# Анализ продаж видеоигр

## Описание проекта

Из открытых источников доступны исторические данные о продажах игр, оценки пользователей и экспертов, жанры и платформы (например, Xbox или PlayStation).

Нам нужно выявить определяющие успешность игры закономерности.

Это позволит сделать ставку на потенциально популярный продукт и спланировать рекламные кампании.

Перед нами данные за 2016 год на основании которых нам необходимо составить прогноз на 2017 год.

## Изучение общей информации

In [1]:
```
import pandas as pd
import numpy as np
from scipy import stats as st
import matplotlib.pyplot as plt
import seaborn as sns

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

In [2]:
```
df = pd.read_csv('/datasets/games.csv')
```

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16715 entries, 0 to 16714
Data columns (total 11 columns):
Name               16713 non-null object
Platform           16715 non-null object
Year_of_Release    16446 non-null float64
Genre              16713 non-null object
NA_sales           16715 non-null float64
EU_sales           16715 non-null float64
JP_sales           16715 non-null float64
Other_sales        16715 non-null float64
Critic_Score       8137 non-null float64
User_Score         10014 non-null object
Rating             9949 non-null object
dtypes: float64(6), object(5)
memory usage: 1.4+ MB
```

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

Out[4]:

| | 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 | 8.45 |
| 1 | Super Mario Bros. | NES | 1985.0 | Platform | 29.08 | 3.58 | 6.81 | 0.77 |
| 2 | Mario Kart Wii | Wii | 2008.0 | Racing | 15.68 | 12.76 | 3.79 | 3.29 |
| 3 | Wii Sports Resort | Wii | 2009.0 | Sports | 15.61 | 10.93 | 3.28 | 2.95 |
| 4 | Pokemon Red/Pokemon Blue | GB | 1996.0 | Role-Playing | 11.27 | 8.89 | 10.22 | 1.00 |

In [5]:
```
df.tail()
```

Out[5]:

| | Name | Platform | Year_of_Release | Genre | NA_sales | EU_sales | JP_sales | Other_sa |
|---|---|---|---|---|---|---|---|---|
| 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 [6]:
```
df.duplicated().sum()
```

Out[6]:
```
0
```

При осмотре данных дубликатов не обнаружено, однако имеется большое количество пропусков в столбцах года выпуска игра, рейтингу критиков, пользователей и рейтинге по международной классификации. Вероятно, по не особо популярным релизам информация в базе данных отсутствует

# Подготовка данных

In [7]:
```
display(df.columns)
```

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

In [8]:
```
df.columns = df.columns.str.lower()
```

In [9]:
```
df.head()
```

Out[9]:

| | name | platform | year_of_release | genre | na_sales | eu_sales | jp_sales | other_sales | criti |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Wii Sports | Wii | 2006.0 | Sports | 41.36 | 28.96 | 3.77 | 8.45 | |
| 1 | Super Mario Bros. | NES | 1985.0 | Platform | 29.08 | 3.58 | 6.81 | 0.77 | |
| 2 | Mario Kart Wii | Wii | 2008.0 | Racing | 15.68 | 12.76 | 3.79 | 3.29 | |
| 3 | Wii Sports Resort | Wii | 2009.0 | Sports | 15.61 | 10.93 | 3.28 | 2.95 | |
| 4 | Pokemon Red/Pokemon Blue | GB | 1996.0 | Role-Playing | 11.27 | 8.89 | 10.22 | 1.00 | |

In [10]:
```
df['name'] = df['name'].str.lower()
df['platform'] = df['platform'].str.lower()
df['genre'] = df['genre'].str.lower()


df['critic_score'] = df['critic_score'].fillna(df['critic_score'].mean())

df.loc[df['user_score'] == "tbd", 'user_score'] = 'NaN'
df['user_score'] = df['user_score'].astype('float')
df['user_score'] = df['user_score'].fillna(df['user_score'].mean())

df['rating'] = df['rating'].fillna('undefined')
```

In [11]:
```
# example
df['user_score'].astype('float').fillna(df['user_score'].mean())
```

Out[11]:
```
0        8.000000
1        7.125046
2        8.300000
3        8.000000
4        7.125046
           ...
16710    7.125046
16711    7.125046
16712    7.125046
16713    7.125046
16714    7.125046
Name: user_score, Length: 16715, dtype: float64
```

In [12]:
```
df.head()
```

Out[12]:

| | name | platform | year_of_release | genre | na_sales | eu_sales | jp_sales | other_sales | critic |
|---|---|---|---|---|---|---|---|---|---|
| 0 | wii sports | wii | 2006.0 | sports | 41.36 | 28.96 | 3.77 | 8.45 | 76. |
| 1 | super mario bros. | nes | 1985.0 | platform | 29.08 | 3.58 | 6.81 | 0.77 | 68. |
| 2 | mario kart wii | wii | 2008.0 | racing | 15.68 | 12.76 | 3.79 | 3.29 | 82. |
| 3 | wii sports resort | wii | 2009.0 | sports | 15.61 | 10.93 | 3.28 | 2.95 | 80. |
| 4 | pokemon red/pokemon blue | gb | 1996.0 | role-playing | 11.27 | 8.89 | 10.22 | 1.00 | 68. |

Все наши действия выше по замене пропусков в столбцах с рейтингом были призваны дать возможность удалить пропуски в столбце с годом релиза игры, поскольку в дальнейшем нам этот столбец нам понадобится и оставлять его с пропусками не очень хорошо для нас.

Заменив нулями rating пропуски, мы ничего не лишаемся, поскольку по-прежнему понятно, что данные с нулями в этих столбцах получены в связи с отсутствием информации о рейтинге.

In [13]:
```
df = df.dropna().reset_index()
```

In [14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16444 entries, 0 to 16443
Data columns (total 12 columns):
index              16444 non-null int64
name               16444 non-null object
platform           16444 non-null object
year_of_release    16444 non-null float64
genre              16444 non-null object
na_sales           16444 non-null float64
eu_sales           16444 non-null float64
jp_sales           16444 non-null float64
other_sales        16444 non-null float64
critic_score       16444 non-null float64
user_score         16444 non-null float64
rating             16444 non-null object
dtypes: float64(7), int64(1), object(4)
memory usage: 1.5+ MB
```

In [15]:
```
df.head()
```

Out[15]:

| | index | name | platform | year_of_release | genre | na_sales | eu_sales | jp_sales | other_sales |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | wii sports | wii | 2006.0 | sports | 41.36 | 28.96 | 3.77 | 8.45 |
| 1 | 1 | super mario bros. | nes | 1985.0 | platform | 29.08 | 3.58 | 6.81 | 0.77 |
| 2 | 2 | mario kart wii | wii | 2008.0 | racing | 15.68 | 12.76 | 3.79 | 3.29 |
| 3 | 3 | wii sports resort | wii | 2009.0 | sports | 15.61 | 10.93 | 3.28 | 2.95 |
| 4 | 4 | pokemon red/pokemon blue | gb | 1996.0 | role-playing | 11.27 | 8.89 | 10.22 | 1.00 |

In [16]:
```
df['year_of_release'] = df['year_of_release'].astype('int')
df['year_of_release'] = pd.to_datetime(df['year_of_release'], format='%Y')
df['year_of_release'] = df['year_of_release'].dt.year

df.head()
```

Out[16]:

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

In [17]:
```
df.head()
```

Out[17]:

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

Тип столбца year_of_release был преобразован в datetime, так как в нашем будущем анализе может быть важно проводить операции с данным столбцом.

Аббревиатура "tbd" (в столбце user_score) означает неизвестное значение, которое должно быть заполнено позднее. Я обработал его заменив обычным пропуском для дальнейшего заполнения.

In [18]:
```
df['income'] = df['na_sales'] + df['eu_sales'] + df['jp_sales'] + df['other_sales

