Исследование данных о продажах компьютерных игр

В нашем распоряжении данные интернет-магазине «Стримчик», который продаёт по всему миру компьютерные игры. Имеем исторические данные о продажах игр, оценки пользователей и экспертов, жанры и платформы (например, Xbox или PlayStation). Нужно выявить определяющие успешность игры закономерности. Это позволит сделать ставку на потенциально популярный продукт и спланировать рекламные кампании.

Откроем файл с данными и изучим общую информацию.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import math
import seaborn as sns
from scipy import stats as st
#подключаем необходимые библиотеки
```

```
df = pd.read_csv('/datasets/games.csv')
df.info() #выводим общую информацию
df.head(5) #выводим первые пять строк таблицы
```

```
<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
```

	Name	Platform	Year_of_Release	Genre	NA_sales	EU_sales	JP_sales	Other_sales	Critic_Score	User_Score	Rating
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	8.45	76.0	8	E
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	0.77	NaN	NaN	NaN
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	3.29	82.0	8.3	E
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	2.95	80.0	8	E
4	Pokemon Red/Pokemon	GB	1996.0	Role-Playing	11.27	8.89	10.22	1.00	NaN	NaN	NaN

Необходимо привести названия столбцов к нижнему регистру для удобства работы, а также поменять типы данных столбцам Year_of_Release - interger (годы должны быть целыми числами), User_Score - float (приведем к числовому формату). Проверим на пропуски и дубликаты.

Подготовим данные

Заменим названия столбцов

```
df.columns = df.columns.str.lower() #приведем все названия столбцов к нижнему регистру
df.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
```

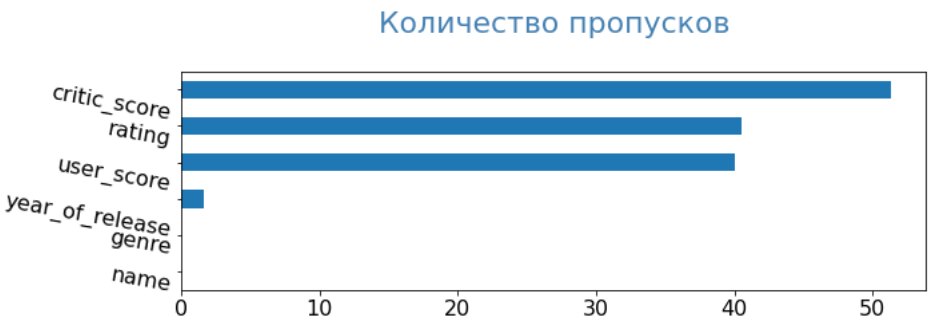
Обработаем пропуски и преобразуем данные в нужные типы

```
df.isna().sum() #проверим пропуски
```

```
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
```

```
def pass_value_barh(df):
    try:
        (
            (df.isna().mean()*100)
            .to_frame()
            .rename(columns = {0:'space'})
            .query('space > 0')
            .sort_values(by = 'space', ascending = True)
            .plot(kind = 'barh', figsize = (10,3), rot = -10, legend = False, fontsize = 16)
            .set_title('Количество пропусков' + "\n", fontsize = 22, color = 'SteelBlue')
        );
    except:
        print('пропусков не осталось :) ')
```

```
pass_value_barh(df)
```



Строки с неизвестными названиями игр удалим. Их всего две, это не повлияет на исследование. Проверим столбец с годом выпуска

```
df = df.dropna(subset = ['name']) # удаляем строки с неизвестными играми
```

```
df['year_of_release'].unique() #определим уникальные значения по столбцу Год выпуска
```

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

Чтобы поменять тип столбца на int (т.к года могут быть только целочисленными), заменим все значения nan на 0

```
df['year_of_release'] = df['year_of_release'].fillna(0)
df['year_of_release'] = df['year_of_release'].astype(int)
```

Столбец user_score нужно привести к численному типу. Проверим его уникальные значения

```
df['user_score'].unique()
```

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

Заменим значение tbd на NaN и приведем значения столбца к вещественному типу

```
df['user_score'] = df['user_score'].replace('tbd', np.NaN)
df['user_score'] = df['user_score'].astype(float)
```

Пропуски в critic_score 8578, user_score 6701, rating 6766 оставим как есть. Их слишком много, удалять их нельзя, на 0 заменять тоже некорректно - это негативно повлияет на дальнейшее исследование.

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

Проверим данные на дубликаты

```
dupl = df[df.duplicated()]
dupl
```

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	rating

Полных дубликатов нет. Поищем неявные дубликаты

```
df['name'].value_counts() #проверим уникальные значения названия игр
```

```
Need for Speed: Most Wanted                          12
LEGO Marvel Super Heroes                              9
Madden NFL 07                                         9
Ratatouille                                           9
FIFA 14                                               9
                                                     ..
Petz: Dogz Family                                     1
World Championship Poker: Howard Lederer - All In     1
Transformers: War for Cybertron (DS Version)          1
Godzilla: Domination!                                 1
Sumikko Gurashi: Koko ga Ochitsukundesu               1
Name: name, Length: 11559, dtype: int64
```

```
df['name'].duplicated().sum() #посчитаем количество дубликатов в названиях игр
```

```
5154
```

Очень много строк с дубликатами по названию игры. Посмотрим, чем отличаются строки

```
df.query('name == "Need for Speed: Most Wanted"')
```

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	rating
253	Need for Speed: Most Wanted	PS2	2005	Racing	2.03	1.79	0.08	0.47	82.0	9.1	T
523	Need for Speed: Most Wanted	PS3	2012	Racing	0.71	1.46	0.06	0.58	NaN	NaN	unknown
1190	Need for Speed: Most Wanted	X360	2012	Racing	0.62	0.78	0.01	0.15	83.0	8.5	T
1591	Need for Speed: Most Wanted	X360	2005	Racing	1.00	0.13	0.02	0.10	83.0	8.5	T
1998	Need for Speed: Most Wanted	XB	2005	Racing	0.53	0.46	0.00	0.05	83.0	8.8	T
2048	Need for Speed: Most Wanted	PSV	2012	Racing	0.33	0.45	0.01	0.22	NaN	NaN	unknown

Видим, что название одно, а остальные данные разные. Значит мы не можем удалить эти дубликаты. По факту игры разные.

