.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16713 entries, 0 to 16714
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   name                  16712 non-null  object
1   platform              16713 non-null  object
2   year_of_release       16444 non-null  Int64
3   genre                 16712 non-null  object
4   NA_sales              16713 non-null  float64
5   EU_sales              16713 non-null  float64
6   JP_sales              16713 non-null  float64
7   other_sales           16713 non-null  float64
8   critic_score          8136 non-null   Int64
9   user_score            10013 non-null  object
10  rating                9948 non-null   object
dtypes: Int64(2), float64(4), object(5)
memory usage: 1.6+ MB
```

Выводы по разделу:

Явных дубликатов не выявлено

Выявлены и удалены неявные дубликаты в количестве 2х строк.

Работа с пропусками

In [19]: `games_data.isna().sum()`

```
Out[19]: name          1
platform        0
year_of_release 269
genre           1
NA_sales        0
EU_sales        0
JP_sales        0
other_sales     0
critic_score    8577
user_score      6700
rating          6765
dtype: int64
```

```
In [23]: games_data = games_data.dropna(subset=['year_of_release', 'name']).reset_index(drop=True)
games_data
```

```
Out[23]:
```

| | name | platform | year_of_release | genre | NA_sales | EU_sales | JP_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 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 16438 | Samurai Warriors: Sanada Maru | PS3 | 2016 | Action | 0.00 | 0.00 | 0.01 |
| 16439 | LMA Manager 2007 | X360 | 2006 | Sports | 0.00 | 0.01 | 0.00 |
| 16440 | Haitaka no Psychedelica | PSV | 2016 | Adventure | 0.00 | 0.00 | 0.01 |
| 16441 | Spirits & Spells | GBA | 2003 | Platform | 0.01 | 0.00 | 0.00 |
| 16442 | Winning Post 8 2016 | PSV | 2016 | Simulation | 0.00 | 0.00 | 0.01 |

16443 rows × 11 columns

```
In [24]: games_data.isna().sum()
```



```
Out[24]: name          0
platform        0
year_of_release 0
genre           0
NA_sales        0
EU_sales        0
JP_sales        0
other_sales     0
critic_score    8461
user_score      6605
rating          6676
dtype: int64
```

Удалили все пропуски в значениях года выпуска.

Далее приступаю к обработке пропусков в значениях оценок критиков и пользователей, а так же значения рейтинга для игр.

```
In [25]: print(f'Года издания игр, в которых пропущены значения рейтинга и оценок для все
for year, group in games_data.groupby('year_of_release'):

    if (group['critic_score'].isna().all() and
        group['user_score'].isna().all() and
        group['rating'].isna().all()):

        print(year)
```

Года издания игр, в которых пропущены значения рейтинга и оценок для всех игр, выпущенных в этом году:

```
1980
1981
1982
1983
1984
1986
1987
1989
1990
1991
1993
1995
```

В цикле выше выяснили, что для почти всех годов в интервале от 1980 до 1995 пропущены все значения рейтингов и оценок, кроме 1985, 1988, 1992 и 1994. Это можно объяснить тем, что в ранние годы популярности компьютерных игр не придавали особого значения оценкам и рейтингам. Ограничений не было, так как в начале эры развития основная аудитория игр состояла из детей, в частности школьников. За 10 лет в 80-е мы имеем 2 пропуска, и такое же число отсутствия данных на 5 лет в 90-е до 1995 года. После 1995 года полного отсутствия данных совсем не наблюдается.

В данной работе применить простые способы замены пропусков медианой, средним или модой не представляется возможным, так как рейтинги и оценки в данной группе не зависимы друг от друга, и заполнить их каким-либо из изученных способов ранее не представляется возможным. Предположу, что в

будущем с такими пропусками возможно будет работать через парсинг данных (загрузить данные из открытых источников в интернете, либо передать информацию разработчикам и получить дополненную информацию от них). В этой работе пропуски я оставляю неизменными.

```
In [26]: games_data[games_data['user_score'] == 'tbd']
```

Out[26]:

| | name | platform | year_of_release | genre | NA_sales | EU_sales | JP_sales |
|-------|--------------------------------|----------|-----------------|------------|----------|----------|----------|
| 119 | Zumba Fitness | Wii | 2010 | Sports | 3.45 | 2.59 | 0.0 |
| 300 | Namco Museum: 50th Anniversary | PS2 | 2005 | Misc | 2.08 | 1.35 | 0.0 |
| 516 | Zumba Fitness 2 | Wii | 2011 | Sports | 1.51 | 1.03 | 0.0 |
| 639 | uDraw Studio | Wii | 2010 | Misc | 1.65 | 0.57 | 0.0 |
| 709 | Just Dance Kids | Wii | 2010 | Misc | 1.52 | 0.54 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 16423 | Planet Monsters | GBA | 2001 | Action | 0.01 | 0.00 | 0.0 |
| 16425 | Bust-A-Move 3000 | GC | 2003 | Puzzle | 0.01 | 0.00 | 0.0 |
| 16426 | Mega Brain Boost | DS | 2008 | Puzzle | 0.01 | 0.00 | 0.0 |
| 16432 | Plushees | DS | 2008 | Simulation | 0.01 | 0.00 | 0.0 |
| 16434 | Men in Black II: Alien Escape | GC | 2003 | Shooter | 0.01 | 0.00 | 0.0 |

2376 rows × 11 columns



В столбце "user_score" встречаем аббревиатуру "tbd", что является сокращением с английского языка и означает "to be determined", что в переводе означает "предстоит определить". Данное значение представляет собой пропуск в данных датасета. поэтому считаю необходимым в данном столбце привести замену типа данных, все аббревиатуры "tbd" привести к нулевому значению "NaN"

```
In [27]: games_data['user_score'] = pd.to_numeric(games_data['user_score'], errors='coerc
```

```
In [28]: games_data[games_data['user_score'] == 'tbd']
```

Out[28]:

| name | platform | year_of_release | genre | NA_sales | EU_sales | JP_sales | other_sales | crit |
|------|----------|-----------------|-------|----------|----------|----------|-------------|------|
|------|----------|-----------------|-------|----------|----------|----------|-------------|------|

Выводы по разделу:

Удалили все пропуски в значениях года выпуска, так как объем пропусков не превышает 1,6 % от общего объема инфомайии.

Отсутствие оценок и рейтингов в период 80-х и первой половине 90-х можно объяснить не высоким уровнем развития игровой индустрии, возможно, не серьезное отношение к ней, тк в большинстве случаев воспринималось как развлечение для детей. Рейтингов не было, так как это совсем недавнее нововедение в индустрию игр и других развлекательных контентов.

- *Лишь В 1992 году была учреждена японская организация Ethics Organization of Computer Software (EOCS), которая стала первой организацией, присваивающей возрастные рейтинги видеоигр в мире. В 1994 году был создан национальный регулятор видеоигровой продукции в США.*

В оценке пользователей и критиков встречается множество пропусков, что можно объяснить тем, что на ранних этапах развития игр данные не собирались и не оценивались.

Так же к появлению пропусков может приветсти:

- В процессе загрузки данных могла произойти потеря информации.
- Данные о играх собираются из разных источников, и не для всех игр удалось найти информацию об оценках. Возможно, в некоторых случаях просто не было возможности получить эти данные.
- Данные были получены из внешних источников, они уже изначально могли содержать пропуски. **

Добавление доп. информации в датасет

In [29]: `games_data['total_sales'] = games_data['EU_sales'] + games_data['NA_sales'] + ga`

Добавила в датасет столбец "total_sales" с суммой продаж данной игры по всем регионам.

In [30]: `games_data['rating'] = games_data['rating'].fillna('NULL')`

Заполняю специально строковым значением нуля, так как все остальные значения так же представлены типом object

общие выводы по разделу

1. Стандартизировали названия столбцов
2. Проверили явные и неявные дубликаты и выполнили их удаление

3. Обработали пропуски, пропуски в оценках и ретинга оставили без изменений, так как корректное заполнение не представляется возможным.
4. Добавили дополнительный столбец "total_sales", в котором указали общее кол-во проданных копий по всем представленным странам.
5. По рекомендации проверяющего заполнила заглушками значения возрастных рейтингов.

```
In [31]: # Комментарий ревьюера
# Посмотрим, что осталось
temp = games_data.copy()
list_c = ['name', 'platform', 'year_of_release', 'genre', 'critic_score', 'user_
print(temp.info())
for col_l in list_c:
    print('-'* 25)
    print(col_l, temp[col_l].sort_values().unique())
    print(col_l, ': кол-во NaN', temp[col_l].isna().sum(),
          ', процент NaN', round(temp[col_l].isna().mean()*100,2), '%')
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 16443 entries, 0 to 16442
```

```
Data columns (total 12 columns):
```

| # | Column | Non-Null Count | Dtype |
|----|-----------------|----------------|---------|
| 0 | name | 16443 non-null | object |
| 1 | platform | 16443 non-null | object |
| 2 | year_of_release | 16443 non-null | Int64 |
| 3 | genre | 16443 non-null | object |
| 4 | NA_sales | 16443 non-null | float64 |
| 5 | EU_sales | 16443 non-null | float64 |
| 6 | JP_sales | 16443 non-null | float64 |
| 7 | other_sales | 16443 non-null | float64 |
| 8 | critic_score | 7982 non-null | Int64 |
| 9 | user_score | 7462 non-null | float64 |
| 10 | rating | 16443 non-null | object |
| 11 | total_sales | 16443 non-null | float64 |

```
dtypes: Int64(2), float64(6), object(4)
```

```
memory usage: 1.5+ MB
```

```
None
```

```
-----
name ['Beyblade Burst' 'Fire Emblem Fates' "Frozen: Olaf's Quest" ...
      'uDraw Studio' 'uDraw Studio: Instant Artist'
      '¡Shin Chan Flipa en colores!']
```

```
name : кол-во NaN 0 , процент NaN 0.0 %
```

```
-----
platform ['2600' '3DO' '3DS' 'DC' 'DS' 'GB' 'GBA' 'GC' 'GEN' 'GG' 'N64' 'NES' 'N
G'
          'PC' 'PCFX' 'PS' 'PS2' 'PS3' 'PS4' 'PSP' 'PSV' 'SAT' 'SCD' 'SNES' 'TG16'
          'WS' 'Wii' 'WiiU' 'X360' 'XB' 'XOne']
```

```
platform : кол-во NaN 0 , процент NaN 0.0 %
```

```
-----
year_of_release <IntegerArray>
```

```
[1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992,
 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,
 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016]
```

```
Length: 37, dtype: Int64
```

```
year_of_release : кол-во NaN 0 , процент NaN 0.0 %
```

```
-----
genre ['Action' 'Adventure' 'Fighting' 'Misc' 'Platform' 'Puzzle' 'Racing'
       'Role-Playing' 'Shooter' 'Simulation' 'Sports' 'Strategy']
```

```
genre : кол-во NaN 0 , процент NaN 0.0 %
```

```
-----
critic_score <IntegerArray>
```

```
[ 13,  17,  19,  20,  21,  23,  24,  25,  26,  27,  28,  29,  30,
  31,  32,  33,  34,  35,  36,  37,  38,  39,  40,  41,  42,  43,
  44,  45,  46,  47,  48,  49,  50,  51,  52,  53,  54,  55,  56,
  57,  58,  59,  60,  61,  62,  63,  64,  65,  66,  67,  68,  69,
  70,  71,  72,  73,  74,  75,  76,  77,  78,  79,  80,  81,  82,
  83,  84,  85,  86,  87,  88,  89,  90,  91,  92,  93,  94,  95,
  96,  97,  98, <NA>]
```

```
Length: 82, dtype: Int64
```

```
critic_score : кол-во NaN 8461 , процент NaN 51.46 %
```

```
-----
user_score [0.  0.2 0.3 0.5 0.6 0.7 0.9 1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9
2.  
```

```
2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.  3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8
3.9 4.  4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.  5.1 5.2 5.3 5.4 5.5 5.6
5.7 5.8 5.9 6.  6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7.  7.1 7.2 7.3 7.4
7.5 7.6 7.7 7.8 7.9 8.  8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.8 8.9 9.  9.1 9.2
```

```
9.3 9.4 9.5 9.6 9.7 nan]
user_score : кол-во NaN 8981 , процент NaN 54.62 %
-----
rating ['AO' 'E' 'E10+' 'EC' 'K-A' 'M' 'NULL' 'RP' 'T']
rating : кол-во NaN 0 , процент NaN 0.0 %
```

Исследовательский анализ данных

Этот раздел важен для достижения поставленных целей и аналитического ответа на озвученную задачу. Мы погружаемся в исследование представленных данных, анализируя их по ключевым параметрам с целью выявления закономерностей и взаимосвязей. В процессе этой тщательной работы мы стремимся к формулированию ясных, обоснованных рекомендаций для нашего заказчика, которые будут служить надежным ориентиром в принятии решений. Понимание нюансов данных позволит не только глубже осмыслить текущую ситуацию, но и предугадать возможные пути развития, спланировав стратегии, отвечающие на вызовы времени. В результате получится целостное видение, обрамлённое аналитической точностью и практической полезностью, что станет основой для эффективного и успешного функционирования бизнеса.

Анализ игр по годам релиза

```
In [32]: games_data.groupby('year_of_release')['year_of_release'].count().plot(kind='bar',
plt.xlabel('Год релиза')
plt.ylabel('Количество игр')
plt.xticks(rotation=45, ha='right')
plt.title('Общее число релизов игра по годам')
plt.show()
```



```
In [33]: games_data['year_of_release'].value_counts()
```

```
Out[33]: 2008    1427
          2009    1426
          2010    1255
          2007    1197
          2011    1136
          2006    1006
          2005     939
          2002     829
          2003     775
          2004     762
          2012     652
          2015     606
          2014     581
          2013     544
          2016     502
          2001     482
          1998     379
          2000     350
          1999     338
          1997     289
          1996     263
          1995     219
          1994     121
          1993      60
          1981      46
          1992      43
          1991      41
          1982      36
          1986      21
          1983      17
          1989      17
          1990      16
          1987      16
          1988      15
          1985      14
          1984      14
          1980       9
Name: year_of_release, dtype: Int64
```

Выводы по разделу

По представленным данным, 2008 год явился апогеем популярности, когда было выпущено рекордное количество разнообразных игр. Все игры, появившиеся на свет в XXI веке, по количеству превосходят произведения XX века, где 2000 год является частью последнего. С 1994 года наблюдается стремительный подъем, а до 2008 года продолжается активный рост. Затем, однако, следует явная динамика снижения.

В рамках данного датасета нет смысла анализировать всю доступную информацию за столь длительный период; целесообразнее сосредоточиться на тенденциях и закономерностях среди игр, выпущенных не менее чем за десятилетний отрезок, начиная с указанного срока. Далее информация утратит свою актуальность, поскольку со временем меняются запросы аудитории и развиваются технологии, которые совершенствуют визуальные

аспекты игр. Дальнейший анализ сможет более точно указать на подходящий временной отрезок, который будет наиболее показательным для изучения.

Проданные копии игр, упорядоченные по платформам их релиза

```
In [34]: print('Всего в данных представлено', len(games_data.groupby('platform')['total_s
top10_platform_data = games_data.groupby('platform')['total_sales'].sum().reset_
top10_platform_data.reset_index(drop=True))
```

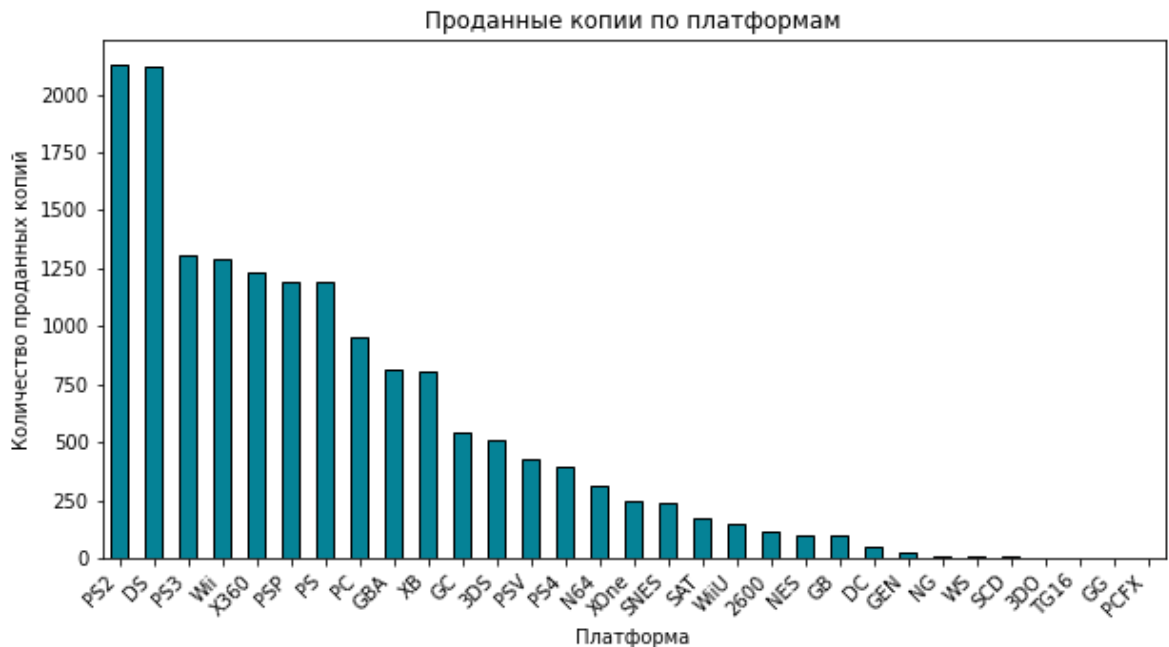
Всего в данных представлено 31 платформа

Выведем 10 с самой большой прибылью в течении всего срока:

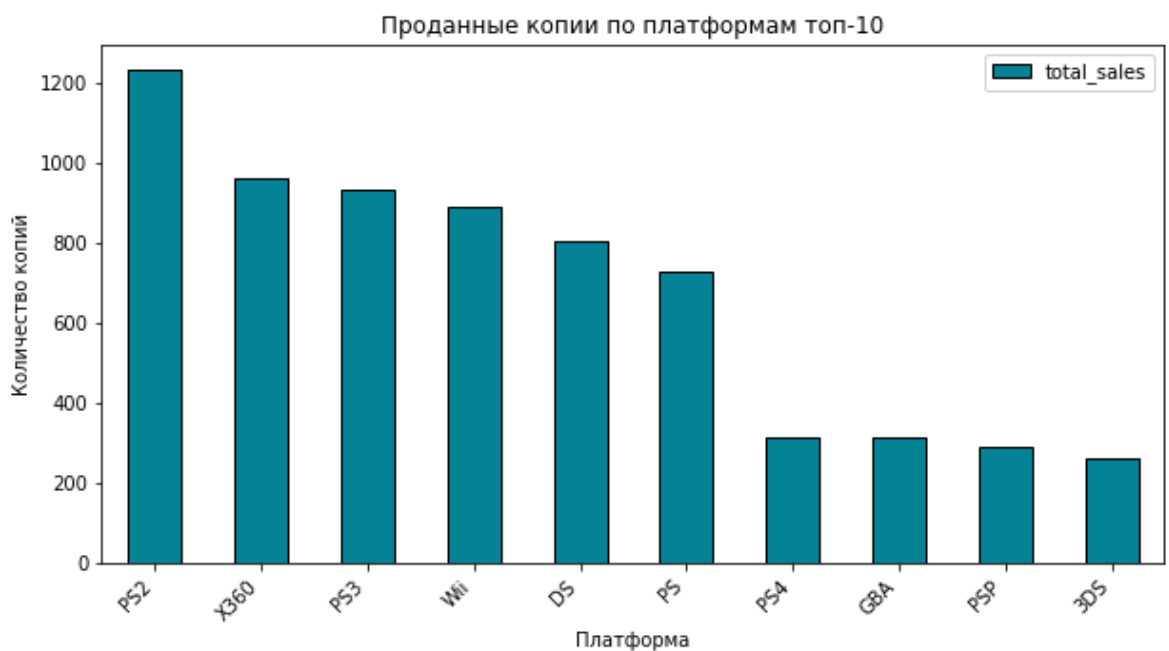
```
Out[34]:
```

| | platform | total_sales |
|---|----------|-------------|
| 0 | PS2 | 1233.56 |
| 1 | X360 | 961.24 |
| 2 | PS3 | 931.33 |
| 3 | Wii | 891.18 |
| 4 | DS | 802.78 |
| 5 | PS | 727.58 |
| 6 | PS4 | 314.14 |
| 7 | GBA | 312.88 |
| 8 | PSP | 289.53 |
| 9 | 3DS | 257.81 |

```
In [35]: games_data['platform'].value_counts().plot(kind='bar', figsize=(10,5), color='#0
plt.xlabel('Платформа')
plt.ylabel('Количество проданных копий')
plt.title('Проданные копии по платформам')
plt.xticks(rotation=45, ha='right')
plt.show()
```

```
In [36]: top10_platform_data.plot(kind='bar', figsize=(10,5), color='#088699', edgecolor=
plt.xlabel('Платформа')
plt.ylabel('Количество копий')
plt.title('Проданные копии по платформам топ-10')
plt.xticks(rotation=45, ha='right')
plt.show()
```



По выбранным платформам с наибольшими суммарными продажами построим распределение по годам.

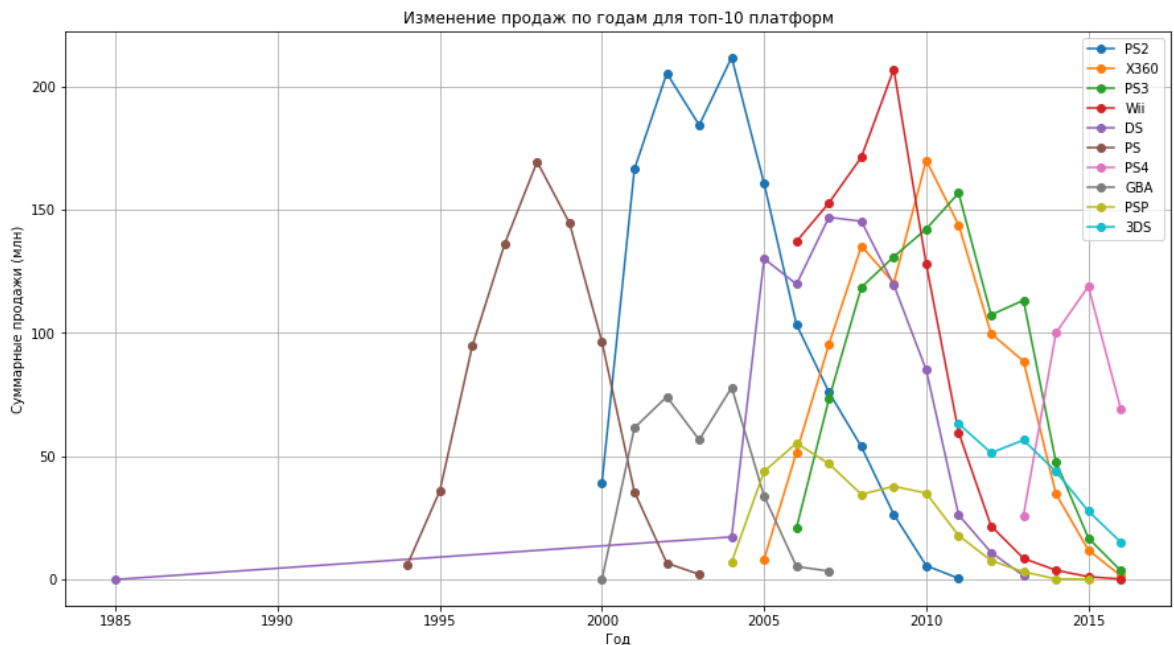
```
In [37]: platform_sales = games_data.groupby('platform')['total_sales'].sum()

top_platforms = platform_sales.nlargest(10).index

grouped_year_sales = games_data[games_data['platform'].isin(top_platforms)].groupby('year')
```

```
plt.figure(figsize=(15, 8))
for platform in top_platforms:
    plt.plot(grouped_year_sales.loc[platform].index, grouped_year_sales.loc[platform].values)

plt.xlabel('Год')
plt.ylabel('Суммарные продажи (млн)')
plt.title('Изменение продаж по годам для топ-10 платформ')
plt.legend()
plt.grid()
plt.show()
```



Платформа Nintendo DS была выпущена в 2004 г, представленный на графике 1985 г является ошибкой.

```
In [38]: games_data[(games_data['platform'] == 'DS') & (games_data['year_of_release'] ==
```

```
Out[38]:
```

| | name | platform | year_of_release | genre | NA_sales | EU_sales | JP_sales | other |
|-------|-------------------------------------|----------|-----------------|--------|----------|----------|----------|-------|
| 15704 | Strongest Tokyo University Shogi DS | DS | 1985 | Action | 0.0 | 0.0 | 0.02 | |



```
In [39]: games_data = games_data.drop(index=15703).reset_index(drop=True)
```

Выводы по разделу

1. За весь период указанный в датасете наибольшее кол-во проданных копий наблюдается у PS2 более 1200 млн копий
2. Выявили не обычный артефакт представленный в данных платформой DS, в записи которой был указ 1985 год релиза игры, данный артефакт было решено удалить.
3. Выбрали топ-10 платформ

Выбор актуального периода

Данный график хорошо отражает пики продаж самых популярных платформ среди представленных в датасете. По графику видно, что пики продаж чаще не превышают 5 лет; примерная длительность роста продаж платформы составляет от 3 до 6 лет, после чего начинается фаза снижения стоимости. Средняя длительность популярности (с первого появления до последнего) составляет примерно 10 лет, если не учитывать такие платформы, как "3DS" и "DS", так как на них после первых релизов наблюдается длительная пауза. Однако после второго релиза на данных платформах срок популярности не превышает 8 лет.

```
In [40]: # Комментарий ревьюера
temp = games_data.copy()
time_life = temp.pivot_table(index='platform', values='year_of_release', aggfunc=
time_life['life'] = time_life['max'] - time_life['min'] + 1 # в срок жизни платф
# поэтому +1
time_life['life'].median()
```

Out[40]: 7.0

Примерным сроком актуальности платформы равен 7 годам. Для дальнейшего анализа ограничу актуальный срок в 5 лет, так как для целей исследования нам необходимо спрогнозировать продажи и сформировать запрос на рекламу. По предварительному анализу длительность в 5 лет будет достаточной для определения платформ, которые находятся на пике, которые уже снижаются или, наоборот, возрастают в популярности. За этот срок сможем увидеть более полный профиль популярности платформ. Нам необходимо спрогнозировать наиболее популярные платформы, поэтому для нашего исследования актуальны будут те из них, которые набирают популярность, или уже находятся на пике.

```
In [41]: selected_platforms = games_data[games_data['year_of_release'] >= 2012]
```

```
In [42]: selected_platforms = selected_platforms.reset_index(drop=True)
```

```
In [43]: selected_platforms
```

Out[43]:

| | name | platform | year_of_release | genre | NA_sales | EU_sales | JP_sales | |
|------|-------------------------------|----------|-----------------|--------------|----------|----------|----------|--|
| 0 | Grand Theft Auto V | PS3 | 2013 | Action | 7.02 | 9.09 | 0.98 | |
| 1 | Grand Theft Auto V | X360 | 2013 | Action | 9.66 | 5.14 | 0.06 | |
| 2 | Call of Duty: Black Ops 3 | PS4 | 2015 | Shooter | 6.03 | 5.86 | 0.36 | |
| 3 | Pokemon X/Pokemon Y | 3DS | 2013 | Role-Playing | 5.28 | 4.19 | 4.35 | |
| 4 | Call of Duty: Black Ops II | PS3 | 2012 | Shooter | 4.99 | 5.73 | 0.65 | |
| ... | ... | ... | ... | ... | ... | ... | ... | |
| 2880 | Strawberry Nauts | PSV | 2016 | Adventure | 0.00 | 0.00 | 0.01 | |
| 2881 | Aiyoku no Eustia | PSV | 2014 | Misc | 0.00 | 0.00 | 0.01 | |
| 2882 | Samurai Warriors: Sanada Maru | PS3 | 2016 | Action | 0.00 | 0.00 | 0.01 | |
| 2883 | Haitaka no Psychedelica | PSV | 2016 | Adventure | 0.00 | 0.00 | 0.01 | |
| 2884 | Winning Post 8 2016 | PSV | 2016 | Simulation | 0.00 | 0.00 | 0.01 | |

2885 rows × 12 columns

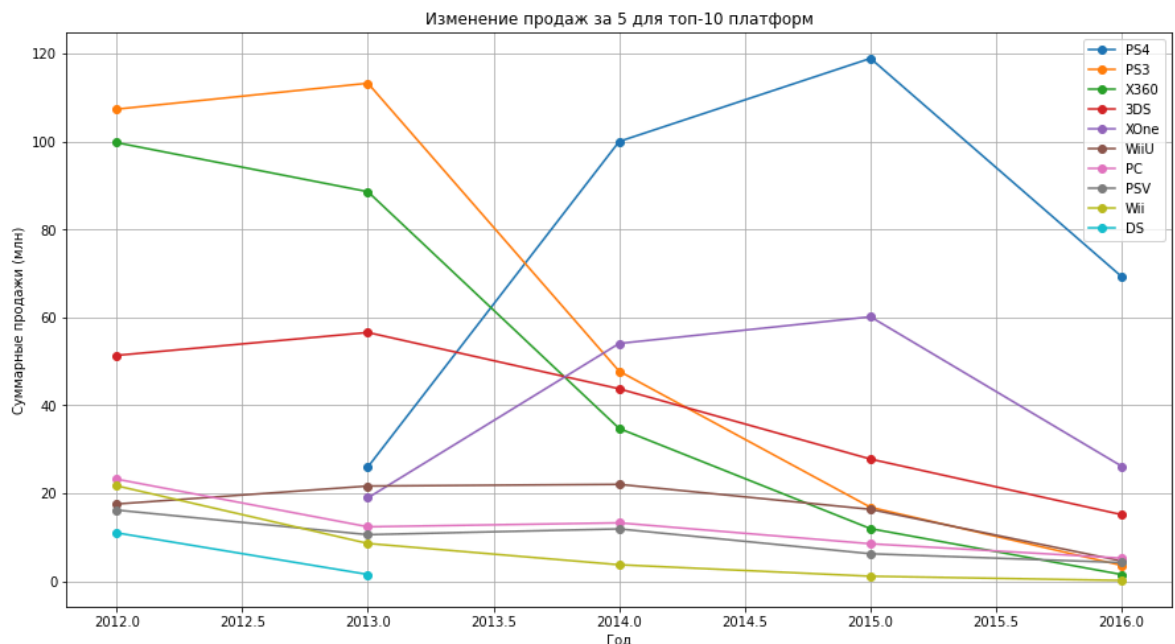


In [44]:

```
platform_sales = selected_platforms.groupby('platform')['total_sales'].sum()
top_platforms = platform_sales.nlargest(10).index
grouped_sales = selected_platforms[selected_platforms['platform'].isin(top_platforms)]

plt.figure(figsize=(15, 8))
for platform in top_platforms:
    plt.plot(grouped_sales.loc[platform].index, grouped_sales.loc[platform].values)

plt.xlabel('Год')
plt.ylabel('Суммарные продажи (млн)')
plt.title('Изменение продаж за 5 для топ-10 платформ')
plt.legend()
plt.grid()
plt.show()
```



По графику можно сделать такие выводы:

Платформы, начавшие набирать популярность после 2013 года, ещё примерно 3 года будут актуальны; Платформы, которые мы видим в начале графика, уже можно считать неактуальными или недостаточно актуальными, чтобы фокусировать внимание на них;

Платформы PS4 и XOne набирают популярность. По предыдущим найденным закономерностям можно предположить, что стоит сосредоточиться именно на этих платформах;

На 2016 год лидером продаж является PS4

Выводы по разделу:

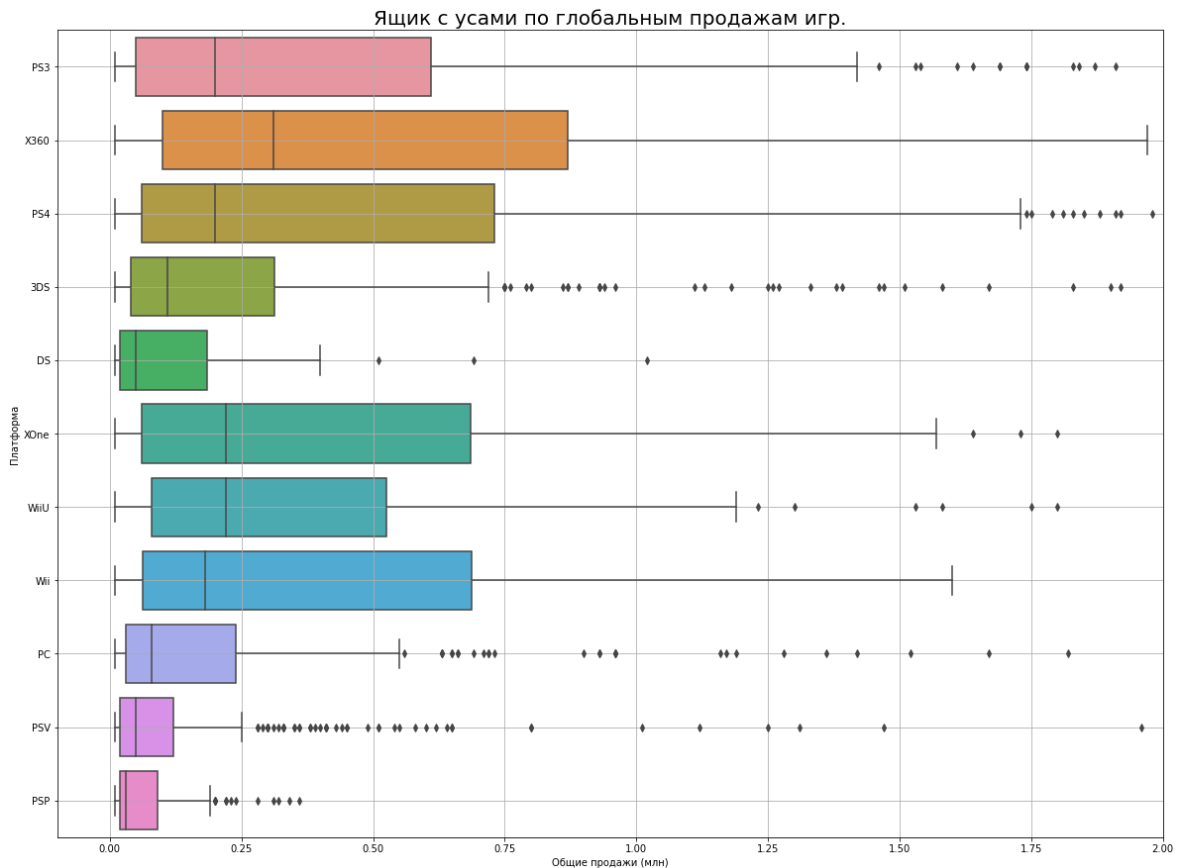
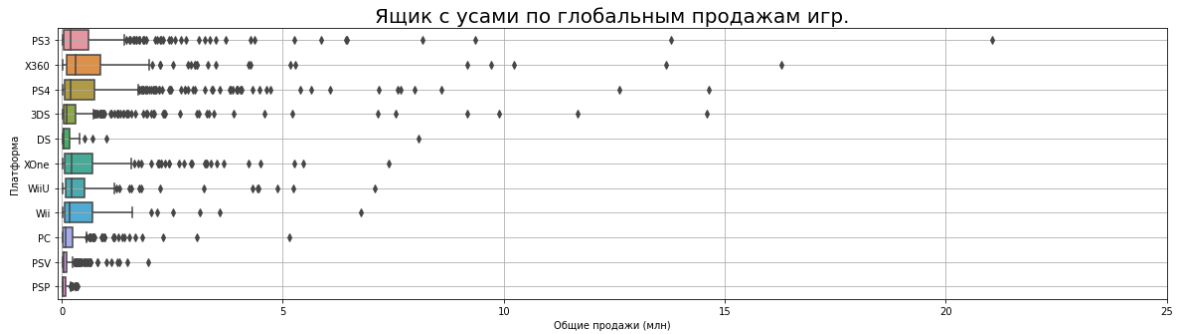
На этом этапе нашего исследования мы провели анализ по количеству проданных копий игр на различных платформах, что позволило нам установить временные рамки для дальнейшего анализа. Мы отобрали десять ведущих платформ, представленных в наших данных. С отфильтрованными данными теперь более целесообразно продолжить углубленный анализ, который позволит сформулировать более обоснованные и рациональные рекомендации.

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

```
In [45]: plt.figure(figsize=(20, 5))
plt.xlim(-0.1, 25)
plt.title('Ящик с усами по глобальным продажам игр.', fontsize=20)
sns.boxplot(x='total_sales', y='platform', data=selected_platforms, orient='h')
plt.xlabel('Общие продажи (млн)')
plt.ylabel('Платформа')
plt.grid(True);

plt.figure(figsize=(20, 15))
```

```
plt.xlim(-0.1, 2)
plt.title('Ящик с усами по глобальным продажам игр.', fontsize=20)
sns.boxplot(x='total_sales', y='platform', data=selected_platforms, orient='h')
plt.xlabel('Общие продажи (млн)')
plt.ylabel('Платформа')
plt.grid(True);
```



По графику выше мы видим, что лишь 4 платформы из выбранного диапазона времени превышали продажу в миллион копий за этот срок: это PS4, XOne, PS3 и X360. Это показывает размах "усов": он больше, чем у остальных, поэтому в них входит больше нормальных значений без отклонений. Все жанры, находящиеся за пределами, показывают аномалии и выбросы.

Влияние отзывов пользователей и критиков

Далее изучим, как отзывы пользователей и критиков влияют на продажи внутри одной популярной платформы. Построим диаграмму рассеяния и посчитаем корреляцию между отзывами и продажами.

```
In [46]: selected_platforms[['total_sales', 'user_score', 'critic_score']].corr()
```

```
Out[46]:
```

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | 0.004194 | 0.308633 |
| user_score | 0.004194 | 1.000000 | 0.518573 |
| critic_score | 0.308633 | 0.518573 | 1.000000 |

Среди всех выбранных платформ корреляции между продажами и оценками не выявлено; имеется очень слабая положительная корреляция между оценками критиков и продажами — 0,27. Связь слабая, поэтому говорить о её значимости не имеет смысла. Зато существует положительная связь между оценками критиков и пользователей. Корреляция положительная и выше среднего, её значение равно 0,57, что логично, ведь оценивают один и тот же продукт. Хорошему продукту ставят одинаково высокие оценки как пользователи, так и критики, а в случае не очень качественного продукта ситуация обстоит аналогично.

Посмотрим корреляцию на выбранной самой перспективной платформе на 2016 год имеющей наибольшее кол-во проданных копий то платформа PS4

```
In [47]: selected_platforms[selected_platforms['platform']=='PS4'][['total_sales', 'user_
```

```
Out[47]:
```

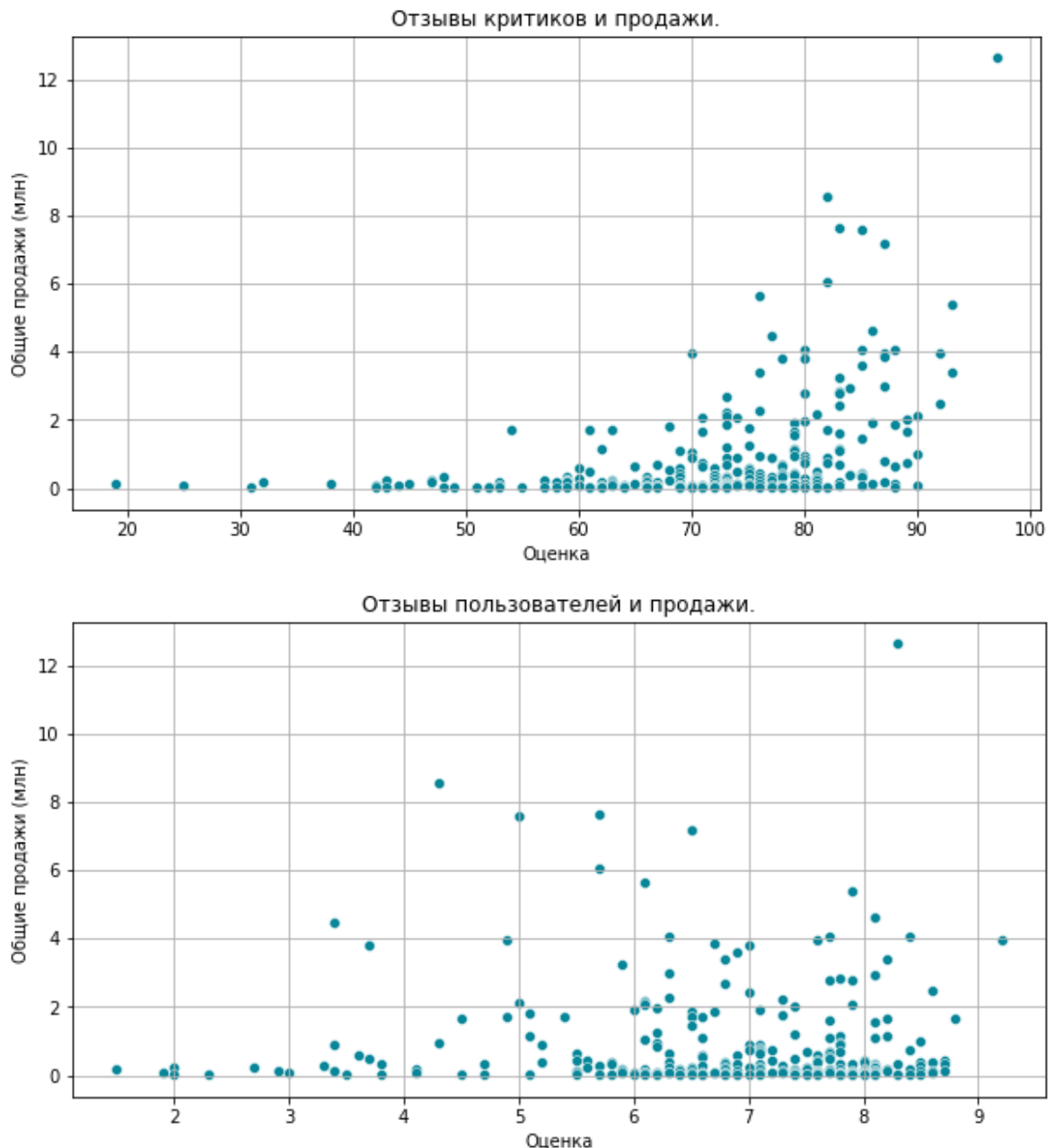
| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | -0.031957 | 0.406568 |
| user_score | -0.031957 | 1.000000 | 0.557654 |
| critic_score | 0.406568 | 0.557654 | 1.000000 |

На данной таблице можно увидеть значительную связь между общими продажами и оценками критиков — 0.43. Корреляция положительная и приближается к среднему значению 0.43, что выше, чем по всем платформам, однако связь остается слабой. Похожие результаты наблюдаются в корреляции между оценками пользователей и критиков, как показано в таблице выше.

```
In [48]: plt.figure(figsize=(10, 5))
plt.title('Отзывы критиков и продажи.')
sns.scatterplot(x='critic_score', y='total_sales', data=selected_platforms[selecte
plt.xlabel('Оценка')
plt.ylabel('Общие продажи (млн)')
plt.grid()
plt.show()

plt.figure(figsize=(10, 5))
plt.title('Отзывы пользователей и продажи.')
sns.scatterplot(x='user_score', y='total_sales', data=selected_platforms[selecte
plt.xlabel('Оценка')
plt.ylabel('Общие продажи (млн)')
```

```
plt.grid()
plt.show()
```



По диаграммам рассеивания мы видим, что постепенное увеличение от нуля к максимальной оценке с повышением до 5 млн лишь в редких случаях выбивается из общей тенденции. У пользователей этот разброс больше: на низких оценках 3-4 (что явно ниже среднего) мы наблюдаем выбросы с продажами более 2,5 млн копий, тогда как у критиков такой выброс можно увидеть после оценки 70 (что соответствует оценке выше среднего) с продажами, приближенными к 4 млн копий и даже превышающими это значение. В графике рассеивания оценок критиков мы видим больше порядка и постепенное увеличение; это может быть связано с тем, что критики оценивают продукт по объективным критериям, и поэтому их оценки более согласованы. Пользователи же не ограничены в своих возможностях оценивать и часто используют эмоциональный фактор при выставлении оценки игре, поэтому в графике рассеивания наблюдаем больше хаоса.

Посмотрим, насколько точно выводы подтверждаются на проверке других выбранных платформах.

Посмотрим корреляцию для ещё нескольких выбранных платформ, представленных в актуальном периоде:

```
In [49]: for name in selected_platforms['platform'].unique():
          if name == 'PS4':
              continue
          else:
              print(f'Посмотрим корреляцию для платформы {name}')

              print(selected_platforms[selected_platforms['platform']==name][['total_s
              print('-'*45)
```

Посмотрим корреляцию для платформы PS3

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | -0.006206 | 0.334152 |
| user_score | -0.006206 | 1.000000 | 0.544510 |
| critic_score | 0.334152 | 0.544510 | 1.000000 |

Посмотрим корреляцию для платформы X360

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | 0.006164 | 0.360573 |
| user_score | 0.006164 | 1.000000 | 0.557352 |
| critic_score | 0.360573 | 0.557352 | 1.000000 |

Посмотрим корреляцию для платформы 3DS

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | 0.197583 | 0.320803 |
| user_score | 0.197583 | 1.000000 | 0.722762 |
| critic_score | 0.320803 | 0.722762 | 1.000000 |

Посмотрим корреляцию для платформы DS

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | 0.882709 | NaN |
| user_score | 0.882709 | 1.000000 | NaN |
| critic_score | NaN | NaN | NaN |

Посмотрим корреляцию для платформы XOne

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | -0.068925 | 0.416998 |
| user_score | -0.068925 | 1.000000 | 0.472462 |
| critic_score | 0.416998 | 0.472462 | 1.000000 |

Посмотрим корреляцию для платформы WiiU

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | 0.400219 | 0.34838 |
| user_score | 0.400219 | 1.000000 | 0.77008 |
| critic_score | 0.348380 | 0.770080 | 1.00000 |

Посмотрим корреляцию для платформы Wii

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | 0.296197 | -0.424341 |
| user_score | 0.296197 | 1.000000 | 0.816295 |
| critic_score | -0.424341 | 0.816295 | 1.000000 |

Посмотрим корреляцию для платформы PC

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | -0.121867 | 0.237243 |
| user_score | -0.121867 | 1.000000 | 0.432587 |
| critic_score | 0.237243 | 0.432587 | 1.000000 |

Посмотрим корреляцию для платформы PSV

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | -0.004339 | 0.094488 |
| user_score | -0.004339 | 1.000000 | 0.699199 |
| critic_score | 0.094488 | 0.699199 | 1.000000 |

Посмотрим корреляцию для платформы PSP

| | total_sales | user_score | critic_score |
|--------------|-------------|------------|--------------|
| total_sales | 1.000000 | -0.802302 | NaN |
| user_score | -0.802302 | 1.000000 | NaN |
| critic_score | NaN | NaN | NaN |

```
In [50]: len(selected_platforms[(selected_platforms['platform']=='PSP') & (~selected_plat
```

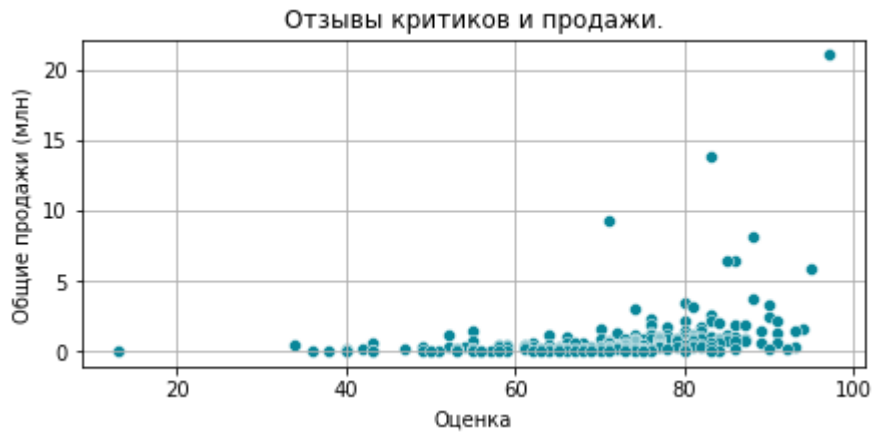
```
Out[50]: 1
```

```
In [51]: len(selected_platforms[(selected_platforms['platform']=='DS') & (~selected_platf
```

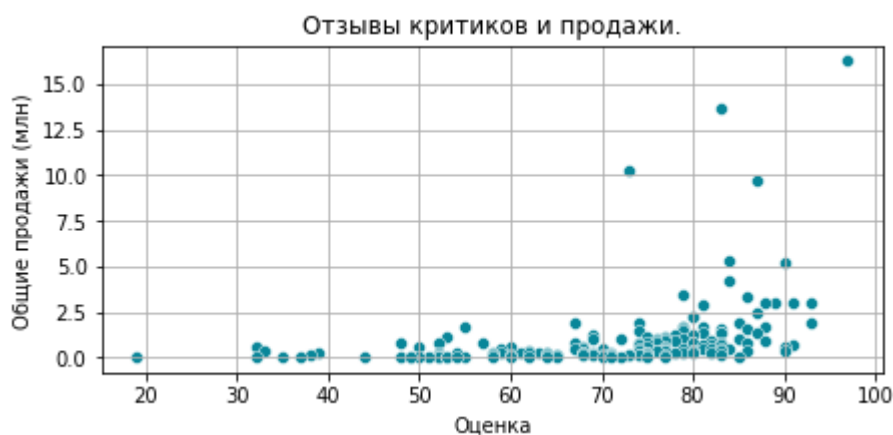
```
Out[51]: 1
```

```
In [52]: for name in selected_platforms['platform'].unique():
    if name == 'PS4':
        continue
    else:
        print(f'Посмотрим график рассеивания для платформы {name}')
        plt.figure(figsize=(7, 3))
        plt.title('Отзывы критиков и продажи.')
        sns.scatterplot(x='critic_score', y='total_sales', data=selected_platforms)
        plt.xlabel('Оценка')
        plt.ylabel('Общие продажи (млн)')
        plt.grid()
        plt.show()
```

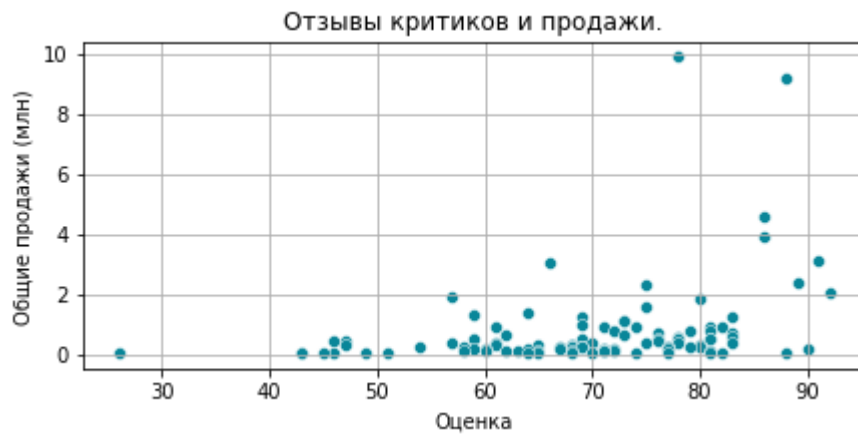
Посмотрим график рассеивания для платформы PS3



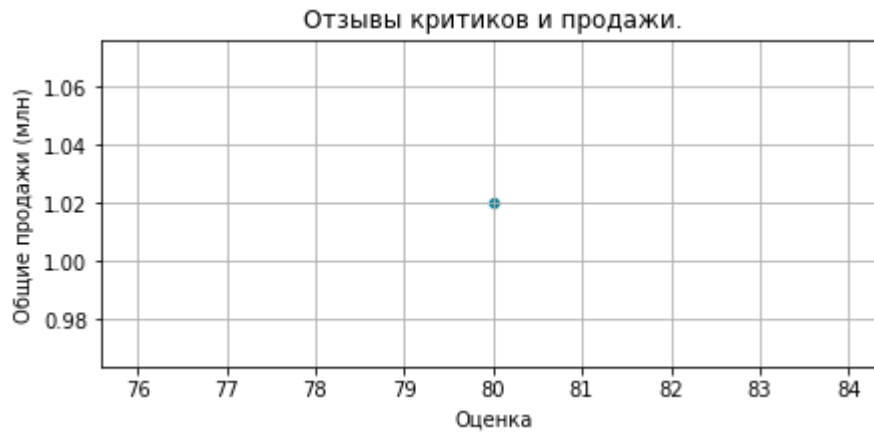
Посмотрим график рассеивания для платформы X360



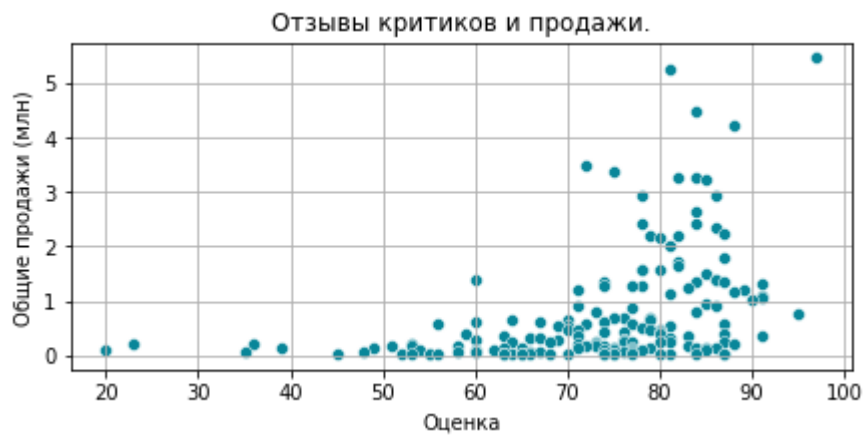
Посмотрим график рассеивания для платформы 3DS



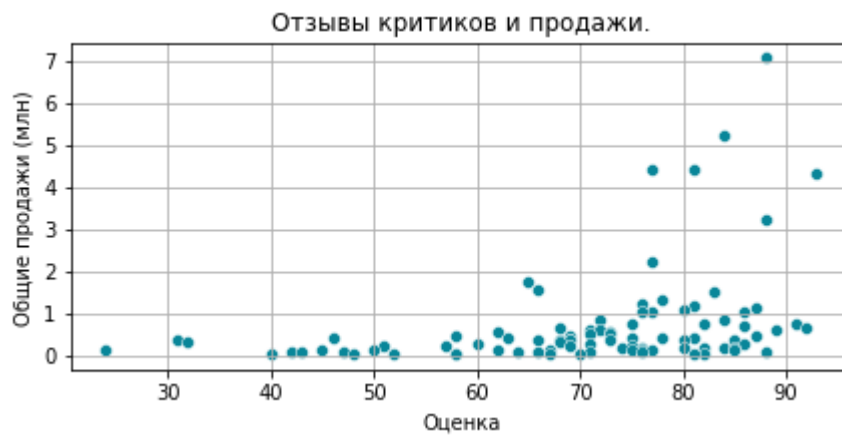
Посмотрим график рассеивания для платформы DS



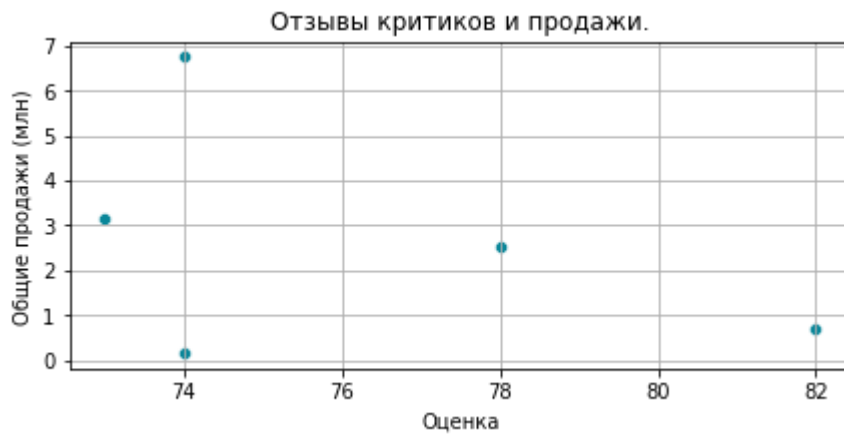
Посмотрим график рассеивания для платформы XOne



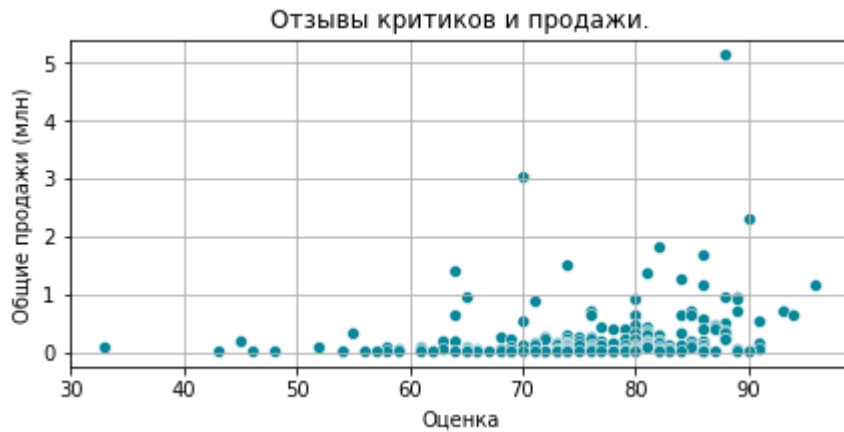
Посмотрим график рассеивания для платформы WiiU



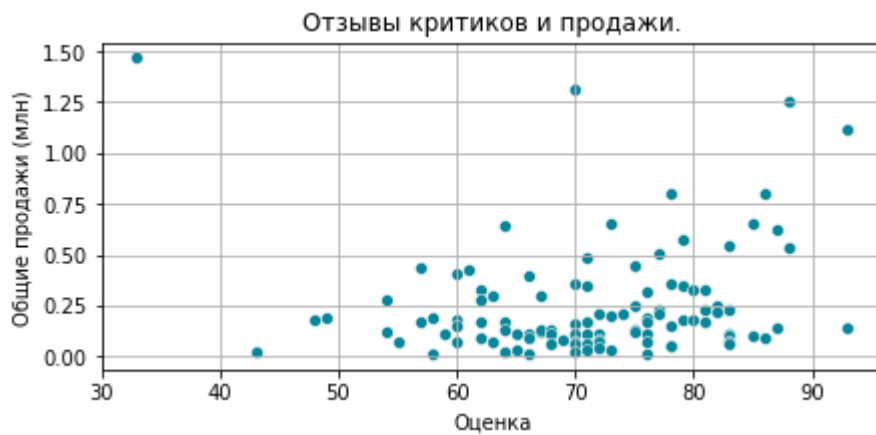
Посмотрим график рассеивания для платформы Wii



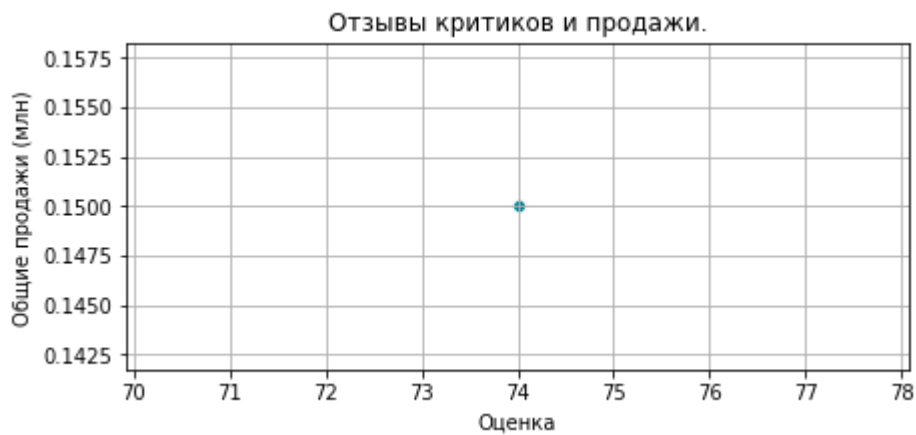
Посмотрим график рассеивания для платформы PC



Посмотрим график рассеивания для платформы PSV



Посмотрим график рассеивания для платформы PSP



```
In [53]: for name in selected_platforms['platform'].unique():
          if name == 'PS4':
```

```

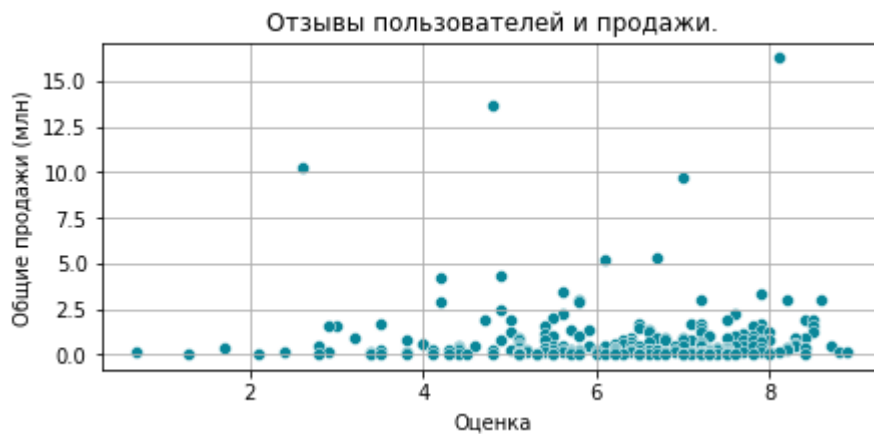
        continue
    else:
        print(f'Посмотрим график рассеивания для платформы {name}')
        plt.figure(figsize=(7, 3))
        plt.title('Отзывы пользователей и продажи.')
        sns.scatterplot(x='user_score', y='total_sales', data=selected_platforms)
        plt.xlabel('Оценка')
        plt.ylabel('Общие продажи (млн)')
        plt.grid()
        plt.show()

```

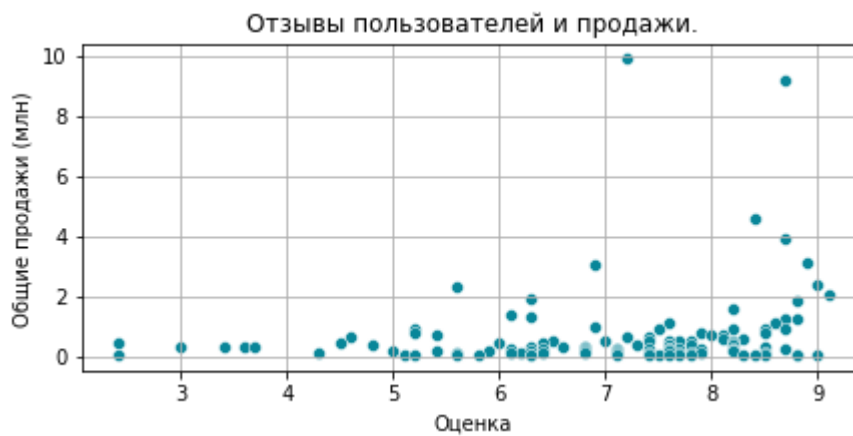
Посмотрим график рассеивания для платформы PS3



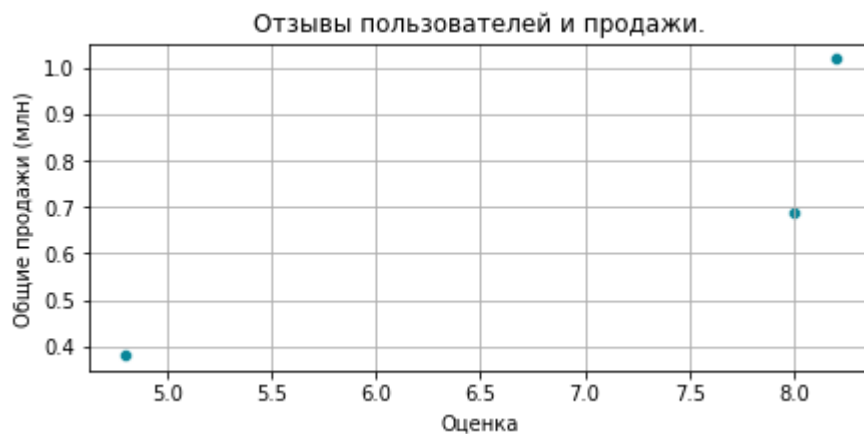
Посмотрим график рассеивания для платформы X360



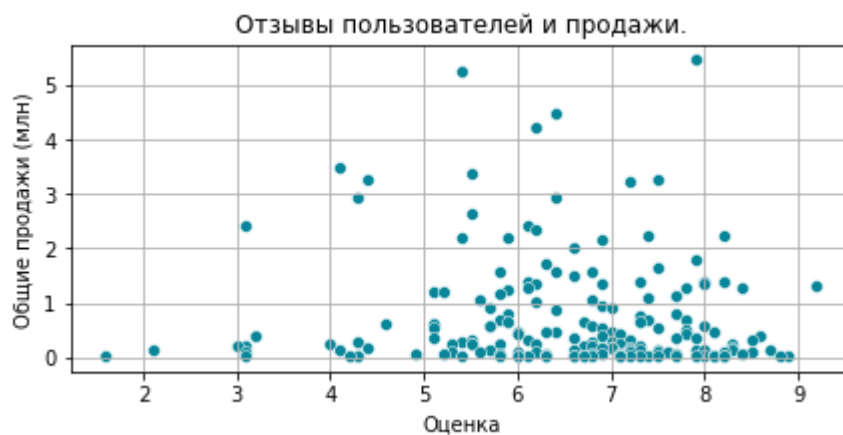
Посмотрим график рассеивания для платформы 3DS



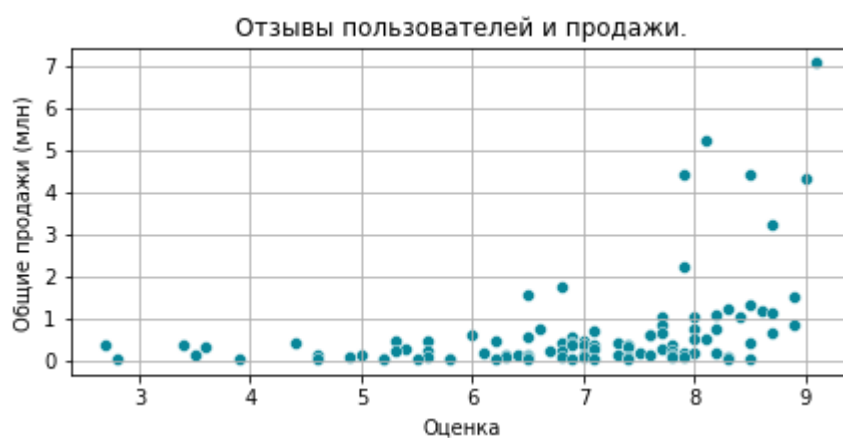
Посмотрим график рассеивания для платформы DS



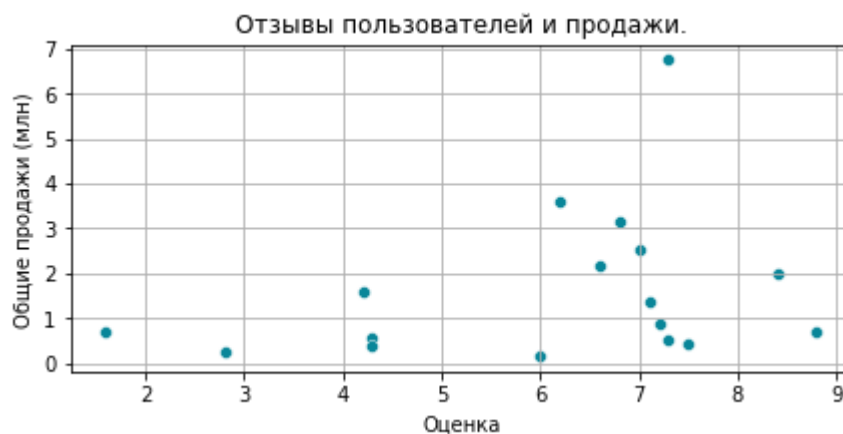
Посмотрим график рассеивания для платформы XOne



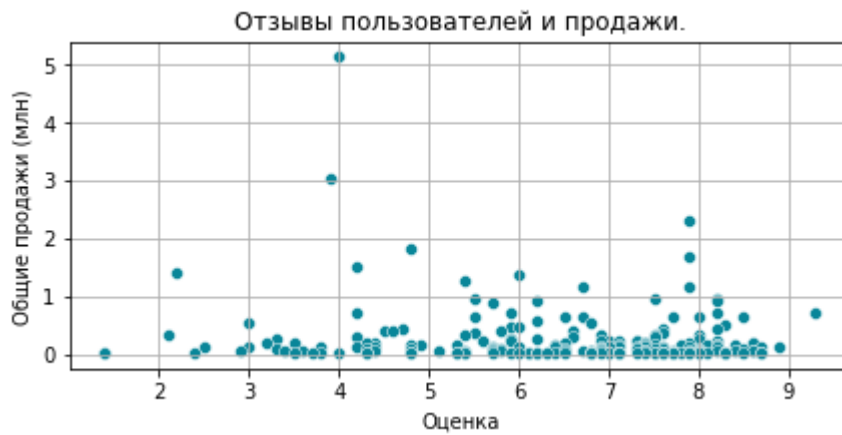
Посмотрим график рассеивания для платформы WiiU



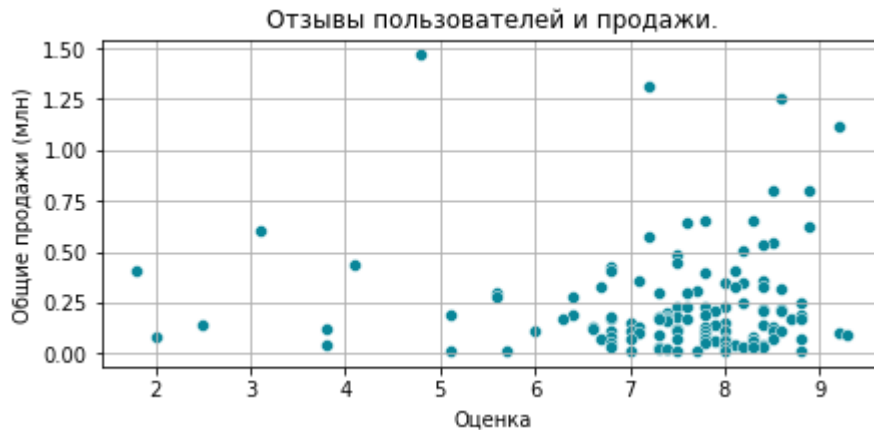
Посмотрим график рассеивания для платформы Wii



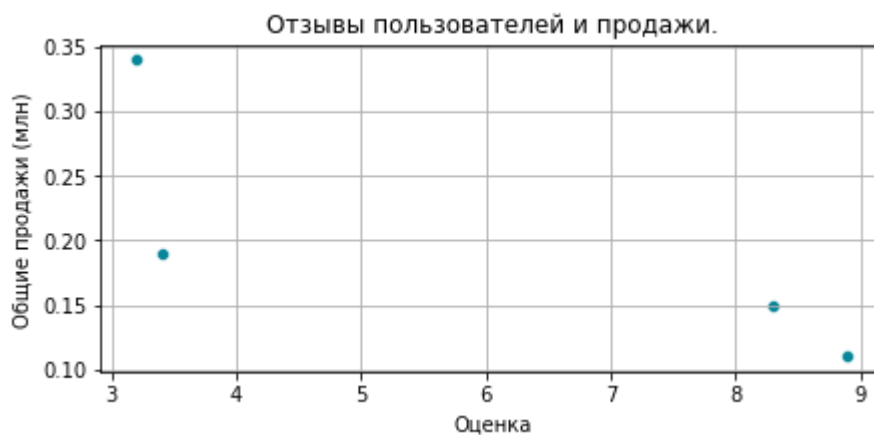
Посмотрим график рассеивания для платформы PC



Посмотрим график рассеивания для платформы PSV

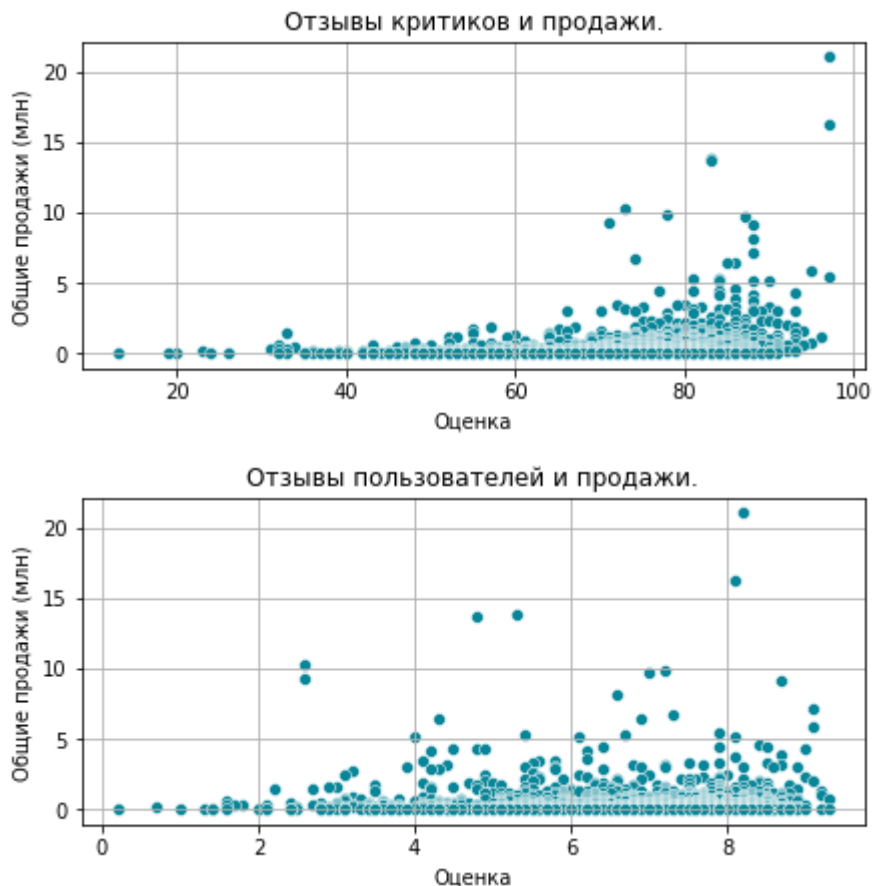


Посмотрим график рассеивания для платформы PSP



```
In [54]: plt.figure(figsize=(7, 3))
plt.title('Отзывы критиков и продажи.')
sns.scatterplot(x='critic_score', y='total_sales', data=selected_platforms[select
plt.xlabel('Оценка')
plt.ylabel('Общие продажи (млн)')
plt.grid()
plt.show()

plt.figure(figsize=(7, 3))
plt.title('Отзывы пользователей и продажи.')
sns.scatterplot(x='user_score', y='total_sales', data=selected_platforms[selecte
plt.xlabel('Оценка')
plt.ylabel('Общие продажи (млн)')
plt.grid()
plt.show()
```

Выводы по разделу:

На таблице корреляции прослеживается, что взаимосвязь между объемами продаж и отзывами в большинстве случаев отсутствует. Лишь одна заметная связь выделяется между продажами платформы DS и пользовательскими оценками. Однако критические отзывы в этих данных не представлены, поскольку рассматривается всего одна игра с рецензией критика на данной платформе — такая же ситуация наблюдается и у PSP. В целом, для большинства платформ наблюдается только слабая или средняя корреляция между оценками критиков и отзывами пользователей. Исключение составляют лишь платформы WiiU и Wii, где мы можем заметить сильную связь между мнениями критиков и пользователей.

Дальнейшее построение графиков рассеивания объема продаж и оценок пользователей и критиков, раскрывают более подробно результаты корреляции:

- По графикам рассеивания можно увидеть, что показатели корреляции для платформ DS и PSP нельзя считать показательными, тк представлено всего 3 отзыва пользователей. Платформы Wii представлено очень мало игр, в которых указаны оценки от критиков (5 игр) и пользователей (17)
- Исключая из анализа рассеивание платформ, в которых незначительное кол-во отзывов критиков или пользователей, можно сделать вывод, что выбросы с большими продажами и низкими оценками у пользователей, а также более постепенный и ровный рост у критиков с выбросами после оценок выше среднего, что

подтверждает объективность оценок критиков. Можно предположить, что оценкам критиков, при их достаточном количестве стоит доверять больше, чем оуенкам от пользователей.**

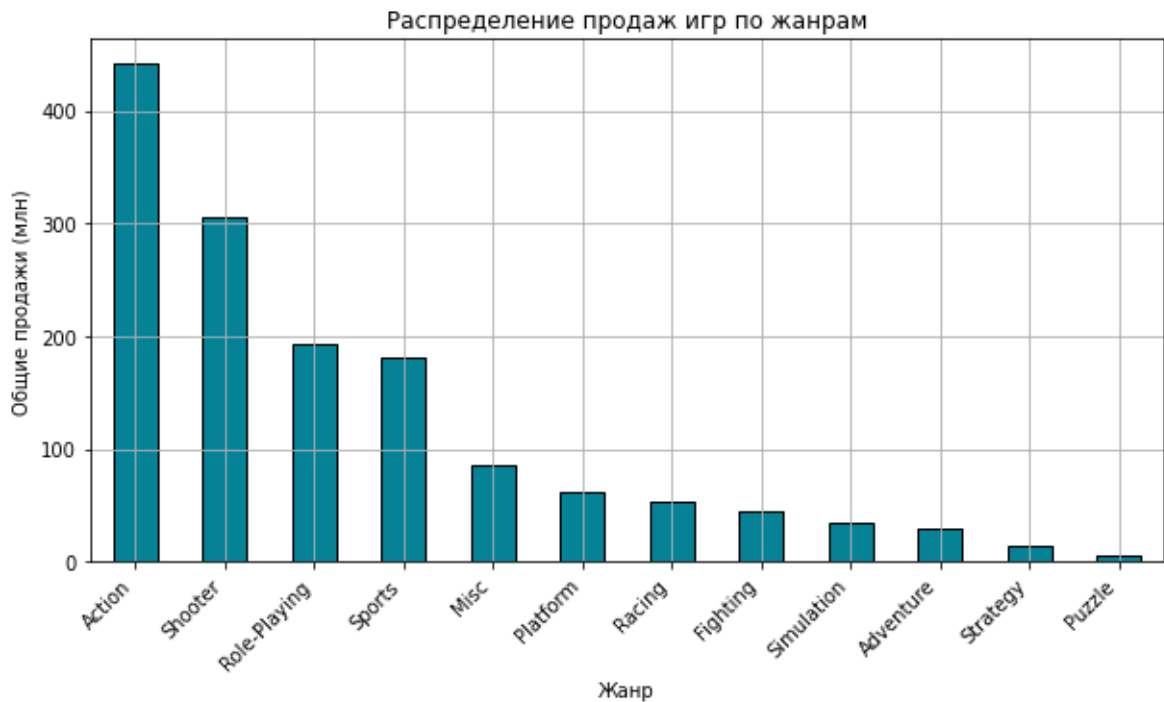
Анализ игр по жанрам

```
In [55]: plt.figure(figsize=(10, 5))
plt.title('Распределение медианы продаж игр по жанрам')
selected_platforms.groupby('genre')['total_sales'].median().plot(kind='bar', edge
plt.xlabel('Жанр')
plt.ylabel('Общее кол-во игр')
plt.xticks(rotation=45, ha='right')
plt.grid()
plt.show()
```



С большим отрывом по кол-ву представленных игр в датасете лидируют игры представленные в жанре шутера, на втором месте спортивные игры, закрывает тройку лидеров игры-платформеры. Наименьшее медианное значение имеют игры-приключения

```
In [56]: plt.figure(figsize=(10, 5))
plt.title('Распределение продаж игр по жанрам')
selected_platforms.groupby('genre')['total_sales'].sum().sort_values(ascending=F
plt.xlabel('Жанр')
plt.ylabel('Общие продажи (млн)')
plt.xticks(rotation=45, ha='right')
plt.grid()
plt.show()
```

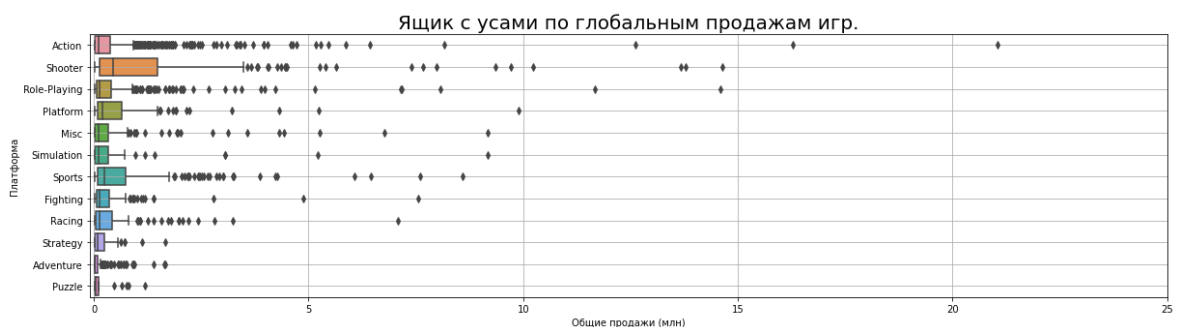


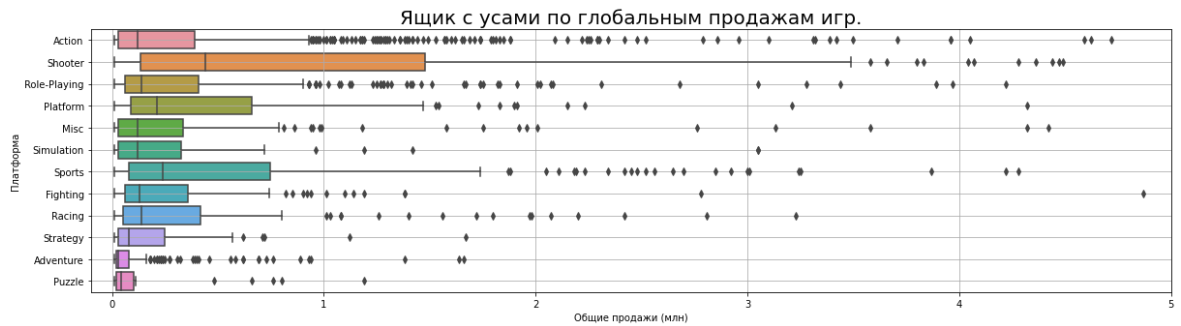
В распределении сумм продаж по жанрам лидирует экшен, данный жанр представлен примерно 30% игр в выборке. После него с количеством более 300 млн копий располагаются шутеры. Аутсайдерами текущего рейтинга по общей сумме продаж становятся пазлы.

Более явно распределение и разброс мы увидим на графике ящик с усами.

```
In [57]: plt.figure(figsize=(20, 5))
plt.xlim(-0.1, 25)
plt.title('Ящик с усами по глобальным продажам игр.', fontsize=20)
sns.boxplot(x='total_sales', y='genre', data=selected_platforms, orient='h')
plt.xlabel('Общие продажи (млн)')
plt.ylabel('Платформа')
plt.grid(True);

plt.figure(figsize=(20, 5))
plt.xlim(-0.1, 5)
plt.title('Ящик с усами по глобальным продажам игр.', fontsize=20)
sns.boxplot(x='total_sales', y='genre', data=selected_platforms, orient='h')
plt.xlabel('Общие продажи (млн)')
plt.ylabel('Платформа')
plt.grid(True);
```

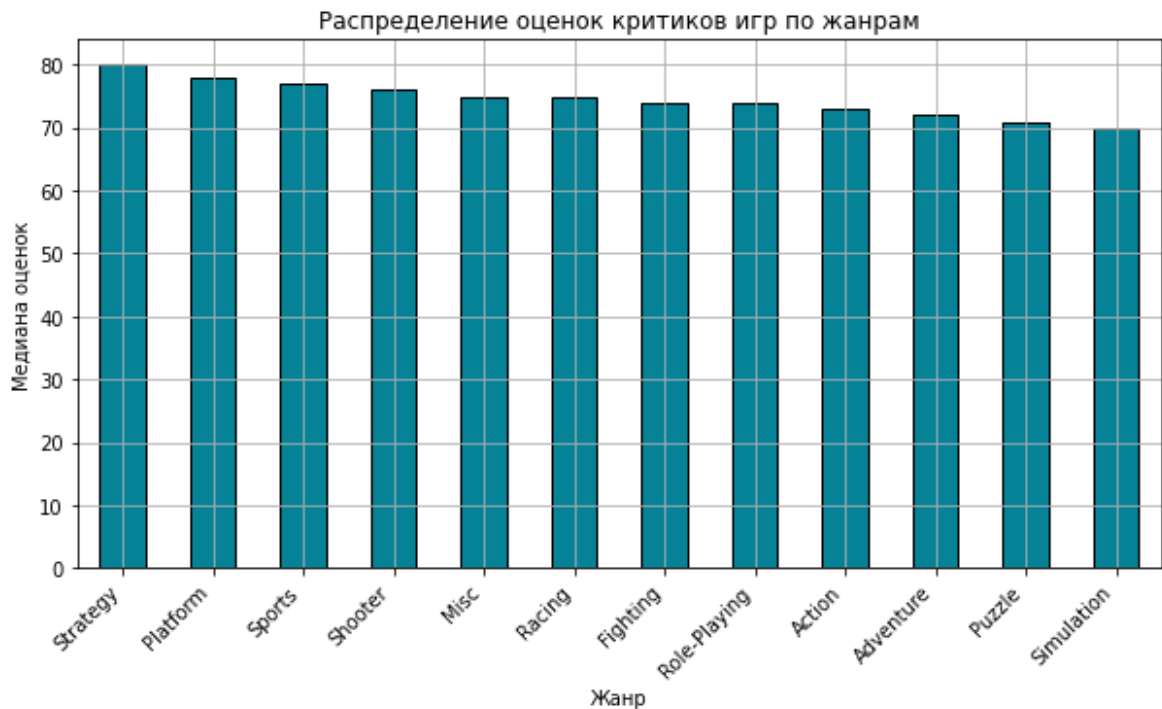




```
In [58]: plt.figure(figsize=(10, 5))
plt.title('Распределение оценок пользователей игр по жанрам')
selected_platforms.groupby('genre')['user_score'].median().sort_values(ascending=True)
plt.xlabel('Жанр')
plt.ylabel('Медиана оценок')
plt.xticks(rotation=45, ha='right')
plt.grid()
plt.show()
```



```
In [59]: plt.figure(figsize=(10, 5))
plt.title('Распределение оценок критиков игр по жанрам')
selected_platforms.groupby('genre')['critic_score'].median().sort_values(ascending=True)
plt.xlabel('Жанр')
plt.ylabel('Медиана оценок')
plt.xticks(rotation=45, ha='right')
plt.grid()
plt.show()
```

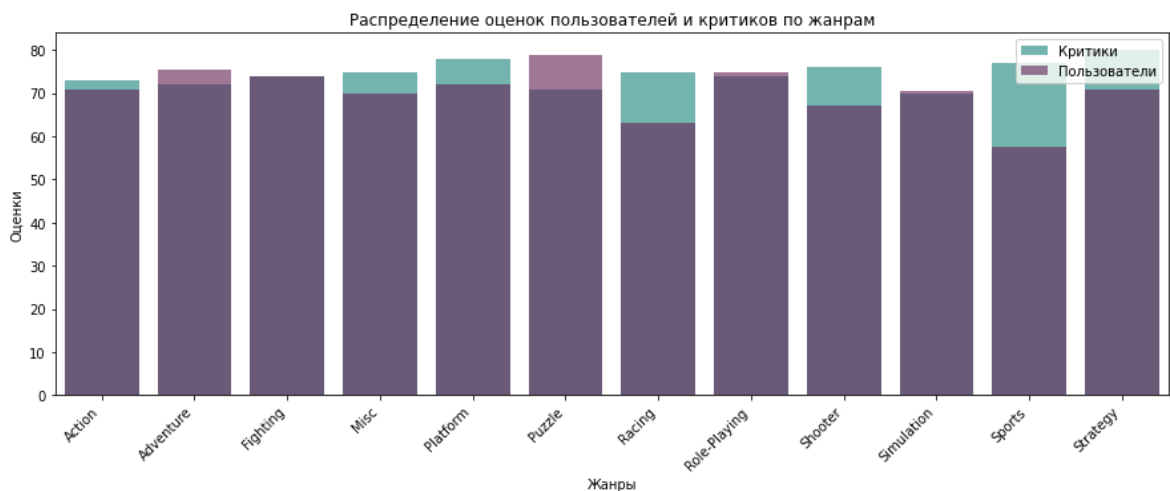


Удобнее рассмотреть и сравнить данные распределения будет на совмещенном графике оценок критиков и пользователей:

```
In [60]: plt.figure(figsize=(15, 5))

critic_score = selected_platforms.groupby('genre')['critic_score'].median()
user_score = selected_platforms.groupby('genre')['user_score'].median()

sns.barplot(y=critic_score, x=critic_score.index, label='Критики', color='#0c9688')
sns.barplot(y=user_score*10, x=user_score.index, label='Пользователи', color='#6a3d9a')
plt.xlabel('Жанры')
plt.ylabel('Оценки')
plt.title('Распределение оценок пользователей и критиков по жанрам')
plt.xticks(rotation=45, ha='right')
plt.legend()
plt.show()
```



В графике выше за счет наложения распределений оценок пользователей (приведены к размерности оценок критиков, умножив значение на 10) и критиков мы можем увидеть отличия в их распределении. У пользователей по рейтингу

лидирует жанр приключений, тогда как у критиков лидирует платформер. В отстающих – спорт у пользователей и симуляторы у критиков соответственно.

Общие выводы по разделу

- Мы видим различия в оценках почти по всем жанрам, но самые значительные наблюдаются у приключений и спорта, а также у шутеров и гонок.
- По оценкам, мы не можем выделить явных лидеров и аутсайдеров среди прошлых распределений, так как они смещаются по графикам, и заметить какую-либо закономерность с продажами и оценками в жанрах не удаётся.
- Самым продаваемым можно считать игры в жанре шутер.

Составим портрет пользователя каждого региона

Для более точного создания портрета пользователя и выработки рекомендаций крайне необходимо учесть региональную специфику распространения игры, а также запросы и предпочтения её аудитории. Опираясь на уже имеющиеся данные, мы сможем углубить наше понимание их интересов, что позволит более эффективно интерпретировать полученные результаты. Выстраивая анализ на основе этих критериев, мы не только уточним детали представления о целевой аудитории, но и сможем четко сформулировать окончательные выводы и рекомендации, касающиеся рекламной стратегии. Такой подход обеспечит глубокую аналитику, благодаря которой реклама станет более целенаправленной и актуальной, способствуя устойчивому развитию бизнеса и повышению продаж.

Доли продаж по регионам

Определим доли продаж в представленных регионах, чтобы выявить самый обширный рынок среди наших данных.

```
In [61]: plt.figure(figsize=(15, 5))
plt.pie(selected_platforms[['NA_sales', 'EU_sales', 'JP_sales']].sum(), labels=['
plt.axis('equal')
plt.title('Доли продаж по странам')
plt.show()
```



Самым большим рынком по количеству проданных копий является Северная Америка.

Доли продаж в регионах в зависимости от платформы.

Для того чтобы выявить наиболее востребованные и продаваемые игры на различных платформах в зависимости от регионов, имеет смысл обратиться к долям продаж. Эффективным способом будет распределение данных по каждому региону с их последующей системой организации в зависимости от выпускаемой платформы. Такой подход позволит не только охватить широкий спектр информативных показателей, но и выделить тенденции и предпочтения, присущие каждому специфическому рынку. Статистический анализ показывает, как каждая платформа завоевывает лояльность аудитории, выявляя наибольшее внимание к конкретным жанрам и играм. Фокусируясь на этих аспектах, мы сможем глубже понять динамику игрового рынка и его многообразие, что, в свою очередь, откроет новые горизонты для дальнейших исследований и стратегического планирования.

```
In [62]: top_by_country = selected_platforms.groupby('platform')[['NA_sales', 'EU_sales'],  
top_by_country
```

Out[62]:

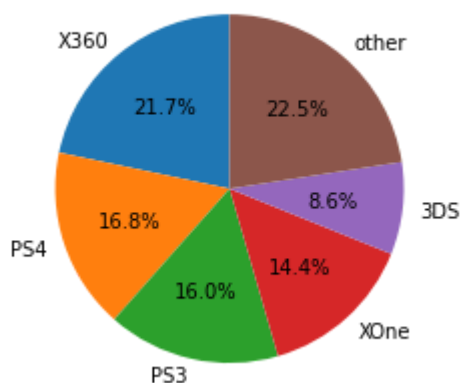
| | NA_sales | EU_sales | JP_sales |
|----------|----------|----------|----------|
| platform | | | |
| 3DS | 55.31 | 42.64 | 87.79 |
| DS | 4.59 | 3.53 | 3.72 |
| PC | 19.12 | 37.76 | 0.00 |
| PS3 | 103.38 | 106.85 | 35.29 |
| PS4 | 108.74 | 141.09 | 15.96 |
| PSP | 0.13 | 0.42 | 10.47 |
| PSV | 10.98 | 11.36 | 21.04 |
| Wii | 17.45 | 11.92 | 3.39 |
| WiiU | 38.10 | 25.13 | 13.01 |
| X360 | 140.05 | 74.52 | 1.57 |
| XOne | 93.12 | 51.59 | 0.34 |

In [82]: *# Функция для построения "пирога" долей стран по разным параметрам.*

```
def build_a_pie(data):  
    top5 = data.sort_values(ascending=False).head(5)  
    other = sum(data.sort_values(ascending=False).tail(7))  
    pie_data = top5.append(pd.Series({'other': other}))  
    return pie_data.values, pie_data.index
```

```
In [89]: data_NA, labels_NA = build_a_pie(top_by_country['NA_sales'])  
plt.pie(data_NA, labels=labels_NA, autopct='%1.1f%%', startangle=90)  
plt.title('Доли продаж платформ (NA)')  
plt.show()
```

Доли продаж платформ (NA)



```
In [84]: fig, axs = plt.subplots(1, 3, figsize=(12, 4))  
  
data_NA, labels_NA = build_a_pie(top_by_country['NA_sales'])  
axs[0].pie(data_NA, labels=labels_NA, autopct='%1.1f%%', startangle=90)  
axs[0].set_title('Доли продаж платформ (NA)')  
  
data_EU, labels_EU = build_a_pie(top_by_country['EU_sales'])  
axs[1].pie(data_EU, labels=labels_EU, autopct='%1.1f%%', startangle=90)
```



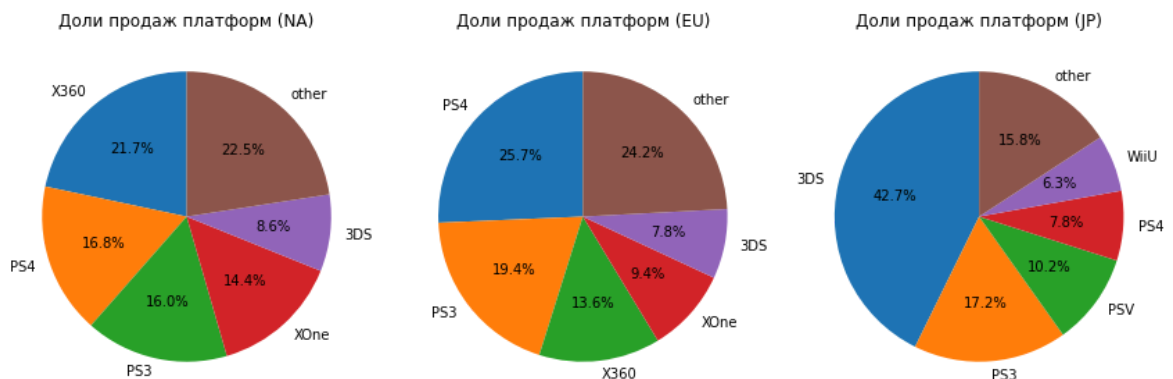
```

axs[1].set_title('Доли продаж платформ (EU)')

data_JP, labels_JP = build_a_pie(top_by_country['JP_sales'])
axs[2].pie(data_JP, labels=labels_JP, autopct='%1.1f%%', startangle=90)
axs[2].set_title('Доли продаж платформ (JP)')

plt.tight_layout()

```



По графикам выше мы видим, что общие и частные данные о продажах игр по платформам преобладают в Северной Америке. Ей принадлежит больше половины проданных копий из представленного датасета. В Северной Америке самой популярной платформой является X360, на неё приходится 22% проданных копий от общего числа продаж в Северной Америке. Для Европы самой популярной платформой за выбранный период стала PS4, а в Японии лидером рейтинга становится платформа 3DS. Единственная платформа, которая вошла в рейтинг по продажам во всех странах, это PS3. Однако, исходя из выявленных данных выше, мы понимаем, что данный период пришёлся на максимальный пик продаж, и далее, по найденным закономерностям, ожидается снижение продаж на данной платформе в пользу более современных.

Доли продаж в регионах в зависимости от жанра игры.

Для выявления наиболее востребованного и продаваемого жанра по регионам целесообразно обратиться к долям продаж, распределяя данные по каждому региону и систематизируя их в зависимости от жанра игры.

```

In [66]: top_by_genre = selected_platforms.groupby('genre')[['NA_sales', 'EU_sales', 'JP_s
top_by_genre

```

Out[66]:

| | NA_sales | EU_sales | JP_sales |
|---------------------|----------|----------|----------|
| genre | | | |
| Action | 177.84 | 159.34 | 52.80 |
| Adventure | 8.92 | 9.46 | 8.24 |
| Fighting | 19.79 | 10.79 | 9.44 |
| Misc | 38.19 | 26.32 | 12.86 |
| Platform | 25.38 | 21.41 | 8.63 |
| Puzzle | 1.13 | 1.40 | 2.14 |
| Racing | 17.22 | 27.29 | 2.50 |
| Role-Playing | 64.00 | 48.53 | 65.44 |
| Shooter | 144.77 | 113.47 | 9.23 |
| Simulation | 7.97 | 14.55 | 10.41 |
| Sports | 81.53 | 69.08 | 8.01 |
| Strategy | 4.23 | 5.17 | 2.88 |

In [67]:

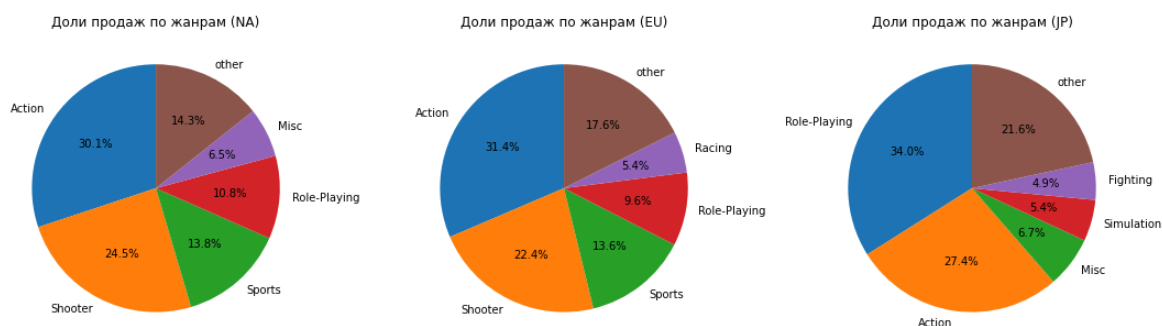
```
fig, axs = plt.subplots(1, 3, figsize=(15, 6))

# Строим графики на разных осях
data_NA, labels_NA = build_a_pie(top_by_genre['NA_sales'])
axs[0].pie(data_NA, labels=labels_NA, autopct='%1.1f%%', startangle=90)
axs[0].set_title('Доли продаж по жанрам (NA)')

data_EU, labels_EU = build_a_pie(top_by_genre['EU_sales'])
axs[1].pie(data_EU, labels=labels_EU, autopct='%1.1f%%', startangle=90)
axs[1].set_title('Доли продаж по жанрам (EU)')

data_JP, labels_JP = build_a_pie(top_by_genre['JP_sales'])
axs[2].pie(data_JP, labels=labels_JP, autopct='%1.1f%%', startangle=90)
axs[2].set_title('Доли продаж по жанрам (JP)')

plt.tight_layout()
```



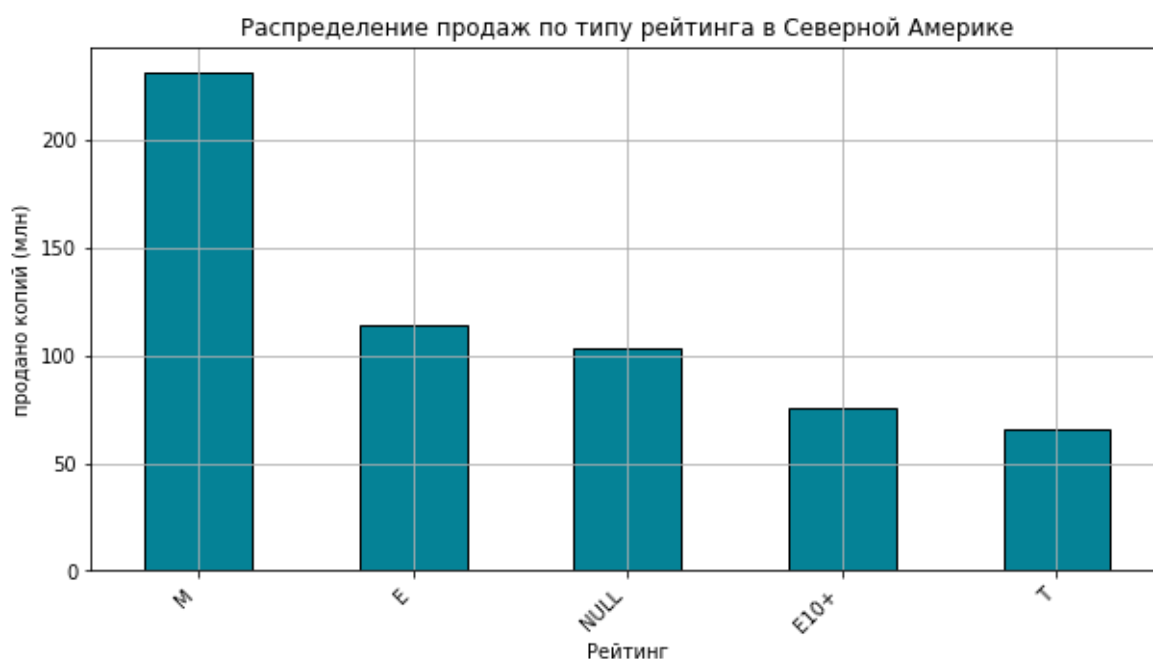
В Европе и Северной Америке лидирует по продажам жанр экшен, как и в остальных пунктах рейтинга. Северная Америка и Европа одинаково популярны в жанрах игр. Для Японии в лидерах находятся ролевые игры.

Такие жанры, как экшен, ролевые игры, миссии и спортивные игры, вошли в рейтинги всех стран по продажам.

Доли продаж в регионах в зависимости от типа рейтинга.

При выходе продукции на рынок необходимо учитывать множество факторов, чтобы обеспечить успешный старт продаж. В игровой индустрии одним из наиболее значимых аспектов становится возрастной рейтинг игр. Например, в зависимости от региона, игры, ориентированные на детскую или взрослую аудиторию, могут демонстрировать различные уровни успешности. Эта тонкость крайне важна для формирования финальных рекомендаций для рекламной кампании интернет-магазина. Рекламируя продукт, магазины должны внимательно изучить целевую аудиторию и адаптировать свои усилия соответственно. Успех не заключается лишь в качественном контенте, но и в умении находить подход к каждому сегменту рынка. Учитывая возрастные предпочтения потребителей, можно значительно повысить эффективность рекламных стратегий и, как следствие, ускорить процесс продаж. В этой динамичной среде, насыщенной конкурентами, знание своеобразия рынка и способности к адаптации становятся важнейшими инструментами для достижения долгожданного успеха.

```
In [68]: plt.figure(figsize=(10, 5))
plt.title('Распределение продаж по типу рейтинга в Северной Америке')
selected_platforms.groupby('rating')['NA_sales'].sum().sort_values(ascending=False)
plt.xlabel('Рейтинг')
plt.ylabel('продано копий (млн)')
plt.xticks(rotation=45, ha='right')
plt.grid()
plt.show()
```

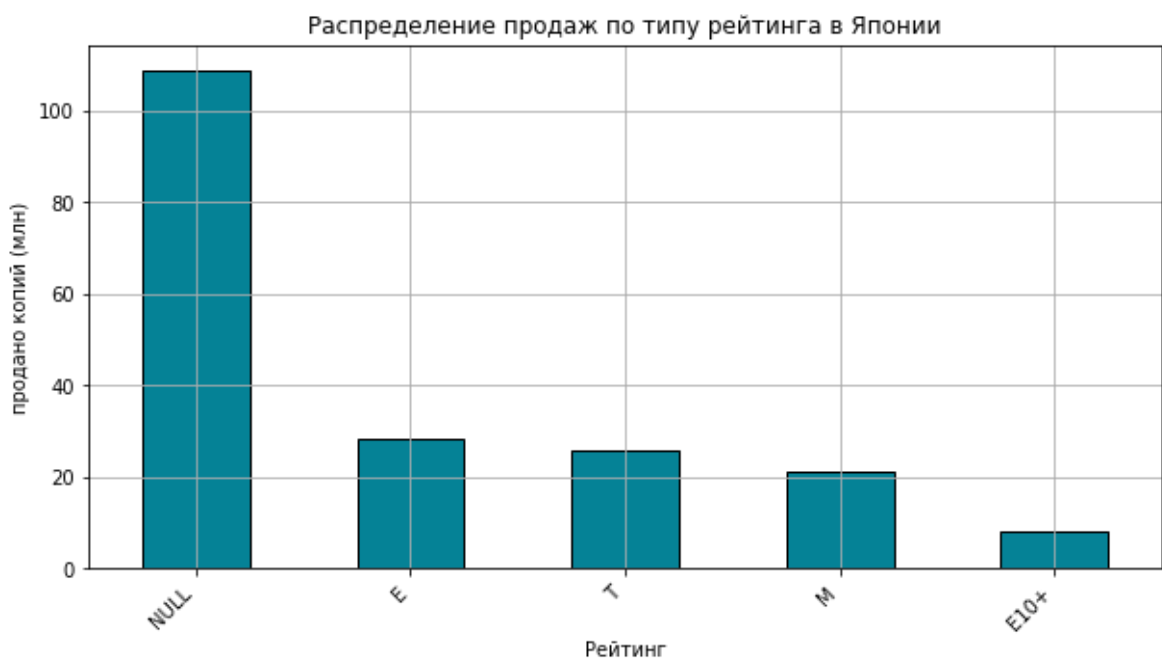


```
In [69]: plt.figure(figsize=(10, 5))
plt.title('Распределение продаж по типу рейтинга в Европе')
selected_platforms.groupby('rating')['EU_sales'].sum().sort_values(ascending=False)
```

```
plt.xlabel('Рейтинг')
plt.ylabel('продано копий (млн)')
plt.xticks(rotation=45, ha='right')
plt.grid()
plt.show()
```



```
In [70]: plt.figure(figsize=(10, 5))
plt.title('Распределение продаж по типу рейтинга в Японии')
selected_platforms.groupby('rating')['JP_sales'].sum().sort_values(ascending=False)
plt.xlabel('Рейтинг')
plt.ylabel('продано копий (млн)')
plt.xticks(rotation=45, ha='right')
plt.grid()
plt.show()
```



```
In [71]: plt.figure(figsize=(15, 5))

NA = selected_platforms.groupby('rating')['NA_sales'].sum()
EU = selected_platforms.groupby('rating')['EU_sales'].sum()
```

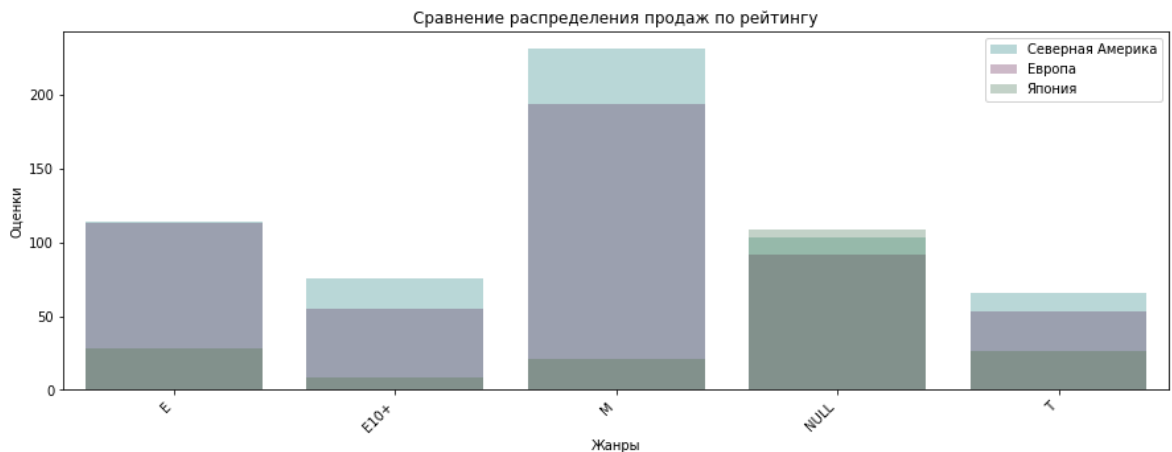
```

JP = selected_platforms.groupby('rating')['JP_sales'].sum()

sns.barplot(y=NA, x=NA.index, label='Северная Америка', color='#0c9689', alpha=0.3)
sns.barplot(y=EU, x=EU.index, label='Европа', color='#6b185d', alpha=0.3)
sns.barplot(y=JP, x=JP.index, label='Япония', color='#417843', alpha=0.3)

plt.xlabel('Жанры')
plt.ylabel('Оценки')
plt.title('Сравнение распределения продаж по рейтингу')
plt.xticks(rotation=45, ha='right')
plt.legend()
plt.show()

```



И в данном сравнении количества продаж в зависимости от рейтинга Северная Америка и Европа близки. Первое место занимают игры, имеющие рейтинг М (для лиц старше 17 лет), на втором месте для этих регионов расположились игры с рейтингом Е (подходящий для всех возрастов). В Японии лидируют игры без возрастного рейтинга, это можно объяснить тем, что ESRB это американская система рейтинга. Игры, которые продаются только на других территориях, могут не иметь рейтинга ESRB. В некоторых случаях, разработчики могут отказаться от рейтинга, если считают, что игра будет неправильно интерпретирована системой рейтинга. Поэтому Япония отличается гораздо больше и выделяется в рейтингах.

```

In [72]: # Комментарий ревьюера
temp = selected_platforms.copy()
print(temp.rating.isna().sum(), temp.rating.isna().sum()/len(temp))
temp.rating.value_counts(dropna=False)

```

0 0.0

```

Out[72]: NULL    1275
         M        498
         T        412
         E        394
         E10+     306
         Name: rating, dtype: int64

```

Проверка гипотез

Для более полного анализа необходимо проверить гипотезы предложенные заказчиком.

Гипотезы 1

Гипотезы:

H0: Средние пользовательские рейтинги платформ Xbox One и PC одинаковые;

H1: Средние пользовательские рейтинги платформ Xbox One и PC отличаются.

```
In [73]: alpha = 0.05

filtr_XOne = games_data[(games_data['platform']=='XOne') & (~games_data['user_score'])
filtr_PC = games_data[(games_data['platform']=='PC') & (~games_data['user_score'])

result = stats.ttest_ind(filtr_XOne, filtr_PC, equal_var=False)
print(f"p-value: {result.pvalue}")

if result.pvalue < alpha:
    print("Отвергаем нулевую гипотезу: \nСредний пользовательский рейтинг Xbox One и PC отличаются")
else:
    print("Не отвергаем нулевую гипотезу: \nЕсть основания утверждать, что средние рейтинги Xbox One и PC одинаковы")
```

p-value: 4.5385802963771835e-06

Отвергаем нулевую гипотезу:

Средний пользовательский рейтинг Xbox One и PC отличаются

В данной проверке гипотез я выбрала метод `ttest_ind`, так как мы проверяем 2 независимые выборки

- `stats.ttest_ind`: Функция для проведения двустороннего t-теста для независимых выборок, в параметрах указываю:
 - `equal_var=False`: Указываем, что дисперсии выборок могут быть не равны.
- `alpha=0.05`: Устанавливаем уровень значимости.

Гипотезы были сформированы исходя из требований проекта. Нулевая гипотеза составлена с общепринятыми нормами и утверждает, что данные показали равно. В качестве альтернативной гипотезы тестируем двустороннюю гипотезу, поэтому параметр "alternative" нет необходимости указывать.

```
In [74]: # Ручная проверка средних
print(f'посмотрим средние значения рейтинга платформ: \nXbox One = {filtr_XOne.mean()} \nPC = {filtr_PC.mean()}')
```

посмотрим средние значения рейтинга платформ:

Xbox One = 6.521428571428572, PC = 7.065960264900661

Видим, что средний пользовательский рейтинг Xbox One и PC отличаются (PC больше)

Гипотезы 2

2. Гипотезы:

H0: Средние пользовательские рейтинги жанров экшен и спорт одинаковые;
H1: Средние пользовательские рейтинги жанров экшен и спорт отличаются.

```
In [75]: alpha = 0.05

filtr_action = games_data[(games_data['genre']=='Action') & (~games_data['user_score']==0)]
filtr_sport = games_data[(games_data['genre']=='Sports') & (~games_data['user_score']==0)]

result = stats.ttest_ind(filtr_action, filtr_sport, equal_var=False)
print(f"p-value: {result.pvalue}")

if result.pvalue < alpha:
    print("Отвергаем нулевую гипотезу: \nСредний пользовательский рейтинг жанров экшен и спорт отличаются.")
else:
    print("Не отвергаем нулевую гипотезу: \nЕсть основания утверждать, что средние рейтинги жанров экшен и спорт равны.")
```

p-value: 0.07751671595536253

Не отвергаем нулевую гипотезу:

Есть основания утверждать, что средние пользовательские рейтинги жанров экшен и спорт равны.

В данной проверке гипотез я выбрала метод `ttest_ind`, так как мы проверяем 2 независимые выборки

- `stats.ttest_ind`: Функция для проведения двустороннего t-теста для независимых выборок, в параметрах указываю:
 - `equal_var=False`: Указываем, что дисперсии выборок могут быть не равны.
- `alpha=0.05`: Устанавливаем уровень значимости.

Гипотезы были сформулированы исходя из требований проекта. Нулевая гипотеза составлена с общепринятыми нормами и утверждает, что данные показали равны. В качестве альтернативной гипотезы я рассматриваю вариант, что проверяемый параметр будет отличаться, без уточнений в малую или большую сторону. Проверка подтверждает, что нулевая гипотеза верна, пользовательские рейтинги для игр в жанре экшена и спорта равны.

```
In [76]: # Ручная проверка средних
print(f'посмотрим средние значения рейтинга платформ: \nXbox One = {filtr_action.mean()} \nPC = {filtr_sport.mean()}')
```

посмотрим средние значения рейтинга платформ:

Xbox One = 7.058129175946549, PC = 6.9527777777777775

В ручной проверке мы можем увидеть разницу средних значений в случае с сравнением средних для рейтинга Xbox One и PC мы получаем средние отличающиеся более чем на 0.5, в случае с средними пользовательскими рейтингами жанров экшен и спорт получаем разницу средних не более 0.1, что в нашем случае при проверке гипотезы не является существенным.

Общий вывод

Провели предварительную обработку данных:

- * Замена пропусков, где это было возможно; остальные пропуски оставили без изменений, за исключением изменения строковых значений пользовательского рейтинга 'tbd', так как в будущем это могло помешать в работе с этим столбцом;
- * Добавили дополнительный столбец в данные, в котором указали прибыль по всем странам, представленным в датасете;
- * Остальные столбцы приведены к стандарту именования;

На этапе исследовательского анализа данных изучили зависимости и выявили, выявили некоторые тенденции и зависимости вкратце о них:

- * В число актуальных периодов, за которые рассматриваем продажи игр попали все года включая 2012;
- * В датасете представлено всего 31 платформа, самая популярная из представленных – PS2, продано 1055.68 млн копий за весь период (в актуальный период не вошла);
- * Выявили, что примерный период активных продаж платформы составляет 10 лет, из которых примерно 5 лет приходятся на пик продаж. Поэтому выбран был актуальный период за 8 лет, он позволит увидеть снижение пика популярности;
- * Лидером продаж на 2016 год стала платформа PS4 (почти 50 млн копий) и 3DS (менее 25 млн копий);
- * Выделили 4 платформы: PS4, XOne, WiiU и PSV, которые вписываются в тенденции и имеют шансы на высокие продажи;*
- * Можно выделить платформу 3DS, она на втором месте по продажам, но уже достаточно долго (больше всех из представленных) находится в топе и пик популярности начал снижаться;
- * Выявили, что самой покупаемой частью мира стала Северная Америка (51%), Европа – на втором месте с 36%, завершает рейтинг Япония с 13%;*
- * Северная Америка и Европа имеют схожие предпочтения в играх, популярны одни и те же жанры и возрастные рейтинги (экшен, старше 17 лет);

По итогам проверки гипотез выявили, что средние рейтинги пользователей Xbox One меньше, чем у PC (за весь представленный период);

И узнали, что средние пользовательские рейтинги жанров экшен и спорт не имеют статистически важных отличий (схожи)

Рекомендации

Предлагаю сосредоточиться на рекламе игр на такие платформы, как PS4, XOne. Данные платформы набирают популярность и являются наиболее распространёнными в Европе, частью которой наша страна и является. Рекомендуется рекламировать игры в жанре шутера, и спортивных играх, так как они являются лидерами продаж в выбранный актуальный период и и также на

территории Европы, имеют одни из самых высоких значений продаж по медиане. Обратите внимание на продукты с рейтингом М (для лиц старше 17 лет) и Е (для всех возрастов).

Примерный портрет игры на рекламе которой стоит сосредоточиться: Шутер или спортивная игра на платформу PS4, XOne с возрастным рейтингом М, реклама преимущественно в регионе Северной Америки и Европе. Рынка этих регионов считаю будет достаточно, так как бюджет заложенный на рекламу ограничен, а данные регионы имеют очень высокий процент продаж почти 85% от общего числа продаж.