Анализ продаж в интернет-магазине «Стримчик»

Цель исследования:

Проанализировать данные и проверить некоторые гипотезы, которые могут помочь интернет-магазину «Стримчик» вырасти в продажах.

Ход исследования:

В работе я буду использовать три датафрейма: /datasets/games.csv., откуда получу необходимую информацию для анализа. Однако перед работой мне придется проверить качетсво полученной информации: найти дубликаты и пропущенные значения. После предобработки, данные будут визуализированны.

Таким образовам, мое исследование будет состоять из следующих действий:

```
Загрузка данных
Предобработка данных
Исследовательский анализ данных
Составление портрета пользователей
Проверка гипотезы
Вывод
```

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

Загрузка данных

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

In [3]:
```
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
```

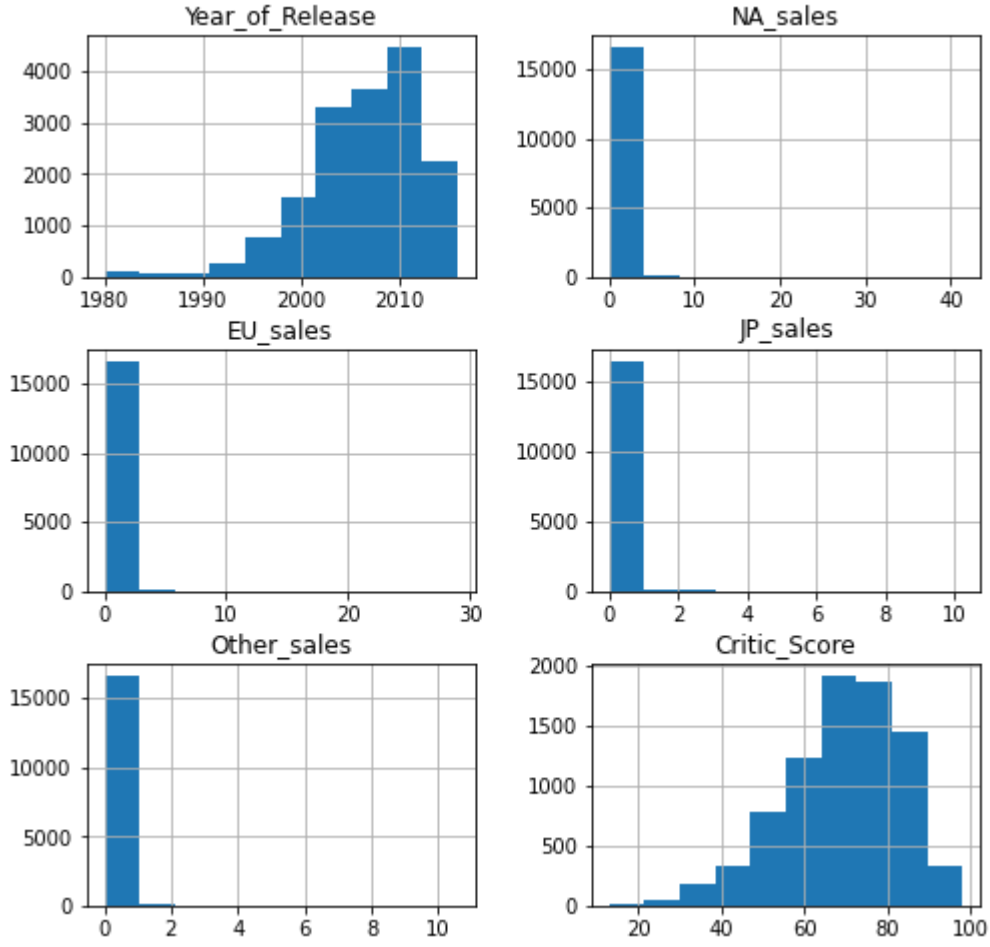
In [4]:
```
data.head()
```

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]:
```
data.hist(figsize = (8,8))
```

Out[5]:
```
array([[<AxesSubplot:title={'center':'Year_of_Release'}>,
        <AxesSubplot:title={'center':'NA_sales'}>],
       [<AxesSubplot:title={'center':'EU_sales'}>,
        <AxesSubplot:title={'center':'JP_sales'}>],
       [<AxesSubplot:title={'center':'Other_sales'}>,
        <AxesSubplot:title={'center':'Critic_Score'}>]], dtype=object)
```



К сожалению, таблица нуждается в доработке, так как встречаются пропущенные данные и дубликаты. Поэтому перед анализом и проверкой гипотез, нам необходимо подчистить датасет

Предобработка данных

Заменяем названия столбцов

In [6]:
```
#Переименуем некрасивые столбцы
data.columns = map(str.lower, data.columns)
```

In [7]:
```
#Проверим
data.head()
```

Out[7]:

	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

Отлично, названия столбцов выглядят образцово!

Обработаем пропуски

In [8]:
```
data.head()
```

Out[8]:

	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 [9]:
```
#Поиск пропущенных значений, сортировка от большего к меньшему
data.isna().sum().sort_values(ascending = False)
```

Out[9]:
```
critic_score       8578
rating             6766
user_score         6701
```

```
year_of_release     269
name                  2
genre                 2
platform              0
na_sales              0
eu_sales              0
jp_sales              0
other_sales           0
dtype: int64
```

In [10]:
```
#Поиск доли пропусков
pd.DataFrame(round(data.isna().mean()*100,1)).style.background_gradient('cool
```

Out[10]:

	0
name	0.000000
platform	0.000000
year_of_release	1.600000
genre	0.000000
na_sales	0.000000
eu_sales	0.000000
jp_sales	0.000000
other_sales	0.000000
critic_score	51.300000
user_score	40.100000
rating	40.500000

- Начнем с малого - name и genre

In [11]:
```
#Находим пропущенные значения
data[data['name'].isna()]
```

Out[11]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales	critic
659	NaN	GEN	1993.0	NaN	1.78	0.53	0.00	0.08	
14244	NaN	GEN	1993.0	NaN	0.00	0.00	0.03	0.00	

In [12]:
```
#Удаляем
data = data.dropna(subset = ['name'])
```

- Удаление пропусков в 'year_of_release'

In [13]:
```
#Находим пропущенные значения
data[data['year_of_release'].isna()]
```

Out[13]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_
183	Madden NFL 2004	PS2	NaN	Sports	4.26	0.26	0.01	

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_
377	FIFA Soccer 2004	PS2	NaN	Sports	0.59	2.36	0.04	
456	LEGO Batman: The Videogame	Wii	NaN	Action	1.80	0.97	0.00	
475	wwe Smackdown vs. Raw 2006	PS2	NaN	Fighting	1.57	1.02	0.00	
609	Space Invaders	2600	NaN	Shooter	2.36	0.14	0.00	
...	...	...	...	...	...	...	...	
16373	PDC World Championship Darts 2008	PSP	NaN	Sports	0.01	0.00	0.00	
16405	Freaky Flyers	GC	NaN	Racing	0.01	0.00	0.00	
16448	Inversion	PC	NaN	Shooter	0.01	0.00	0.00	
16458	Hakuouki: Shinsengumi Kitan	PS3	NaN	Adventure	0.01	0.00	0.00	
16522	Virtua Quest	GC	NaN	Role-Playing	0.01	0.00	0.00	

269 rows × 11 columns

In [14]:
```
#Удаляем
data = data.dropna(subset = ['year_of_release'])
```

- Удаление пропусков в "critic_score"

In [15]:
```
#Находим пропущенные значения
data[data['critic_score'].isna()]
```

Out[15]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_s
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	11.27	8.89	10.22	
5	Tetris	GB	1989.0	Puzzle	23.20	2.26	4.22	
9	Duck Hunt	NES	1984.0	Shooter	26.93	0.63	0.28	
10	Nintendogs	DS	2005.0	Simulation	9.05	10.95	1.93	
...	...	...	...	...	...	...	...	
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	

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_s
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	

8461 rows × 11 columns

- Удаление пропусков в "user_score" через медианну

In [16]:
```
data['user_score']=data['user_score'].replace({'tbd': 0})
```

Заменим дубликаты

In [17]:
```
#Удаление явных дубликатов
data.drop_duplicates()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16444 entries, 0 to 16714
Data columns (total 11 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  float64
 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       9839 non-null   object 
 10  rating           9768 non-null   object 
dtypes: float64(6), object(5)
memory usage: 1.5+ MB
```

In [18]:
```
#Поиск неявных дубликатов в 'genre'
data['genre'].unique()
```

Out[18]:
```
array(['Sports', 'Platform', 'Racing', 'Role-Playing', 'Puzzle', 'Misc',
       'Shooter', 'Simulation', 'Action', 'Fighting', 'Adventure',
       'Strategy'], dtype=object)
```

In [19]:
```
#Поиск неявных дубликатов в 'name'
data['name'].unique()
```

Out[19]:
```
array(['Wii Sports', 'Super Mario Bros.', 'Mario Kart Wii', ...,
       'Woody Woodpecker in Crazy Castle 5', 'LMA Manager 2007',
       'Haitaka no Psychedelica'], dtype=object)
```

In [20]:
```
#Поиск неявных дубликатов в 'platform'
data['platform'].unique()
```

Out[20]:
```
array(['Wii', 'NES', 'GB', 'DS', 'X360', 'PS3', 'PS2', 'SNES', 'GBA',
       'PS4', '3DS', 'N64', 'PS', 'XB', 'PC', '2600', 'PSP', 'XOne',
       'WiiU', 'GC', 'GEN', 'DC', 'PSV', 'SAT', 'SCD', 'WS', 'NG', 'TG16',
       '3DO', 'GG', 'PCFX'], dtype=object)
```

In [21]:
```
#Поиск неявных дубликатов в 'year_of_release'
data['year_of_release'].unique()
```

Out[21]:
```
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., 1995., 1991., 1981., 1987., 1980.,
       1983.])
```

Неявные дубликаты не обнаружены

In [22]:
```
#Код ревьюера
data[data[['name', 'platform','year_of_release']].duplicated(keep=False)]
```

Out[22]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales	cri
604	Madden NFL 13	PS3	2012.0	Sports	2.11	0.22	0.0	0.23	
16230	Madden NFL 13	PS3	2012.0	Sports	0.00	0.01	0.0	0.00	

Преобразуем тип данных

In [23]:
```
#Препобразуем типы данных
data['year_of_release'] = data['year_of_release'].astype('int64')
# data['na_sales'] = data['na_sales'].astype('int64')
# data['eu_sales'] = data['eu_sales'].astype('int64')
# data['jp_sales'] = data['jp_sales'].astype('int64')
data['user_score'] = data['user_score'].astype('float64')
#data['other_sales'] = data['other_sales'].astype('int64')
```

Добавляем столбец с суммарными продажами

In [24]:
```
data['total_sales'] = (data['na_sales'] + data['eu_sales'] + data['jp_sales']
```

Вывод

1. Для начала я переименовала таблицы (привела к нижнему регистру)
2. Затем приступила к поиску пропущенных значений
    - Пропущенные строки в "name" и "genre" я удалила, так как их значения были пустыми и не имели никакой ценности для исследования
    - Пропущенные строки в "year_of_release" также пришлось удалить, так как воссоздать информацию нельзя, а удаление 269 строк для такой огромной таблицы не сильно поменяют ситуацию
    - Данные "user_score", "rating" и "critic_score" обладают светло оранжевым цветом в таблице долей пропуской, так что просто удалить эти значения мы не можем, а поставить медианное значение может стать дезинформацией, придется оставить данные в таком виде

3. Работа с дубликатами (удаление явных)
4. Преобразовала тип данных
    - "year_of_release", "na_sales", "eu_sales", "user_score", "jp_sales" и "other_sales" были преобразованны в целые числа для удобства дальшейшей работы с ними
5. Добавляем столбец с суммарными продажами
6. "tbd" – недостаточно данных для оценки

Исследовательский анализ данных

Сколько игр выпускалось в разные годы?