df.head()
```

Out[18]:

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

# Исследовательский анализ данных

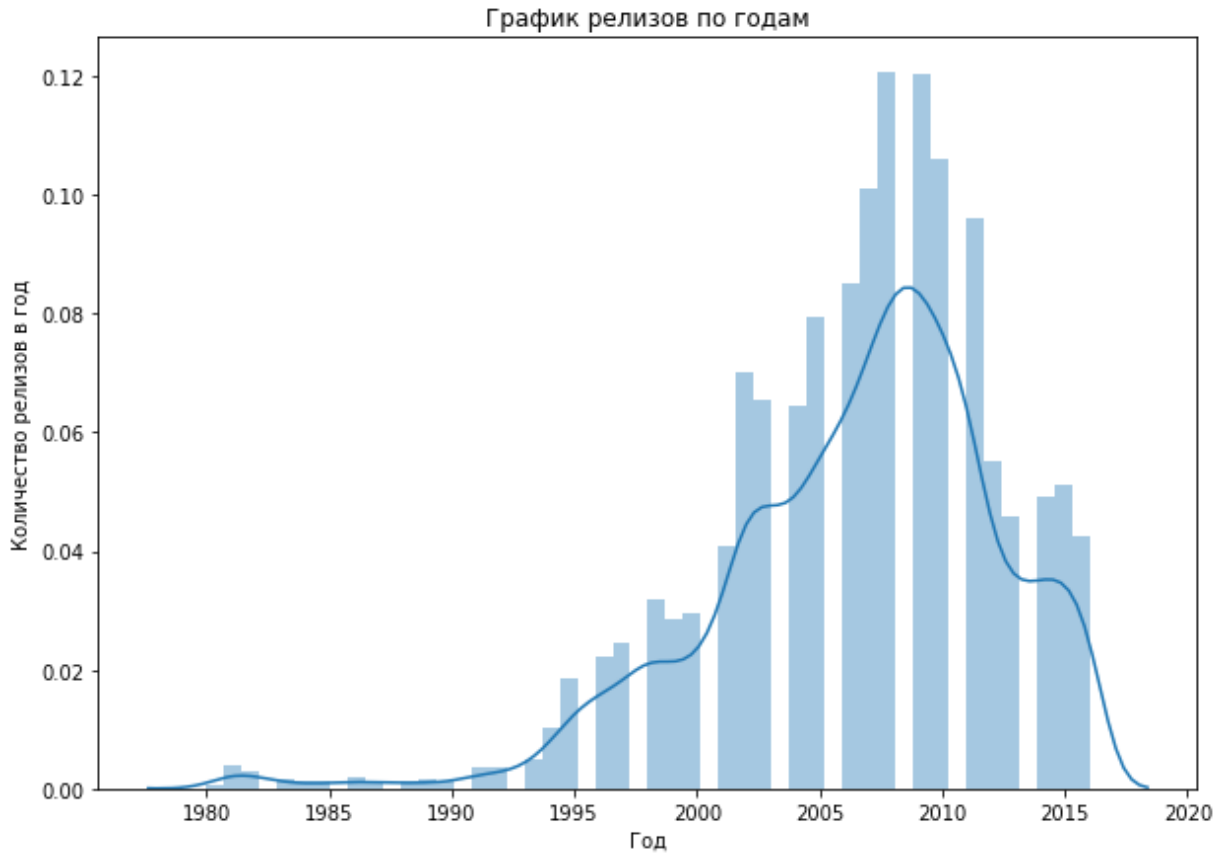
In [19]:
```
df.head()
```

Out[19]:

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

In [20]:
```
plt.figure(figsize=(10,7))
sns.distplot(df['year_of_release'], bins=50).set(title='График релизов по годам',
```

Out[20]:
```
[Text(0, 0.5, 'Количество релизов в год'),
 Text(0.5, 0, 'Год'),
 Text(0.5, 1.0, 'График релизов по годам')]
```




На графике мы видим, что докризисный период знаменует собой бурный рост индустрии видеоигр, однако наш период для анализа - это последние 5 лет (2012-2016), поскольку

общая обстановка на рынке по релизам похожа от года к году в течение данного периода. Соответственно, можно проводить анализ и прогнозировать следующие периоды.

In [21]:
```
#считаем количество релизов по годам
year_releases = pd.pivot_table(df,
                               index='year_of_release',
                               values='name',
                               aggfunc='count').reset_index()
display(year_releases)
```

| | year_of_release | name |
|---|---|---|
| 0 | 1980 | 9 |
| 1 | 1981 | 46 |
| 2 | 1982 | 36 |
| 3 | 1983 | 17 |
| 4 | 1984 | 14 |
| 5 | 1985 | 14 |
| 6 | 1986 | 21 |
| 7 | 1987 | 16 |
| 8 | 1988 | 15 |
| 9 | 1989 | 17 |
| 10 | 1990 | 16 |
| 11 | 1991 | 41 |
| 12 | 1992 | 43 |
| 13 | 1993 | 60 |
| 14 | 1994 | 121 |
| 15 | 1995 | 219 |
| 16 | 1996 | 263 |
| 17 | 1997 | 289 |
| 18 | 1998 | 379 |
| 19 | 1999 | 338 |
| 20 | 2000 | 350 |
| 21 | 2001 | 482 |
| 22 | 2002 | 829 |
| 23 | 2003 | 775 |
| 24 | 2004 | 762 |
| 25 | 2005 | 939 |
| 26 | 2006 | 1006 |
| 27 | 2007 | 1197 |
| 28 | 2008 | 1427 |
| 29 | 2009 | 1426 |
| 30 | 2010 | 1255 |

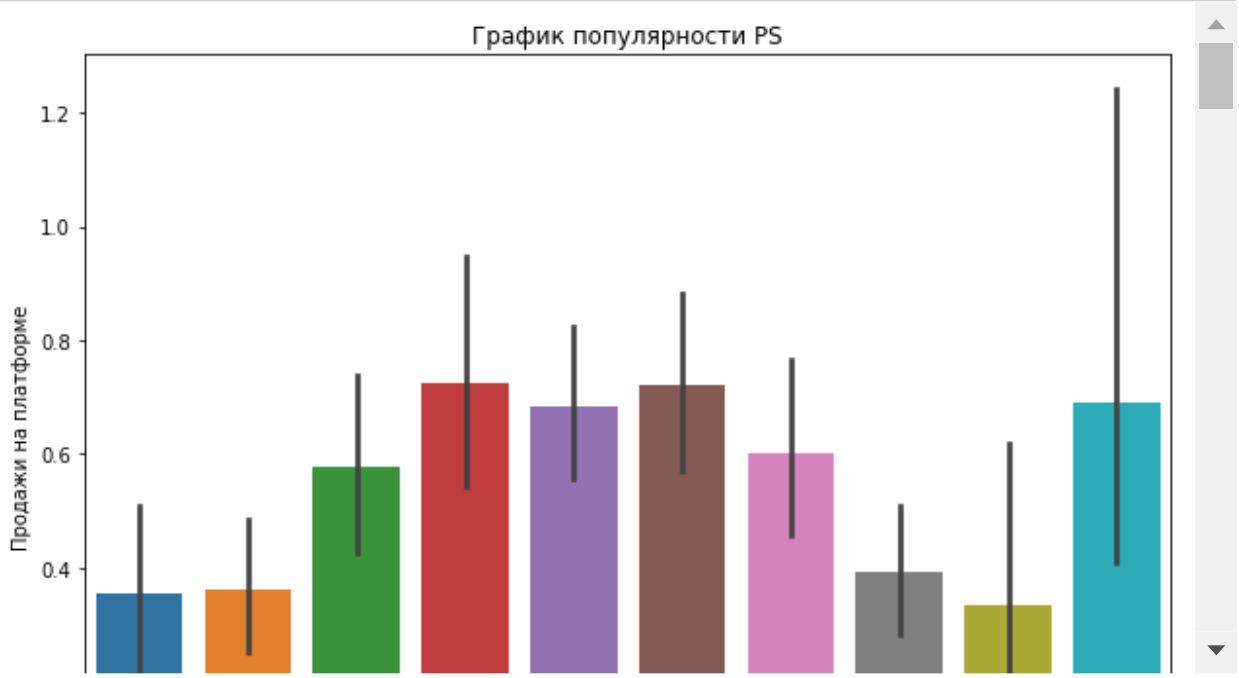
| | year_of_release | name |
|---|---|---|
| 31 | 2011 | 1136 |
| 32 | 2012 | 653 |
| 33 | 2013 | 544 |
| 34 | 2014 | 581 |
| 35 | 2015 | 606 |
| 36 | 2016 | 502 |

In [22]:
```
#платформы и продажи
platform_income = pd.pivot_table(df,
                                 index='platform',
                                 values='income',
                                 aggfunc='sum').reset_index()
display(platform_income.sort_values('income'))
```

| | platform | income |
|---|---|---|
| 14 | pcfx | 0.03 |
| 9 | gg | 0.04 |
| 1 | 3do | 0.10 |
| 24 | tg16 | 0.16 |
| 27 | ws | 1.42 |
| 12 | ng | 1.44 |
| 22 | scd | 1.86 |
| 3 | dc | 15.95 |
| 8 | gen | 28.35 |
| 21 | sat | 33.59 |
| 20 | psv | 53.81 |
| 26 | wiiu | 82.19 |
| 0 | 2600 | 86.48 |
| 30 | xone | 159.32 |
| 7 | gc | 196.73 |
| 23 | snes | 200.04 |
| 10 | n64 | 218.01 |
| 11 | nes | 251.05 |
| 29 | xb | 251.57 |
| 5 | gb | 254.43 |
| 13 | pc | 255.76 |
| 2 | 3ds | 257.81 |
| 19 | psp | 289.53 |
| 6 | gba | 312.88 |
| 18 | ps4 | 314.14 |
| 15 | ps | 727.58 |
| 4 | ds | 802.78 |
| 25 | wii | 891.18 |
| 17 | ps3 | 931.34 |
| 28 | x360 | 961.24 |
| 16 | ps2 | 1233.56 |

Берем 6 последних платформ для анализа на предмет вычисления периода популярности

In [23]:
```
ps = df.query('platform == "ps"')
ds = df.query('platform == "ds"')
wii = df.query('platform == "wii"')
ps3 = df.query('platform == "ps3"')
x360 = df.query('platform == "x360"')
ps2 = df.query('platform == "ps2"')