```
df['genre'].unique()
```

```
array(['Sports', 'Platform', 'Racing', 'Role-Playing', 'Puzzle', 'Misc',
       'Shooter', 'Simulation', 'Action', 'Fighting', 'Adventure',
       'Strategy'], dtype=object)
```

```
df['rating'].unique()
```

```
array(['E', 'unknown', 'M', 'T', 'E10+', 'K-A', 'AO', 'EC', 'RP'],
      dtype=object)
```

Дубликаты по жанру и рейтингу не обнаружены

Посчитаем суммарные продажи во всех регионах

```
df['total_sales'] = df[['na_sales','eu_sales','jp_sales', 'other_sales']].sum(axis = 1)
#добавим столбец с общими продажами
df.head()
```

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	rating	to
0	Wii Sports	Wii	2006	Sports	41.36	28.96	3.77	8.45	76.0	8.0	E	
1	Super Mario Bros.	NES	1985	Platform	29.08	3.58	6.81	0.77	NaN	NaN	unknown	
2	Mario Kart Wii	Wii	2008	Racing	15.68	12.76	3.79	3.29	82.0	8.3	E	
3	Wii Sports Resort	Wii	2009	Sports	15.61	10.93	3.28	2.95	80.0	8.0	E	

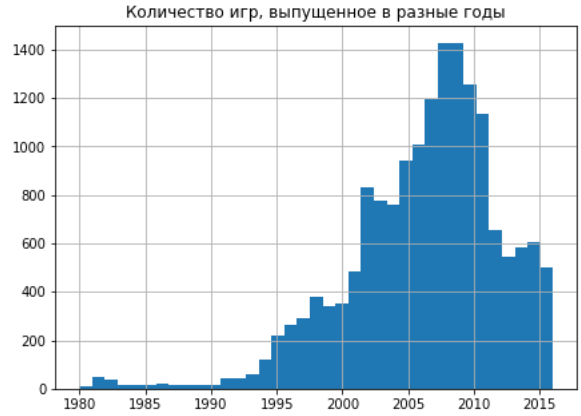
Проведем исследовательский анализ данных

Исследуем сколько игр выпускалось в разные годы

```
df['year_of_release'].describe()
```

```
count    16713.000000
mean      1974.191348
std        252.574959
min          0.000000
25%       2003.000000
50%       2007.000000
75%       2010.000000
max       2016.000000
Name: year_of_release, dtype: float64
```

```
df['year_of_release'].hist(bins = 37, figsize = (7,5), range = (1980,2016))
plt.title('Количество игр, выпущенное в разные годы')
plt.show()
```



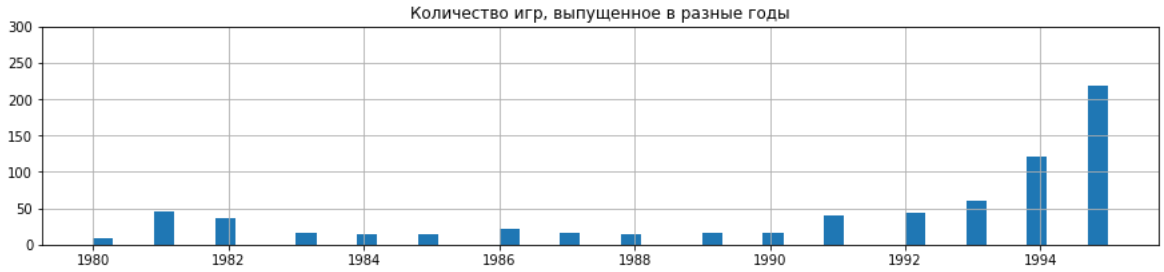

```
df.year_of_release.value_counts().head(36) # в порядке убывания количества продаж по годам
```

```
2008    1427
2009    1426
2010    1255
2007    1197
2011    1136
2006    1006
2005     939
2002     829
2003     775
2004     762
2012     653
2015     606
2014     581
2013     544
2016     502
2001     482
1998     379
2000     350
1999     338
1997     289
0        269
1996     263
1995     219
1994     121
1993      60
1981      46
1992      43
1991      41
1982      36
1986      21
1983      17
1989      17
1987      16
1990      16
1988      15
1984      14
Name: year_of_release, dtype: int64
```

По графику видим, как сильно увеличилось количество проданных игр со временем. Это связано с развитием технологий и индустрии игр. Также видно просадку продаж после 2008 года. Скорее всего это связано с мировым финансовым кризисом и его влиянием на индустрию. Хотя точно сказать мы не можем. Этот график дает нам понять, что с 1980 по 1995 года данных критически мало.

Посчитаем количество строк и посмотрим, можно ли их исключить из исследования

```
df.hist(column = 'year_of_release', bins = 50, figsize = (15,3), range = (1980,1995))
plt.title('Количество игр, выпущенное в разные годы')
plt.ylim(0, 300);
```




```
df[['year_of_release']].apply (['count']).style.format("{:,.2f}")# общее количество строк
```

	year_of_release
count	16,713.00

```
df.query('year_of_release < 1995').count()
#количество строк со значением меньше 1995
# сюда же включаем проверку на 0 - это те значения NaN, которые мы поменяли.
```

```
name               755
platform           755
year_of_release    755
genre              755
na_sales           755
eu_sales           755
jp_sales           755
other_sales        755
critic_score       158
user_score         131
rating             755
total_sales        755
dtype: int64
```

Количество строк меньше 5%. Исключим их.

```
df = df.loc[(df['year_of_release'] >= 1995)]
#записываем в таблицу только данные с подходящими условиями
```

Исследуем как менялись продажи по платформам

Сгруппируем данные по платформам и найдем общую сумму продаж по каждой из них

```
df_pl = df.groupby('platform')['total_sales'].sum().sort_values(ascending = False)
```