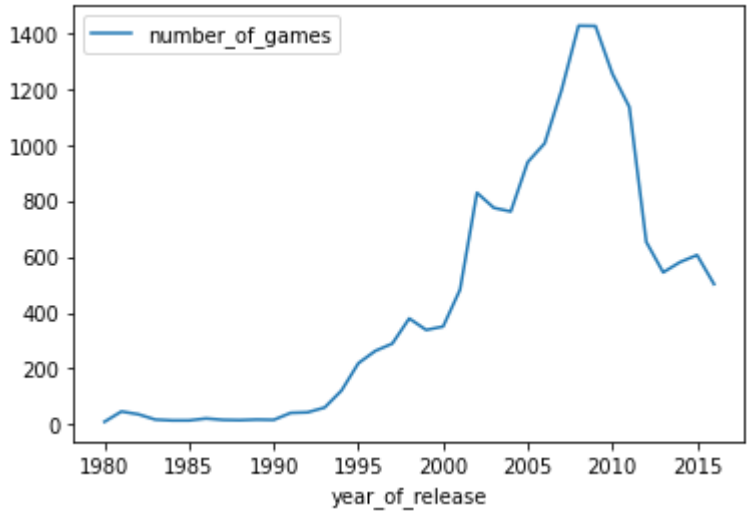
In [25]:
```
#Создаем сводную таблицу "data_year"
data_year = pd.pivot_table(data, index='year_of_release', values='name', aggf
data_year.columns = ['number_of_games']
data_year
```

Out[25]:

year_of_release	number_of_games
1980	9
1981	46
1982	36
1983	17
1984	14
1985	14
1986	21
1987	16
1988	15
1989	17
1990	16
1991	41
1992	43
1993	60
1994	121
1995	219
1996	263
1997	289
1998	379
1999	338
2000	350
2001	482
2002	829
2003	775
2004	762

year_of_release	number_of_games
2005	939
2006	1006
2007	1197
2008	1427
2009	1426
2010	1255
2011	1136
2012	653
2013	544
2014	581
2015	606
2016	502

In [26]:
```
#Строим график
data_year.plot()
plt.show()
```



Вывод: резкий скачек выпусков новых игр можно разделить на три периода: конец 90х, 2003г и самый пик - 2008г. После, с 2010 года количество выпущенных игр понемногу начало спускаться

Как менялись продажи по платформам?

In [27]:
```
#Создаем сводную таблицу "data_sales"
data_sales = pd.pivot_table(data, index=['platform'], values='total_sales', a
data_sales = data_sales.sort_values('total_sales',ascending=False)
data_sales
```

Out[27]:

platform	total_sales
PS2	1233.56

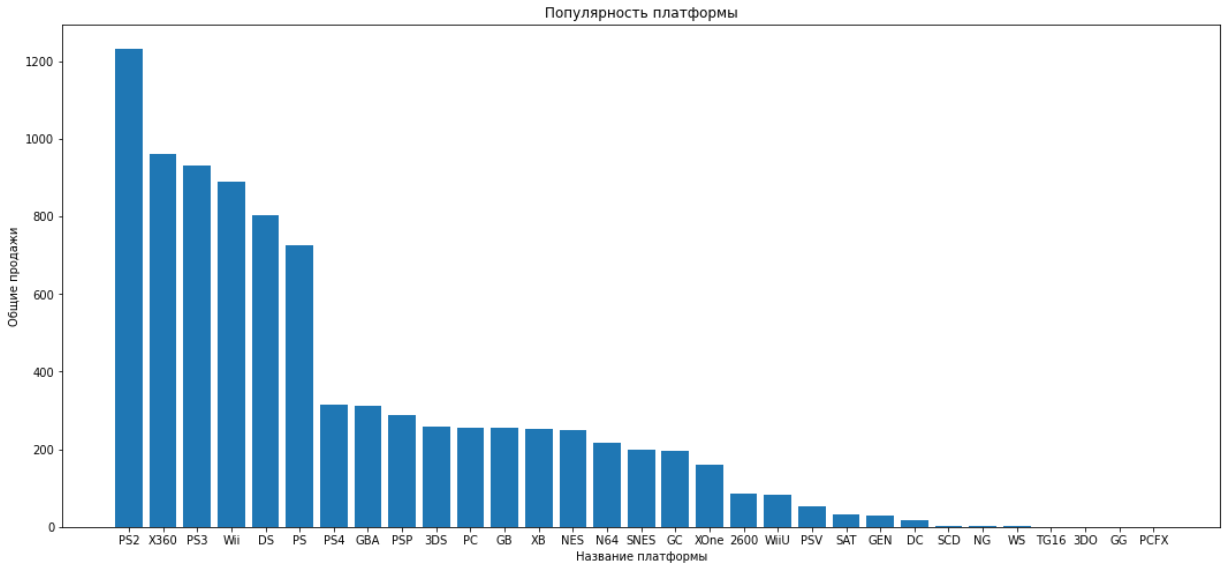
platform	total_sales
X360	961.24
PS3	931.34
Wii	891.18
DS	802.78
PS	727.58
PS4	314.14
GBA	312.88
PSP	289.53
3DS	257.81
PC	255.76
GB	254.43
XB	251.57
NES	251.05
N64	218.01
SNES	200.04
GC	196.73
XOne	159.32
2600	86.48
WiiU	82.19
PSV	53.81
SAT	33.59
GEN	28.35
DC	15.95
SCD	1.86
NG	1.44
WS	1.42
TG16	0.16
3DO	0.10
GG	0.04
PCFX	0.03

In [28]:
```
best_platform = list(data_sales.index[:5])
best_platform
```

Out[28]:
```
['PS2', 'X360', 'PS3', 'Wii', 'DS']
```

In [29]:
```
#Строим график "data_sales"
plt.figure(figsize=(18,8))
```

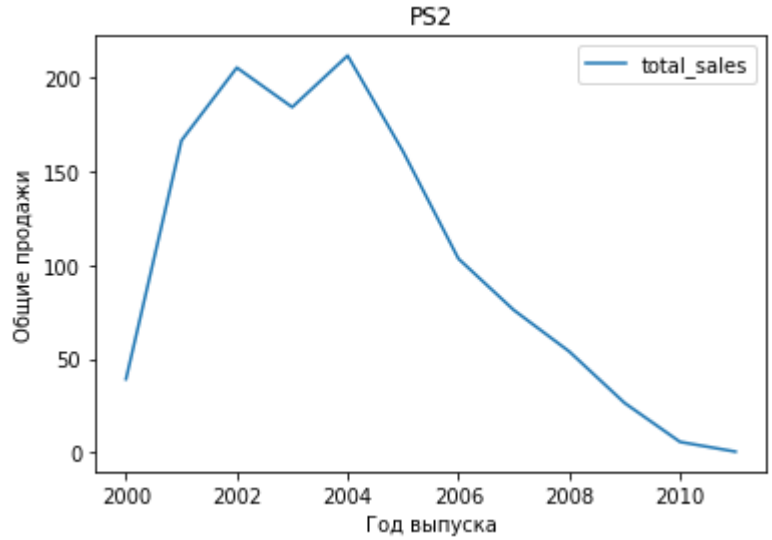
```
plt.bar(data_sales.index, data_sales['total_sales'])
plt.title('Популярность платформы')
plt.xlabel('Название платформы')
plt.ylabel('Общие продажи')
plt.show()
```

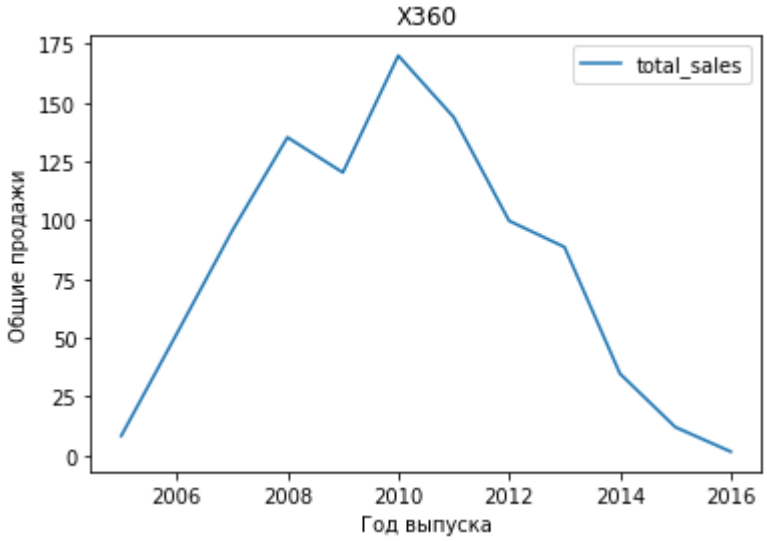


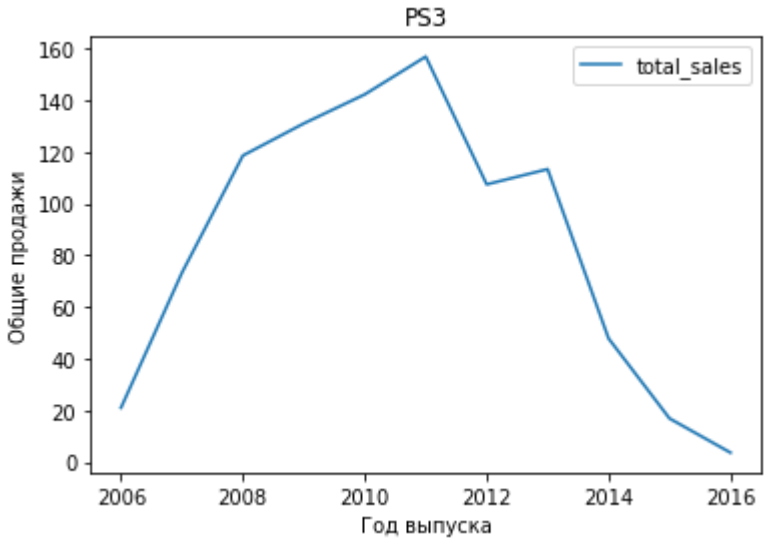
In [30]:

```
#Именуем топовые платформы и строим для каждого отдельный график
best_platform = list(data_sales.index[:3])

for platform in best_platform:
    platform_sales = pd.pivot_table(data.query('platform == @platform'), inde
    platform_sales.plot()
    plt.title(platform)
    plt.xlabel('Год выпуска')
    plt.ylabel('Общие продажи')
    plt.show()
```