platform_list = {'PS':ps, 'Nintendo':ds, 'Wii':wii, 'PS3':ps3, 'Xbox360':x360, 'P

for name, value in platform_list.items():
    title_platform='График популярности ' + name
    plt.figure(figsize=(10,7))
    sns.barplot(data=value, x='year_of_release', y='income').set(title=title_plat
    plt.show()
```




Как видно из графиков, в отдельных случаях популярность платформы может длиться всего год.

Однако, в среднем, платформы ранних поколений в среднем такие, как PS и PS2 популярны около 5-6 лет, затем следует снижение продаж.

Популярность более поздних поколений платформ - PS3 и Xbox360 длилась намного больше их предшественниц - 8-9 лет. Это может быть связано с выпуском разных версий платформ и их модернизацией под запросы времени.

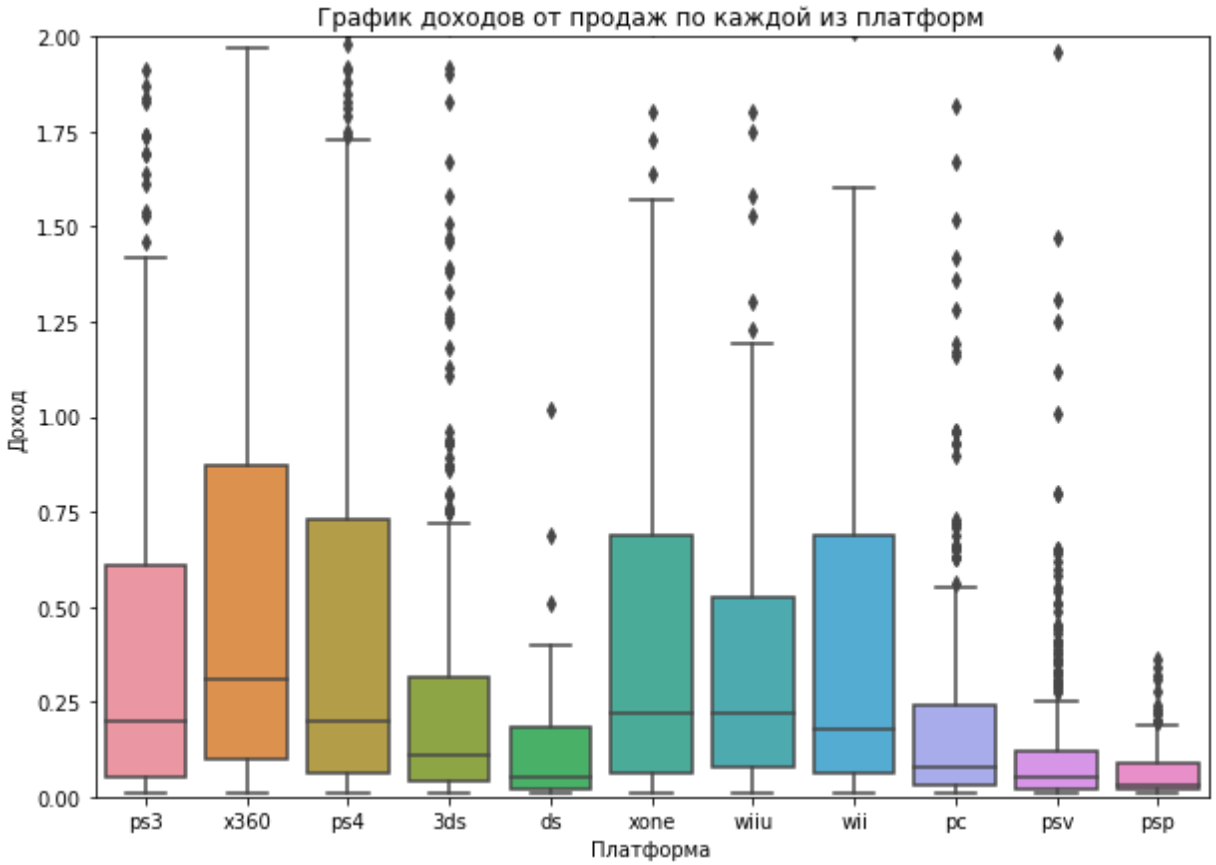
Поскольку большинство продаж среди самых популярных консолей принадлежит Xbox и ps, нам следует обратить внимание на следующее поколение этих консолей, поскольку Xbox360 и PS3 уже прожили свой пик популярности.

In [24]:
```
#считаем релизы в контексте платформ
platform_5y = df.query('year_of_release >= 2012')
platform = pd.pivot_table(platform_5y,
                                index=['platform'],
                                values='income',
                                aggfunc='count').sort_values('income').reset_index()
display(platform)
```

| | platform | income |
|---|---|---|
| 0 | ds | 31 |
| 1 | wii | 54 |
| 2 | wiiu | 147 |
| 3 | psp | 173 |
| 4 | xone | 247 |
| 5 | pc | 250 |
| 6 | x360 | 292 |
| 7 | ps4 | 392 |
| 8 | 3ds | 396 |
| 9 | psv | 411 |
| 10 | ps3 | 493 |

In [25]:
```
plt.figure(figsize=(10,7))
sns.boxplot(x=platform_5y['platform'], y=platform_5y['income']).set(ylim=(0,2))
plt.title('График доходов от продаж по каждой из платформ')
plt.xlabel('Платформа')
plt.ylabel('Доход')
```

Out[25]:
```
Text(0, 0.5, 'Доход')
```




С 2012 года сохраняют свою популярность консоли третьего поколения, однако перспективные платформы нового поколения как PS4 и Xbox One уже имеют большой хвост в сторону больших значений выручки и вскоре могут перегнать по продажам консоли третьего поколения.

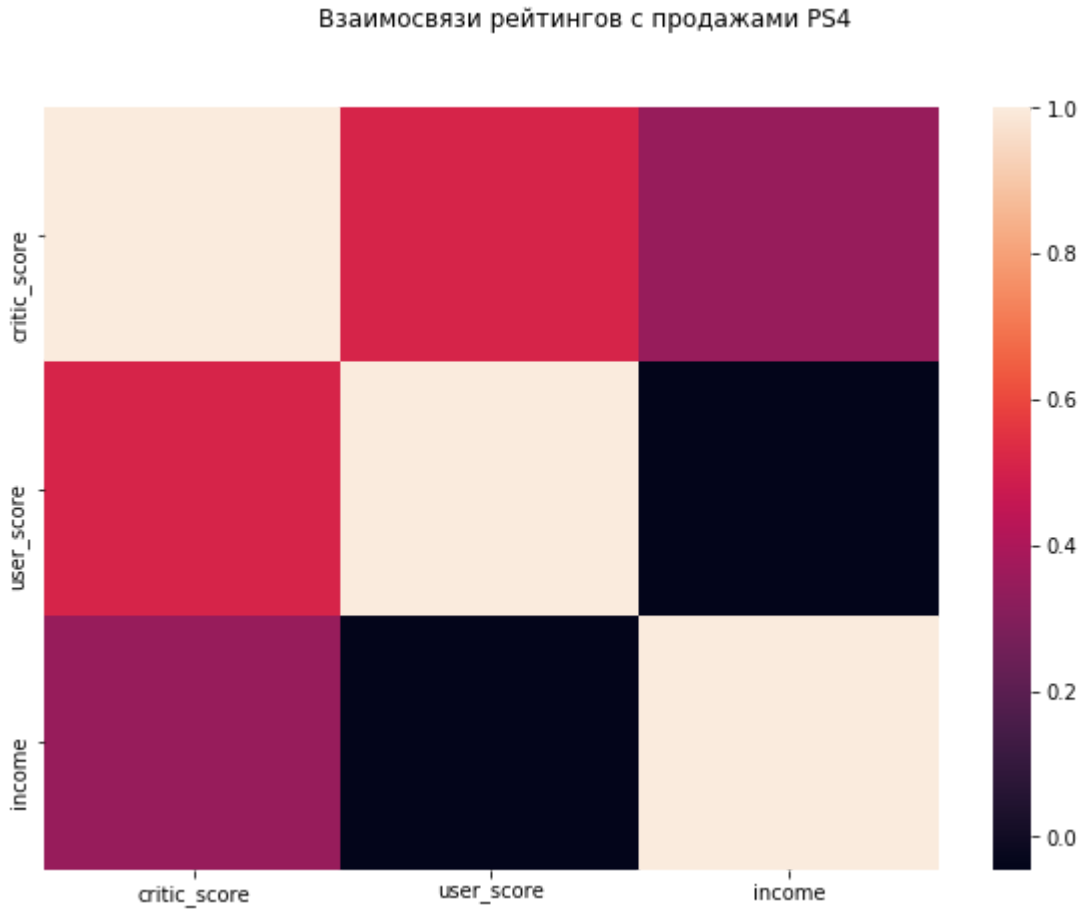
ПК остается на низком уровне по продажам.

In [26]:
```
corr_df = platform_5y.query('platform == "ps4"')
ps4_corr = corr_df[['critic_score', 'user_score', 'income']].corr()