```
df_pl.plot(kind = 'bar', figsize = (10,5), title = 'Продажи по платформам')
plt.xlabel("platform")
plt.show()
```

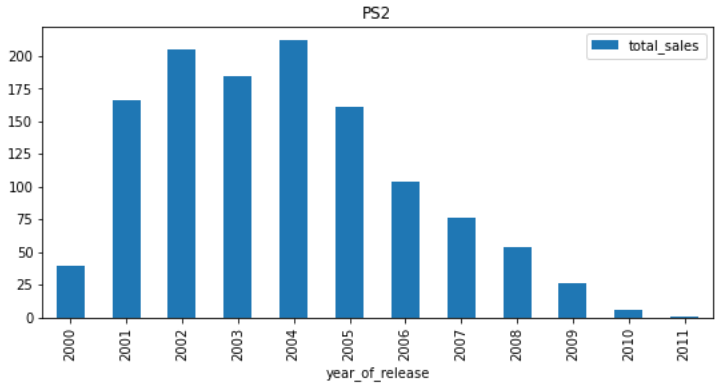
Сформируем топ 10 платформ с самыми высокими продажами

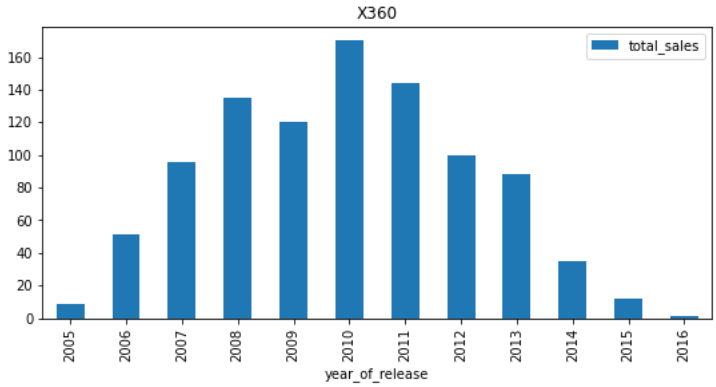
```
df_pl = df_pl.reset_index().head(10)
df_pl
```

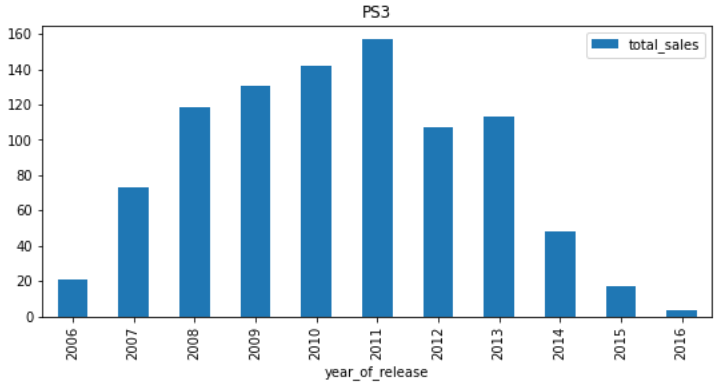
	platform	total_sales
0	PS2	1233.56
1	X360	961.24
2	PS3	931.34
3	Wii	891.18
4	DS	802.76
5	PS	721.55
6	PS4	314.14
7	GBA	312.88
8	PSP	289.53
9	3DS	257.81

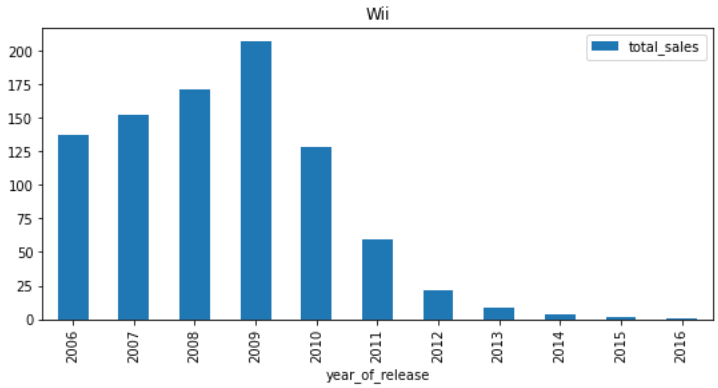
Построим графики по каждой из этих платформ

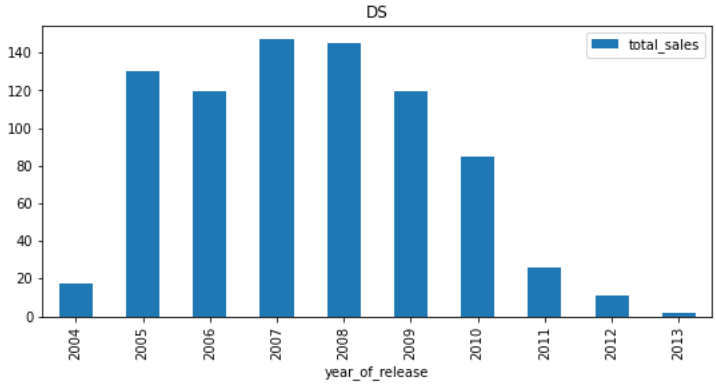
```
for index in df_pl['platform']:
    df[df['platform'] == index].pivot_table(index='year_of_release', values='total_sales', aggfunc='sum')\
    .plot(kind='bar', figsize=(9,4))
    plt.title(index)
```

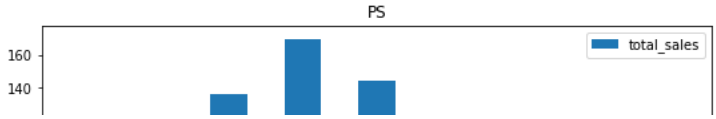

PS2
total_sales
year_of_release

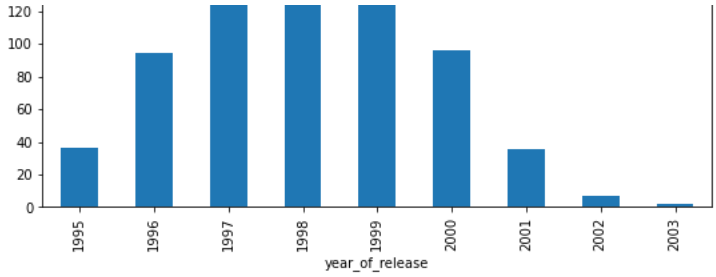

X360
total_sales
year_of_release

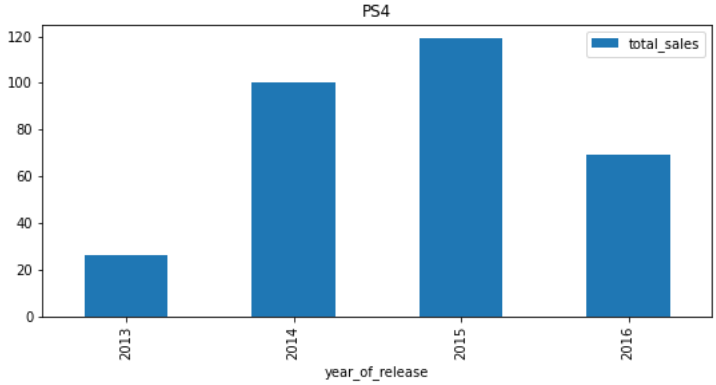

PS3
total_sales
year_of_release

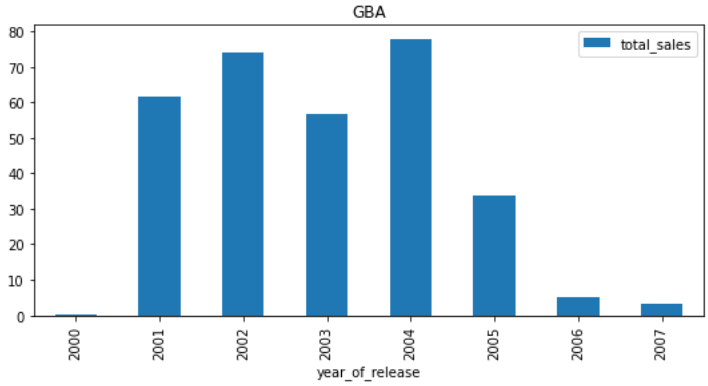

Wii
total_sales
year_of_release

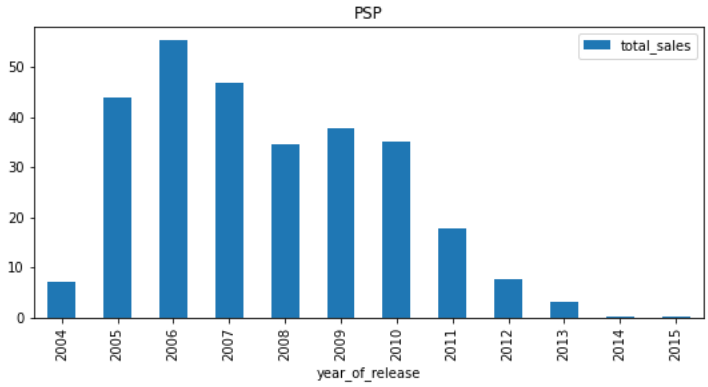

DS
total_sales
year_of_release

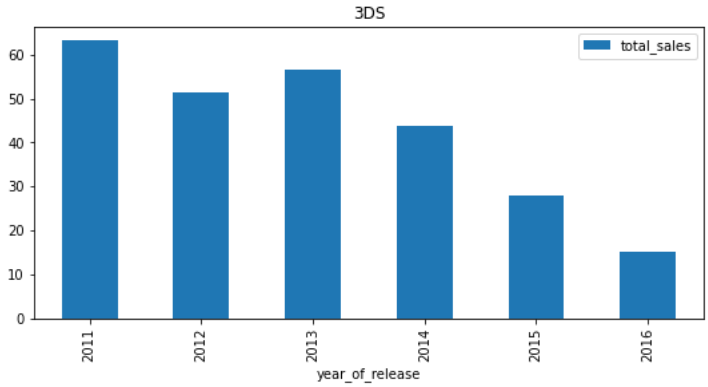

PS
total_sales











Чаще всего жизненный цикл платфоомы занимает около 10ти лет. При этом пик продаж случается на 3-5 год существования платформы. На основании этого возьмем последние 4 года за актуальный период.

Запишем таблицу только с актуальными данными