У каждой платформыы был свой пик продаж, самый молодой - X360. Самая продоваемая платформа - Wii, чей пик пришелся на 2006г.

За какой характерный срок появляются новые и исчезают старые платформы?

In [31]:
```
#Создаем сводную таблицу "data_year"
data_year = pd.pivot_table(data, index=['platform'], values='year_of_release'

data_year
```

Out[31]:

platform	year_of_release
2600	1982.137931
3DO	1994.666667
3DS	2013.126953
DC	1999.942308
DS	2008.185290
GB	1995.958763
GBA	2003.210851
GC	2003.400369

platform	year_of_release
GEN	1993.037037
GG	1992.000000
N64	1998.531646
NES	1987.153061
NG	1994.500000
PC	2008.914316
PCFX	1996.000000
PS	1998.005882
PS2	2004.583921
PS3	2010.840735
PS4	2015.145408
PSP	2008.731769
PSV	2014.132867
SAT	1996.028902
SCD	1993.833333
SNES	1993.845188
TG16	1995.000000
WS	2000.000000
Wii	2008.966563
WiiU	2013.659864
X360	2009.880682
XB	2003.636364
XOne	2014.951417

In [32]:
```
#Посмотрим на частоту появления новых платформ
data_year.hist(figsize = (10,5))
plt.title("Частота появления новых платформ")
plt.xlabel('Год выпуска')
plt.ylabel('Количество платформ')
plt.show()
```

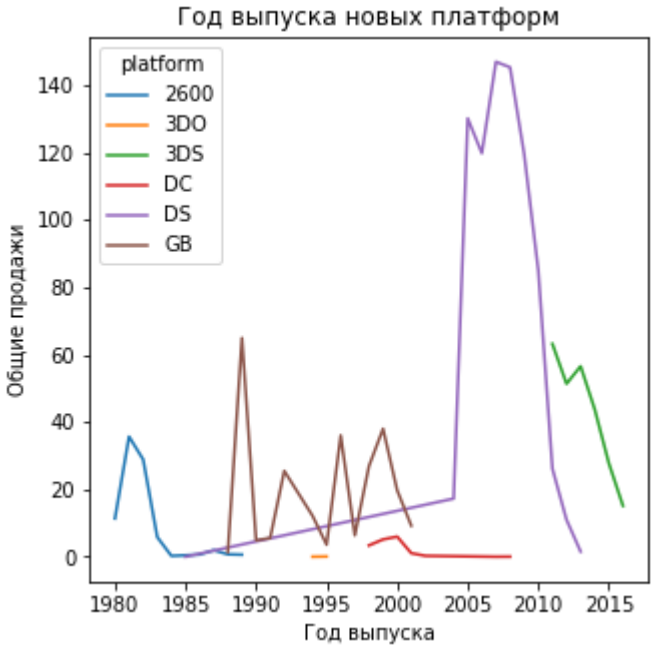
Новые платформы появляются каждый год. Нужно узнать за какой период исчезают старые платформы

In [33]:
```
#Делим платформы на группы
new_platform = list(data_year.index[:6])
middle_platform = list(data_year.index[6:11])
old_platform = list(data_year.index[11:])
```

In [34]:
```
#Строим график для новых платформ
new_platform_sales = pd.DataFrame(data.query('platform in @new_platform'))

new_platform_sales_graph = new_platform_sales.groupby(['platform', 'year_of_r
plt.figure(figsize=(5,5))
plt.xlabel('Год выпуска')
plt.ylabel('Общие продажи')
plt.title('Год выпуска новых платформ')
sns.lineplot(x='year_of_release',y='total_sales', hue='platform', data=new_pl
```

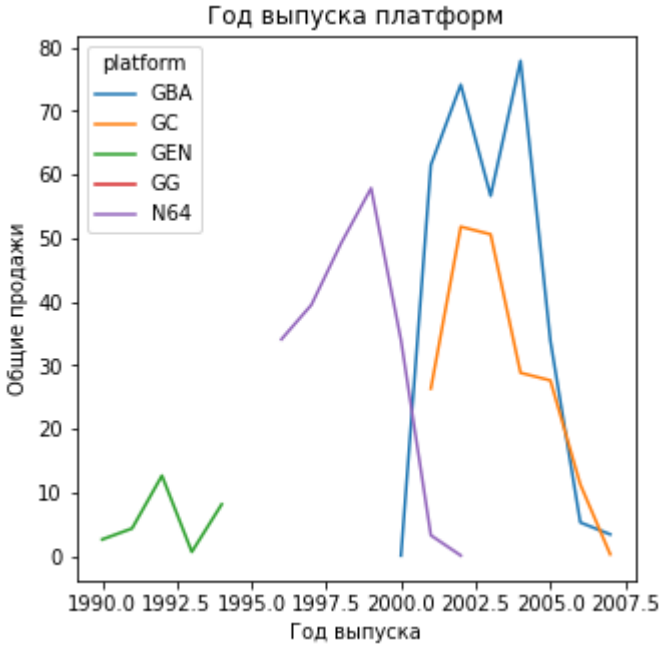
Out[34]:
```
<AxesSubplot:title={'center':'Год выпуска новых платформ'}, xlabel='Год выпус
ка', ylabel='Общие продажи'>
```



In [35]:
```
#Строим график для платформ
middle_platform_sales = pd.DataFrame(data.query('platform in @middle_platform

middle_platform_sales_graph = middle_platform_sales.groupby(['platform', 'yea
plt.figure(figsize=(5,5))
plt.xlabel('Год выпуска')
plt.ylabel('Общие продажи')
plt.title('Год выпуска платформ')
sns.lineplot(x='year_of_release',y='total_sales', hue='platform', data=middle
```

Out[35]:
```
<AxesSubplot:title={'center':'Год выпуска платформ'}, xlabel='Год выпуска', y
label='Общие продажи'>
```



In [36]:
```
#Строим график для старых платформ
old_platform_sales = pd.DataFrame(data.query('platform in @new_platform'))

old_platform_sales_graph = old_platform_sales.groupby(['platform', 'year_of_r
plt.figure(figsize=(5,5))
plt.xlabel('Год выпуска')
plt.ylabel('Общие продажи')
```

```
plt.title('Год выпуска старых платформ')
sns.lineplot(x='year_of_release',y='total_sales', hue='platform', data=old_pl
```

Out[36]:
```
<AxesSubplot:title={'center':'Год выпуска старых платформ'}, xlabel='Год выпу
ска', ylabel='Общие продажи'>
```

Нас интересуют данные последнего графика. В среднем старая платформа начинает пропадать спустя 10 лет

Выстраиваем актуальный период

In [37]:
```
current_data = data[data['year_of_release'] > 2011]
```

Какие платформы лидируют по продажам, растут или падают?

In [38]:
```
#Таблица лидеров продаж
current_data_1 = pd.pivot_table(current_data, index=['platform','year_of_rele
                                values='total_sales', aggfunc = 'sum')
current_data_1
```

Out[38]:

		total_sales
platform	**year_of_release**	
3DS	**2012**	51.36
	2013	56.57
	2014	43.76
	2015	27.78
	2016	15.14
DS	**2012**	11.01
	2013	1.54
PC	**2012**	23.22
	2013	12.38
	2014	13.28
	2015	8.52

platform	year_of_release	total_sales
	2016	5.25
PS3	2012	107.36
	2013	113.25
	2014	47.76
	2015	16.82
	2016	3.60
PS4	2013	25.99
	2014	100.00
	2015	118.90
	2016	69.25
PSP	2012	7.69
	2013	3.14
	2014	0.24
	2015	0.12
PSV	2012	16.19
	2013	10.59
	2014	11.90
	2015	6.25
	2016	4.25
Wii	2012	21.71
	2013	8.59
	2014	3.75
	2015	1.14
	2016	0.18
WiiU	2012	17.56
	2013	21.65
	2014	22.03
	2015	16.35
	2016	4.60
X360	2012	99.74
	2013	88.58
	2014	34.74
	2015	11.96
	2016	1.52
XOne	2013	18.96
	2014	54.07

		total_sales
platform	year_of_release	
	2015	60.14
	2016	26.15

In [39]:
```
current_data_2 = pd.pivot_table(current_data_1, index='platform', values='tot
current_data_2.sort_values('total_sales', ascending=False)
```

Out[39]:

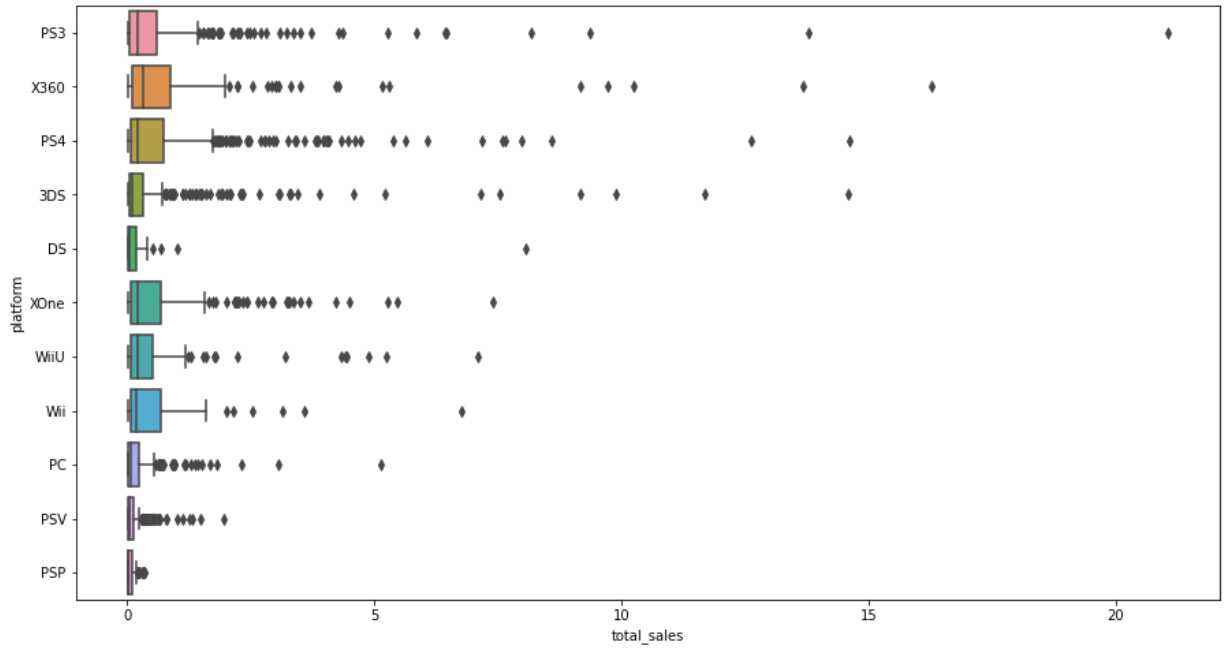
	total_sales
platform	
PS4	314.14
PS3	288.79
X360	236.54
3DS	194.61
XOne	159.32
WiiU	82.19
PC	62.65
PSV	49.18
Wii	35.37
DS	12.55
PSP	11.19

Вывод:

Лидерами продаж стали такие платформы, как PS4, X360 и PS3. Чаще характерно, что продажи платформы падают, это обьяснимо тем, что на их смену ежегодно приходят новые и более усовершественные технологии, однако PS4 на протяжении 5 лет неплохо держит планку

«Ящик с усами»

In [40]:
```
plt.figure(figsize=(15,8))
sns.boxplot(x="total_sales", y="platform", data=current_data.reset_index());
```



"Ящик с усами" - это тот самый график, который иллюстрирует продажи каждой платформы. На графике отчетливо видно как PS3 обгоняет своих конкурентов в продажах, и старается его догнать X360. Медианное значение у каждой платформы проходит по-разному. Самые слабенькие платформы изображены небольшими ящиками с левой стороны, то есть PSP, PSV

Корреляция между отзывами и продажами

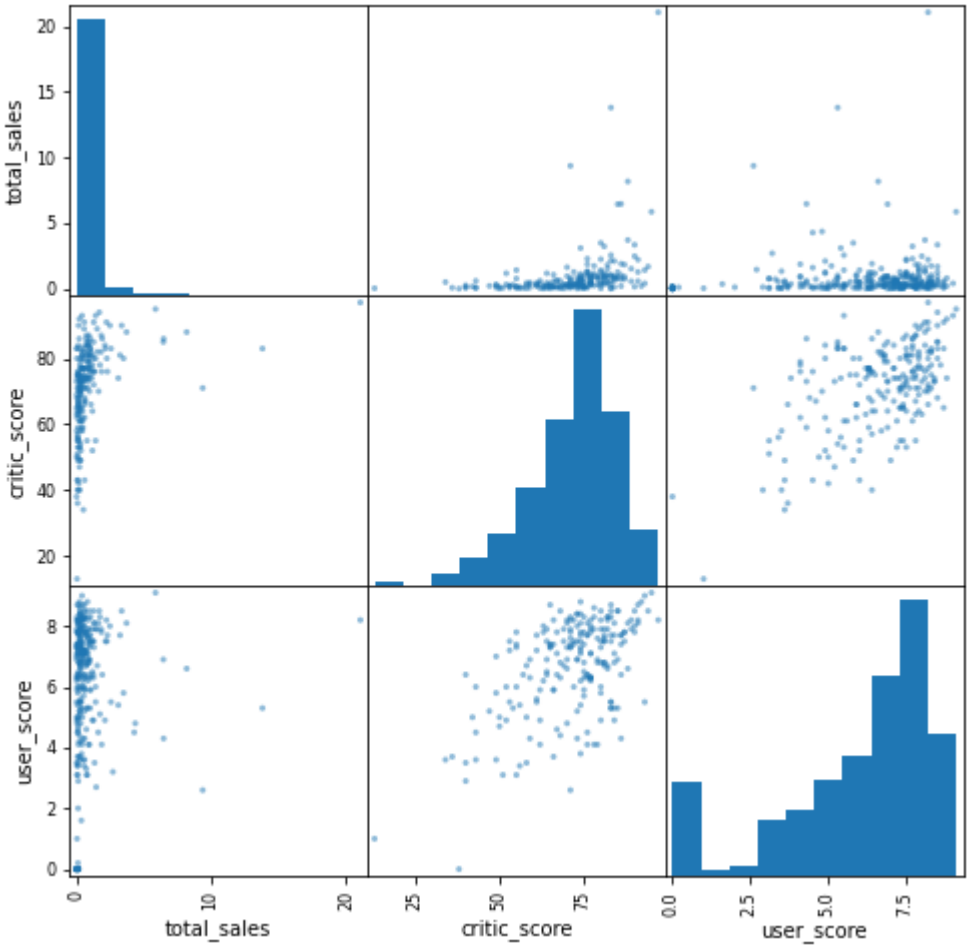
PS3

In [43]:
```
current_data_PS3 = current_data[current_data['platform'] == 'PS3']
current_data_PS3 = current_data_PS3.loc[:,['total_sales', 'critic_score', 'us
display(current_data_PS3.corr())

pd.plotting.scatter_matrix(current_data_PS3, figsize=(8, 8), alpha = 0.5)
```

	total_sales	critic_score	user_score
total_sales	1.000000	0.331497	0.094424
critic_score	0.331497	1.000000	0.558611
user_score	0.094424	0.558611	1.000000

Out[43]:
```
array([[<AxesSubplot:xlabel='total_sales', ylabel='total_sales'>,
        <AxesSubplot:xlabel='critic_score', ylabel='total_sales'>,
        <AxesSubplot:xlabel='user_score', ylabel='total_sales'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='critic_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='critic_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='critic_score'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='user_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='user_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='user_score'>]],
      dtype=object)
```



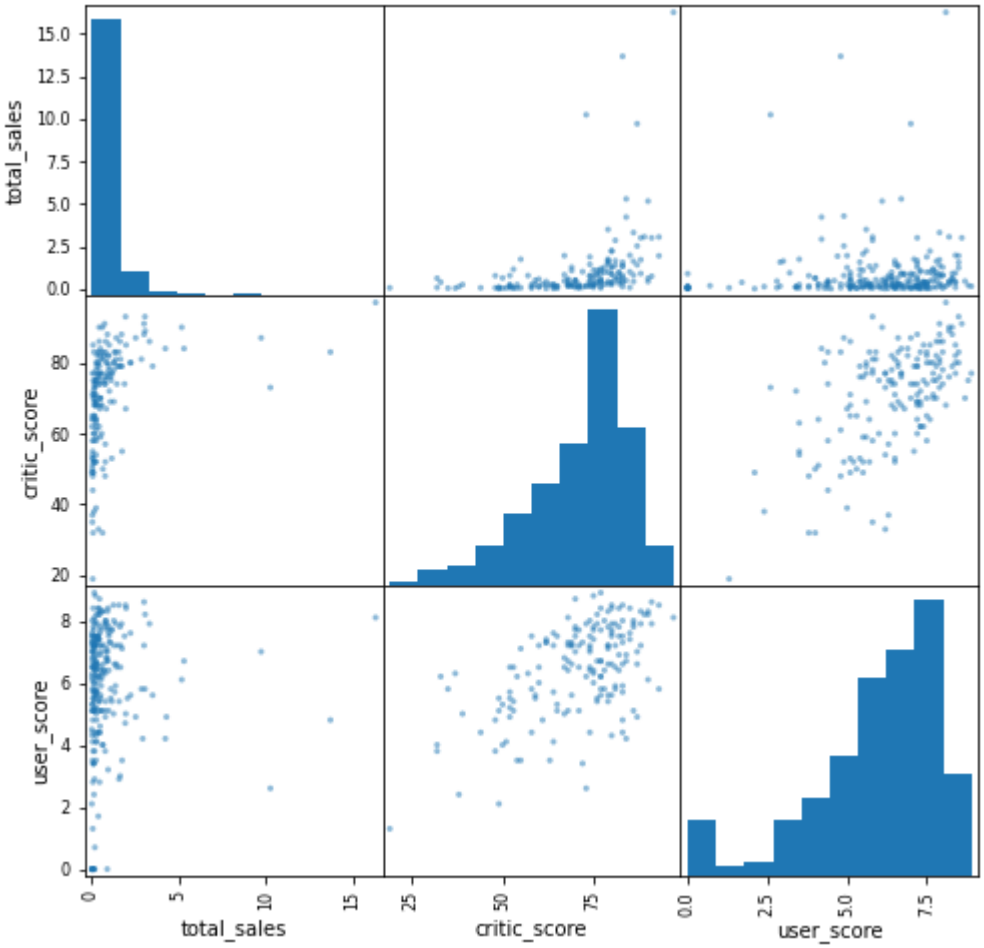
X360

In [44]:
```
current_data_X360 = current_data[current_data['platform'] == 'X360']
current_data_X360 = current_data_X360.loc[:,['total_sales', 'critic_score', '
display(current_data_X360.corr())

pd.plotting.scatter_matrix(current_data_X360, figsize=(8, 8), alpha = 0.5)
```

	total_sales	critic_score	user_score
total_sales	1.000000	0.360573	0.062215
critic_score	0.360573	1.000000	0.557352
user_score	0.062215	0.557352	1.000000

Out[44]:
```
array([[<AxesSubplot:xlabel='total_sales', ylabel='total_sales'>,
        <AxesSubplot:xlabel='critic_score', ylabel='total_sales'>,
        <AxesSubplot:xlabel='user_score', ylabel='total_sales'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='critic_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='critic_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='critic_score'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='user_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='user_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='user_score'>]],
      dtype=object)
```