display("Корреляция PS4", ps4_corr)
```

'Корреляция PS4'

| | critic_score | user_score | income |
|---|---|---|---|
| critic_score | 1.00000 | 0.513400 | 0.349010 |
| user_score | 0.51340 | 1.000000 | -0.045391 |
| income | 0.34901 | -0.045391 | 1.000000 |

In [27]:
```
plt.figure(figsize=(10,7))
sns.heatmap(ps4_corr)
plt.suptitle('Взаимосвязи рейтингов с продажами PS4')
plt.show()
```




Обнаружена крайне слабая отрицательная зависимость пользовательского рейтинга и продаж, а также средняя зависимость рейтинга критиков с продажами на платформе.

Рассмотрим вывод на примере корреляции по другим платформам

In [28]:
```
ps3 = platform_5y.query('platform == "ps3"')
ps3_corr = ps3[['critic_score', 'user_score', 'income']].corr()

display("Корреляция PS3", ps3_corr)
```

'Корреляция PS3'

| | critic_score | user_score | income |
|---|---|---|---|
| critic_score | 1.000000 | 0.396384 | 0.335720 |
| user_score | 0.396384 | 1.000000 | -0.057529 |
| income | 0.335720 | -0.057529 | 1.000000 |

In [29]:
```
psp = platform_5y.query('platform == "psp"')
psp_corr = psp[['critic_score', 'user_score', 'income']].corr()

display("Корреляция PSP", psp_corr)
```

'Корреляция PSP'

| | critic_score | user_score | income |
|---|---|---|---|
| critic_score | 1.000000 | 0.207725 | 0.091510 |
| user_score | 0.207725 | 1.000000 | -0.251851 |
| income | 0.091510 | -0.251851 | 1.000000 |

In [30]:
```
wii = platform_5y.query('platform == "wii"')
wii_corr = wii[['critic_score', 'user_score', 'income']].corr()

display("Корреляция Wii", wii_corr)
```

'Корреляция Wii'

| | critic_score | user_score | income |
|---|---|---|---|
| critic_score | 1.000000 | 0.178168 | 0.379920 |
| user_score | 0.178168 | 1.000000 | -0.019465 |
| income | 0.379920 | -0.019465 | 1.000000 |

In [31]:
```
x360 = platform_5y.query('platform == "x360"')
x360_corr = x360[['critic_score', 'user_score', 'income']].corr()

display("Корреляция Xbox360", x360_corr)
```

'Корреляция Xbox360'

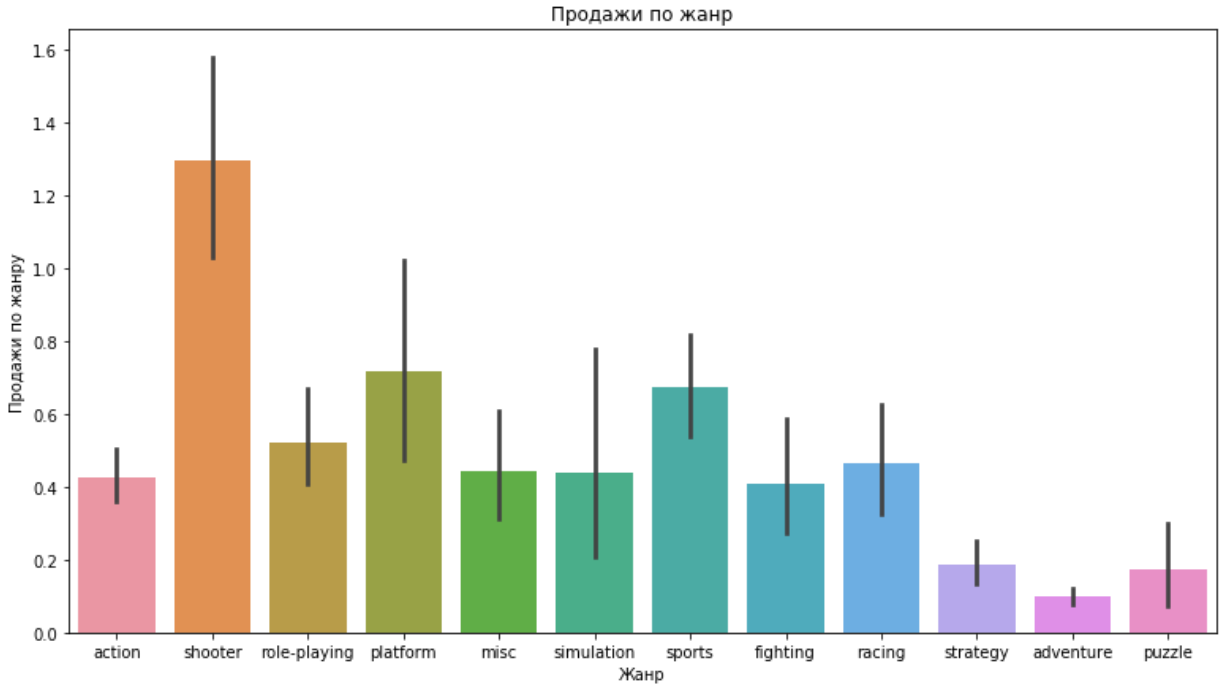
| | critic_score | user_score | income |
|---|---|---|---|
| critic_score | 1.000000 | 0.430835 | 0.341724 |
| user_score | 0.430835 | 1.000000 | -0.021819 |
| income | 0.341724 | -0.021819 | 1.000000 |

При рассмотрении взаимосвязей по другим платформам обнаружено следующее: чем менее популярна платформа - тем более пользовательский рейтинг влияет на продажи по сравнению с рейтингом критиков.

А также чем более популярна платформа, тем более рейтинг критиков влияет на продажи по сравнению с рейтингом пользователей.

По PSP обнаружена слабая отрицательная зависимость рейтинга пользователей к продажам.

In [32]:
```
plt.figure(figsize=(13,7))
genre = sns.barplot(data=platform_5y, x='genre', y='income')
sns.set(rc={'figure.figsize':(15,9)})
genre.set(title='Продажи по жанр', xlabel='Жанр', ylabel='Продажи по жанру')