```
df = df.loc[(df['year_of_release'] >= 2013)]
```

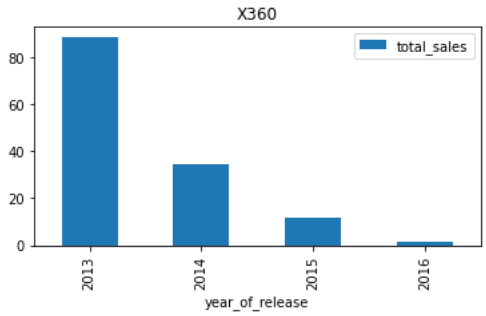
Выясним какие платформы лидируют по продажам

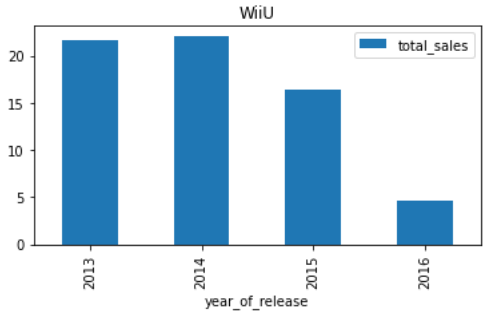
```
df_pl = df.groupby('platform')['total_sales'].sum().sort_values(ascending = False)
df_pl = df_pl.reset_index()
df_pl
```

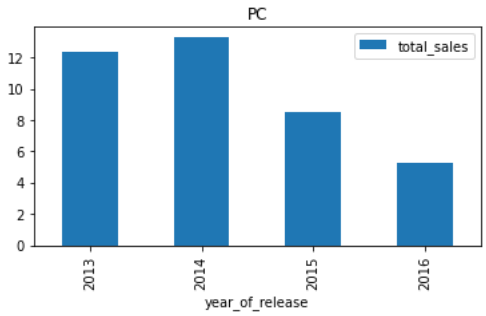
	platform	total_sales
0	PS4	314.14
1	PS3	181.43
2	XOne	159.32
3	3DS	143.25
4	X360	136.80
5	WiiU	64.63
6	PC	39.43
7	PSV	32.99
8	Wii	13.66
9	PSP	3.50
10	DS	1.54

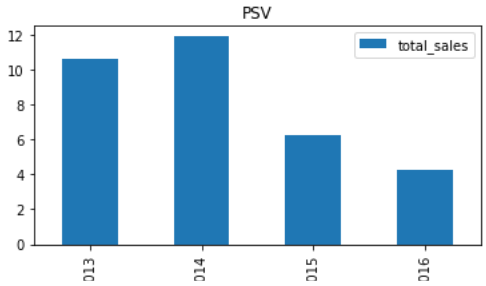
```
for index in df_pl['platform']:
    df[df['platform'] == index].pivot_table(index='year_of_release', values='total_sales', aggfunc='sum')\
    .plot(kind='bar', figsize=(6,3))
    plt.title(index)
```


0
2013
2014
2015
2016
year_of_release


X360
total_sales
80
60
40
20
0
2013
2014
2015
2016
year_of_release


WiiU
total_sales
20
15
10
5
0
2013
2014
2015
2016
year_of_release


PC
total_sales
12
10
8
6
4
2
0
2013
2014
2015
2016
year_of_release


PSV
total_sales
12
10
8
6
4
2
0

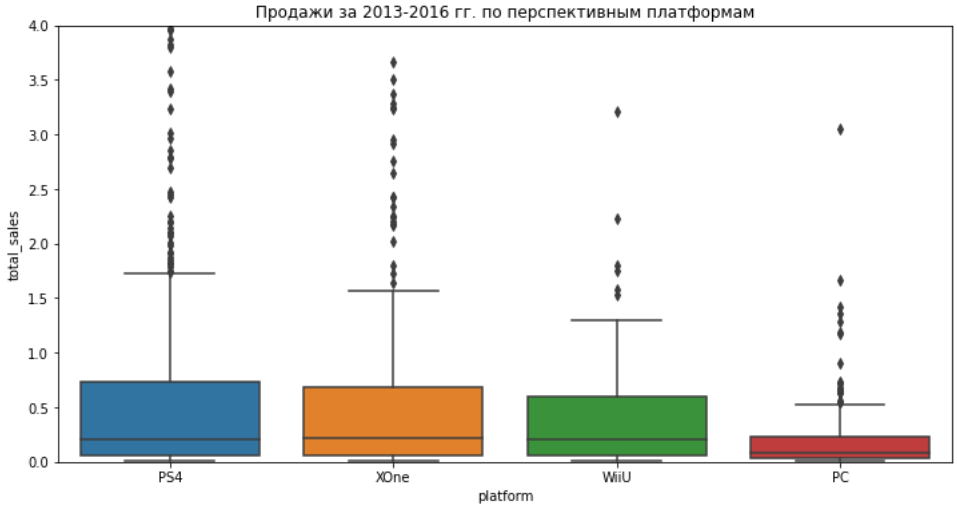
Кажется, что по всем платформам на данный момент продажи падают. Но нужно иметь в виду, что данные за 2016 не полные и год еще не закончен. Соответственно, данные могут измениться. По DS у нас есть данные только за 2013 год. По PSP нет данных за 2016 год вообще. Тем не менее, если рассмотреть динамику продаж за предыдущие года, то можно выделить платформы, которые наращивают продажи - это PS4, XOne. Возьмем PC и WiiU как перспективные тоже.

Построим диаграмму размаха

