# Сборный проект 1

## Изучение общей информации

Провести первичный анализ данных Провести предобработку данных (привести к нужным типам и исправить ошибки) Обработать пропуски при необходимости: Объяснить, почему заполнили пропуски определённым образом или почему не стали это делать; Опишите причины, которые могли привести к пропускам; Обратите внимание на аббревиатуру 'tbd' в столбце с оценкой пользователей. Отдельно разобрать это значение и описать, как его обработать; Посчитать суммарные продажи во всех регионах и записать их в отдельный столбец. Посмотреть, сколько игр выпускалось в разные годы. Важны ли данные за все периоды? Посмотреть, как менялись продажи по платформам. Выбрать платформы с наибольшими суммарными продажами и построить распределение по годам. Определить за какой характерный срок появляются новые и исчезают старые платформы. Взять данные за соответствующий актуальный период. Актуальный период определить самостоятельно в результате исследования предыдущих вопросов. Основной фактор — эти данные помогут построить прогноз на 2017 год. Определить,какие платформы лидируют по продажам, растут или падают. Выбрать несколько потенциально прибыльных платформ. Построить график «ящик с усами» по глобальным продажам игр в разбивке по платформам. Описать результат. Посмотреть, как влияют на продажи внутри одной популярной платформы отзывы пользователей и критиков. Построить диаграмму рассеяния и посчитать корреляцию между отзывами и продажами. Сформулировать выводы. Соотнести выводы с продажами игр на других платформах. Посмотреть на общее распределение игр по жанрам. Что можно сказать о самых прибыльных жанрах? Выделяются ли жанры с высокими и низкими продажами? Определить для пользователя каждого региона (NA, EU, JP): Самые популярные платформы (топ-5). Описать различия в долях продаж. Самые популярные жанры (топ-5). Пояснить разницу. Определить,влияет ли рейтинг ESRB на продажи в отдельном регионе. Проверить гипотезы: Средние пользовательские рейтинги платформ Xbox One и PC одинаковые; Средние пользовательские рейтинги жанров Action (англ. «действие», экшен-игры) и Sports (англ. «спортивные соревнования») разные. Написать общий вывод

In [1]:
```
#импортируем необходимые библиотеки
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as st
%matplotlib inline
import seaborn as sns
from scipy import stats as st
```

In [2]:
```
# скачиваем файлы
data = pd.read_csv('/datasets/games.csv')
```

In [3]:
```
data.head()
```

Out[3]:

| | Name | Platform | Year_of_Release | Genre | NA_sales | EU_sales | JP_sales | Other_sales | Critic_Score | User |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Wii Sports | Wii | 2006.0 | Sports | 41.36 | 28.96 | 3.77 | 8.45 | 76.0 | |
| **1** | Super Mario Bros. | NES | 1985.0 | Platform | 29.08 | 3.58 | 6.81 | 0.77 | NaN | |
| **2** | Mario Kart Wii | Wii | 2008.0 | Racing | 15.68 | 12.76 | 3.79 | 3.29 | 82.0 | |

| | Name | Platform | Year_of_Release | Genre | NA_sales | EU_sales | JP_sales | Other_sales | Critic_Score | User |
|---|---|---|---|---|---|---|---|---|---|---|
| **3** | Wii Sports Resort | Wii | 2009.0 | Sports | 15.61 | 10.93 | 3.28 | 2.95 | 80.0 | |
| **4** | Pokemon Red/Pokemon Blue | GB | 1996.0 | Role-Playing | 11.27 | 8.89 | 10.22 | 1.00 | NaN | |

In [4]:
```
#получаем информацию
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 [5]:
```
# проверим столбцы
data.columns
```

Out[5]:
```
Index(['Name', 'Platform', 'Year_of_Release', 'Genre', 'NA_sales', 'EU_sales',
       'JP_sales', 'Other_sales', 'Critic_Score', 'User_Score', 'Rating'],
      dtype='object')
```

In [6]:
```
#Подсчитаем количество пустых значений
data.isna().sum()
```

Out[6]:
```
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 [7]:
```
#Посмотрим какие платформы для игры у нас имеются
data['Platform'].value_counts()
```

Out[7]:
```
PS2     2161
DS      2151
PS3     1331
Wii     1320
X360    1262
PSP     1209
PS      1197
PC       974
XB       824
```

```
GBA      822
GC       556
3DS      520
PSV      430
PS4      392
N64      319
XOne     247
SNES     239
SAT      173
WiiU     147
2600     133
NES       98
GB        98
DC        52
GEN       29
NG        12
SCD        6
WS         6
3DO        3
TG16       2
GG         1
PCFX       1
Name: Platform, dtype: int64
```

In [8]:
```
#Посмотрим какие жанры игр мы имеем и нет ли повторений
data['Genre'].value_counts()
```

Out[8]:
```
Action          3369
Sports          2348
Misc            1750
Role-Playing    1498
Shooter         1323
Adventure       1303
Racing          1249
Platform         888
Simulation       873
Fighting         849
Strategy         683
Puzzle           580
Name: Genre, dtype: int64
```

In [9]:
```
#Посчитаем количество дубликатов
data.duplicated().sum()
```

Out[9]:
```
0
```

Вывод по изучению общей информации:

Необходимо привести к правильному типу столбец : Year of Release и User_Score Также нужно привести к нижнему регистру столбцы нашей таблицы, а так же сами названия колонок. Тип данных года выпуска указан float64, a User_Score - object. Имеются пустые значения в столбцах rating, user_score, critic_score. Дубликатов не имеется

# Предобработка данных

## Приведем названия столбцов к нижнему регистру

In [10]:
```
# приведем названия столбцов к нижнему регистру
data.columns = map(str.lower, data.columns)
data.head()
```

Out[10]:

| | name | platform | year_of_release | genre | na_sales | eu_sales | jp_sales | other_sales | critic_score | user_s |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Wii Sports | Wii | 2006.0 | Sports | 41.36 | 28.96 | 3.77 | 8.45 | 76.0 | |

| | name | platform | year_of_release | genre | na_sales | eu_sales | jp_sales | other_sales | critic_score | user_s |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | Super Mario Bros. | NES | 1985.0 | Platform | 29.08 | 3.58 | 6.81 | 0.77 | NaN | |
| **2** | Mario Kart Wii | Wii | 2008.0 | Racing | 15.68 | 12.76 | 3.79 | 3.29 | 82.0 | |
| **3** | Wii Sports Resort | Wii | 2009.0 | Sports | 15.61 | 10.93 | 3.28 | 2.95 | 80.0 | |
| **4** | Pokemon Red/Pokemon Blue | GB | 1996.0 | Role-Playing | 11.27 | 8.89 | 10.22 | 1.00 | NaN | |

## Приведем к нижнему регистру следующие столбцы:

In [11]:
```
# Приведем к нижнему регистру следующие столбцы:
for column in data[['name','platform','genre','rating']]:
    data[column] = data[column].str.lower()
```

In [12]:
```
# получаем информацию
data['name']
```

Out[12]:
```
0                          wii sports
1                   super mario bros.
2                      mario kart wii
3                   wii sports resort
4              pokemon red/pokemon blue
                      ...
16710    samurai warriors: sanada maru
16711                  lma manager 2007
16712          haitaka no psychedelica
16713                 spirits & spells
16714               winning post 8 2016
Name: name, Length: 16715, dtype: object
```

## Поменяем тип данных

При попытке поменять тип данных столбца user_score на float64, выскаквает ошибка, тк встречается аббревиатура tbd. Обратим на нее внимание.Отдельно разберем это значение и опишем, как его обработать. Аббревиатура tbd значит to be determined, to be done. То есть, данные были нарочно не заполнены, так как не определились с рейтингом. Поэтому предлагаю заменить tbd на Nan

In [13]:
```
#Заменим tbd на Nan
data['user_score'] = data['user_score'].replace('tbd', np.nan, regex=True)
```

In [14]:
```
# Поменяем формат столбца user_score на float
data['user_score'] = data['user_score'].astype(float)
data['user_score'].dtype
```

Out[14]:
```
dtype('float64')
```

In [15]:
```
# определяем пропущенные значения по name
print(data['name'].isna().sum())
```

```
2
```

In [16]:
```
# определяем пропущенные значения по rating
print(data['rating'].isna().sum())
```

```
6766
```

In [17]:
```
#Заменим пропуски на unknow
data['rating'] = data['rating'].fillna('unknow')
```

In [18]:
```
# проверяем пропущенные значения по rating
print(data['rating'].isna().sum())
```

```
0
```

In [19]:
```
# получаем информацию
data['year_of_release']
```

Out[19]:
```
0        2006.0
1        1985.0
2        2008.0
3        2009.0
4        1996.0
          ...
16710    2016.0
16711    2006.0
16712    2016.0
16713    2003.0
16714    2016.0
Name: year_of_release, Length: 16715, dtype: float64
```

In [20]:
```
# заменили тип данных на целые числа
In [125]: data = data.astype({"year_of_release": "Int64"})
```

In [21]:
```
#проверяем
data['year_of_release'].sort_values()
```

Out[21]:
```
1764     1980
546      1980
1968     1980
6300     1980
6875     1980
         ...
16373    <NA>
16405    <NA>
16448    <NA>
16458    <NA>
16522    <NA>
Name: year_of_release, Length: 16715, dtype: Int64
```

заменили тип данных года выпуска, тк он может быть только целый

In [22]:
```
# получаем информацию
data['platform']
```

Out[22]:
```
0         wii
1         nes
2         wii
3         wii
4          gb
         ...
16710     ps3
16711    x360
16712     psv
16713     gba
16714     psv
Name: platform, Length: 16715, dtype: object
```

In [23]:
```
# получаем информацию
```

```
data['user_score']
```

Out[23]:
```
0        8.0
1        NaN
2        8.3
3        8.0
4        NaN
        ...
16710    NaN
16711    NaN
16712    NaN
16713    NaN
16714    NaN
Name: user_score, Length: 16715, dtype: float64
```

In [24]:
```
data['rating']
```

Out[24]:
```
0            e
1        unknow
2            e
3            e
4        unknow
          ...
16710    unknow
16711    unknow
16712    unknow
16713    unknow
16714    unknow
Name: rating, Length: 16715, dtype: object
```

# Проверяем пропуски

In [25]:
```
#проверяем пропуски в столбце Год выпуска
print(data['year_of_release'].isna().sum())
```

```
269
```

пропусков в столбцах year_of_release, genre и name меньше 2%. Можно пренебречь.