plt.show()
```




Среди продаж по жанрам самыми прибыльными являются - Sports, Platform, Shooter.

Самыми низкими по продажам являются жанры - Strategy, Adventure

Эти жанры сильно выделяются на фоне остальных, в особенности жанр Shooter Можно сказать, что пользователям больше по душе наиболее активные игры.

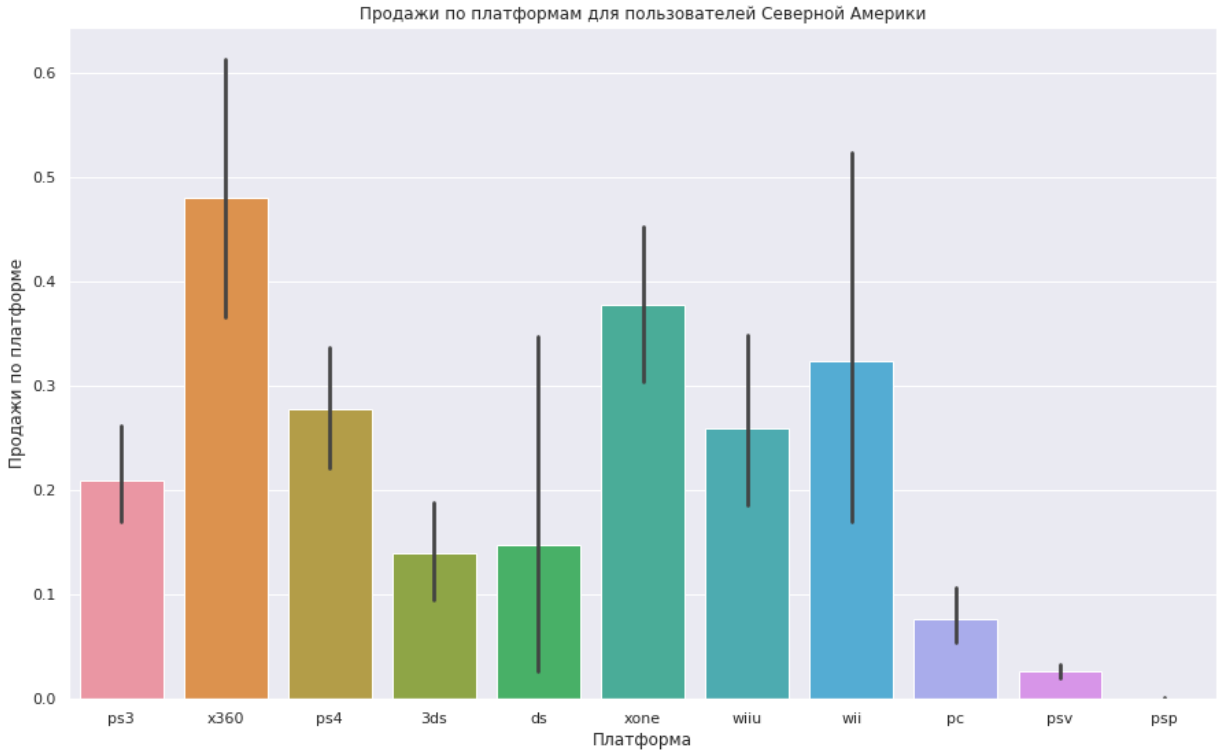
# Портрет пользователя каждого региона

In [33]:
```
platform_5y.head()
```

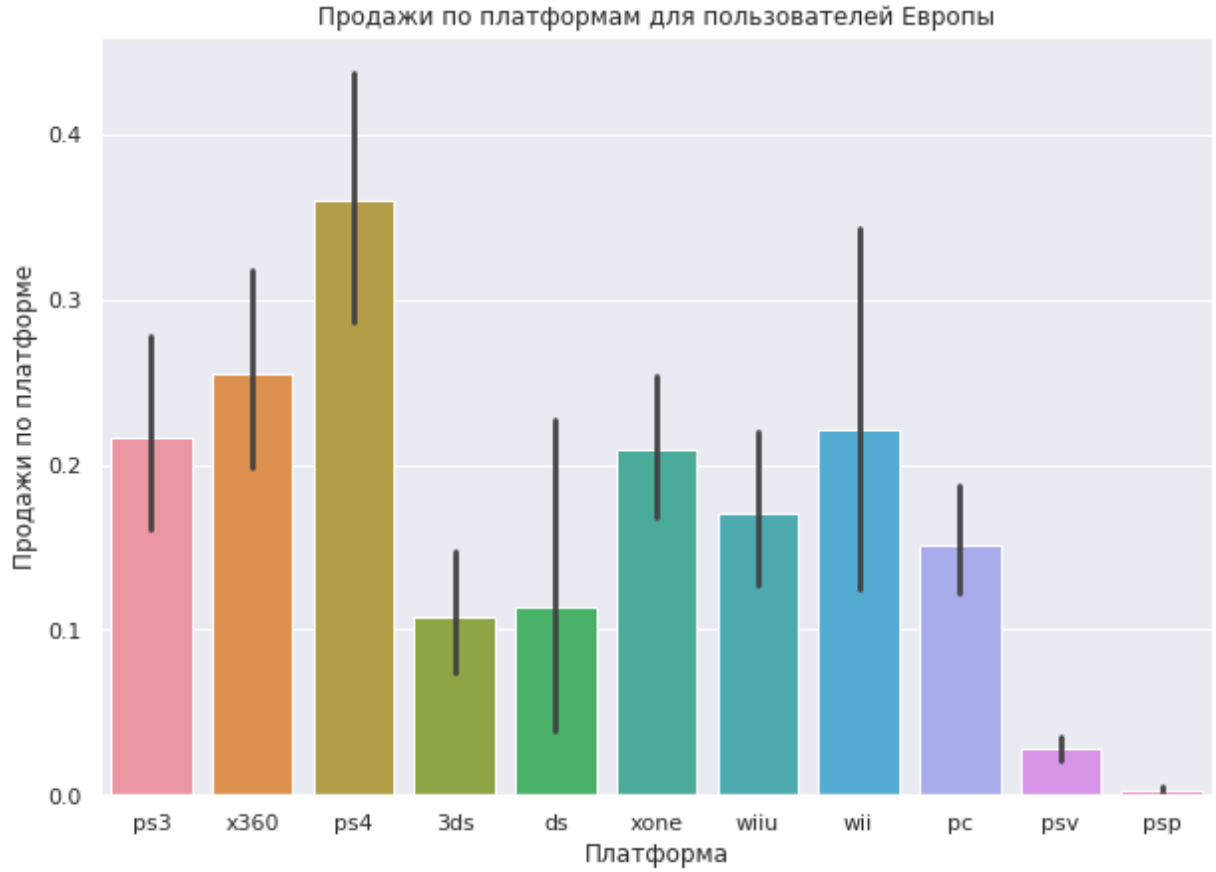
Out[33]:

| | index | name | platform | year_of_release | genre | na_sales | eu_sales | jp_sales | other_sales |
|---|---|---|---|---|---|---|---|---|---|
| 16 | 16 | grand theft auto v | ps3 | 2013 | action | 7.02 | 9.09 | 0.98 | 3.96 |
| 23 | 23 | grand theft auto v | x360 | 2013 | action | 9.66 | 5.14 | 0.06 | 1.41 |
| 31 | 31 | call of duty: black ops 3 | ps4 | 2015 | shooter | 6.03 | 5.86 | 0.36 | 2.38 |
| 33 | 33 | pokemon x/pokemon y | 3ds | 2013 | role-playing | 5.28 | 4.19 | 4.35 | 0.78 |
| 34 | 34 | call of duty: black ops ii | ps3 | 2012 | shooter | 4.99 | 5.73 | 0.65 | 2.42 |

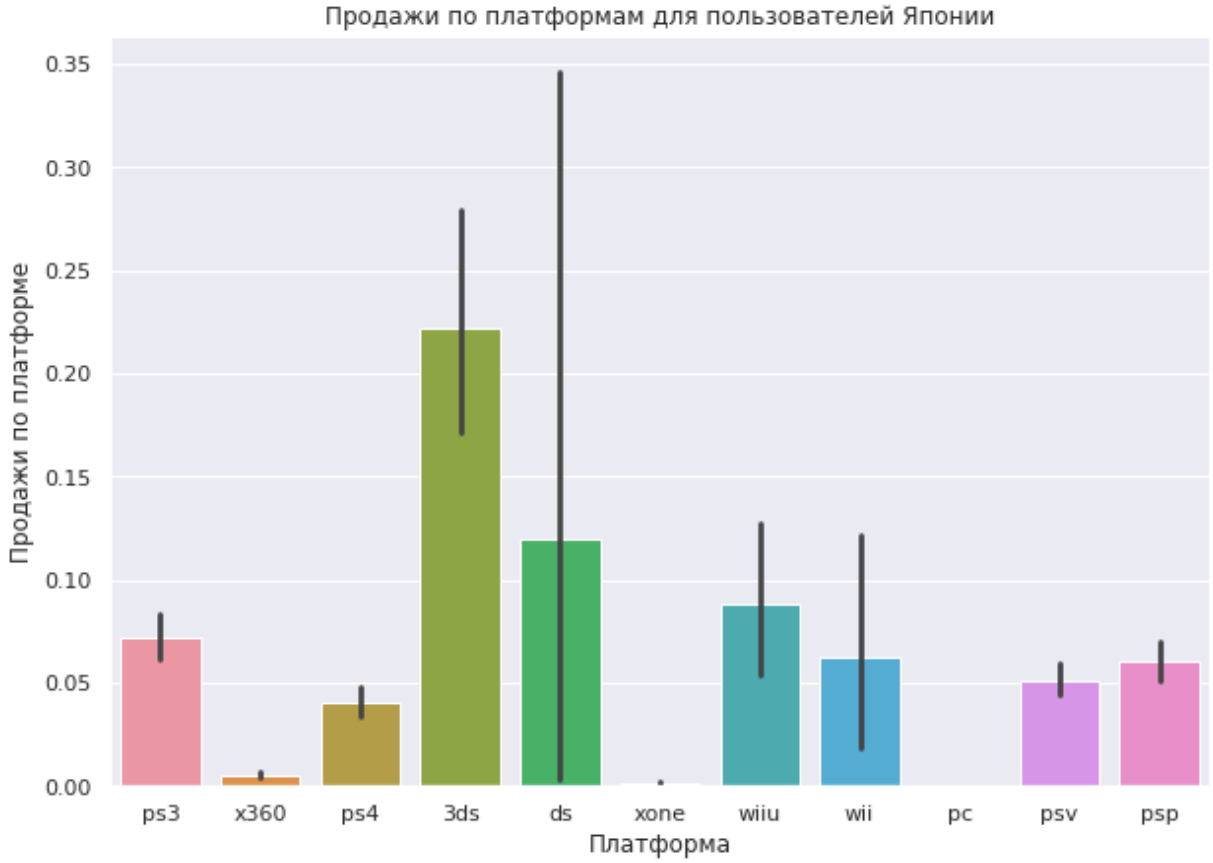
In [34]:
```
ax = sns.barplot(data=platform_5y, x='platform', y='na_sales')
sns.set(rc={'figure.figsize':(10,7)})