```
perspective = df.query('platform == "PS4" or platform == "XOne" or platform == "PC" or platform == "WiiU"')
# срез по перспективным платформам
perspective.describe()
```

	year_of_release	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	total_sales
count	943.000000	943.000000	943.000000	943.000000	943.000000	640.000000	674.000000	943.000000
mean	2014.867444	0.256819	0.252269	0.028823	0.074517	73.173438	6.608902	0.612428
std	1.018111	0.536214	0.579797	0.098201	0.178949	12.389843	1.518478	1.248627
min	2013.000000	0.000000	0.000000	0.000000	0.000000	19.000000	1.400000	0.010000
25%	2014.000000	0.000000	0.020000	0.000000	0.000000	67.000000	5.825000	0.050000
50%	2015.000000	0.060000	0.070000	0.000000	0.020000	75.000000	6.900000	0.170000
75%	2016.000000	0.250000	0.230000	0.020000	0.060000	82.000000	7.700000	0.600000
max	2016.000000	6.030000	6.310000	1.460000	2.380000	97.000000	9.300000	14.630000

```
plt.figure(figsize=(12, 6))
sns.boxplot(data = perspective, x='platform', y = 'total_sales').set(ylim=(0,4))
plt.title('Продажи за 2013-2016 гг. по перспективным платформам')
plt.show()
```



На перспективных платформах с 2013 по 2016 гг. продавалось в основном от 0,2 до 1 млн копий игр. Минимум 10 тыс копий, максимум 12 млн. Разброс очень большой, поэтому мы видим множество точек за ""усами".

```
plt.figure(figsize=(12, 6))
sns.boxplot(data = perspective, x='platform', y = 'total_sales')
plt.title('Продажи за 2013-2016 гг. по перспективным платформам')
plt.show()
```

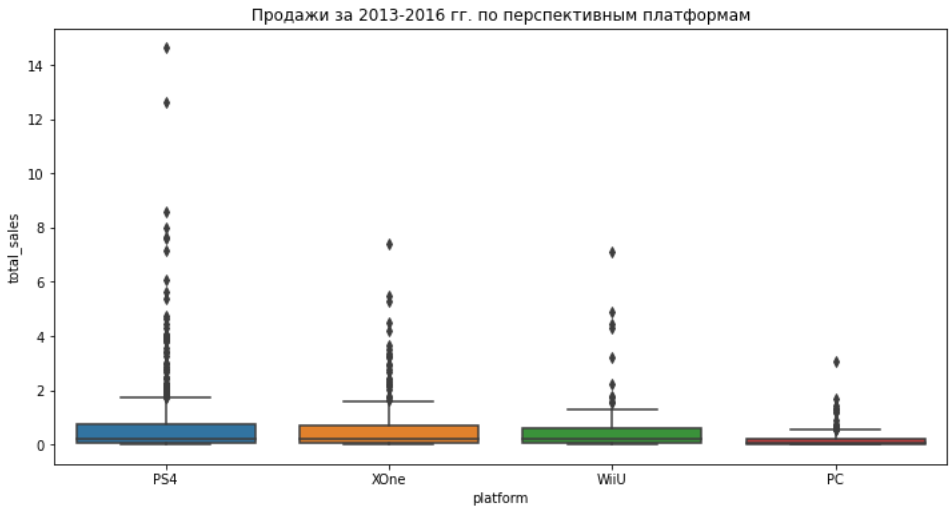


График без предела по оси продаж позволяет оценить разброс.

Рассмотрим, как влияют отзывы на продажи

Возьмем в качестве исследуемой платформу с самыми высокими продажами PS4

```
ps4 = df.loc[(df['platform'] == 'PS4')]
```

```
print('Коэффициент корреляции продаж и отзывов критиков', ps4['critic_score'].corr(ps4['total_sales']))
```

```
Коэффициент корреляции продаж и отзывов критиков 0.40656790206178095
```

Корреляция низкая, зависимость от отзывов критиков слабая.

```
ps4.plot(kind='scatter',
        y='total_sales' , x='critic_score', alpha=0.5, subplots=True, figsize=(10,4), c = 'b', s = 15)
plt.title('Диаграмма рассеяния — зависимость продаж от отзывов критиков')
plt.show()
```

Большая часть критиков ставила оценки от 60 до 80 баллов. Опираясь на диаграмму рассеяния, можем сказать о существующей, но слабой зависимости количества продаж от отзывов критиков.

```
print('Коэффициент корреляции продаж и отзывов пользователей по платформе PS4',ps4['user_score'].corr(ps4['total_sales']))
```

```
Коэффициент корреляции продаж и отзывов пользователей по платформе PS4 -0.031957110204556376
```

Корреляция очень слабая

```
ps4.plot(kind='scatter',
        y='total_sales' , x='user_score', alpha=0.4, subplots=True, figsize=(10,4), c = 'r', s = 15)
plt.title('Диаграмма рассеяния — зависимость продаж от отзывов пользователей')
plt.show()
```

Большая часть игр получает оценку пользователя от 6 до 8 баллов. Как мы видим по корреляции и графику, зависимость продаж от оценок пользователей настолько мала, что несущественна.

Сравним с общей картиной в мире

```
print('Коэффициент корреляции продаж и отзывов критиков', df['critic_score'].corr(df['total_sales']))
```

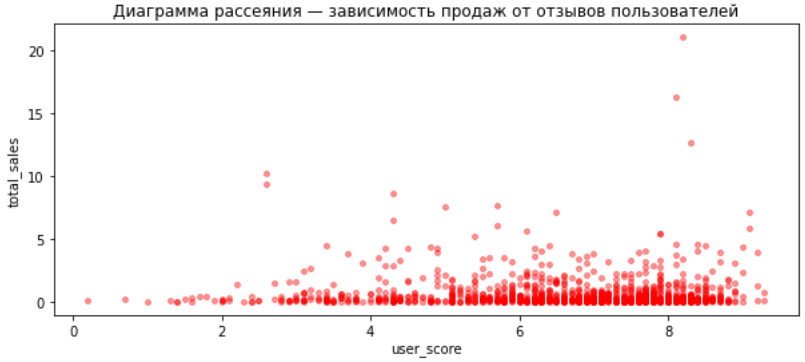
```
Коэффициент корреляции продаж и отзывов критиков 0.3136995151027371
```

```
df.plot(kind='scatter',
        y='total_sales' , x='critic_score', alpha=0.5, subplots=True, figsize=(10,4), c = 'b', s = 15)
plt.title('Диаграмма рассеяния — зависимость продаж от отзывов критиков')
plt.show()
```


```
print('Коэффициент корреляции продаж и отзывов пользователей',df['user_score'].corr(df['total_sales']))
```

```
Коэффициент корреляции продаж и отзывов пользователей -0.0026078133545982744
```

```
df.plot(kind='scatter',
        y='total_sales' , x='user_score', alpha=0.4, subplots=True, figsize=(10,4), c = 'r', s = 15)
plt.title('Диаграмма рассеяния — зависимость продаж от отзывов пользователей')
plt.show()
```



Если сравнить с корреляцией продаж и отзывов по каждой отдельной игре, выводы останутся прежними.

Сравним с продажами игр на других платформах

Возьмем перспективную XOne и 2 платформы из начала списка с высокими продажами . Это PS3 и 3DS

```
xone = df.query('platform == "XOne"')
ps3 = df.query('platform == "PS3"')
ds3 = df.query('platform == "3DS"')
```

```
print('Коэффициент корреляции продаж и отзывов критиков по платформе XOne', xone['total_sales'].corr(xone['critic_score']))
```

```
Коэффициент корреляции продаж и отзывов критиков по платформе XOne 0.41699832800840175
```

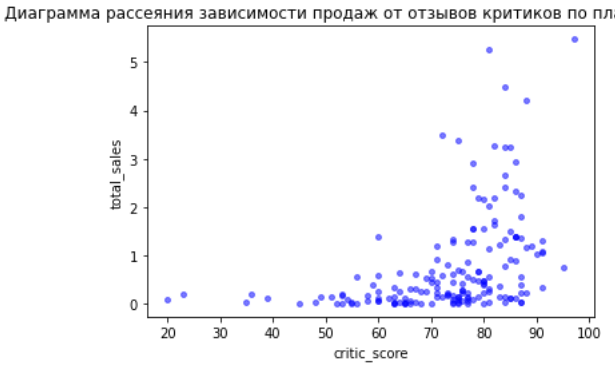
```
print('Коэффициент корреляции продаж и отзывов критиков по платформе PS3', ps3['total_sales'].corr(ps3['critic_score']))
```

```
Коэффициент корреляции продаж и отзывов критиков по платформе PS3 0.3342853393371919
```

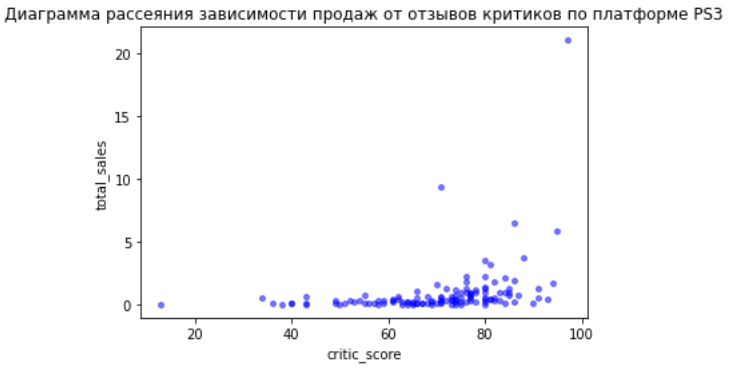
```
print('Коэффициент корреляции продаж и отзывов критиков по платформе 3DS', ds3['total_sales'].corr(ds3['critic_score']))
```

```
Коэффициент корреляции продаж и отзывов критиков по платформе 3DS 0.35705661422881035
```

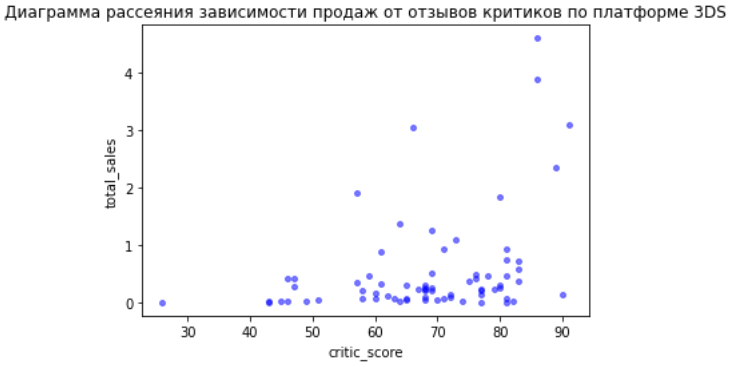
```
xone.plot(kind='scatter', y='total_sales' , x='critic_score', alpha=0.5, subplots=True, figsize=(6,4), c = 'b', s = 15)
plt.title('Диаграмма рассеяния зависимости продаж от отзывов критиков по платформе XOne')
plt.show()
```




```
ps3.plot(kind='scatter', y='total_sales' , x='critic_score', alpha=0.5, subplots=True, figsize=(6,4), c = 'b', s = 15)
plt.title('Диаграмма рассеяния зависимости продаж от отзывов критиков по платформе PS3')
plt.show()
```




```
ds3.plot(kind='scatter', y='total_sales' , x='critic_score', alpha=0.5, subplots=True, figsize=(6,4), c = 'b', s = 15)
plt.title('Диаграмма рассеяния зависимости продаж от отзывов критиков по платформе 3DS')
plt.show()
```



Видим, что зависимость продаж от отзывов критиков примерно одинакова и не превышает 41%. Это совпадает с предыдущими результатами. И говорит о том, что критики не сильно влияют на продажи.

```
print('Коэффициент корреляции продаж и отзывов пользователей по платформе XOne', xone['total_sales'].corr(xone['user_score']))
```

```
Коэффициент корреляции продаж и отзывов пользователей по платформе XOne -0.06892505328279412
```

```
print('Коэффициент корреляции продаж и отзывов пользователей по платформе PS3',ps3['total_sales'].corr(ps3['user_score']))
```

```
Коэффициент корреляции продаж и отзывов пользователей по платформе PS3 0.0023944027357566925
```

```
print('Коэффициент корреляции продаж и отзывов пользователей по платформе 3DS', ds3['total_sales'].corr(ds3['user_score']))
```

```
Коэффициент корреляции продаж и отзывов пользователей по платформе 3DS 0.24150411773563016
```

```
xone.plot(kind='scatter', y='total_sales' , x='user_score', alpha=0.4, subplots=True, figsize=(6,4), c = 'r', s = 15)
plt.title('Диаграмма рассеяния зависимости продаж от отзывов пользователей по платформе XOne')
plt.show()
```

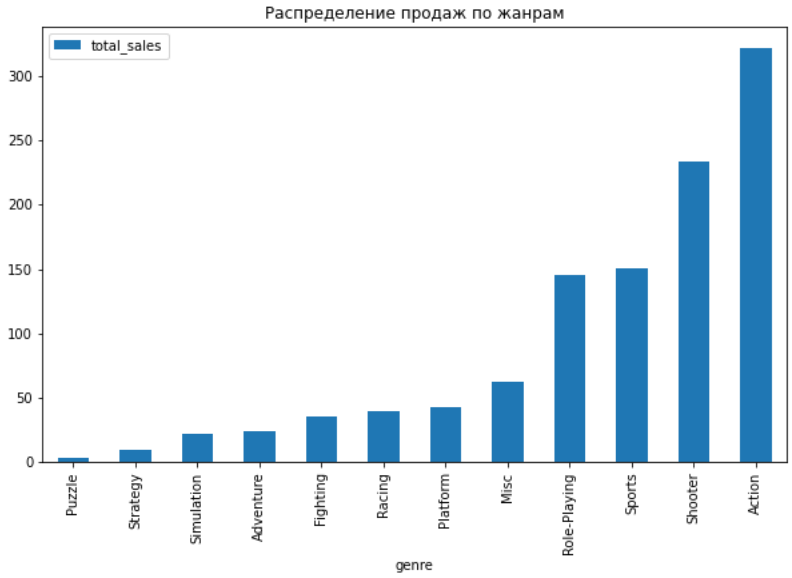

```
ps3.plot(kind='scatter', y='total_sales' , x='user_score', alpha=0.4, subplots=True, figsize=(6,4), c = 'r', s = 15)
plt.title('Диаграмма рассеяния зависимости продаж от отзывов пользователей по платформе PS3')
plt.show()
```


```
ds3.plot(kind='scatter', y='total_sales' , x='user_score', alpha=0.4, subplots=True, figsize=(6,4), c = 'r', s = 15)
plt.title('Диаграмма рассеяния зависимости продаж от отзывов пользователей по платформе 3DS')
plt.show()
```

Построенные графики и значения корреляции подтверждают предыдущие выводы. Только зависимость продаж от отзывов пользователей игр 3DS неожиданно выше других. Но она все еще низкая, чтобы быть существенной.

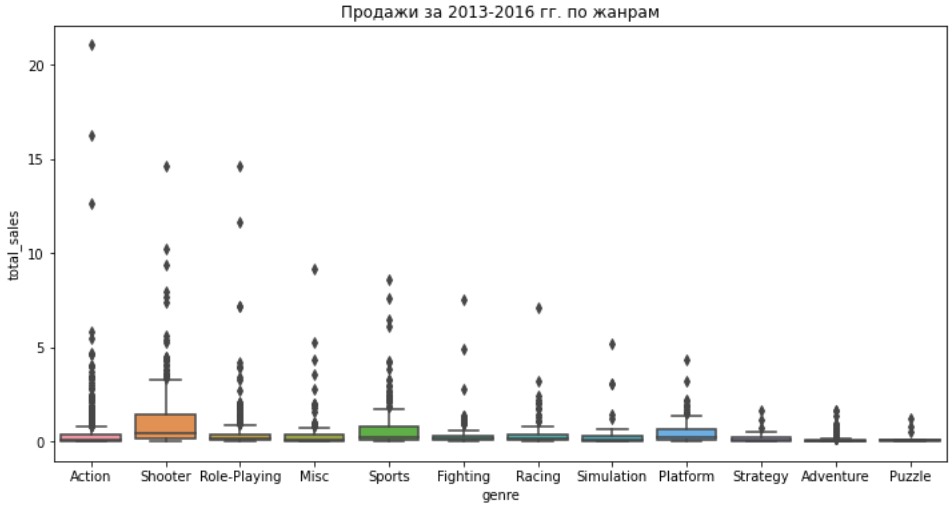
Распределение игр по жанрам

```
df.pivot_table(index = 'genre', values = 'total_sales', aggfunc = 'sum').sort_values('total_sales').plot(kind = 'bar', figsize = (10,6))
plt.title('Распределение продаж по жанрам')
plt.show()
```



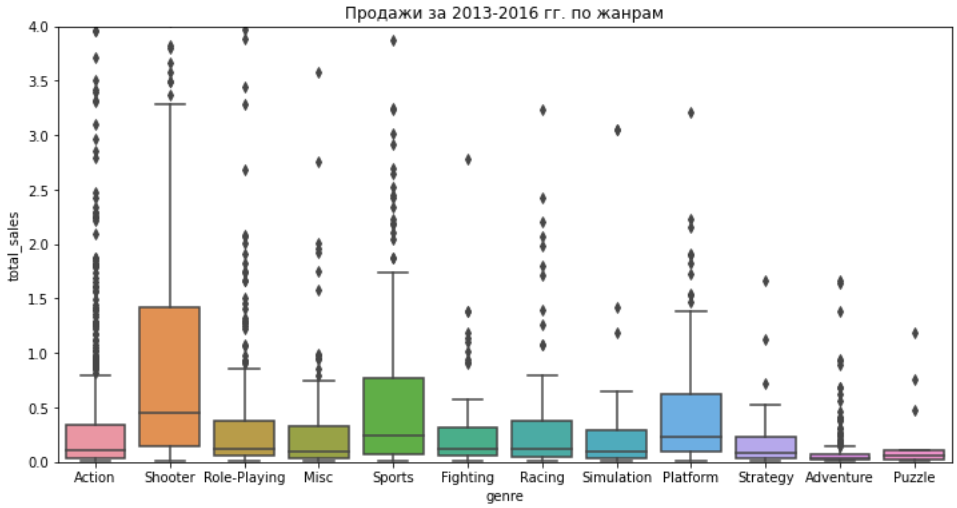
Распределив игры по жанрам, можно увидеть, что самые популярные жанры - это экшн, шутеры и спортивные. Самые непопулярные - пазлы, стратегии, симуляторы

```
plt.figure(figsize=(12, 6))
sns.boxplot(data = df, x='genre', y = 'total_sales')
plt.title('Продажи за 2013-2016 гг. по жанрам')
plt.show()
```



Рассмотрим поближе

```
plt.figure(figsize=(12, 6))
sns.boxplot(data = df, x='genre', y = 'total_sales').set(ylim=(0,4))
plt.title('Продажи за 2013-2016 гг. по жанрам')
plt.show()
```

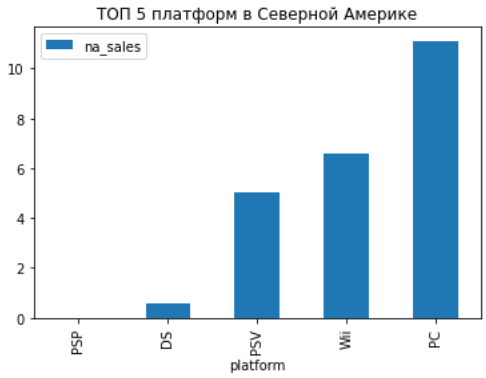


Медианы продаж шутеров и спортивных игр выше остальных. Эти жанры самые перспективные. Остальные медианы примерно на одном уровне. Интересно, что несмотря на самые высокие продажи в жанре экшн, его медиана намного ниже шутеров. Это говорит о том, что есть значения очень высоких продаж экшн игр, которые на графике попадают в выбросы. Они как бы компенсируют большее количество непопулярных игр в жанре.

Портрет пользователя каждого региона

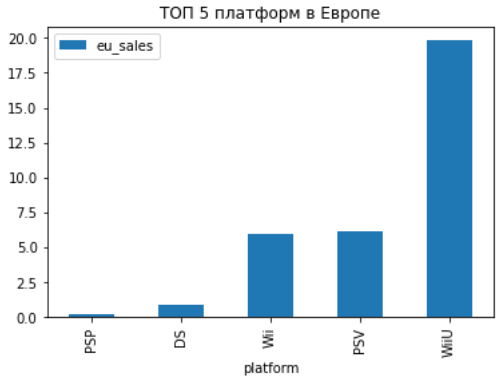
Самые популярные платформы для каждого региона

```
df_na = df.pivot_table(index = 'platform', values = 'na_sales', aggfunc = 'sum')\
.sort_values('na_sales').head(5)
df_na.plot(kind = 'bar', figsize=(6,4))
plt.title('ТОП 5 платформ в Северной Америке')
plt.show()
```



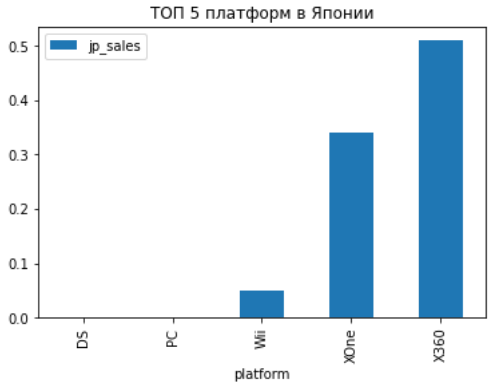
В Северной Америке самые популярные платформы PC, Wii, PSV

```
df_eu = df.pivot_table(index = 'platform', values = 'eu_sales', aggfunc = 'sum')\
.sort_values('eu_sales').head(5)
df_eu.plot(kind = 'bar', figsize=(6,4))
plt.title('ТОП 5 платформ в Европе')
plt.show()
```



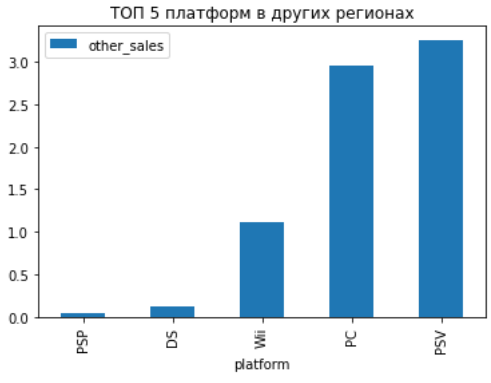
В Европе самые популярные платформы WiiU, PSV, Wii

```
df_jp = df.pivot_table(index = 'platform', values = 'jp_sales', aggfunc = 'sum')\
.sort_values('jp_sales').head(5)
df_jp.plot(kind = 'bar', figsize=(6,4))
plt.title('ТОП 5 платформ в Японии')
plt.show()
```



В Японии самые популярные платформы X360, XOne, Wii.

```
df_oth = df.pivot_table(index = 'platform', values = 'other_sales', aggfunc = 'sum')\
.sort_values('other_sales').head(5)
df_oth.plot(kind = 'bar', figsize=(6,4))
plt.title('ТОП 5 платформ в других регионах')
plt.show()
```



Первое, что может броситься в глаза, — то, что доля пользователей, играющих на игровых консолях, больше, чем доля играющих на ПК везде, кроме Северной Америки.