In [26]:
```
# удаляем строки с пропущенными значениями года выпуска методом dropna()
data = data.dropna(subset=['year_of_release'])
print('пропуски после:',data['year_of_release'].isna().sum()) #проверяем
```

```
пропуски после: 0
```

In [27]:
```
# удаляем строки с пропущенными значениями name методом dropna()
data = data.dropna(subset=['name'])
print('пропуски после:',data['name'].isna().sum()) #проверяем
```

```
пропуски после: 0
```

In [28]:
```
# удаляем строки с пропущенными значениями genre методом dropna()
data = data.dropna(subset=['genre'])
print('пропуски после:',data['genre'].isna().sum()) #проверяем
```

```
пропуски после: 0
```

Остальные пропуски не будем ничем заполнять. Заполнение исказит результаты кореляционного анализа. Причины, которые могли привести к пропускам:отсутствие полной информации.

# Посчитаем суммарные продажи во всех регионах и запишем их в отдельный столбец.

In [29]:
```
#Создадим новый столбец total_sales и прибавим продажи всех столбцов
```

```
data['total_sales'] = data[['na_sales','eu_sales','jp_sales','other_sales']].sum(axis=1)
data.head()
```

Out[29]:

| | name | platform | year_of_release | genre | na_sales | eu_sales | jp_sales | other_sales | critic_score | user_sc |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | wii sports | wii | 2006 | sports | 41.36 | 28.96 | 3.77 | 8.45 | 76.0 | |
| **1** | super mario bros. | nes | 1985 | platform | 29.08 | 3.58 | 6.81 | 0.77 | NaN | N |
| **2** | mario kart wii | wii | 2008 | racing | 15.68 | 12.76 | 3.79 | 3.29 | 82.0 | |
| **3** | wii sports resort | wii | 2009 | sports | 15.61 | 10.93 | 3.28 | 2.95 | 80.0 | |
| **4** | pokemon red/pokemon blue | gb | 1996 | role-playing | 11.27 | 8.89 | 10.22 | 1.00 | NaN | N |

In [30]:

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16444 entries, 0 to 16714
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   name             16444 non-null  object 
 1   platform         16444 non-null  object 
 2   year_of_release  16444 non-null  Int64  
 3   genre            16444 non-null  object 
 4   na_sales         16444 non-null  float64
 5   eu_sales         16444 non-null  float64
 6   jp_sales         16444 non-null  float64
 7   other_sales      16444 non-null  float64
 8   critic_score     7983 non-null   float64
 9   user_score       7463 non-null   float64
 10  rating           16444 non-null  object 
 11  total_sales      16444 non-null  float64
dtypes: Int64(1), float64(7), object(4)
memory usage: 1.6+ MB
None
```

## Вывод по предобработка данных:

**Вывод по предобработка данных:**

Привели к правильному типу столбец : year_of_release и rating. Также привели к нижнему регистру столбцы нашей таблицы, а так же сами названия колонок. Тип данных года выпуска перевели в целые числа , a user_score - float64. Пустые значения в столбцах user_score, critic_score решили не трогать, а в столбцах year_of_release,genre и name удалили строки с пропусками тк их меньше 2%. Пропуски в столбце rating заполнили на unknow Дубликатов не имеется.

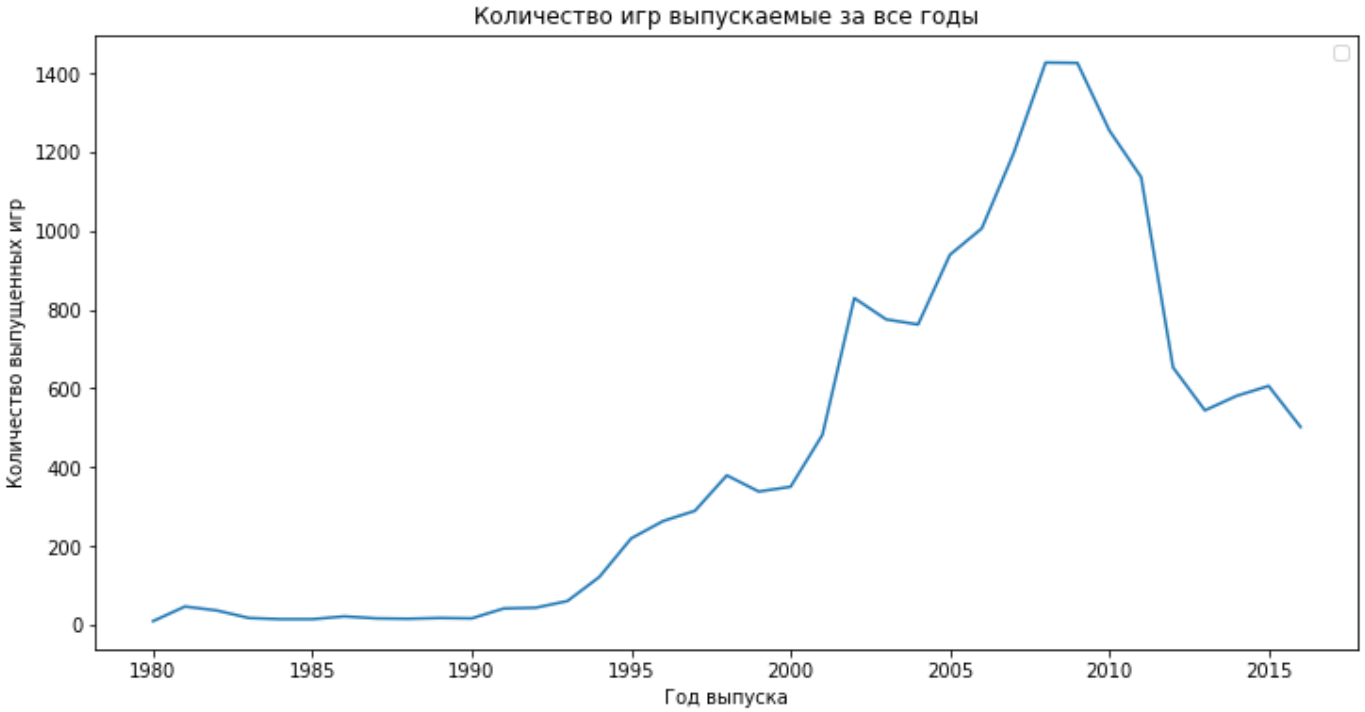
# Исследовательский анализ данных

## Посмотрим, сколько игр выпускалось в разные годы и важны ли данные за все периоды.

In [31]:

```
# Методом pivot отсортируем таблицы и отрисуем график, чтобы просмотреть как менялось количеств
games_for_the_period = data.pivot_table(index='year_of_release', values='name', aggfunc='count'
plt.figure(figsize=(12,6))
sns.lineplot(data=games_for_the_period)
plt.title("Количество игр выпускаемые за все годы")
plt.xlabel("Год выпуска")
plt.ylabel("Количество выпущенных игр")
```

```
plt.legend('')
plt.show()
```




In [91]:

```
#проведем анализ методом describe() и построим гистограмму:
data[['year_of_release']].hist()
plt.title('Количество игр выпускаемые за все годы')
plt.xlabel('Год выпуска')
plt.ylabel('Количество выпущенных игр')
plt.show()
data[['year_of_release']].describe().round(2)
```


Out[91]:

| | year_of_release |
|---|---|
| **count** | 16444.00 |
| **mean** | 2006.49 |
| **std** | 5.88 |
| **min** | 1980.00 |
| **25%** | 2003.00 |
| **50%** | 2007.00 |
| **75%** | 2010.00 |
| **max** | 2016.00 |

Вывод:

Из графика видно, что количество игр на игровые приставки и компьютеры начало расти с большой скоростью с 1992 года до 2010 года. Данные заканчиваются 2016 годом. С 2009 года, после того как массово начали создавать мобильные приложения и мобильные игры, виден резкий спад консольных игр. Возьмем данные за соответствующий актуальный период, который определяем самостоятельно в результате исследования предыдущих вопросов. Основной фактор — рост продаж в 2013 году. С этого года будем считать актуальный период.Полученные данные помогут построить прогноз на 2017 год. Так как продажи игр падают,прогноз на 2017 - тоже снижение. Данные за предыдущие годы не будем учитывать в работе.

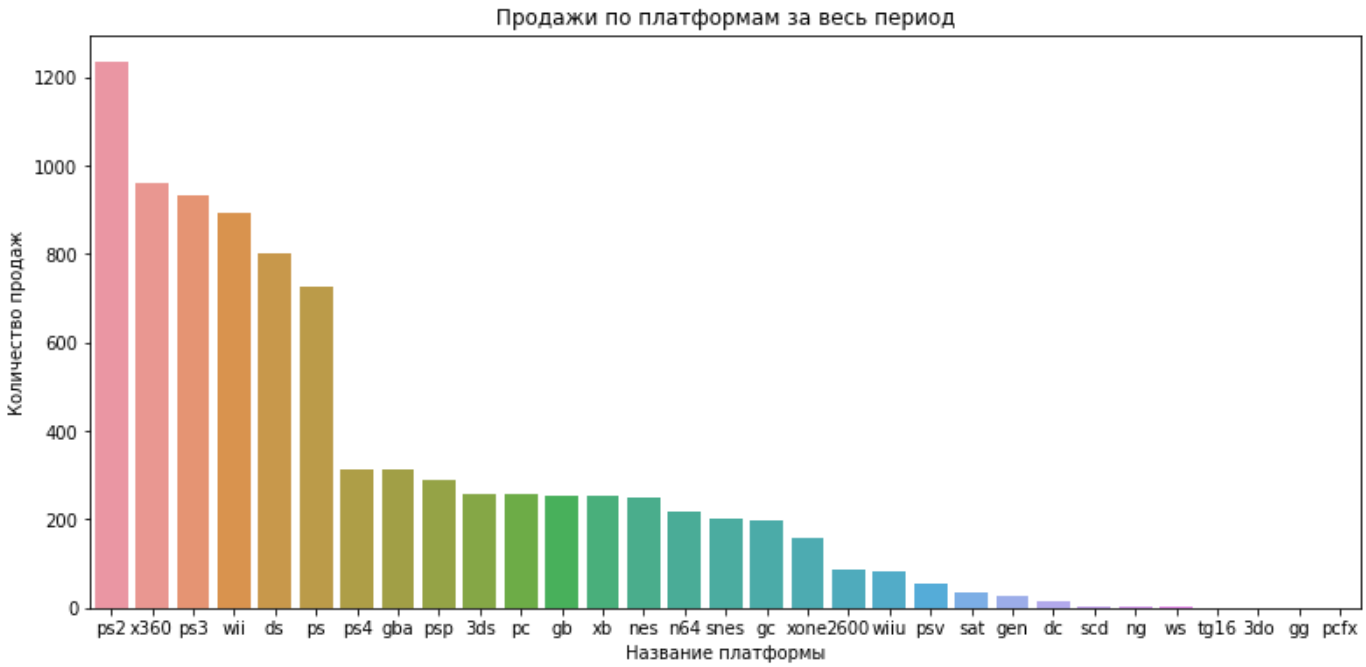
# Посмотрим, как менялись продажи по платформам

Посмотрим, как менялись продажи по платформам. Выберем платформы с наибольшими суммарными продажами и построим распределение по годам. Определим,за какой характерный срок появляются новые и исчезают старые платформы.

## Посмотрим, как менялись продажи по платформам за весь период

In [33]:
```
#Методом pivot отсортируем таблицы и отрисуем график, чтобы просмотреть как менялись продажи иг
sales_by_platform = data.pivot_table(
    index='platform', values='total_sales', aggfunc='sum').sort_values(by='total_sales', ascend

plt.figure(figsize=(13,6))
sns.barplot(x=sales_by_platform.index,y=sales_by_platform['total_sales'])
plt.title("Продажи по платформам за весь период")
plt.xlabel("Название платформы")
plt.ylabel("Количество продаж")
plt.show()
```




По графику видно,что за исследуемый период налицо определилась шестерка лидеров PS2, X360, PS3, WII, DS, PS.

In [34]:
```
 #Напишем функцию, которая будет возвращать нужную сводную таблицу и выводить данные с 2000 год
def year_total_sale_by_platform(name, data):
    segment = data[(data['platform'] == name) & (data['year_of_release'] > 2000)]
    total = segment.pivot_table(index='year_of_release', values='total_sales', aggfunc='sum').s
    return total
```

In [35]:
```
# Создадим  таблицу по платформам и их обшим продажам. отсортируем их по убыванию и оставим тол
top_10_platforms = data.pivot_table(index='platform', values='total_sales', aggfunc='sum').sort
top_10_platforms = top_10_platforms.reset_index().rename_axis(None, axis=1)
```

In [36]:
```
#Отрисуем все игровые платформы и их поведение за последние 16 лет
plt.figure(figsize=(12,6))
plt.title('Количество продаж популярных игровых платформ')
plt.xlabel('Годы игровых релизов')
plt.ylabel('Продажи')

for i in list(top_10_platforms['platform']):
    sns.lineplot(data=year_total_sale_by_platform(i,data)['total_sales'], label=i)
    plt.legend()
```


На графике видно, как менялись продажи по платформам по годам. В основном,после выхода платформы идет рост продаж примерно до 4 лет. Потом резкий спад.Примерно, срок жизни платформы 10 лет. Платформа PS2 прекратила продажи в 2011 году, а DS - в 2013г. Перспективная Ps4.

In [37]:
```
#Выведем топ 10 продаваемых платформ
top_10_platforms
```

Out[37]:

| | platform | total_sales |
|---|---|---|
| **0** | ps2 | 1233.56 |
| **1** | x360 | 961.24 |
| **2** | ps3 | 931.34 |
| **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 |

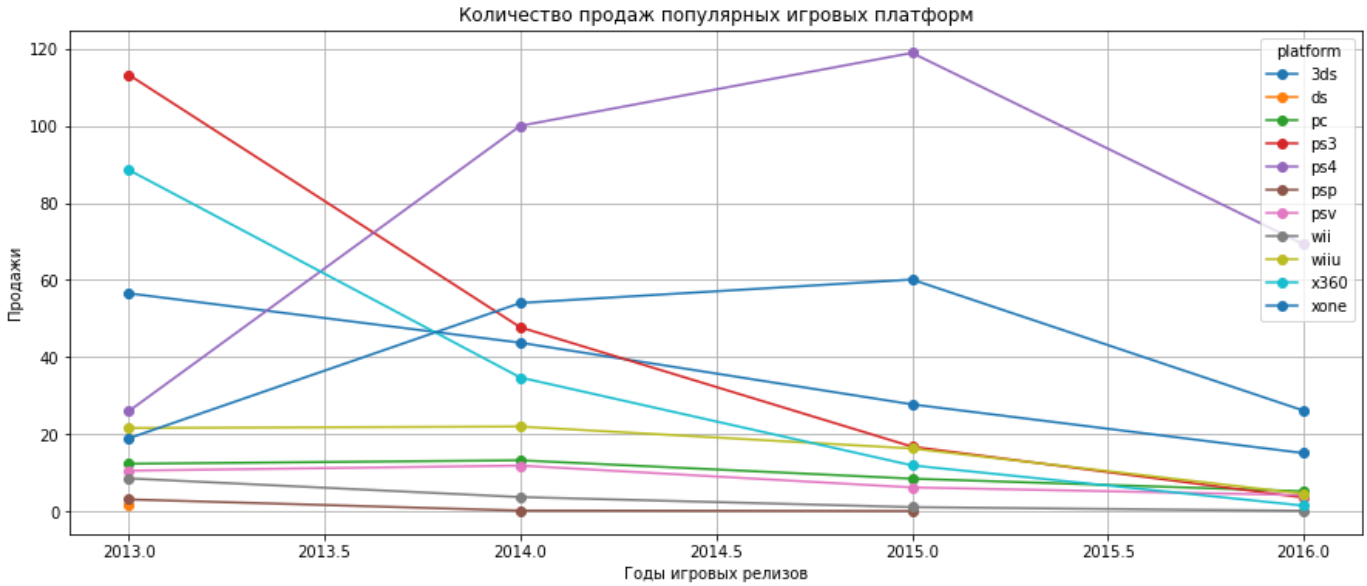
## Посмотрим, как менялись продажи по платформам за актуальный период

In [38]:
```
# Возьмем актуальный период с 2013 года
current_period = data.query('2013<=year_of_release')
current_period.head(10)
```

Out[38]:

| | name | platform | year_of_release | genre | na_sales | eu_sales | jp_sales | other_sales | critic_score | user_ |
|---|---|---|---|---|---|---|---|---|---|---|
| **16** | grand theft auto v | ps3 | 2013 | action | 7.02 | 9.09 | 0.98 | 3.96 | 97.0 | |
| **23** | grand theft auto v | x360 | 2013 | action | 9.66 | 5.14 | 0.06 | 1.41 | 97.0 | |
| **31** | call of duty: black ops 3 | ps4 | 2015 | shooter | 6.03 | 5.86 | 0.36 | 2.38 | NaN | |
| **33** | pokemon x/pokemon y | 3ds | 2013 | role-playing | 5.28 | 4.19 | 4.35 | 0.78 | NaN | |
| **42** | grand theft auto v | ps4 | 2014 | action | 3.96 | 6.31 | 0.38 | 1.97 | 97.0 | |
| **47** | pokemon omega ruby/pokemon alpha sapphire | 3ds | 2014 | role-playing | 4.35 | 3.49 | 3.10 | 0.74 | NaN | |
| **60** | call of duty: ghosts | x360 | 2013 | shooter | 6.73 | 2.56 | 0.04 | 0.91 | 73.0 | |
| **69** | call of duty: ghosts | ps3 | 2013 | shooter | 4.10 | 3.63 | 0.38 | 1.25 | 71.0 | |
| **72** | minecraft | x360 | 2013 | misc | 5.70 | 2.65 | 0.02 | 0.81 | NaN | |
| **77** | fifa 16 | ps4 | 2015 | sports | 1.12 | 6.12 | 0.06 | 1.28 | 82.0 | |

In [39]:
```
# Нарисуем графики платформ за актуальный период.
current_period.pivot_table(index = 'year_of_release',
                values = 'total_sales',
                columns = 'platform',
                aggfunc='sum'
               ).plot(grid=True, figsize=(15, 6), style ='o-')
plt.title('Количество продаж популярных игровых платформ')
plt.xlabel('Годы игровых релизов')
plt.ylabel('Продажи')
plt.show()
```




Вывод: -За исследуемый актуальный период продажи на всех платформах падали.Исключение составила PS4 и 3ds росли до 2015 года. -.Платформа psp вообще прекратила продажи в 2015. году. К 2016 году - осталось 3 потенциальных платформы PS4, 3ds и xone. Их и будем рассматривать.

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

In [40]:
```
#Сохраним в переменной df_top_3_platforms топ 3 платформ и избавимся от выбросов
list_of_top3 = ['ps4','xone','3ds']
df_top_3_platforms = current_period[current_period['platform'].isin(list_of_top3)]
df_top_3_platforms = df_top_3_platforms[df_top_3_platforms['total_sales']<1.4]
```

In [41]:
```
df_top_3_platforms.sort_values(by='total_sales', ascending=False)
df_top_3_platforms.head()
```

Out[41]:

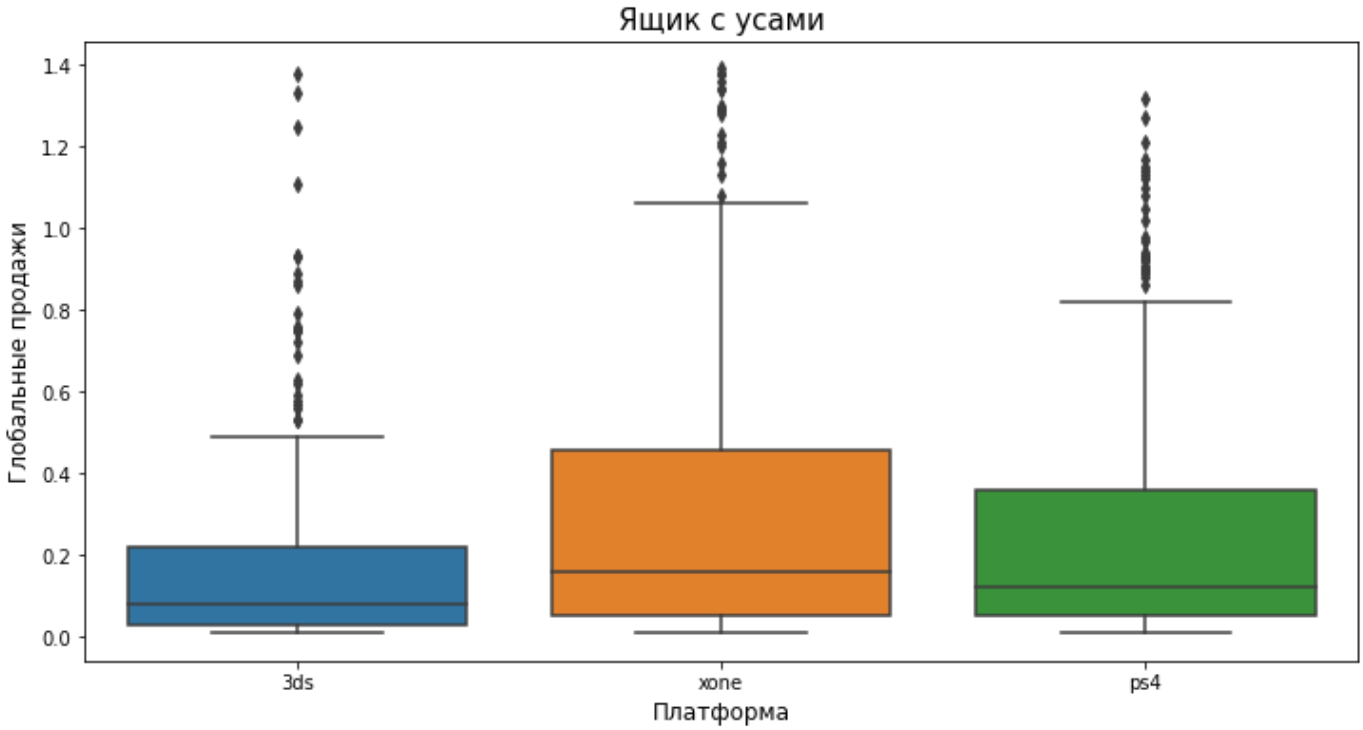
| | name | platform | year_of_release | genre | na_sales | eu_sales | jp_sales | other_sales | critic_score | user_sc |
|---|---|---|---|---|---|---|---|---|---|---|
| **1395** | yoshi's new island | 3ds | 2014 | platform | 0.48 | 0.53 | 0.28 | 0.09 | 64.0 | |
| **1400** | ryse: son of rome | xone | 2013 | action | 0.83 | 0.43 | 0.00 | 0.13 | 60.0 | |
| **1401** | mortal kombat x | xone | 2015 | fighting | 1.03 | 0.21 | 0.00 | 0.14 | 86.0 | |
| **1403** | rise of the tomb raider | xone | 2015 | adventure | 0.55 | 0.70 | 0.02 | 0.11 | 86.0 | |
| **1434** | middle-earth: shadow of mordor | xone | 2014 | action | 0.73 | 0.50 | 0.01 | 0.12 | 87.0 | |

In [89]:
```
# исследуем глобальные продажи методом describe()
df_top_3_platforms['total_sales'].describe().round(2)
```

Out[89]:
```
count    820.00
mean       0.25
std        0.30
min        0.01
25%        0.04
50%        0.11
75%        0.33
max        1.39
Name: total_sales, dtype: float64
```

In [43]:
```
#Отрисуем ящики с усами
plt.figure(figsize=(12,6))
sns.boxplot(data=df_top_3_platforms, x='platform', y='total_sales')
plt.title('Ящик с усами', fontsize=15)
plt.xlabel('Платформа', fontsize=12)
plt.ylabel('Глобальные продажи',fontsize=12)
```

Out[43]:
```
Text(0, 0.5, 'Глобальные продажи')
```




Вывод:

- Провели срез данных для того, чтобы отрисовать "ящики с усами".
- Исходя из графиков видно, что медиана протекает у всех одинаково.
- Больше всех продаж у xone, затем PS4 и 3DS на последнем месте.
- Выбросы - это игры-хиты с аномально высокими продажами,бестселлеры
- Посмотрим на 3DS: маленький ящик и много выбросов. Это говорит о том, что большАя часть объема продаж делается за счет этих выбросов.
- В то же время у XOne ситуация иная: большой бокс, длинный ус и мало выбросов. Это свидетельствует о том, что на XOne покупают самые разные игры, причем, в немалом количестве.

## Посмотрим, как влияют на продажи внутри одной популярной платформы отзывы пользователей и критиков

**Посмотрим, как влияют на продажи внутри одной популярной платформы отзывы пользователей и критиков. Построим диаграмму рассеяния и посчитаем корреляцию между отзывами и продажами. Сформулируем выводы.**

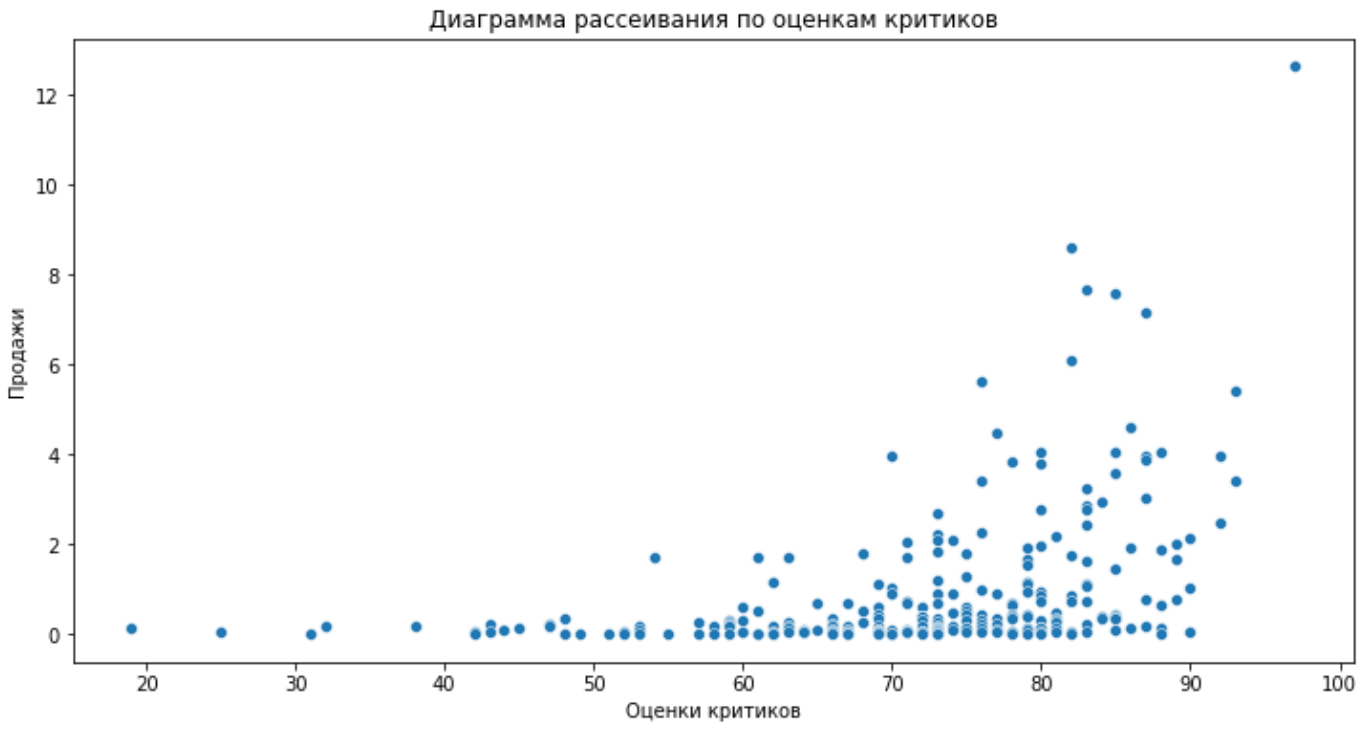
In [87]:
```
#Корреляция между оценками пользователей и продажами PS4
sony_play_station4 = current_period[current_period['platform']=='ps4']
sony_play_station4['user_score'].corr(sony_play_station4['total_sales']).round(3)
```

Out[87]: -0.032

Коэффициет корреляции маленький,оценки пользователей на продажи влияют слабо.

In [45]:
```
#Построим диаграмму рассеивания по оценкам пользователей
plt.figure(figsize=(12,6))
sns.scatterplot(x='user_score', y='total_sales', data=sony_play_station4)
plt.title('Диаграмма рассеивания по оценкам пользователей')
plt.xlabel('Оценки пользователей')
plt.ylabel('Продажи')
plt.show()
```


In [46]:
```
#Построим диаграмму рассеяния по оценкам критиков
plt.figure(figsize=(12,6))
sns.scatterplot(x='critic_score', y='total_sales', data=sony_play_station4)
plt.title('Диаграмма рассеивания по оценкам критиков')
plt.xlabel('Оценки критиков')
plt.ylabel('Продажи')
plt.show()
```



In [86]:
```
#Корреляция между оценкой критиков и продажам PS4
sony_play_station4['critic_score'].corr(sony_play_station4['total_sales']).round(3)
```

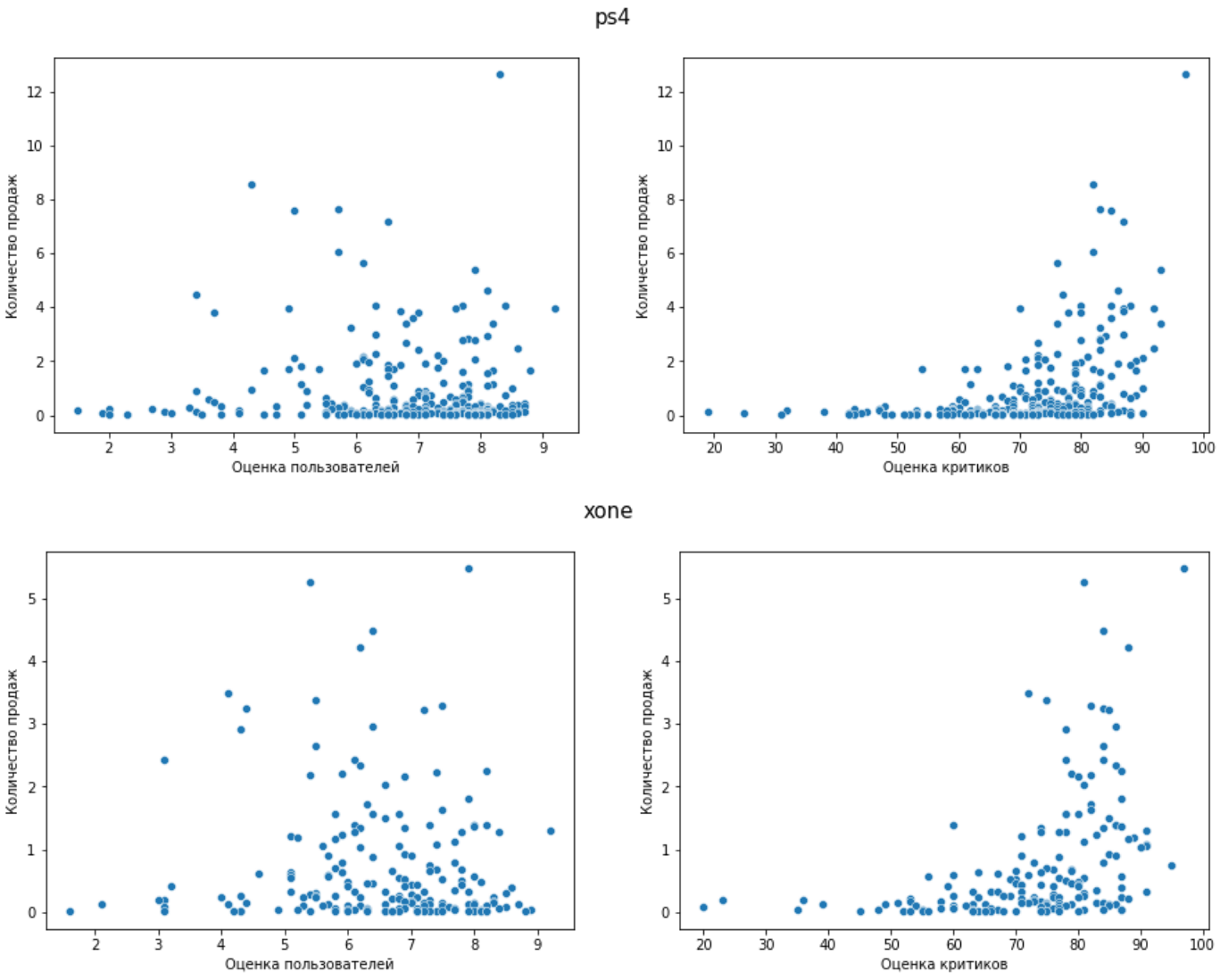
Out[86]: 0.407

Коэффициет корреляции между оценкой критиков и продажам PS4 более чем в 10 раз превышает коэффициет корреляции между оценкой пользователей и продажами. Мнение критиков влияет на продажи, а отзывы пользователей нет.

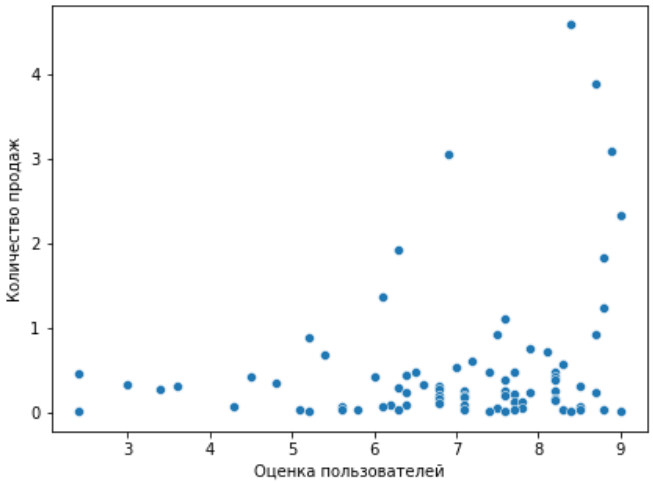
## Соотнесем выводы с продажами игр на других платформах.

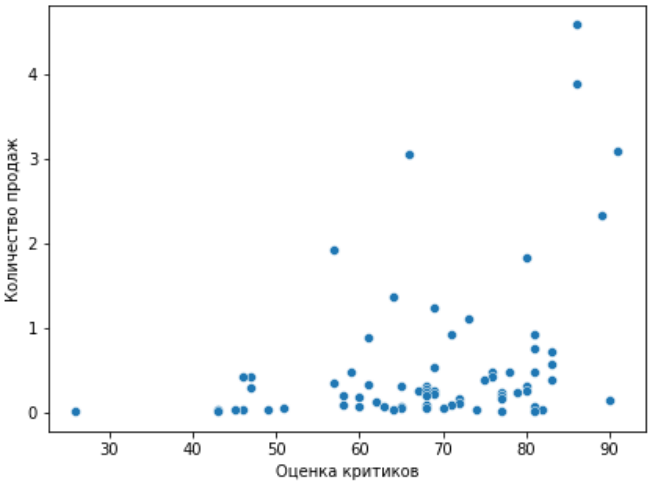
In [48]:
```
#Напишем функцию, которая будет отрисовывать графики рассеивания и считать корреляции
def other_platform(platform_name):
    platform = current_period[current_period['platform']==platform_name]
    fig, ax = plt.subplots(1 ,2, figsize=(15,5))
    sns.scatterplot(x='user_score', y='total_sales', data=platform, ax=ax[0])
    sns.scatterplot(x='critic_score', y='total_sales', data=platform, ax=ax[1])
    fig.suptitle(platform_name, fontsize=15)
    ax[0].set(xlabel='Оценка пользователей')
    ax[1].set(xlabel='Оценка критиков')
    ax[0].set(ylabel='Количество продаж')
    ax[1].set(ylabel='Количество продаж')
    plt.show()
```

In [49]:
```
#С помощью цикла выведем все 3 графика
for platform in list_of_top3:
    other_platform(platform)
```



3ds




## Найдем корреляцию по другим платформам

In [85]:
```
# найдем корреляцию по другим платформам
for platform in list_of_top3:
    print('Корреляция между отзывами пользователей и игровой платформой', platform.upper(),':',
    print('Корреляция между отзывами критиков и игровой платформой', platform.upper(),':', (dat
```

```
Корреляция между отзывами пользователей и игровой платформой PS4 : -0.032
Корреляция между отзывами критиков и игровой платформой PS4 : 0.407
Корреляция между отзывами пользователей и игровой платформой XONE : -0.069
Корреляция между отзывами критиков и игровой платформой XONE : 0.417
Корреляция между отзывами пользователей и игровой платформой 3DS : 0.222
Корреляция между отзывами критиков и игровой платформой 3DS : 0.349
```

Коэффициенты корреляции так же маленькие. Отзывы пользователей не влияют на продажи, оценки критиков влияют незначительно. .Коэффициенты корреляции по отзывам критиков больше, чем корреляция по отзывам пользователей.Покупатели прислушиваются больше к критикам, чем к оценкам других пользователей

## Посчитаем дисперсию, стандартное отклонение, среднее и медиану у топ 5 платформ к оценкам пользователей

In [84]:
```
#Посчитаем дисперсию, стандартное отклонение, среднее и медиану у топ 5 платформ к оценкам поль
for platform in list_of_top3:
    print('Дисперсия', platform.upper(),':', np.var(current_period[current_period['platform']==
    print('Стандартное отклонение', platform.upper(),':', np.std(current_period[current_period[
    print('Среднее',platform.upper(),':',  current_period[current_period['platform']==platform]
    print('Медиана',platform.upper(),':',  current_period[current_period['platform']==platform]
    print('\n')
```

```
Дисперсия PS4 : 2.122
Стандартное отклонение PS4 : 1.457
Среднее PS4 : 6.748
Медиана PS4 : 7.0


Дисперсия XONE : 1.897
Стандартное отклонение XONE : 1.377
Среднее XONE : 6.521
Медиана XONE : 6.8


Дисперсия 3DS : 2.339
Стандартное отклонение 3DS : 1.529
Среднее 3DS : 6.976
Медиана 3DS : 7.3
```

## Посчитаем дисперсию, стандартное отклонение, среднее и медиану у топ 5 платформ к оценкам критиков

In [83]:
```
#Посчитаем дисперсию, стандартное отклонение, среднее и медиану у топ 5 платформ к оценкам крит
for platform in list_of_top3:
    print('Дисперсия', platform.upper(),':', np.var(current_period[current_period['platform']==
    print('Стандартное отклонение', platform.upper(),':', np.std(current_period[current_period[
    print('Среднее',platform.upper(),':',  current_period[current_period['platform']==platform]
    print('Медиана',platform.upper(),':',  current_period[current_period['platform']==platform]
    print('\n')
```

```
Дисперсия PS4 : 155.281
Стандартное отклонение PS4 : 12.461
Среднее PS4 : 72.091
Медиана PS4 : 73.0


Дисперсия XONE : 166.799
Стандартное отклонение XONE : 12.915
Среднее XONE : 73.325
Медиана XONE : 76.0


Дисперсия 3DS : 169.012
Стандартное отклонение 3DS : 13.0
Среднее 3DS : 68.338
Медиана 3DS : 69.0
```

## Посмотрим на общее распределение игр по жанрам

**Посмотрим на общее распределение игр по жанрам. Раберемся,что можно сказать о самых прибыльных жанрах и выделяются ли жанры с высокими и низкими продажами.**

In [53]:
```
# Методом сводных таблиц выведем жанры и их продажи.  отсортируем по убыванию.
genres_of_games = current_period.pivot_table(
    index='genre', values='total_sales', aggfunc='sum').sort_values(by='total_sales', ascending
genres_of_games = genres_of_games.reset_index().rename_axis(None, axis=1)
genres_of_games
```

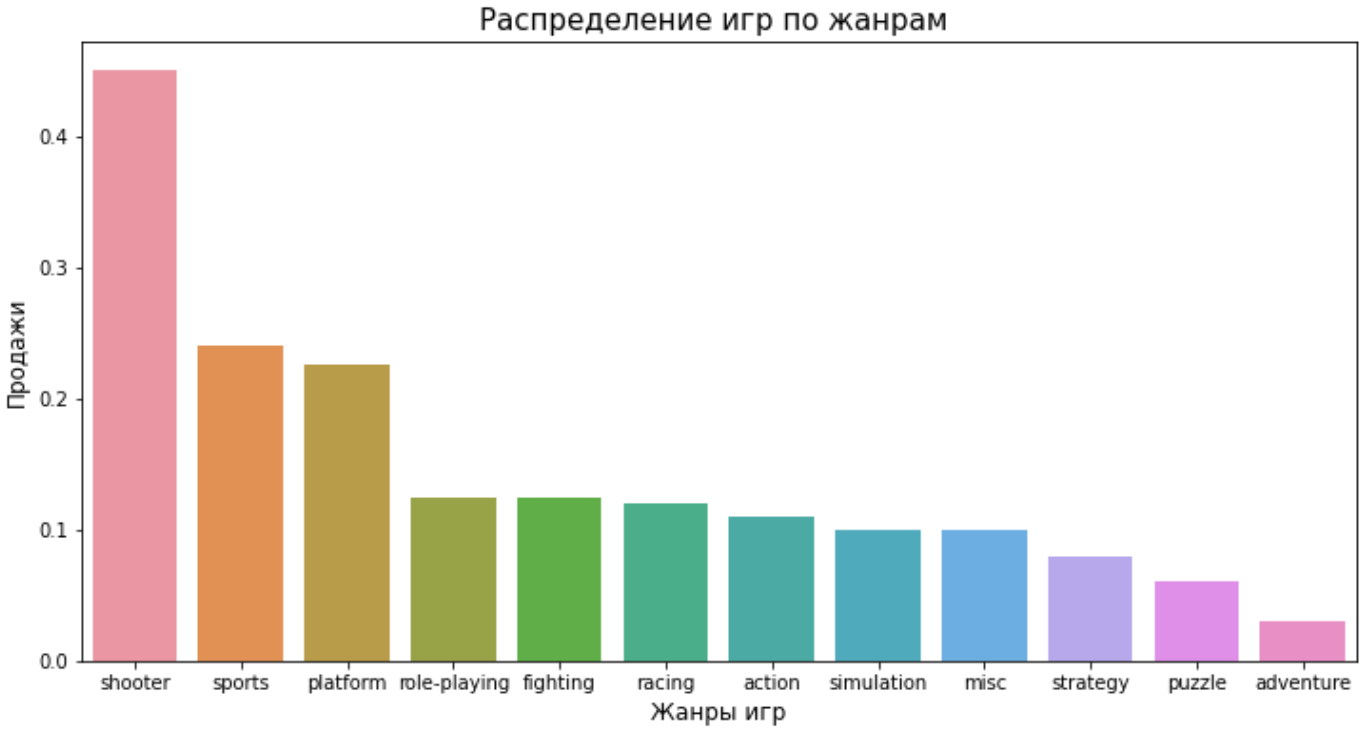
Out[53]:

| | genre | total_sales |
|---|---|---|
| 0 | action | 321.87 |
| 1 | shooter | 232.98 |
| 2 | sports | 150.65 |
| 3 | role-playing | 145.89 |
| 4 | misc | 62.82 |
| 5 | platform | 42.63 |
| 6 | racing | 39.89 |
| 7 | fighting | 35.31 |
| 8 | adventure | 23.64 |
| 9 | simulation | 21.76 |
| 10 | strategy | 10.08 |
| 11 | puzzle | 3.17 |

**Отрисуем барплот**

In [81]:
```
#Отрисуем барплот чтобы наглядно посмотреть какие жанры лидирует, а какие остаются внизу
```

```
plt.figure(figsize=(12,6))
plt.title('Распределение игр по жанрам ',fontsize=15)
sns.barplot(data=genres_of_games, x='genre', y='total_sales')
plt.xlabel('Жанры игр',fontsize=12)
plt.ylabel('Продажи',fontsize=12)
plt.show()
```




In [55]:

```
# Методом сводных таблиц выведем жанры и их продажи по медиане.  отсортируем по убыванию.
genres_of_games = current_period.pivot_table(
    index='genre', values='total_sales', aggfunc='median').sort_values(by='total_sales', ascend
genres_of_games = genres_of_games.reset_index().rename_axis(None, axis=1)
genres_of_games
```

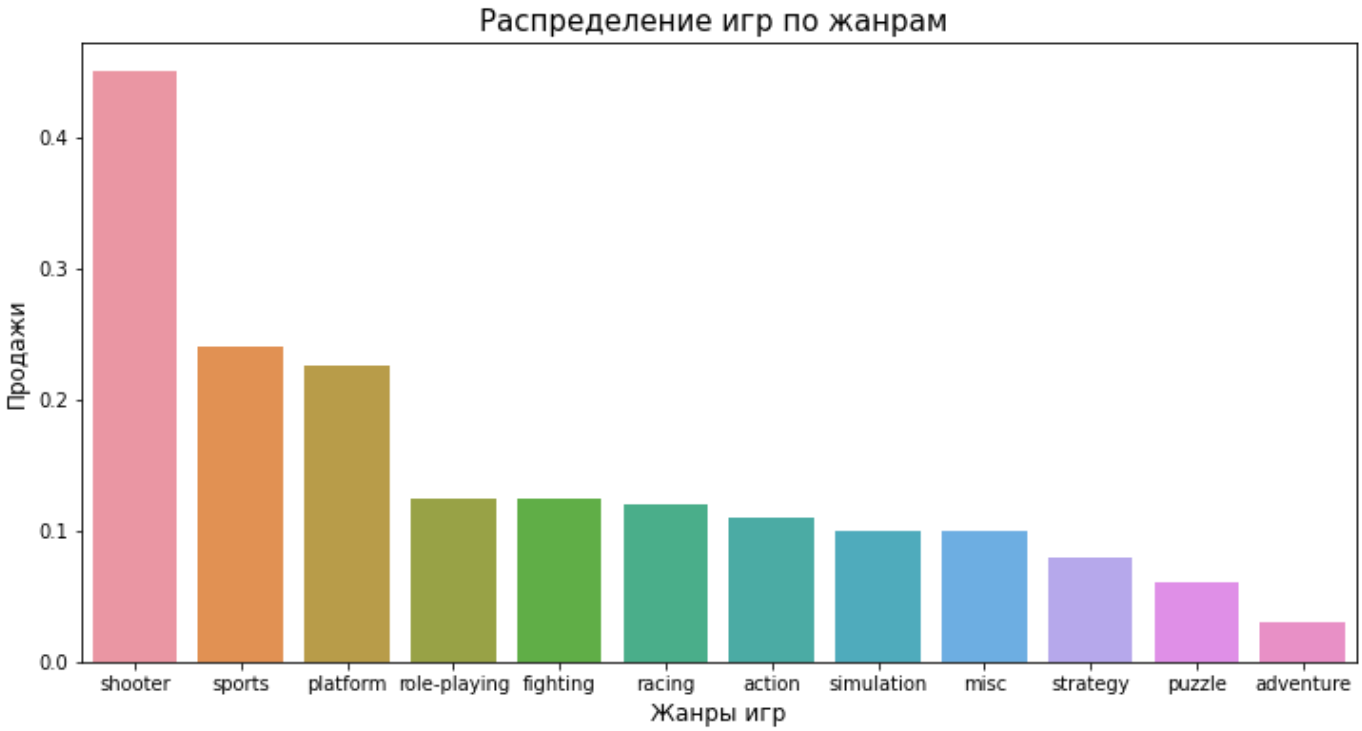
Out[55]:

| | genre | total_sales |
|---|---|---|
| 0 | shooter | 0.450 |
| 1 | sports | 0.240 |
| 2 | platform | 0.225 |
| 3 | role-playing | 0.125 |
| 4 | fighting | 0.125 |
| 5 | racing | 0.120 |
| 6 | action | 0.110 |
| 7 | simulation | 0.100 |
| 8 | misc | 0.100 |
| 9 | strategy | 0.080 |
| 10 | puzzle | 0.060 |
| 11 | adventure | 0.030 |

In [56]:

```
#Отрисуем барплот чтобы наглядно посмотреть какие жанры лидирует, а какие остаются внизу( по ме
plt.figure(figsize=(12,6))
plt.title('Распределение игр по жанрам ',fontsize=15)
sns.barplot(data=genres_of_games, x='genre', y='total_sales')
plt.xlabel('Жанры игр',fontsize=12)
```

```
plt.ylabel('Продажи',fontsize=12)
plt.show()
```




Если взять медианные продажи, то картина меняется. На первом месте shooter , затем спорт , платформ и тд. Action переместилась в середину списка. Последний - adventure.

In [57]:
```
df_top_3_platforms['genre'].describe()
```

Out[57]:
```
count        820
unique        12
top       action
freq         314
Name: genre, dtype: object
```

Лучше всего продаются жанры shooter,sports,платформ ,ролевые. На последнем месте - adventure.

# Вывод по исследовательскому анализу данных

Вывод по исследовательскому анализу данных: Количество игр на игровые приставки и компьютеры начало расти с большой скоростью с 1992 года до 2009 года. Данные заканчиваются 2016 годом. С 2009 года, после того как массово начали создавать мобильные приложения и мобильные игры, виден резкий спад консольных игр. Самые популярные игровые платформы за весь период : Sony PlayStation 2, Xbox 360, Sony Playstation 3, Nintendo WII, Nintendo DS, Sony Playstation.В основном,после выхода платформы идет рост продаж примерно до 4 лет. Потом резкий спад. .Примерно, срок жизни платформы 10 лет. Платформа PS2 прекратила продажи в 2011 году, а DS - в 2013г,psp в 2015 .Больше всего продаются игры на Sony Playstation и XONE.У всех платформ наблюдается отсутствие взаимосвязи между продажами и оценками пользователей. Заметнее всего корреляция между оценками критиков и продажами, но и она незначительна.Лучше всего продаются жанры shooter,sports,платформ ,ролевые. На последнем месте - adventure.

# Составим портрет пользователя каждого региона.

In [58]:
```
def for_pivot_2010(row, title):
    temp = data[data['year_of_release']>2010]
    fig, axes = plt.subplots(1, 3, figsize=(20, 4))
    for pivot, ax in zip(list(['platform','genre','rating']), axes.flatten()[:3]):
        ppivot = temp.pivot_table(index=pivot, values=row, aggfunc='median').sort_values(by=rov
        print(ppivot)
```

```
        print('\n\n')
        sns.set_palette("BuGn_r")
        sns.barplot(data=ppivot, x=pivot, y=row, ax=ax)
        fig.suptitle(title, fontsize=15)
```

## Портрет пользователя North America

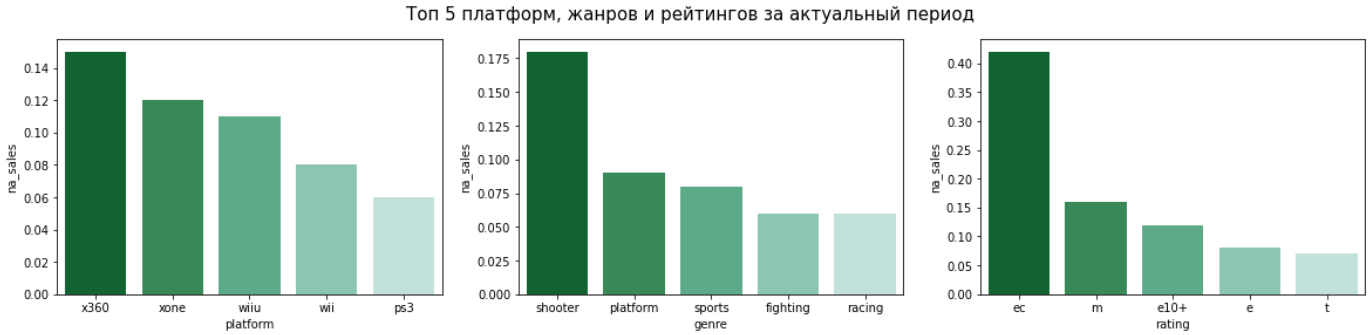
In [59]:

```
#Выведем топ 5 платформ, жанров и рейтингов за актуальный период
for_pivot_2010('na_sales','Топ 5 платформ, жанров и рейтингов за актуальный период')
```

```
  platform  na_sales
0     x360      0.15
1    xone       0.12
2     wiiu      0.11
3      wii      0.08
4      ps3      0.06
```

```
      genre  na_sales
0   shooter      0.18
1  platform      0.09
2    sports      0.08
3  fighting      0.06
4    racing      0.06
```

```
  rating  na_sales
0     ec      0.42
1      m      0.16
2   e10+      0.12
3      e      0.08
4      t      0.07
```




Портрет пользователя:

- Самые популярные платформы x360,xone и wiiu
- Самые популярные жанры shooter,platform и sports
- Самые популярные игры в возрастных категориях : "для младшего возраста". На игры "от 17 лет" и "после 10" приходится гораздо меньше продаж.

## Портрет пользователя European Union

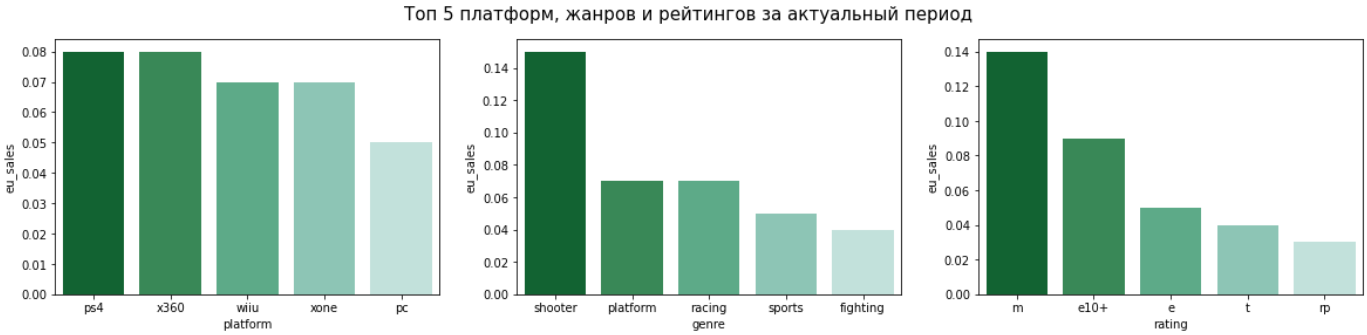
In [60]:

```
#Выведем топ 5 платформ, жанров и рейтингов за актуальный период
for_pivot_2010('eu_sales','Топ 5 платформ, жанров и рейтингов за актуальный период')
```

```
  platform  eu_sales
0      ps4      0.08
1     x360      0.08
2     wiiu      0.07
3     xone      0.07
4       pc      0.05
```

```
      genre  eu_sales
0   shooter      0.15
1  platform      0.07
2    racing      0.07
3    sports      0.05
4  fighting      0.04
```

```
  rating  eu_sales
0      m      0.14
1   e10+      0.09
2      e      0.05
3      t      0.04
4     rp      0.03
```




Портрет пользователя:

- Самые популярные платформы ps4, x360 и wiiu
- Самые популярные жанры shooter ,platform и racing
- Самые популярные игры в возрастных категориях : "после 17","для младшего возраста" и "для всех" приходится гораздо меньше продаж .

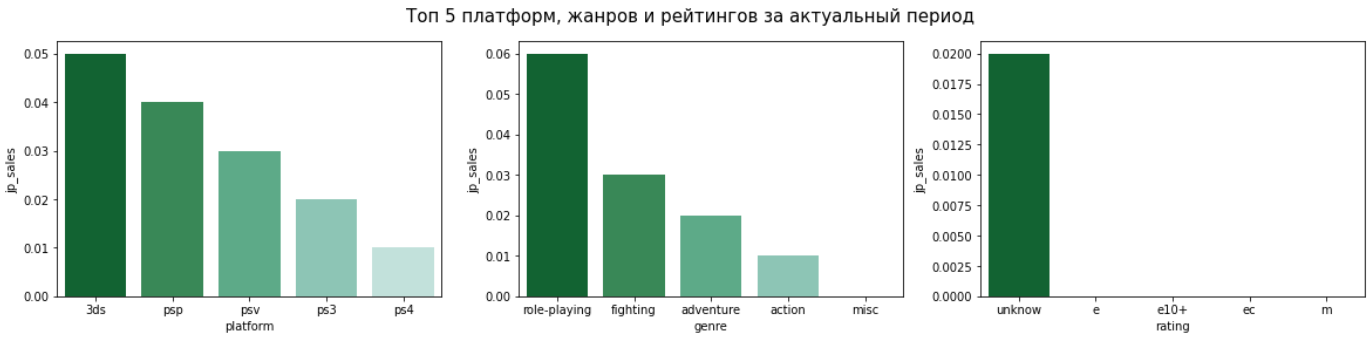
## Портрет пользователя Japanese

In [61]:
```
#Выведем топ 5 платформ, жанров и рейтингов за актуальный период
for_pivot_2010('jp_sales','Топ 5 платформ, жанров и рейтингов за актуальный период')
```

```
  platform  jp_sales
0      3ds      0.05
1      psp      0.04
2      psv      0.03
3      ps3      0.02
4      ps4      0.01
```

```
          genre  jp_sales
0  role-playing      0.06
1      fighting      0.03
2     adventure      0.02
3        action      0.01
4          misc      0.00
```

```
   rating  jp_sales
0  unknow      0.02
1       e      0.00
2    e10+      0.00
3      ec      0.00
4       m      0.00
```




Портрет пользователя:

- Самые популярные платформы 3ds,psp и psv
- Самые популярные жанры role-playing, fighting и adventure
- Все игры с неуказанным рейтингом.

Похоже, что японцы больше любят портативные консоли и родной рынок. И совсем не любят шутеры.

Япония снова отличилась. Скорее всего, дело тут в том, что ESRB работает только на территории СА, в Японии есть аналогичная организация: CERO. Я думаю, что, с одной стороны, иностранным играм они (ESRB) не присваивают рейтинги, поскольку на них уже есть маркировка. Чтобы не было конфликта, так сказать. Так что вполне возможно, что часть игр это продукция Японии или же это корейские игры (там тоже своя организация). С другой стороны, раз они продают игры на своем рынке, то присвоение рейтинга может быть обязательным. Значит, дело еще может быть в том, что наша таблица это склейка двух таблиц: продажи на Западе и на Востоке. Так или иначе, это очень показательный пример. И именно разница в рынках (восточный и западный) наталкивает на мысль о неслучайности пропусков.

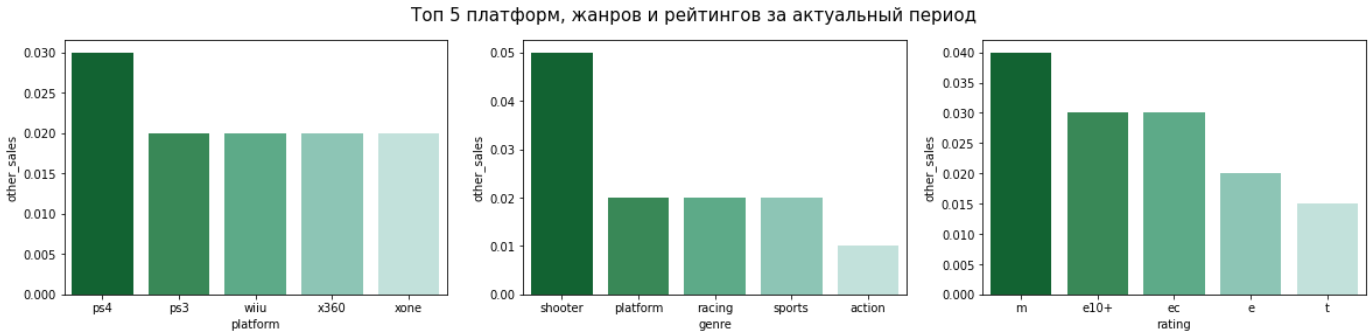
## Портрет пользователя в других странах

In [62]:
```
#Выведем топ 5 платформ, жанров и рейтингов за актуальный период
for_pivot_2010('other_sales','Топ 5 платформ, жанров и рейтингов за актуальный период')
```

```
  platform  other_sales
0      ps4         0.03
1      ps3         0.02
2     wiiu         0.02
3     x360         0.02
4     xone         0.02
```

```
      genre  other_sales
0   shooter         0.05
1  platform         0.02
2    racing         0.02
3    sports         0.02
4    action         0.01
```

```
  rating  other_sales
0      m        0.040
1   e10+        0.030
2     ec        0.030
3      e        0.020
4      t        0.015
```




Портрет пользователя:

- Самые популярные платформы ps4 ,ps3 и wiiu
- Самые популярные жанры shooter,platform и racing
- Самые популярные игры в возрастных категориях : "после 17". Далее идут "после 10" и "для младшего возраста" .

## Общий вывод по составлению портрета пользователя каждого региона

Общий вывод по составлению портрета пользователя каждого региона: Большой объем продаж приходится на игры с необозначенным рейтингом.Так как в этих странах строгое законодательство, можно предположить, что это те же игры " для всех".

Самые популярные платформы в Северной Америке x360,xone и wiiu Самые популярные жанры в Северной Америке : shooter ,platform и racing. По рейтингам видно, что самые популярные игры в возрастных категориях : "для младшего возраста". На игры "от 17 лет" и "после 10" приходится гораздо меньше продаж.

Самые популярные платформы в Европе это: ps4, x360 и wiiu Самые популярные жанры shooter ,platform и racing Самые популярные игры в возрастных категориях : "после 17","для младшего возраста" и "для всех" приходится гораздо меньше продаж .

Самые популярные платформы в Японии это:3ds,psp и psv Самые популярные жанры role-playing, fighting и adventure Все игры с неуказанным рейтингом.

Самые популярные платформы в других странах: ps4 ,ps3 и wiiu Самые популярные жанры shooter,platform и racing Самые популярные игры в возрастных категориях : "после 17". Далее идут "после 10" и "для младшего возраста" .

Для прогноза будущих продаж лучше брать данные за последний год - два. Технологии меняются с оч быстрыми темпами, и вкусы людей также могут меняться оч быстро. Соответственно, для 2017 года большую часть прибыли будут приносить игры жанра shooter , для платформ PS4,3ds и x360, с рейтингом от 17 и выше.

# Проверим гипотезы

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

Сформулируем гипотезы:

Нулевая гипотеза $H_0$: Средние пользовательские рейтинги платформ Xbox One и PC одинаковые; Альтернативная гипотеза $H_1$: Средние пользовательские рейтинги платформ Xbox One и PC различаются

Пороговое значение alpha зададим 0.05(критический уровень статистической значимости). если p-value окажется меньше него - отвергнем гипотезу

In [63]:
```
#Перед проверкой гипотезы проверим дисперсии выборок
for platform in list_of_top3:
    print('Дисперсия', platform.upper(),':', np.var(data[data['genre']=='action']['user_score']
```

```
Дисперсия PS4 : 2.026
Дисперсия XONE : 2.026
Дисперсия 3DS : 2.026
```

In [64]:
```
for platform in list_of_top3:
    print('Дисперсия', platform.upper(),':', np.var(data[data['genre']=='sports']['user_score']
```

```
Дисперсия PS4 : 2.621
Дисперсия XONE : 2.621
Дисперсия 3DS : 2.621
```

In [65]:
```
for platform in list_of_top3:
    print('Дисперсия', platform.upper(),':', np.var(data[data['platform']=='xone']['user_score'
```

```
Дисперсия PS4 : 1.897
Дисперсия XONE : 1.897
Дисперсия 3DS : 1.897
```

In [66]:
```
for platform in list_of_top3:
    print('Дисперсия', platform.upper(),':', np.var(data[data['platform']=='pc']['user_score'])
```

```
Дисперсия PS4 : 2.346
Дисперсия XONE : 2.346
Дисперсия 3DS : 2.346
```

## Первая гипотеза

In [67]:
```
# Сохраним в переменных x_one_hypotheses и pc_hypotheses соответствующие данные (актуальные дан
x_one_hypotheses = data[(data['platform']=='xone') & (data['year_of_release']>2010)]['user_scor
pc_hypotheses = data[(data['platform']=='pc') & (data['year_of_release']>2010)]['user_score']
```

In [74]:
```
#Посчитаем средний рейтинг пользователя для xbox платформ
x_one_hypotheses.mean().round(3)
```

Out[74]:
```
6.521
```

In [75]:
```
#Посчитаем средний рейтинг пользователя для PC платформ
pc_hypotheses.mean().round(3)
```

Out[75]:
```
6.452
```

In [76]:
```
#Выполним проверку гипотезы. Будем использовать метод ttest_ind

alpha = 0.05

results = st.ttest_ind(x_one_hypotheses.dropna(), pc_hypotheses.dropna())

print('p-значение:', results.pvalue)


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

```
p-значение: 0.6267602271422398
Не получилось отвергнуть нулевую гипотезу
```

Вывод: Значение p-value более 62% . Таким образом, не получилось опровергнуть Нулевую гипотезу. То есть, с вероятностью в 62% рейтинги двух платформ равны.

## Вторая гипотеза

Сформулируем гипотезы:

Нулевая гипотеза $H_0$: Средние пользовательские рейтинги жанров Action и Sports одинаковые.
Альтернативная гипотеза $H_1$: Средние пользовательские рейтинги жанров Action и Sports различаются

In [77]:
```
# Сохраним в переменных genre_action_hypotheses и genre_sports_hypotheses соответствующие данны
genre_action_hypotheses = data[(data['genre']=='action') & (data['year_of_release']>2010)]['use
genre_sports_hypotheses = data[(data['genre']=='sports') & (data['year_of_release']>2010)]['use

#выведем среднюю оценку по жанру экшн
genre_action_hypotheses.mean().round(3)
```

Out[77]: 6.776

In [78]:
```
#выведем среднюю оценку по жанру спорт
genre_sports_hypotheses.mean().round(3)
```

Out[78]: 5.651

In [80]:
```
#Выполним проверку гипотезы. Будем использовать метод ttest_ind

alpha = 0.05

results = st.ttest_ind(genre_action_hypotheses.dropna(), genre_sports_hypotheses.dropna())

print('p-значение:', results.pvalue)


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

```
p-значение: 5.1974550252152054e-24
Отвергаем нулевую гипотезу
```

Вывод: Получив p-value, мы отвергли Нулевую гипотезу. Средние рейнтинги по двум жанрам не равны.

## Вывод по проверке гипотез

Вывод по проверке гипотез

Проверили гипотезы :

- Использовали метод "scipy.stats.ttest_ind (array1, array2, equal_var)." - Гипотеза о равенстве средних двух генеральных совокупностей, тк у нас две совокупности.
- Гипотеза: "Средние пользовательские рейтинги платформ Xbox one и PC одинаковые". Нулевую гипотезу не удалось опровергнуть.
- Гипотеза: "Средние пользовательские рейтинги жанров Action и Sports одинаковые". Отвергаем нулевую гипотезу.

# Общий вывод

Перед анализом данных, мы провели подготовку наших данных,привели к правильным данным столбцы, привели к нижнему регистру строки таблицы и сами названия колонок. Привели к правильным типам данных необходимые столбцы.В процессе обработки столкнулись с аббревиатурой TBD (to be determined, to be done).То есть данные по отзывам пользователей были нарочно не заполнены.Поэтому решили заменить tbd на nan для проведения дальнейшего анализа.Пропуски в столбце Рейтинг заполнили заглушкой unknow. Это позволило провести анализ повсем данным столбца.Проведя анализ, мы выявили, что количество игр на игровые приставки и компьютеры начало расти с большой скоростью с 90х до 2009 года. С 2009 года, после того как массово начали создавать мобильные приложения и мобильные игры, произошел резкий спад разработок консольных игр.За весь представленный период среди консольных приставок самые популярные оказались: PS2, X360, PS3, WII, DS, PS.Также мы выявили, что средняя продолжительность жизни игровой приставки составляет порядка 10 лет.Пик продаж наступает примерно через 4 года после выпуска консоли.Так как технологии и пристрастия пользователей меняются, приняли решение проводить анализ не за весь период, а с 2010 года, когда начался рекий спад продаж.Проведя анализ оценок пользователей и критиков, мы выявили, что оценки критиков и пользователей не влияют на продажами самих игр. Так же определили, что самые популярные жанры за все время shooter,sports,платформ ,ролевые. На последнем месте - adventure.

Затем мы составили портреты пользователей каждого региона. Выяснили, что в Северной америке самые популярные жанры это action, sports, shooter. Игровые приставки : x360,ps3 и ps4. По рейтингам определили, что самые популярные игры в возрастных категориях : "после 17". На игры "для всех" и "после 10" приходится почти в 2 раза меньше продаж.Игры с необозначенным рейтингом занимают третье место

В Европе же, популярные жанры в другом порядке: action,shooter и sports. Но приставки по предпочитают по следующей очередности: ps3 ,ps4 и x360 . Рейтинг игр отличается : самые популярные игры в возрастных категориях : "после 17". На игры "для всех" и "после 13" приходится гораздо меньше продаж.Игры с необозначенным рейтингом занимают так же третье место. . В японии люди предпочитают игровые приставки:3ds и ps3. Самые популярные жанры role-playing, action Большинство игр с неуказанным рейтингом, намного меньше в возрастных категориях : "для всех","после 13" и "после 17"

Самые популярные платформы в других странах: ps3 ,ps4 и x360 Самые популярные жанры action,shooter и sports Самые популярные игры в возрастных категориях : "после 17". На игры "для всех" и "после 13" приходится гораздо меньше продаж.Игры с необозначенным рейтингом занимают третье место.

Провели проверку 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. В итоге, обе гипотезы подтвердились.

Исходя из всех данных предполагаем, что лучше всего в 2017 году продавать игры для таких приставок как Sony Playstation 4 Жанр необходимо выбирать shooter,sports,платформ и выбирать игры с рейтингом "от 17 и выше", тогда продажи будут значительно больше.