ax.set(title='Продажи по платформам для пользователей Северной Америки', xlabel='

plt.show()
```



In [35]:
```
ax = sns.barplot(data=platform_5y, x='platform', y='eu_sales')
sns.set(rc={'figure.figsize':(10,7)})

ax.set(title='Продажи по платформам для пользователей Европы', xlabel='Платформа'

plt.show()
```



In [36]:
```
ax = sns.barplot(data=platform_5y, x='platform', y='jp_sales')
sns.set(rc={'figure.figsize':(10,7)})

ax.set(title='Продажи по платформам для пользователей Японии', xlabel='Платформа'

plt.show()
```



In [37]:
```
#топ платформ для Северной Америки
na_top_platforms = platform_5y.groupby('platform')['na_sales'].agg('sum')
na_top5 = na_top_platforms.sort_values(ascending=False).head(5)

print(na_top5)
```

```
platform
x360    140.05
ps4     108.74
ps3     103.38
xone     93.12
3ds      55.31
Name: na_sales, dtype: float64
```

In [38]:
```
#топ платформ для Европы
eu_top_platforms = platform_5y.groupby('platform')['eu_sales'].agg('sum')
eu_top5 = eu_top_platforms.sort_values(ascending=False).head(5)

print(eu_top5)
```

```
platform
ps4     141.09
ps3     106.86
x360     74.52
xone     51.59
3ds      42.64
Name: eu_sales, dtype: float64
```

In [39]:
```
#топ платформ для Японии
jp_top_platforms = platform_5y.groupby('platform')['jp_sales'].agg('sum')
jp_top5 = jp_top_platforms.sort_values(ascending=False).head(5)

print(jp_top5)
```

```
platform
3ds     87.79
ps3     35.29
psv     21.04
ps4     15.96
wiiu    13.01
Name: jp_sales, dtype: float64
```

В Европе и Америке практически весь топ занимают Xbox и PlayStation, лидер в Америке - Xbox, в Европе - PS, Nintendo в обоих случаях замыкает пятерку.

А вот в Японии топ платформ очень отличается - Nintendo занимает первое место, три следующие позиции принадлежат PS, а замыкает Wiiu. Бытует мнение, что Япония - это другой мир :) Видимо, так оно и есть.

In [40]:
```
#топ жанров для Северной Америки
na_top_genres = platform_5y.groupby('genre')['na_sales'].agg('sum')
na_top5_genres = na_top_genres.sort_values(ascending=False).head(5)

print(na_top5_genres)
```

```
genre
action          177.84
shooter         144.77
sports           81.53
role-playing     64.00
misc             38.19
Name: na_sales, dtype: float64
```

In [41]:
```
#топ жанров для Европы
eu_top_genres = platform_5y.groupby('genre')['eu_sales'].agg('sum')
eu_top5_genres = eu_top_genres.sort_values(ascending=False).head(5)