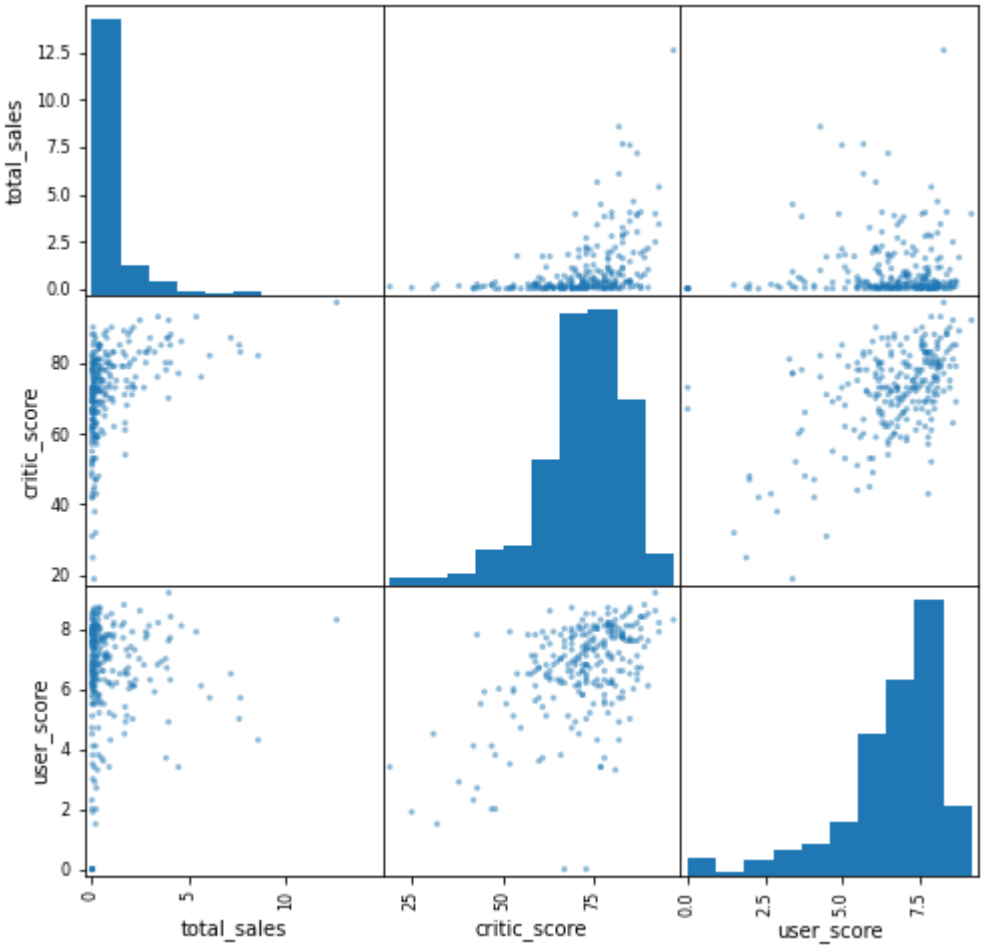
PS4

In [45]:
```
current_data_PS4 = current_data[current_data['platform'] == 'PS4']
current_data_PS4 = current_data_PS4.loc[:,['total_sales', 'critic_score', 'us
display(current_data_PS4.corr())

pd.plotting.scatter_matrix(current_data_PS4, figsize=(8, 8), alpha = 0.5)
```

	total_sales	critic_score	user_score
total_sales	1.000000	0.406568	0.023279
critic_score	0.406568	1.000000	0.520752
user_score	0.023279	0.520752	1.000000

Out[45]:
```
array([[<AxesSubplot:xlabel='total_sales', ylabel='total_sales'>,
        <AxesSubplot:xlabel='critic_score', ylabel='total_sales'>,
        <AxesSubplot:xlabel='user_score', ylabel='total_sales'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='critic_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='critic_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='critic_score'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='user_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='user_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='user_score'>]],
      dtype=object)
```



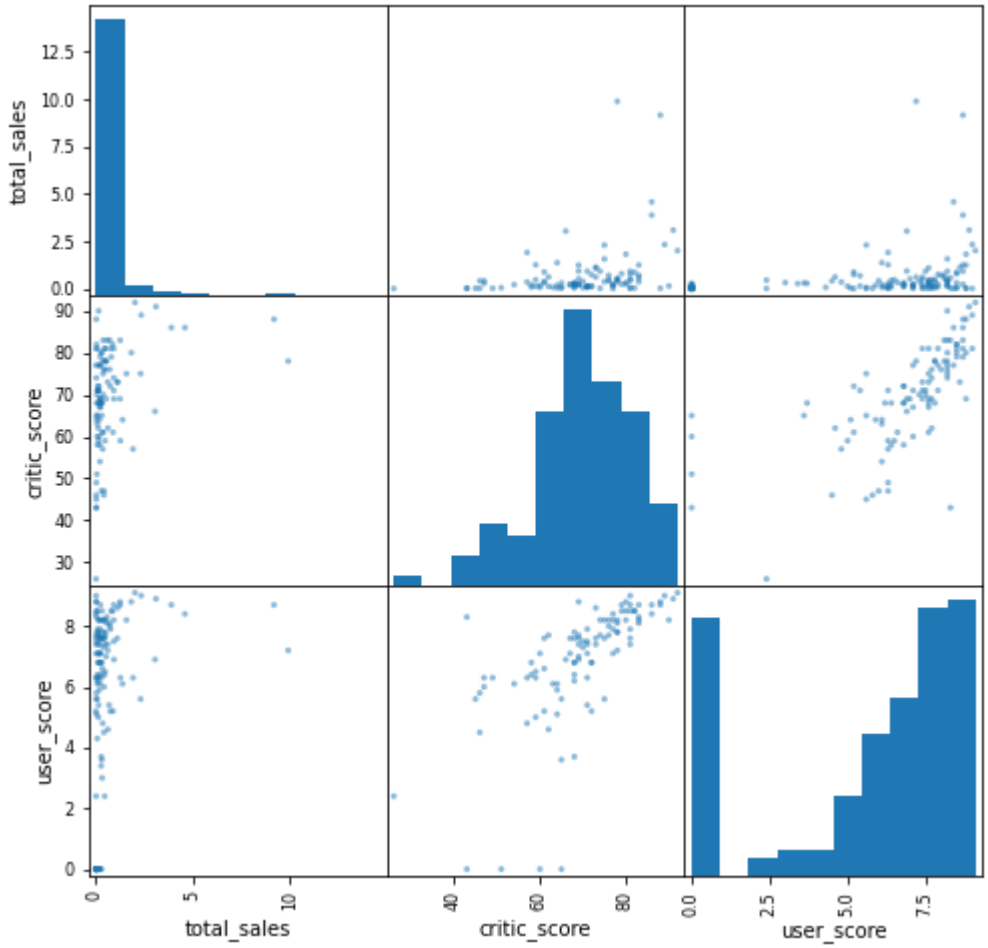
3DS

In [46]:
```
current_data_3DS = current_data[current_data['platform'] == '3DS']
current_data_3DS = current_data_3DS.loc[:,['total_sales', 'critic_score', 'us
display(current_data_3DS.corr())

pd.plotting.scatter_matrix(current_data_3DS, figsize=(8, 8), alpha = 0.5)
```

	total_sales	critic_score	user_score
total_sales	1.000000	0.320803	0.268579
critic_score	0.320803	1.000000	0.657600
user_score	0.268579	0.657600	1.000000

Out[46]:
```
array([[<AxesSubplot:xlabel='total_sales', ylabel='total_sales'>,
        <AxesSubplot:xlabel='critic_score', ylabel='total_sales'>,
        <AxesSubplot:xlabel='user_score', ylabel='total_sales'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='critic_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='critic_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='critic_score'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='user_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='user_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='user_score'>]],
      dtype=object)
```



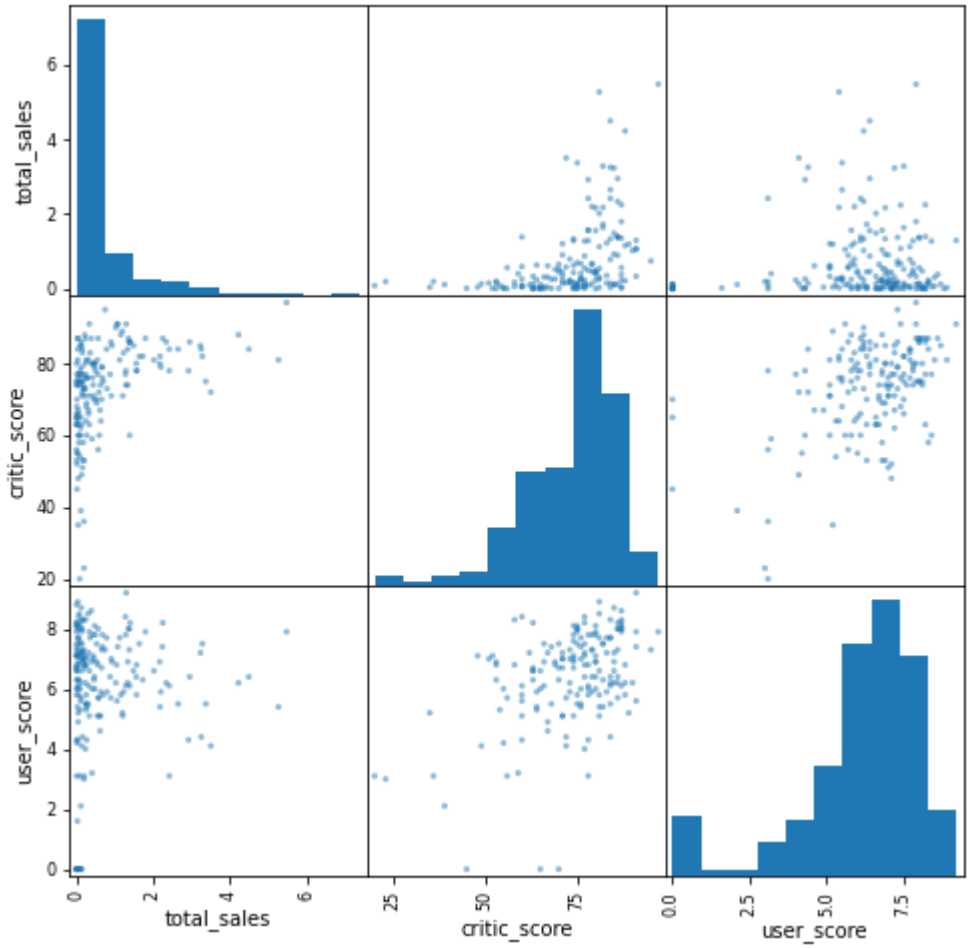
XOne

In [47]:
```
current_data_XOne = current_data[current_data['platform'] == 'XOne']
current_data_XOne = current_data_XOne.loc[:,['total_sales', 'critic_score', '
display(current_data_XOne.corr())

pd.plotting.scatter_matrix(current_data_XOne, figsize=(8, 8), alpha = 0.5)
```

	total_sales	critic_score	user_score
total_sales	1.000000	0.416998	0.074486
critic_score	0.416998	1.000000	0.465368
user_score	0.074486	0.465368	1.000000

Out[47]:
```
array([[<AxesSubplot:xlabel='total_sales', ylabel='total_sales'>,
        <AxesSubplot:xlabel='critic_score', ylabel='total_sales'>,
        <AxesSubplot:xlabel='user_score', ylabel='total_sales'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='critic_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='critic_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='critic_score'>],
       [<AxesSubplot:xlabel='total_sales', ylabel='user_score'>,
        <AxesSubplot:xlabel='critic_score', ylabel='user_score'>,
        <AxesSubplot:xlabel='user_score', ylabel='user_score'>]],
      dtype=object)
```



Вывод: корреляция между отзывами и продажами PS4, PS3, X360, 3DS и XOne не сильно отличается. Видна взаимосвязь между продажами и профессиональной оценкой, чем лучше оценка, тем лучше продается консоль.

Общее распределение игр по жанрам

In [48]:
```
current_genre = pd.pivot_table(current_data, index='genre', values='total_sal
current_genre.sort_values('sum', ascending=False)
```

Out[48]:

genre	mean	median	sum
Action	0.427856	0.120	441.12
Shooter	1.296723	0.440	304.73
Role-Playing	0.521081	0.140	192.80
Sports	0.675634	0.240	181.07
Misc	0.442917	0.120	85.04
Platform	0.717647	0.210	61.00
Racing	0.465217	0.140	53.50
Fighting	0.408165	0.130	44.49
Simulation	0.439000	0.120	35.12
Adventure	0.097450	0.030	29.43
Strategy	0.187887	0.080	13.34

genre	mean	median	sum
Puzzle	0.174643	0.045	4.89

In [49]:
```
current_genre.sort_values('sum', ascending=False)
```

Out[49]:

genre	mean	median	sum
Action	0.427856	0.120	441.12
Shooter	1.296723	0.440	304.73
Role-Playing	0.521081	0.140	192.80
Sports	0.675634	0.240	181.07
Misc	0.442917	0.120	85.04
Platform	0.717647	0.210	61.00
Racing	0.465217	0.140	53.50
Fighting	0.408165	0.130	44.49
Simulation	0.439000	0.120	35.12
Adventure	0.097450	0.030	29.43
Strategy	0.187887	0.080	13.34
Puzzle	0.174643	0.045	4.89

Вывод: самый популярный жанр - Shooter, а менее популярные - Adventure, Puzzle и Strategy

Составление портрета пользователей

NA

In [50]:
```
na_info = data.query('na_sales > 0')
na_info.info()
```

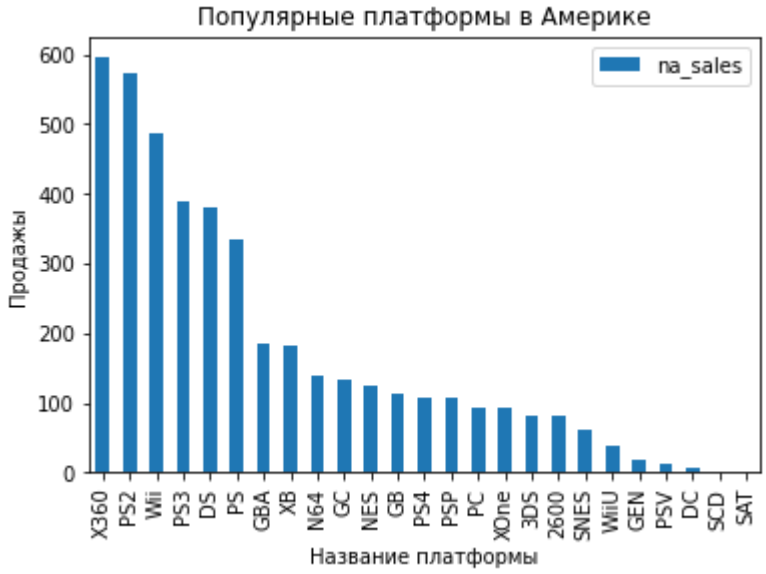
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11995 entries, 0 to 16713
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   name             11995 non-null  object 
 1   platform         11995 non-null  object 
 2   year_of_release  11995 non-null  int64  
 3   genre            11995 non-null  object 
 4   na_sales         11995 non-null  float64
 5   eu_sales         11995 non-null  float64
 6   jp_sales         11995 non-null  float64
 7   other_sales      11995 non-null  float64
 8   critic_score     7366 non-null   float64
 9   user_score       9034 non-null   float64
 10  rating           9007 non-null   object 
 11  total_sales      11995 non-null  float64
dtypes: float64(7), int64(1), object(4)
memory usage: 1.2+ MB
```

Популярные платформы

In [51]:
```
#Создаем таблицу с платформами
na_info_platform = pd.pivot_table(na_info, index='platform', values='na_sales

#Строим график с платформами
na_info_platform.plot.bar()
plt.title("Популярные платформы в Америке")
plt.xlabel('Название платформы')
plt.ylabel('Продажы')
plt.show()

#Добавляем проценты
total_sale_na_pl = na_info_platform['na_sales'].sum()
na_info_platform['percent'] = round(na_info_platform['na_sales']/total_sale_n
print(na_info_platform)
```



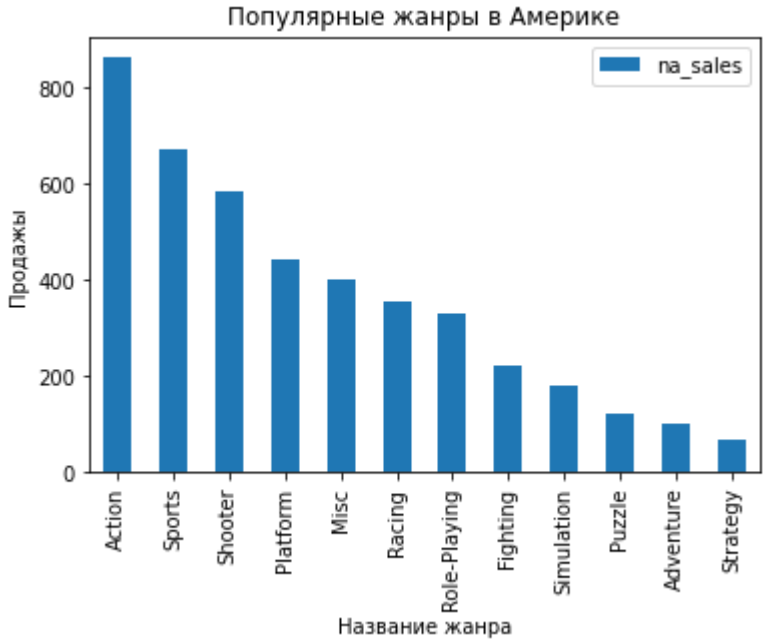
platform	na_sales	percent
X360	595.74	13.72
PS2	572.92	13.20
Wii	486.87	11.21
PS3	390.13	8.99
DS	380.31	8.76
PS	334.72	7.71
GBA	184.12	4.24
XB	182.06	4.19
N64	138.91	3.20
GC	131.94	3.04
NES	125.94	2.90
GB	113.64	2.62
PS4	108.74	2.50
PSP	107.27	2.47
PC	93.34	2.15
XOne	93.12	2.14
3DS	82.65	1.90
2600	80.78	1.86
SNES	61.23	1.41
WiiU	38.10	0.88
GEN	19.27	0.44
PSV	12.47	0.29
DC	5.43	0.13
SCD	1.00	0.02
SAT	0.72	0.02

Популярные жанры

In [52]:
```
#Создаем таблицу с жанрами
na_info_genre = pd.pivot_table(na_info, index='genre', values='na_sales', agg

#Строим график с жанрами
na_info_genre.plot.bar()
plt.title("Популярные жанры в Америке")
plt.xlabel('Название жанра')
plt.ylabel('Продажы')
plt.show()

#Добавляем проценты
total_sale_na = na_info_genre['na_sales'].sum()
na_info_genre['percent'] = round(na_info_genre['na_sales']/total_sale_na * 10
print(na_info_genre)
```




```
              na_sales  percent
genre
Action          863.17    19.88
Sports          671.20    15.46
Shooter         584.83    13.47
Platform        444.44    10.24
Misc            399.57     9.20
Racing          356.86     8.22
Role-Playing    330.04     7.60
Fighting        220.51     5.08
Simulation      180.40     4.16
Puzzle          121.13     2.79
Adventure       101.52     2.34
Strategy         67.75     1.56
```

Рейтинг ESRB

In [53]:
```
#Создаем таблицу с рейтингом
na_info_rating = pd.pivot_table(na_info, index='rating', values='na_sales', a
na_info_rating
```

Out[53]:

rating	na_sales
E	1274.24
T	747.60

rating	na_sales
M	742.89
E10+	345.50
K-A	2.56
EC	1.53
AO	1.26

Вывод про Америку:

- Популярные платформы: X360 16.37%, Wii 13.02%, PS2 12.51%, PS3 7.73%, PS 7.50%
- Популярные жанры: Shooter, Action, Platform, Sports, Misc
- На рейтинг влияет E

EU

In [54]:
```
eu_info = data.query('na_sales >= 1')
eu_info.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 906 entries, 0 to 1953
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   name             906 non-null    object 
 1   platform         906 non-null    object 
 2   year_of_release  906 non-null    int64  
 3   genre            906 non-null    object 
 4   na_sales         906 non-null    float64
 5   eu_sales         906 non-null    float64
 6   jp_sales         906 non-null    float64
 7   other_sales      906 non-null    float64
 8   critic_score     601 non-null    float64
 9   user_score       629 non-null    float64
 10  rating           632 non-null    object 
 11  total_sales      906 non-null    float64
dtypes: float64(7), int64(1), object(4)
memory usage: 92.0+ KB
```

Популярные платформы

In [55]:
```
#Создаем таблицу с платформами
eu_info_platform = pd.pivot_table(eu_info, index='platform', values='eu_sales

#Строим график с платформами
eu_info_platform.plot.bar()
plt.title("Популярные платформы в Европе")
plt.xlabel('Название платформы')
plt.ylabel('Продажы')
plt.show()

#Добавляем проценты
total_sale_eu_pl = eu_info_platform['eu_sales'].sum()
eu_info_platform['percent'] = round(eu_info_platform['eu_sales']/total_sale_e
print(eu_info_platform)
```

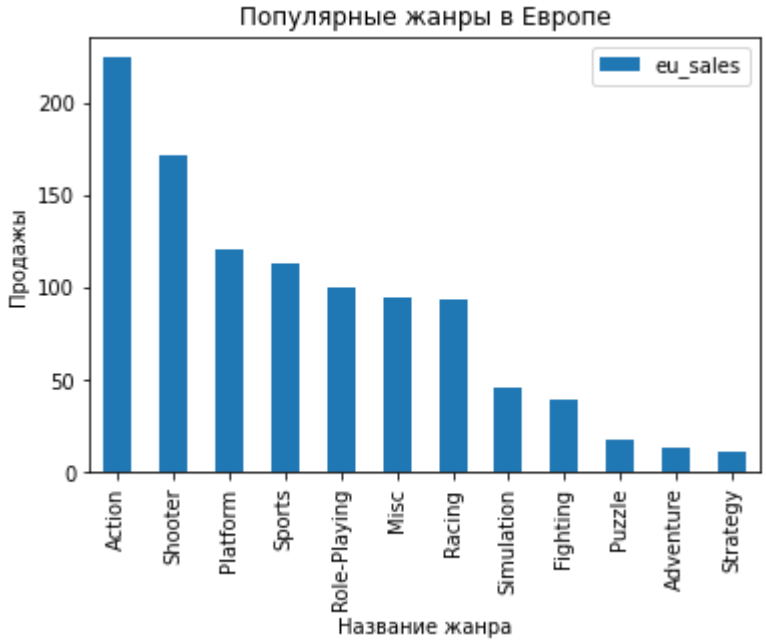
platform	eu_sales	percent
Wii	162.73	15.57
X360	135.26	12.94
PS3	129.26	12.37
PS2	117.46	11.24
DS	98.52	9.43
PS	86.81	8.31
PS4	66.51	6.36
GB	40.35	3.86
3DS	29.10	2.78
PC	26.45	2.53
GBA	24.46	2.34
N64	23.50	2.25
XOne	20.53	1.96
SNES	16.84	1.61
NES	15.85	1.52
XB	11.59	1.11
GC	11.21	1.07
PSP	10.17	0.97
WiiU	9.53	0.91
GEN	5.01	0.48
2600	2.54	0.24
DC	1.22	0.12
SCD	0.36	0.03

Популярные жанры

In [56]:
```
#Создаем таблицу с жанрами
eu_info_genre = pd.pivot_table(eu_info, index='genre', values='eu_sales', agg

#Строим график с жанрами
eu_info_genre.plot.bar()
plt.title("Популярные жанры в Европе")
plt.xlabel('Название жанра')
plt.ylabel('Продажы')
plt.show()

#Добавляем проценты
total_sale_eu = eu_info_genre['eu_sales'].sum()
eu_info_genre['percent'] = round(eu_info_genre['eu_sales']/total_sale_eu * 10
print(eu_info_genre)
```



genre	eu_sales	percent
Action	224.47	21.48
Shooter	171.87	16.44
Platform	120.26	11.51
Sports	112.83	10.79
Role-Playing	99.51	9.52
Misc	94.10	9.00
Racing	93.95	8.99
Simulation	45.71	4.37
Fighting	39.47	3.78
Puzzle	18.12	1.73
Adventure	13.77	1.32
Strategy	11.20	1.07

Рейтинг ESRB

In [57]:
```
#Создаем таблицу с рейтингом
eu_info_rating = pd.pivot_table(eu_info, index='rating', values='eu_sales', a
eu_info_rating
```

Out[57]:

rating	eu_sales
E	312.12
M	268.64
T	145.28
E10+	52.25
AO	0.61
K-A	0.10

Вывод про Европе:

- Популярные платформы: Wii 19.00%, PS3 13.25%, DS 11.19%, PS2 10.75%, X360 10.60%
- Популярные жанры: Action, Shooter, Sports, Platform, Role-Playing
- На рейтинг влияет E

JP

In [58]:
```
jp_info = data.query('jp_sales >= 1')
jp_info.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 243 entries, 0 to 2066
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   name             243 non-null    object
 1   platform         243 non-null    object
 2   year_of_release  243 non-null    int64
 3   genre            243 non-null    object
 4   na_sales         243 non-null    float64
 5   eu_sales         243 non-null    float64
 6   jp_sales         243 non-null    float64
 7   other_sales      243 non-null    float64
 8   critic_score     87 non-null     float64
 9   user_score       87 non-null     float64
 10  rating           87 non-null     object
 11  total_sales      243 non-null    float64
dtypes: float64(7), int64(1), object(4)
memory usage: 24.7+ KB
```

Популярные платформы

In [59]:
```
#Создаем таблицу с платформами
jp_info_platform = pd.pivot_table(jp_info, index='platform', values='jp_sales

#Строим график с платформами
jp_info_platform.plot.bar()
plt.title("Популярные платформы в Японии")
plt.xlabel('Название платформы')
plt.ylabel('Продажы')
plt.show()

#Добавляем проценты
total_sale_jp_pl = jp_info_platform['jp_sales'].sum()
jp_info_platform['percent'] = round(jp_info_platform['jp_sales']/total_sale_j
print(jp_info_platform)
```


```
          jp_sales  percent
platform
DS           89.30    18.09
NES          72.23    14.63
```

```
GB           58.27    11.80
SNES         55.42    11.22
3DS          52.67    10.67
PS           50.69    10.27
Wii          35.93     7.28
PS2          30.52     6.18
N64          15.33     3.10
GBA          11.98     2.43
PSP          11.78     2.39
WiiU          4.01     0.81
PS3           2.95     0.60
GC            1.39     0.28
SAT           1.30     0.26
```

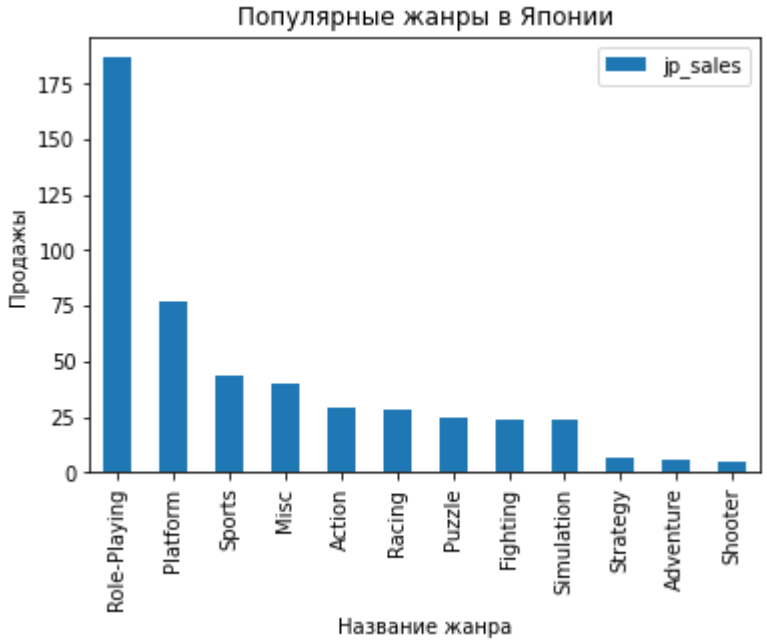
Популярные жанры

In [60]:

```
#Создаем таблицу с жанрами
jp_info_genre = pd.pivot_table(jp_info, index='genre', values='jp_sales', agg

#Строим график с жанрами
jp_info_genre.plot.bar()
plt.title("Популярные жанры в Японии")
plt.xlabel('Название жанра')
plt.ylabel('Продажы')
plt.show()

#Добавляем проценты
total_sale_jp = jp_info_genre['jp_sales'].sum()
jp_info_genre['percent'] = round(jp_info_genre['jp_sales']/total_sale_jp * 10
print(jp_info_genre)
```




```
              jp_sales  percent
genre
Role-Playing    186.79    37.83
Platform         77.28    15.65
Sports           43.21     8.75
Misc             39.98     8.10
Action           29.48     5.97
Racing           28.22     5.72
Puzzle           24.42     4.95
Fighting         23.89     4.84
Simulation       23.62     4.78
Strategy          6.59     1.33
Adventure         5.36     1.09
Shooter           4.93     1.00
```

Рейтинг ESRB

In [61]:
```
#Создаем таблицу с рейтингом
jp_info_rating = pd.pivot_table(jp_info, index='rating', values='jp_sales', a
jp_info_rating
```

Out[61]:

rating	jp_sales
E	110.48
T	47.37
E10+	12.38
M	7.41
K-A	1.46

Вывод про Японии:

- Популярные платформы: DS 18.63%, NES 13.97%, GB 12.50%, SNES 11.03%, 3DS 10.54%
- Популярные жанры: Role-Playing, Platform, Sports, Misc, Action
- На рейтинг влияет E

Other

In [62]:
```
other_info = data.query('other_sales >= 1')
other_info.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 76 entries, 0 to 1511
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   name             76 non-null     object 
 1   platform         76 non-null     object 
 2   year_of_release  76 non-null     int64  
 3   genre            76 non-null     object 
 4   na_sales         76 non-null     float64
 5   eu_sales         76 non-null     float64
 6   jp_sales         76 non-null     float64
 7   other_sales      76 non-null     float64
 8   critic_score     65 non-null     float64
 9   user_score       67 non-null     float64
 10  rating           67 non-null     object 
 11  total_sales      76 non-null     float64
dtypes: float64(7), int64(1), object(4)
memory usage: 7.7+ KB
```

Популярные платформы

In [63]:
```
#Создаем таблицу с платформами
other_info_platform = pd.pivot_table(other_info, index='platform', values='ot

#Строим график с платформами
other_info_platform.plot.bar()
plt.title("Популярные платформы в Японии")
plt.xlabel('Название платформы')
plt.ylabel('Продажы')
plt.show()
```

```
#Добавляем проценты
total_sale_other_pl = other_info_platform['other_sales'].sum()
other_info_platform['percent'] = round(other_info_platform['other_sales']/tot
print(other_info_platform)
```

platform	other_sales	percent
PS2	51.67	35.85
PS3	27.42	19.03
Wii	25.76	17.87
DS	12.10	8.40
PS4	10.38	7.20
X360	10.19	7.07
PSP	2.91	2.02
NES	1.51	1.05
PC	1.18	0.82
GB	1.00	0.69

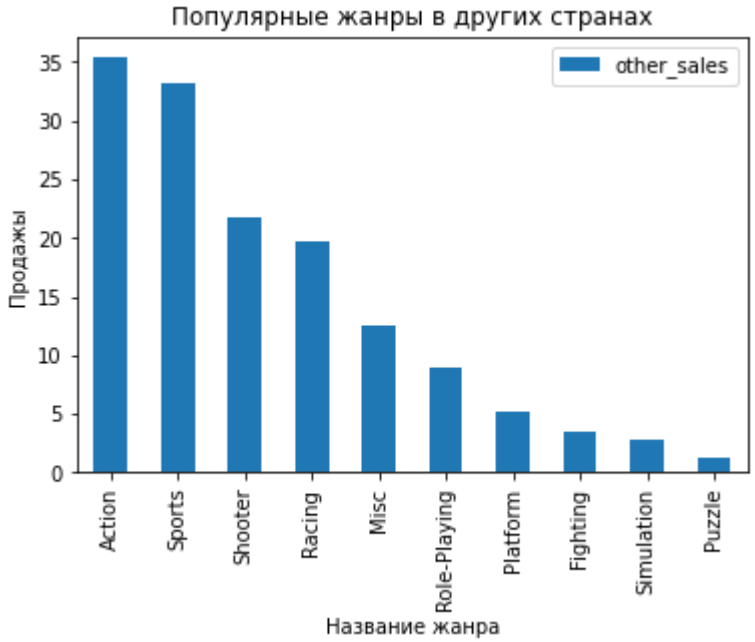
Популярные жанры

In [64]:

```
#Создаем таблицу с жанрами
other_info_genre = pd.pivot_table(other_info, index='genre', values='other_sa

#Строим график с жанрами
other_info_genre.plot.bar()
plt.title("Популярные жанры в других странах")
plt.xlabel('Название жанра')
plt.ylabel('Продажы')
plt.show()

#Добавляем проценты
total_sale_other = other_info_genre['other_sales'].sum()
other_info_genre['percent'] = round(other_info_genre['other_sales']/total_sal
print(other_info_genre)
```



genre	other_sales	percent
Action	35.39	24.56
Sports	33.24	23.06
Shooter	21.69	15.05
Racing	19.78	13.72
Misc	12.55	8.71
Role-Playing	8.92	6.19
Platform	5.12	3.55
Fighting	3.51	2.44
Simulation	2.74	1.90
Puzzle	1.18	0.82

Рейтинг ESRB

In [65]:
```
#Создаем таблицу с рейтингом
other_info_rating = pd.pivot_table(other_info, index='rating', values='other_
other_info_rating
```

Out[65]:

rating	other_sales
E	62.74
M	51.42
T	12.18
E10+	2.74

Вывод по другим странам:

- Популярные платформы: PS2 36.52%, Wii 19.13%, PS3 18.26%, DS 7.83%, PS4 6.96%
- Популярные жанры: Action, Sports, Racing, Shooter, Misc
- На рейтинг влияет E

Вывод

про Америку:
- Популярные платформы: X360 16.37%, Wii 13.02%, PS2 12.51%, PS3 7.73%, PS 7.50%

```
- Популярные жанры: Shooter, Action, Platform, Sports, Misc
- На рейтинг влияет E

про Европе:
- Популярные платформы: Wii 19.00%, PS3 13.25%, DS 11.19%, PS2
10.75%, X360 10.60%
- Популярные жанры: Action, Shooter, Sports, Platform, Role-
Playing
- На рейтинг влияет E

про Японии:
- Популярные платформы: DS 18.63%, NES 13.97%, GB 12.50%, SNES
11.03%, 3DS 10.54%
- Популярные жанры: Role-Playing, Platform, Sports, Misc, Action
- На рейтинг влияет E

по другим странам:
- Популярные платформы: PS2 36.52%, Wii 19.13%, PS3 18.26%, DS
7.83%, PS4 6.96%
- Популярные жанры: Action, Sports, Racing, Shooter, Misc
- На рейтинг влияет E
```

Проверка гипотезы

Средние пользовательские рейтинги платформ Xbox One и PC одинаковые

In [66]:
```
user_xone = current_data.query('platform == "XOne"')
user_xone = user_xone[user_xone['user_score'].notna()]
user_xone['user_score'].describe()
```

Out[66]:
```
count    193.000000
mean       6.149741
std        2.023717
min        0.000000
25%        5.500000
50%        6.600000
75%        7.400000
max        9.200000
Name: user_score, dtype: float64
```

In [67]:
```
user_pc = current_data.query('platform == "PC"')
user_pc = user_pc[user_pc['user_score'].notna()]
user_pc['user_score'].describe()
```

Out[67]:
```
count    211.000000
mean       6.276303
std        1.914458
min        0.000000
25%        5.400000
50%        6.800000
75%        7.700000
max        9.300000
Name: user_score, dtype: float64
```

H_0: средние пользовательские рейтинги платформ Xbox One = PC

H_1: средние пользовательские рейтинги платформ Xbox One ≠ PC

alpha = 0.05

In [68]:
```
results = st.ttest_ind(user_xone['user_score'], user_pc['user_score'])

alpha = 0.05

print('p-значение:', results.pvalue)

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

```
p-значение: 0.5187355060176102
Не получилось отвергнуть нулевую гипотезу
```

Средние пользовательские рейтинги жанров Action разные

In [69]:
```
user_action = current_data.query('genre == "Action"')
user_action = user_action[user_action['user_score'].notna()]
user_action['user_score'].describe()
```

Out[69]:
```
count    571.000000
mean       6.251313
std        2.307139
min        0.000000
25%        5.600000
50%        7.000000
75%        7.750000
max        9.100000
Name: user_score, dtype: float64
```

In [70]:
```
user_sports = current_data.query('genre == "Sports"')
user_sports = user_sports[user_sports['user_score'].notna()]
user_sports['user_score'].describe()
```

Out[70]:
```
count    214.000000
mean       4.971495
std        2.274368
min        0.000000
25%        3.725000
50%        5.600000
75%        6.700000
max        8.800000
Name: user_score, dtype: float64
```

H_0: средние пользовательские рейтинги жанров Action = Sports

H_1: средние пользовательские рейтинги жанров Action ≠ Sports

alpha = 0.05

In [71]:
```
alpha = 0.05

results = st.ttest_ind(user_action['user_score'], user_sports['user_score'])

print('p-значение:', results.pvalue)

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

```
p-значение: 7.819142547903476e-12
Отвергаем нулевую гипотезу
```

Вывод

Проверив первую гипотезу мы выяснили, что средние пользовательские рейтинги платформ Xbox One = PC
Проверив вторую гипотезу мы выяснили, что средние пользовательские рейтинги жанров Action ≠ Sports

Общий вывод

В проекте была проделана огромная работа

1. Были выгружены данные. Таблица нуждалась в доработке, так как встречались пропущенные данные и дубликаты.

2. Предобработка данных
    - Для начала я переименовала таблицы (привела к нижнему регистру)
    - Затем приступила к поиску пропущенных значений
        - Пропущенные строки в "name" и "genre" я удалила, так как их значения были пустыми и не имели никакой ценности для исследования
        - Пропущенные строки в "year_of_release" также пришлось удалить, так как воссоздать информацию нельзя, а удаление 269 строк для такой огромной таблицы не сильно поменяют ситуацию
        - Данные "user_score", "rating" и "critic_score" обладают светло оранжевым цветом в таблице долей пропуской, так что просто удалить эти значения мы не можем, а поставить медианное значение может стать дезинформацией, придется оставить данные в таком виде
    - Работа с дубликатами (удаление явных)
    - Преобразовала тип данных
        - "year_of_release", "na_sales", "eu_sales", "jp_sales" и "other_sales" были преобразованны в целые числа для удобства дальшейшей работы с ними
    - Добавляем столбец с суммарными продажами
    - "tbd" - недостаточно данных для оценки

3. Исследовательский анализ данных
    - резкий скачек выпусков новых игр можно разделить на три периода: конец 90х, 2003г и самый пик - 2008г. После, с 2010 года количество выпущенных игр понемногу начало спускаться
    - У каждой платформыы был свой пик продаж, самый молодой - X360. Самая продоваемая платформа - Wii, чей пик пришелся на 2006г.
    - Новые платформы появляются каждый год. Средняя продолжительность популярности 10 лет
    - На графике "Ящик с усами" отчетливо видно как PS4 обгоняет своих конкурентов в продажах, и старается его догнать X360.
    - Корреляция между отзывами и продажами PS4, PS3, X360, 3DS и XOne не сильно отличается. Видна взаимосвязь между продажами и профессиональной оценкой, чем лучше оценка, тем лучше продается консоль.

4. Составление портрета пользователей

про Америку:
- Популярные платформы: X360 16.37%, Wii 13.02%, PS2 12.51%, PS3 7.73%, PS 7.50%
- Популярные жанры: Shooter, Action, Platform, Sports, Misc
- На рейтинг влияет E

про Европе:
- Популярные платформы: Wii 19.00%, PS3 13.25%, DS 11.19%, PS2 10.75%, X360 10.60%
- Популярные жанры: Action, Shooter, Sports, Platform, Role-Playing
- На рейтинг влияет E

про Японии:
- Популярные платформы: DS 18.63%, NES 13.97%, GB 12.50%, SNES 11.03%, 3DS 10.54%
- Популярные жанры: Role-Playing, Platform, Sports, Misc, Action
- На рейтинг влияет E

по другим странам:
- Популярные платформы: PS2 36.52%, Wii 19.13%, PS3 18.26%, DS 7.83%, PS4 6.96%
- Популярные жанры: Action, Sports, Racing, Shooter, Misc
- На рейтинг влияет E

5. Проверка гипотез

1 – Проверив первую гипотезу мы выяснили, что средние пользовательские рейтинги платформ Xbox One = PC

2 – Проверив вторую гипотезу мы выяснили, что средние пользовательские рейтинги жанров Action ≠ Sports