print(eu_top5_genres)
```

```
genre
action          159.34
shooter         113.47
sports           69.09
role-playing     48.53
racing           27.29
Name: eu_sales, dtype: float64
```

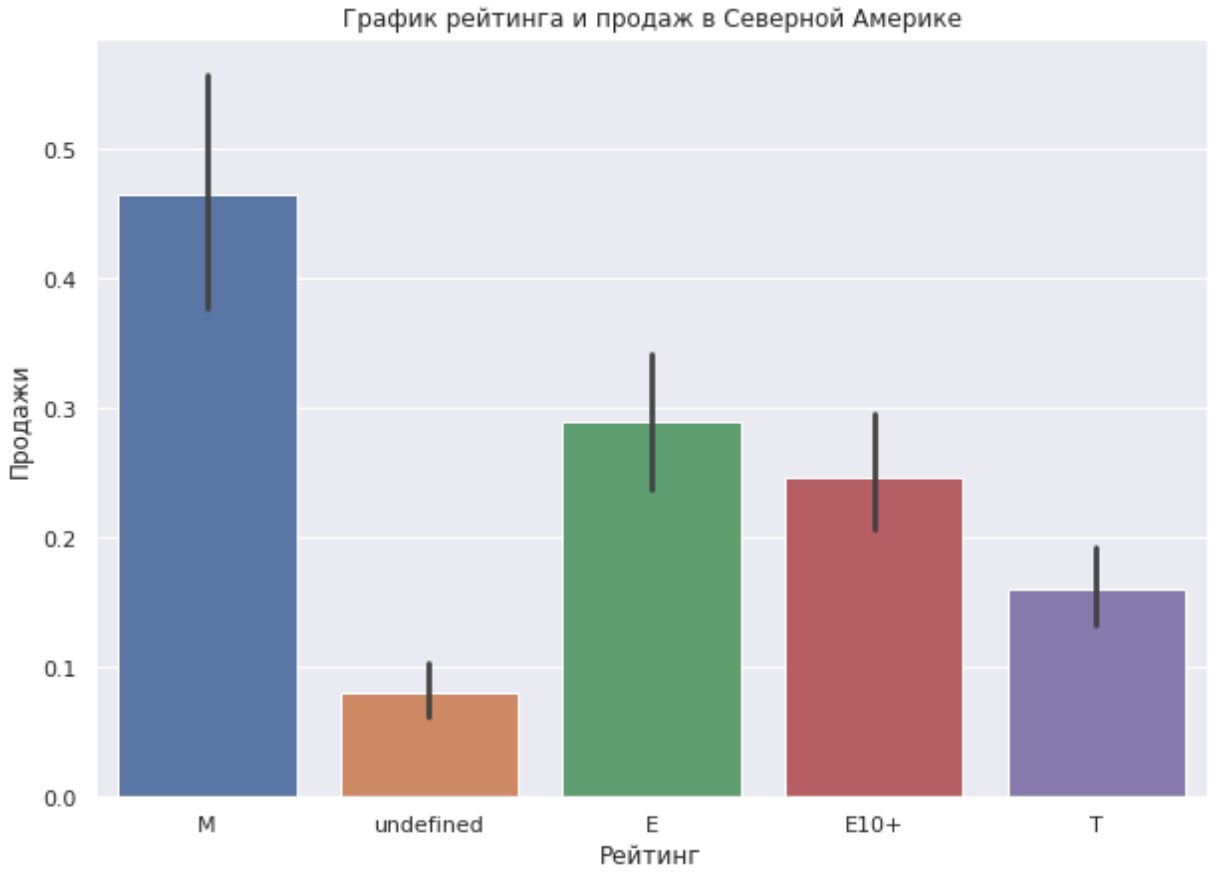
In [42]:
```
#топ жанров для Японии
jp_top_genres = platform_5y.groupby('genre')['jp_sales'].agg('sum')
jp_top5_genres = jp_top_genres.sort_values(ascending=False).head(5)

print(jp_top5_genres)
```

```
genre
role-playing    65.44
action          52.80
misc            12.86
simulation      10.41
fighting         9.44
Name: jp_sales, dtype: float64
```

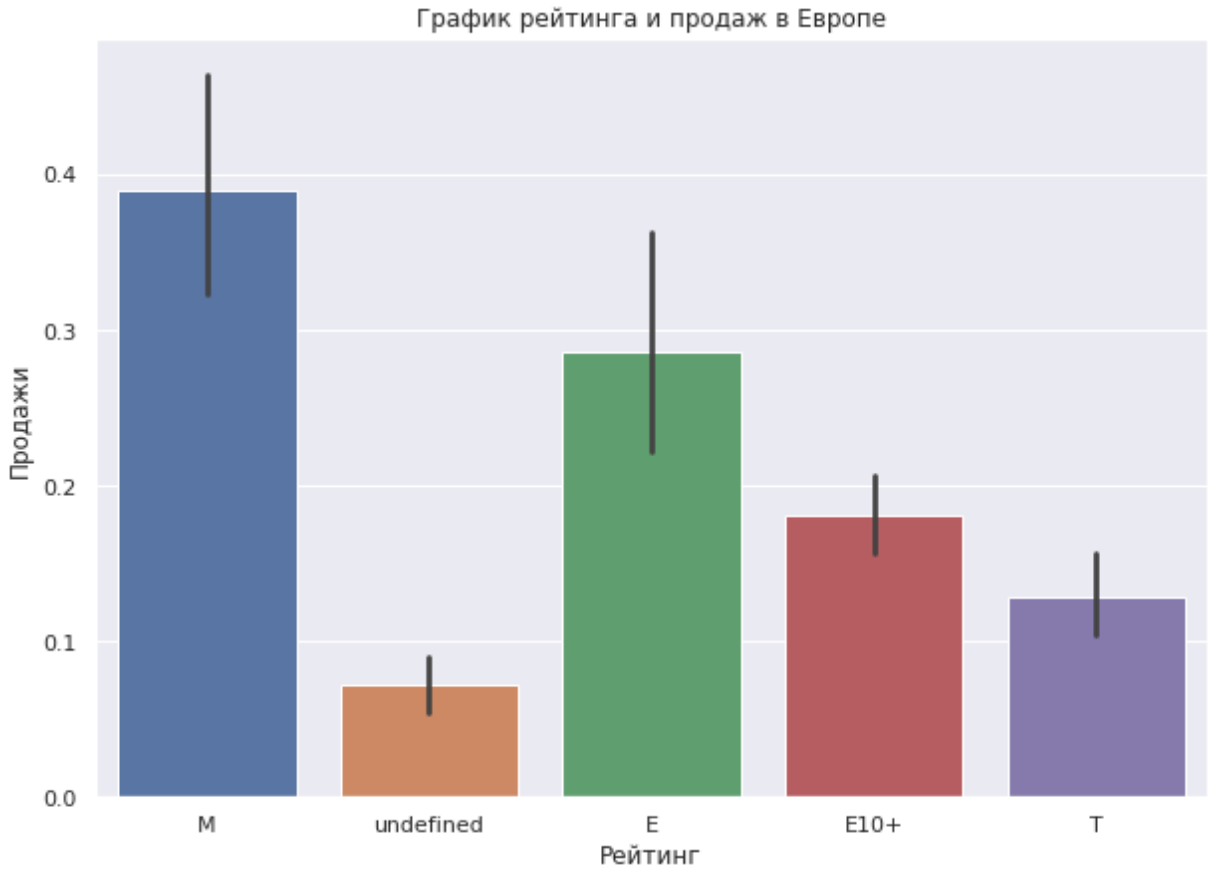
Топ жанров по доходам в Америке и Европе очень похож и за исключением пятой позиции почти не отличается. А вот в Японии на первых трех местах стоит не action, shooter, sports, a role-playing, action, misc.

Такое различие может быть связано с тем, что в таких жанрах в Японии множество игр выпускаются их национальными разработчиками, тогда как Европа и Америка потребляет не только национальные игры, но и любые другие.

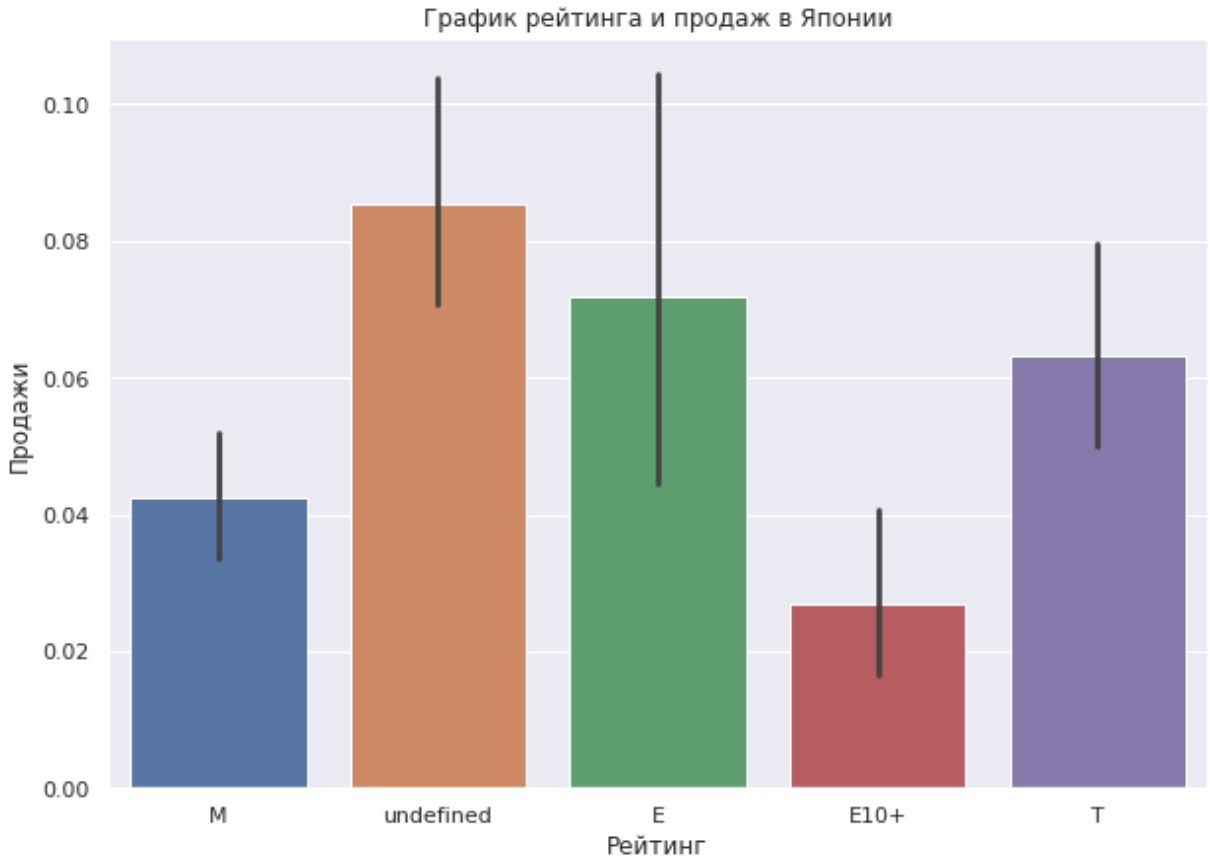
In [43]:
```
#создаем распределение рейтинга и продаж
sns.barplot(data=platform_5y, x='rating', y='na_sales').set(title='График рейтинг
                                                            xlabel='Рейтинг',
                                                            ylabel='Продажи')
plt.show()
```



In [44]:
```
sns.barplot(data=platform_5y, x='rating', y='eu_sales').set(title='График рейтинг
                                                            xlabel='Рейтинг',
                                                            ylabel='Продажи')
plt.show()
```



In [45]:
```
sns.barplot(data=platform_5y, x='rating', y='jp_sales').set(title='График рейтинг
                                                            xlabel='Рейтинг',
                                                            ylabel='Продажи')
plt.show()
```




Как видно из графиков продаж в зависимости от рейтинга по каждому региону прослеживается следующее:

В Европе и Америке рейтинг существенно влияет на продажи игр.

1. Чем большему количеству лиц по рейтингу доступна игра, тем выше ее продажи.
2. А также, чем игра более открыта к реалистичности происходящего, включая сцены насилия и сквернословие, тем большую популярность она приобретает, поскольку рейтинг M допускает к покупке аудиторию 17+ лет и именно он лидирует по продажам среди всех остальных.

В Японии же наибольшую популярность имеют игры с неопределенным рейтингом, а также игры доступные для каждого, либо же 13+ лет. Можно сказать, что в Японии абсолютно точно свои вкусы на все, и это требует дополнительного изучения.

Из проведенного анализа можно сделать вывод, что рейтинг определенно влияет на продажи.

# Проверка гипотез

## Средние пользовательские рейтинги платформ Xbox One и PC одинаковые

Нулевая гипотеза - пользовательские рейтинги Xbox One и PC равны (задана по общему правилу)

Альтернативная гипотеза - пользовательские рейтинги Xbox One и PC отличаются

In [46]:
```
xone_score = platform_5y[platform_5y['platform'] == 'xone']['user_score']
pc_score = platform_5y[platform_5y['platform'] == 'pc']['user_score']


xone_var = np.var(xone_score, ddof=1)
pc_var = np.var(pc_score, ddof=1)

print('Дисперсия Xbox One:', xone_var)
print('Дисперсия PC:', pc_var)

results = st.ttest_ind(xone_score, pc_score, equal_var=False)

alpha = .05

print('p-значение:', results.pvalue)

if results.pvalue < alpha:
    print('Нулевая гипотеза отвергнута')
else:
    print('Нулевая гипотеза подтверждена')
```

```
Дисперсия Xbox One: 1.4740531466415872
Дисперсия PC: 2.3515038897223457
p-значение: 0.2984368020965234
Нулевая гипотеза подтверждена
```

Поскольку нулевая гипотеза подтверждена - пользовательские рейтинги двух платформ могут быть равны.

## Средние пользовательские рейтинги жанров Action и Sports разные

Нулевая гипотеза - пользовательские рейтинги жанров Action и Sports равны (задана по общему правилу)

Альтернативная гипотеза - пользовательские рейтинги жанров Action и Sports отличаются

In [47]:
```
action_score = platform_5y[platform_5y['genre'] == 'action']['user_score']
sports_score = platform_5y[platform_5y['genre'] == 'sports']['user_score']

action_var = np.var(action_score, ddof=1)
sports_var = np.var(sports_score, ddof=1)

print('Дисперсия Action:', action_var)
print('Дисперсия Sports:', sports_var)

results = st.ttest_ind(action_score, sports_score, equal_var=False)

alpha = .05

print('p-значение:', results.pvalue)

if results.pvalue < alpha:
    print('Нулевая гипотеза отвергнута')
else:
    print('Нулевая гипотеза подтверждена')
```

```
Дисперсия Action: 0.9798972840341581
Дисперсия Sports: 2.750651771754206
p-значение: 9.330082550071701e-21
Нулевая гипотеза отвергнута
```

Поскольку нулевая гипотеза отвергнута, имеет место быть альтернативная гипотеза, а стало быть - пользовательские рейтинги двух жанров могут отличаться.

Мною был выбран стандартный критерий критической величины для проверки этих гипотез.

Ddof параметр выбран мною исходя из стандартного значения по госту.

Параметр equal_Var был выбран в качестве False в связи с отличием дисперсий каждых двух выборок.

# Выводы

Поскольку консоли нового поколения PS4, Xbox One могут жить значительно дольше, чем первые поколения, можно сделать ставку на продаже игр для этих платформ, поскольку в следующие годы их популярность падать не будет.

Также из исследования становится ясно, что большинство продаж видеоигр приходится на два региона из трех - это Северная Америка и Европа, а в этих регионах наиболее популярными являются игры жанров shooter, action и sports. Продажи этих трех жанров будут наибольшими в следующие годы.

Также игры рейтингов M и E являются наиболее популярными в двух регионах из трех.

Говоря о статистическом пользователе каждого из регионов можно сказать, что предпочтения пользователей из Северной Америки и Европы схожи, на эти регионы приходится большинство продаж.

Среднестатистический пользователь этих регионов предпочитает активные игры жанров Shooter, Action and Sports, а также игры рейтинга M или E и предпочитает играть на следующих платформах: Xbox 360, PS3, PS4, Xbox One.

Среднестатистический игрок же Японии предпочитает игры жанров role-playing, action, misc поскольку многие из этих игр производятся национальными разработчиками, также предпочитают игры рейтинга E, T и M и играют в основном на платформах Nintendo, PS3, PS Vita, PS4 и Wiiu, вполне возможно, также по национальным причинам.

Также в моем исследовании я прихожу к выводу, что в наибольшей степени продажи видеоигр могут зависеть от рейтинга критиков, однако его влияние ограничивается 35%.

In [ ]: