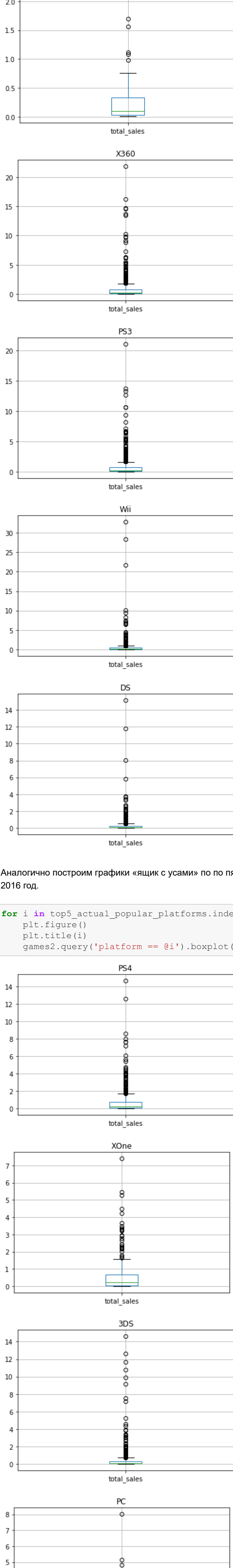
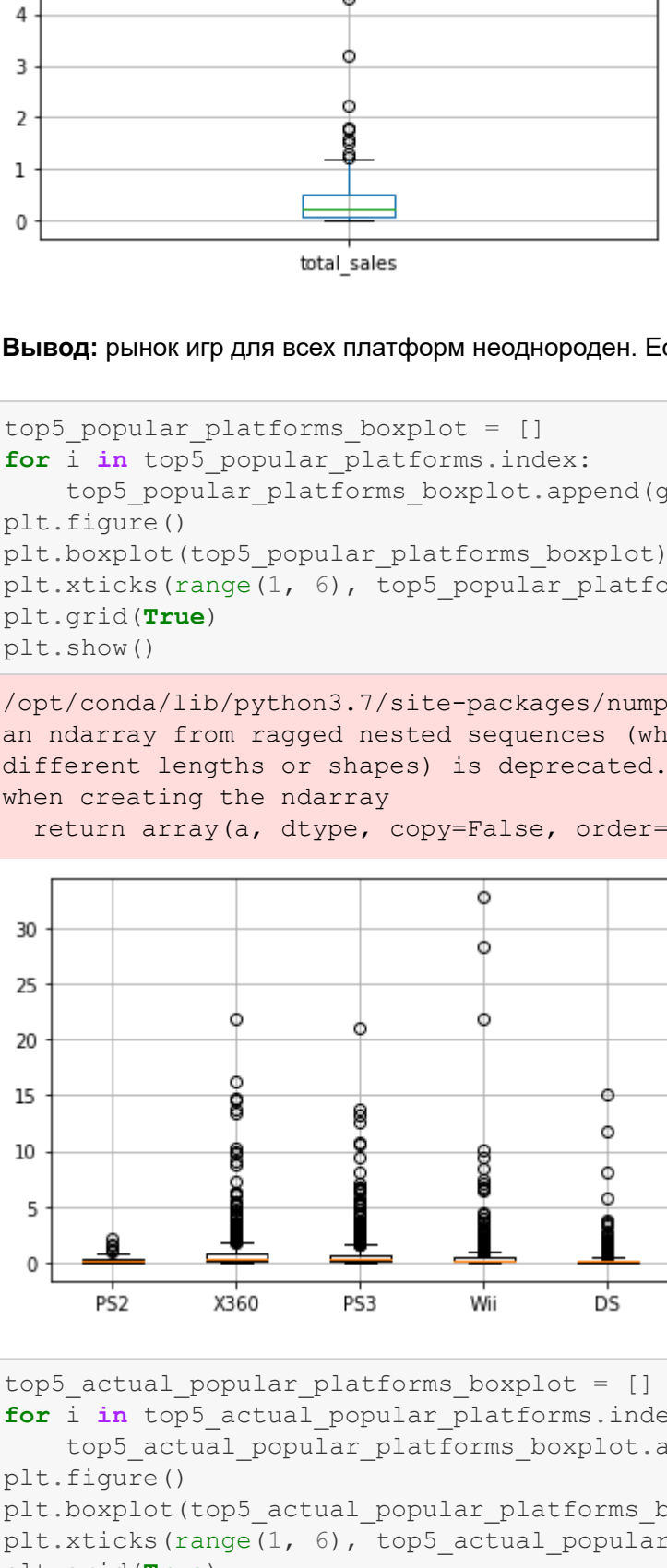



```
In [31]: for i in top5_popular_platforms.index:
plt.figure()
plt.title(i)
games2.query('platform == @i').boxplot(column=['total_sales'])
```



Аналогично построим графики «ящик с усами» по пяти актуальным платформам с наибольшими суммарными продажами за 2016 год.

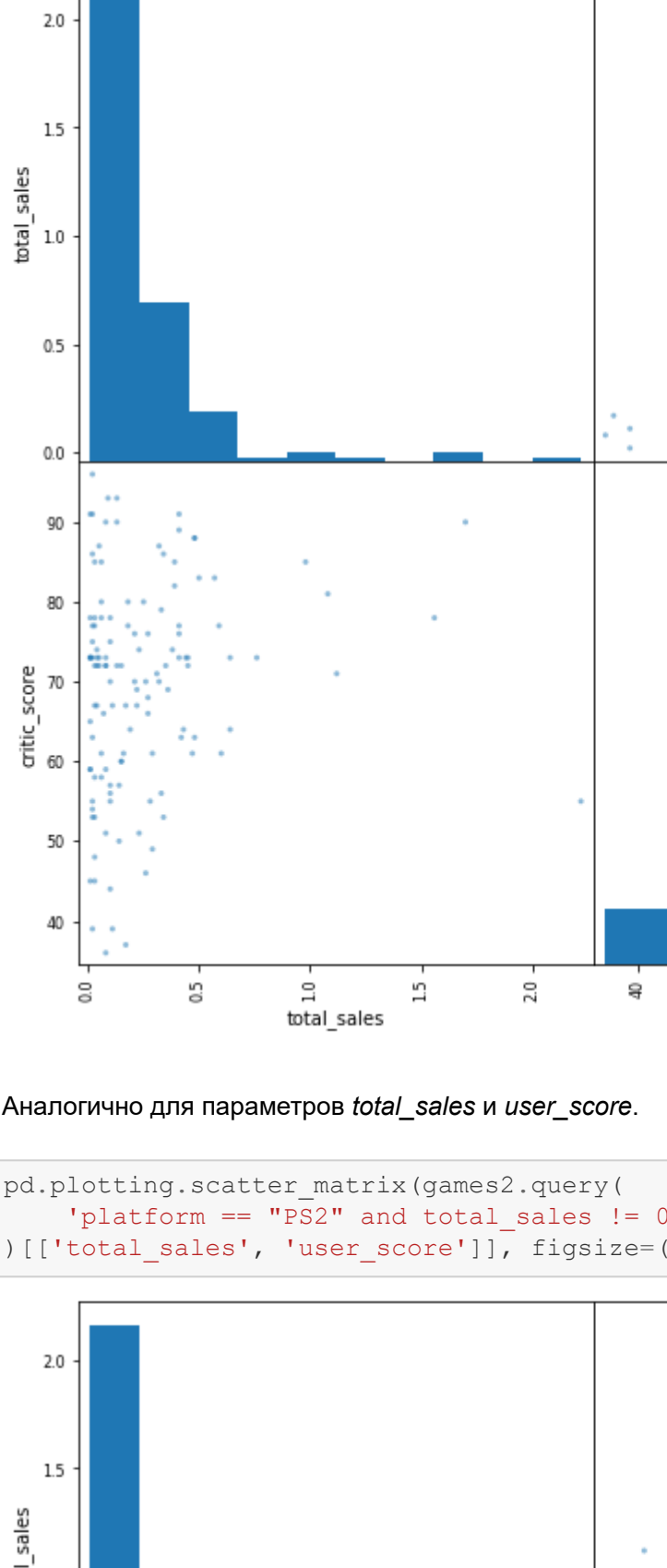
```
In [32]: for i in top5_actual_popular_platforms.index:
plt.figure()
plt.title(i)
games2.query('platform == @i').boxplot(column=['total_sales'])
```



Вывод: рынок игр для всех платформ неоднороден. Есть очевидные хиты продаж и множество проходных игр.

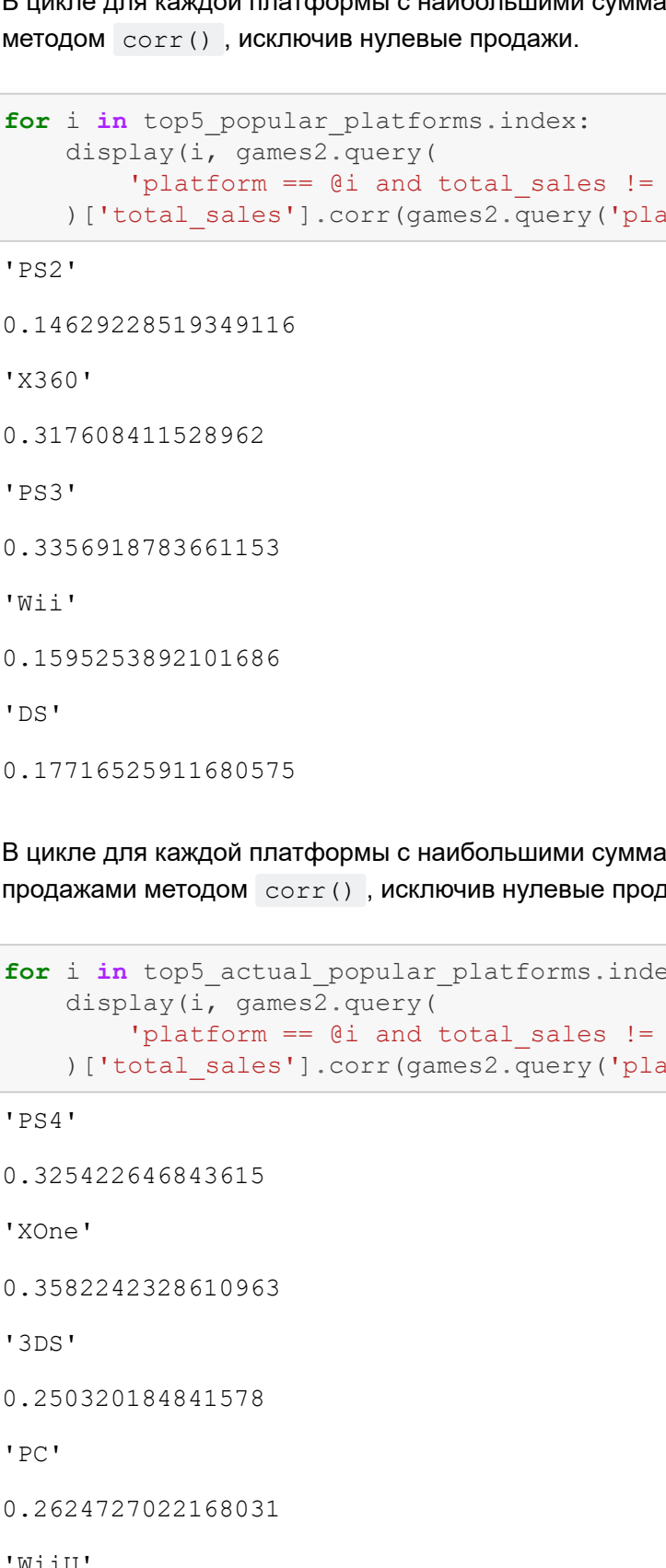
```
In [33]: top5_popular_platforms_boxplot = []
for i in top5_popular_platforms.index:
    top5_popular_platforms_boxplot.append(games2.query('platform == @i')['total_sales'])
plt.figure()
plt.boxplot(top5_popular_platforms_boxplot)
plt.xticks(range(1, 6), top5_popular_platforms.index)
plt.grid(True)
plt.show()

/opt/conda/lib/python3.7/site-packages/numpy/core/_asarray.py:83: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray
return array(a, dtype, copy=False, order=order)
```



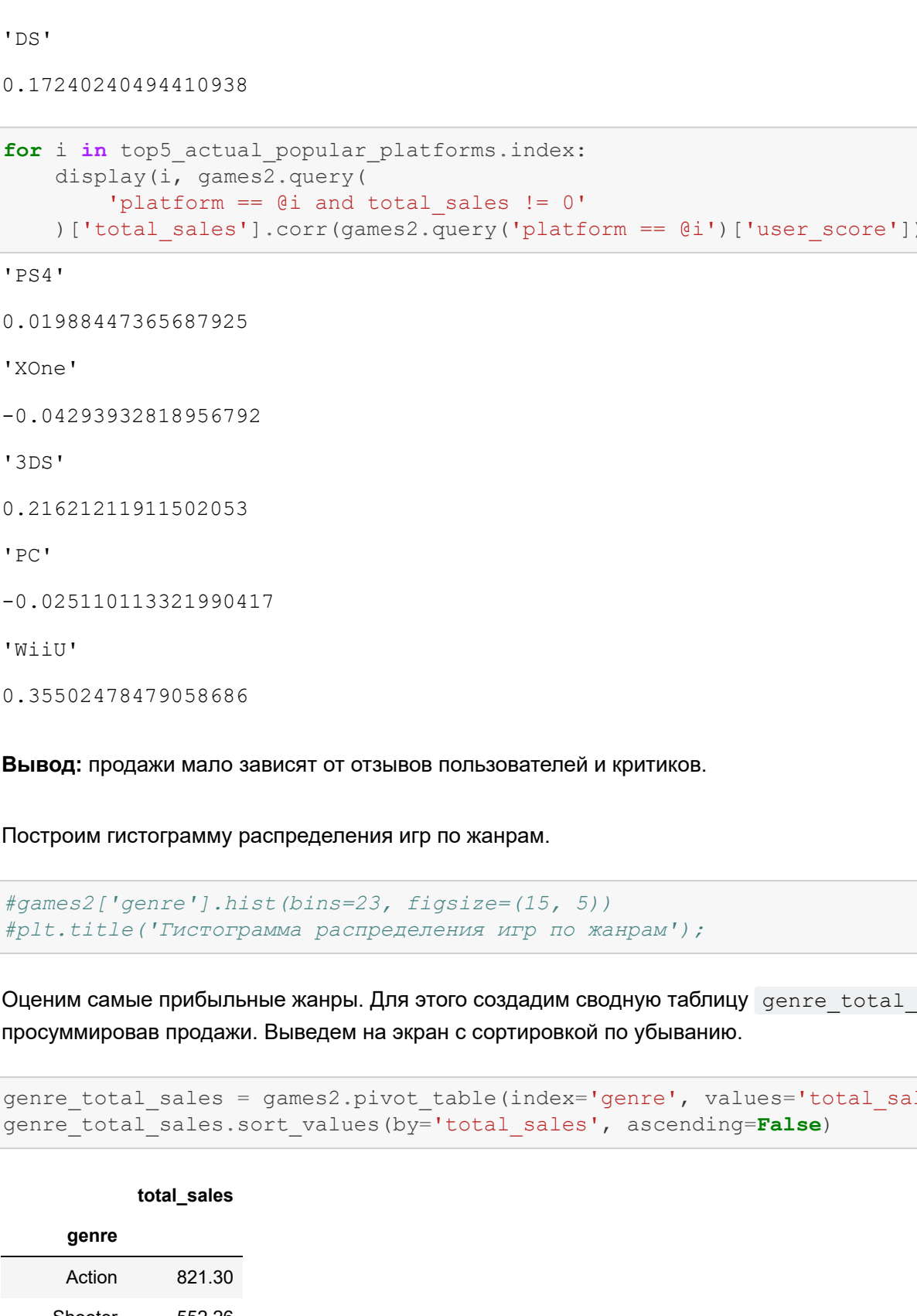
```
In [34]: top5_actual_popular_platforms_boxplot = []
for i in top5_actual_popular_platforms.index:
    top5_actual_popular_platforms_boxplot.append(games2.query('platform == @i')['total_sales'])
plt.figure()
plt.boxplot(top5_actual_popular_platforms_boxplot)
plt.xticks(range(1, 6), top5_actual_popular_platforms.index)
plt.grid(True)
plt.show()

/opt/conda/lib/python3.7/site-packages/numpy/core/_asarray.py:83: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray
return array(a, dtype, copy=False, order=order)
```



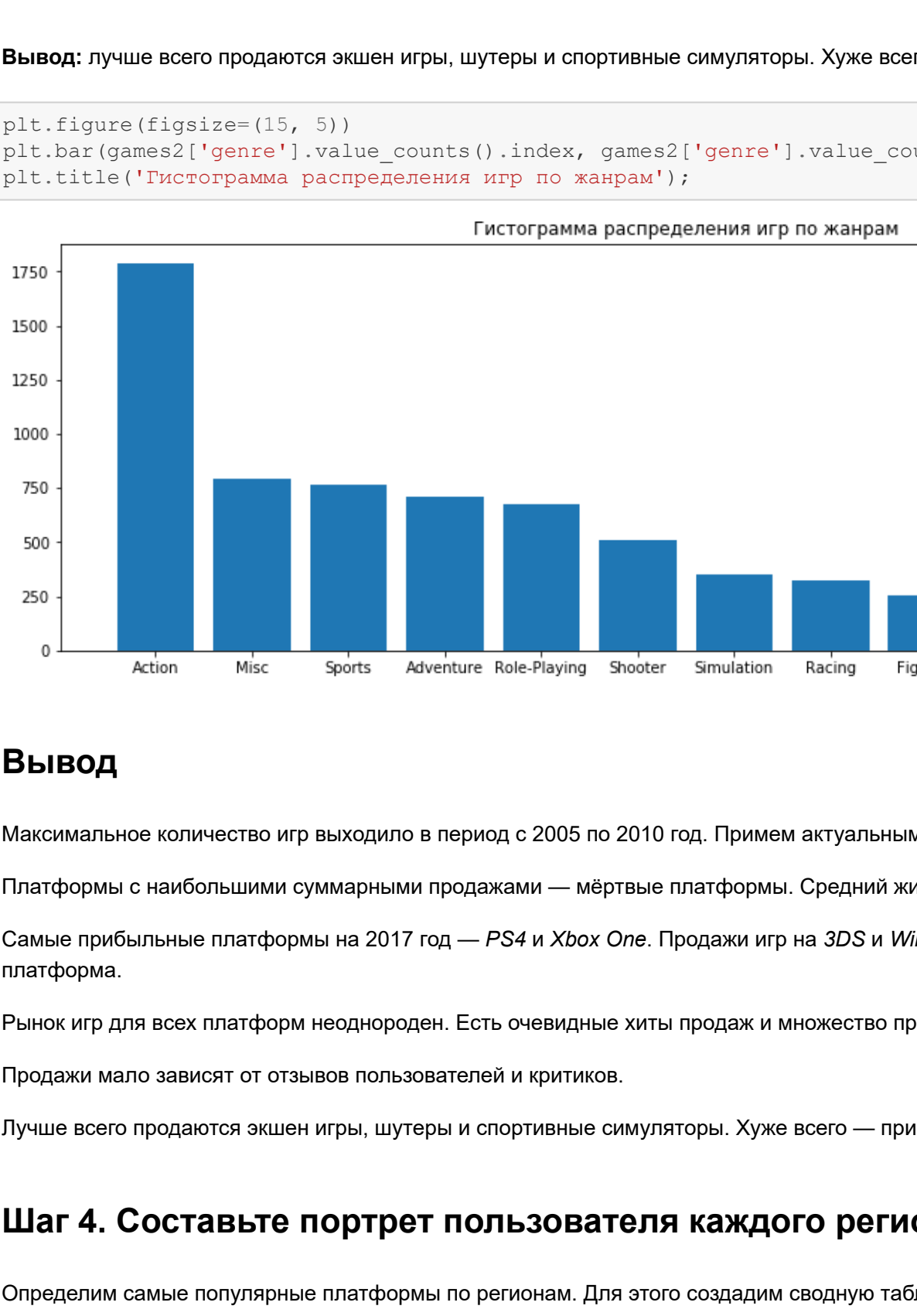
Построим, как влияют на продажи отзывы пользователей и критиков внутри платформы PS2. Для этого построим диаграмму рассеивания методом `pd.plotting.scatter_matrix(df)` для параметров `total_sales` и `critic_score`, отобразив их для PS2 методом `query()` и исключив нулевые продажи.

```
In [35]: pd.plotting.scatter_matrix(games2.query(
    'platform == "PS2" and total_sales != 0'
)[['total_sales', 'critic_score']], figsize=(10, 10))
```



Аналогично для параметров `total_sales` и `user_score`.

```
In [36]: pd.plotting.scatter_matrix(games2.query(
    'platform == "PS2" and total_sales != 0'
)[['total_sales', 'user_score']], figsize=(10, 10))
```



В цикле для каждой платформы с наибольшими суммарными продажами посчитаем корреляцию между отзывами и продажами методом `corr()`, исключив нулевые продажи.

```
In [37]: for i in top5_popular_platforms.index:
display(i, games2.query(
    'platform == @i and total_sales != 0'
)[['total_sales']].corr(games2.query('platform == @i')['critic_score']))
```

```
'PS2'
0.14629228519349116
'X360'
0.317608411528962
'PS3'
0.3356918783661153
'Wii'
0.15952538921016866
'DS'
0.17716525911680575
```

В цикле для каждой платформы с наибольшими суммарными продажами за 2016 год посчитаем корреляцию между отзывами и продажами методом `corr()`, исключив нулевые продажи.

```
In [38]: for i in top5_actual_popular_platforms.index:
display(i, games2.query(
    'platform == @i and total_sales != 0'
)[['total_sales']].corr(games2.query('platform == @i')['critic_score']))
```

```
'PS2'
0.0496239951986465
'X360'
0.07823308948884448
'PS3'
0.08580678593641865
'Wii'
0.1328602575196307
'DS'
0.17240240494410938
```

```
In [40]: for i in top5_actual_popular_platforms.index:
display(i, games2.query(
    'platform == @i and total_sales != 0'
)[['total_sales']].corr(games2.query('platform == @i')['user_score']))
```

```
'PS4'
0.1988447365687925
'XOne'
-0.04293932818956792
'3DS'
0.2162121191502053
'PC'
-0.025110113321990417
'WiiU'
0.35502478479058686
```

Вывод: продажи мало зависят от отзывов пользователей и критиков.

Построим гистограмму распределения игр по жанрам.

```
In [41]: #games2['genre'].hist(bins=23, figsize=(15, 5))
plt.title('Гистограмма распределения игр по жанрам')
```

Оценим самые прибыльные жанры. Для этого создадим сводную таблицу `genre_total_sales`, сгруппировав по жанру и просуммировав продажи. Выведем на экран в сортировкой по убыванию.

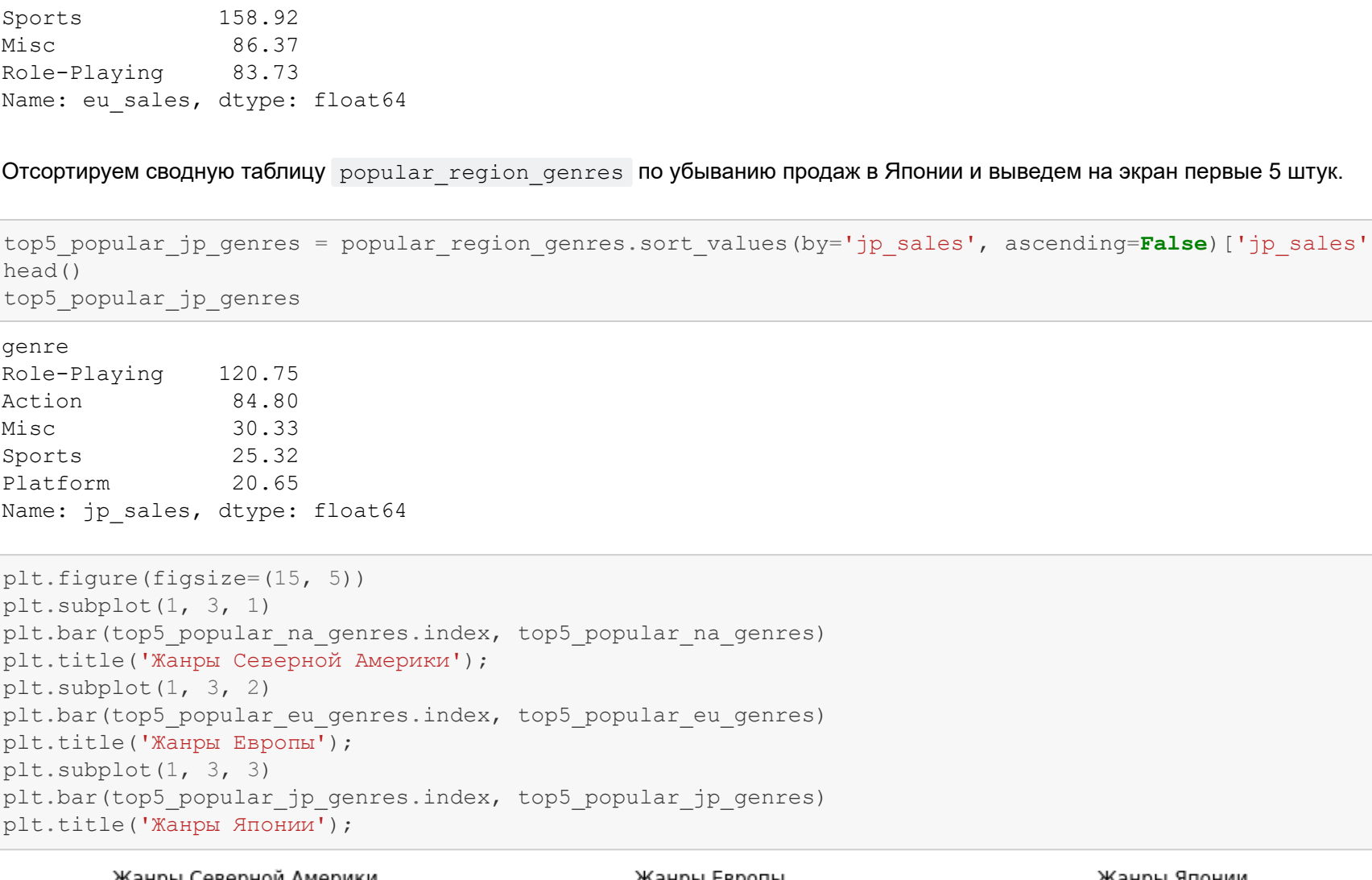
```
In [42]: genre_total_sales = games2.pivot_table(index='genre', values='total_sales', aggfunc='sum')
genre_total_sales.sort_values(by='total_sales', ascending=False)
```

```
Out [42]:
```

total_sales	
Action	821.30
Shooter	552.26
Sports	467.77
Role-Playing	363.34
Misc	311.37
Platform	160.30
Racing	157.96
Fighting	113.49
Simulation	105.56
Adventure	81.50
Strategy	48.87
Puzzle	41.01

Вывод: лучше всего продаются экшен игры, шутеры и спортивные симуляторы. Хуже всего — приключения, стратегии и puzzle.

```
In [43]: plt.figure(figsize=(15, 5))
plt.bar(games2['genre'].value_counts().index, games2['genre'].value_counts())
plt.title('Гистограмма распределения игр по жанрам')
```



Вывод

Максимальное количество игр выходило в период с 2005 по 2010 год. Примем актуальный период с 2000 по 2016 год.

Платформы с наибольшими суммарными продажами — мертвые платформы. Средний жизненный цикл платформы — 8.5 лет.

Самые прибыльные платформы на 2017 год — PS4 и Xbox One. Продажи игр на 3DS и WiiU падают. PC — самая стабильная платформа.

Рынок игр для всех платформ неоднороден. Есть очевидные хиты продаж и множество проходных игр.

Продажи мало зависят от отзывов пользователей и критиков.

Лучше всего продаются экшен игры, шутеры и спортивные симуляторы. Хуже всего — приключения, стратегии и puzzle.

Шаг 4. Составьте портрет пользователя каждого региона

Определим самые популярные платформы по регионам. Для этого создадим сводную таблицу `popular_region_platforms`, сгруппировав по платформам и просуммировав продажи по каждому региону.

```
In [44]: popular_region_platforms = games2.pivot_table(index='platform', values=['na_sales', 'eu_sales', 'jp_sales'],
    aggfunc='sum')
popular_region_platforms
```

```
Out [44]:
```

platform	na_sales	jp_sales	na_sales
2600	0.07	0.00	1.10
3DS	61.48	100.65	83.13
DS	54.18	52.11	121.61
PC	84.28	0.00	41.59
PS2	6.61	2.74	11.68
PS3	256.72	69.00	293.44
PS4	141.09	15.86	108.74
PSP	18.34	50.88	22.37
PSV	13.12	21.91	12.58
Wii	124.65	34.34	238.38
WiiU	25.13	13.01	38.10
X360	198.11	7.46	410.99
XB	0.38	0.00	0.76
XOne	51.59	0.34	93.12

Отсортируем сводную таблицу `popular_region_platforms` по убыванию продаж в Северной Америке и выведем на экран первые 5 штук.

```
In [45]: top5_ESRB_na_platforms = popular_region_platforms.sort_values(by='na_sales', ascending=False)['na_sales'].head()
top5_ESRB_na_platforms
```

```
Out [45]:
```

platform	na_sales
X360	410.99
PS3	293.44
Wii	238.38
DS	121.61
PS4	108.74

Name: na_sales, dtype: float64

Отсортируем сводную таблицу `popular_region_platforms` по убыванию продаж в Европе и выведем на экран первые 5 штук.

```
In [46]: top5_popular_eu_platforms = popular_region_platforms.sort_values(by='eu_sales', ascending=False)['eu_sales'].head()
top5_popular_eu_platforms
```

```
Out [46]:
```

platform	eu_sales
PS3	256.72
X360	198.11
PS4	141.09
Wii	124.65
PC	84.28

Name: eu_sales, dtype: float64

Отсортируем сводную таблицу `popular_region_platforms` по убыванию продаж в Японии и выведем на экран первые 5 штук.

```
In [47]: top5_popular_na_platforms = popular_region_platforms.sort_values(by='jp_sales', ascending=False)['jp_sales'].head()
top5_popular_na_platforms
```

```
Out [47]:
```

platform	jp_sales
3DS	100.65
PS3	69.00
DS	52.11
PSP	50.88
Wii	34.34

Name: jp_sales, dtype: float64

```
In [48]: plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
plt.bar(top5_popular_na_platforms.index, top5_popular_na_platforms)
plt.title('Платформы Северной Америки');
plt.subplot(1, 3, 2)
plt.bar(top5_popular_eu_platforms.index, top5_popular_eu_platforms)
plt.title('Платформы Европы');
plt.subplot(1, 3, 3)
plt.bar(top5_popular_na_platforms.index, top5_popular_na_platforms)
plt.title('Платформы Японии');
```


Вывод: в Северной Америке популярнее Xbox, в Европе — PlayStation, в Японии — более экзотические приставки, такие как Nintendo.

Определим самые популярные жанры по регионам. Для этого создадим сводную таблицу `popular_region_genres`, сгруппировав по жанрам и просуммировав продажи по каждому региону.

```
In [49]: popular_region_genres = games2.pivot_table(index='genre', values=['na_sales', 'eu_sales', 'jp_sales'],
aggfunc='sum')
popular_region_genres
```

```
Out [49]:
```

genre	na_sales	jp_sales	na_sales
Action	275.51	84.80	367.06
Adventure	24.34	19.04	30.90
Fighting	27.74	16.92	56.56
Misc	86.37	30.33	165.01
Platform	49.68	20.65	74.98
Puzzle	12.04	5.05	20.74
Racing	66.39	7.19	61.63
Role-Playing	83.73	120.75	131.19
Shooter	193.65	15.22	277.80
Simulation	37.90	14.27	44.54
Sports	158.92	25.32	229.06
Strategy	17.48	8.86	18.12

Отсортируем сводную таблицу `popular_region_genres` по убыванию продаж в Северной Америке и выведем на экран первые 5 штук.

```
In [50]: top5_popular_na_genres = popular_region_genres.sort_values(by='na_sales', ascending=False)['na_sales'].head()
top5_popular_na_genres
```

```
Out [50]:
```

genre	na_sales
Action	367.06
Shooter	277.80
Sports	229.06
Misc	165.01
Role-Playing	131.19

Name: na_sales, dtype: float64

Отсортируем сводную таблицу `popular_region_genres` по убыванию продаж в Европе и выведем на экран первые 5 штук.

```
In [51]: top5_popular_eu_genres = popular_region_genres.sort_values(by='eu_sales', ascending=False)['eu_sales'].head()
top5_popular_eu_genres
```

```
Out [51]:
```

genre	eu_sales
Action	275.51
Shooter	193.65
Sports	158.92
Misc	86.37
Role-Playing	83.73

Name: eu_sales, dtype: float64

Отсортируем сводную таблицу `popular_region_genres` по убыванию продаж в Японии и выведем на экран первые 5 штук.

```
In [52]: top5_popular_na_genres = popular_region_genres.sort_values(by='jp_sales', ascending=False)['jp_sales'].head()
top5_popular_na_genres
```

```
Out [52]:
```

genre	jp_sales
Role-Playing	120.75
Action	84.80
Misc	30.33
Sports	25.32
Platform	20.65

Name: jp_sales, dtype: float64

Отсортируем сводную таблицу `popular_region_genres` по убыванию продаж в Японии и выведем на экран первые 5 штук.

```
In [53]: plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
plt.bar(top5_popular_na_genres.index, top5_popular_na_genres)
plt.title('Жанры Северной Америки');
plt.subplot(1, 3, 2)
plt.bar(top5_popular_eu_genres.index, top5_popular_eu_genres)
plt.title('Жанры Европы');
plt.subplot(1, 3, 3)
plt.bar(top5_popular_na_genres.index, top5_popular_na_genres)
plt.title('Жанры Японии');
```


Вывод: жанровые предпочтения игроков из Северной Америки и Европы схожи, это — экшен игры, спортивные симуляторы и шутеры. Игры в Японии популярнее всего ролевые игры.

Определим рейтинги ESRB в каждом регионе. Для этого создадим сводную таблицу `popular_region_genres`, сгруппировав по жанрам и просуммировав продажи по каждому региону.

```
In [54]: ESRB_regions = games2.pivot_table(index='rating', values=['na_sales', 'eu_sales', 'jp_sales'], aggfunc='sum')
popular_regions
```

```
Out [54]:
```

rating	na_sales	jp_sales	na_sales
E	261.79	61.49	405.75
E10+	123.52	20.82	205.37
EC	0.00	0.00	1.32
M	329.57	36.29	444.13
RP	0.07	0.00	0.00
T	154.32	50.76	229.07

Отсортируем сводную таблицу `ESRB_regions` по убыванию продаж в Северной Америке и выведем на экран первые 5 штук.

```
In [55]: top5_ESRB_na = ESRB_regions.sort_values(by='na_sales', ascending=False)['na_sales'].head()
top5_ESRB_na
```

```
Out [55]:
```

rating	na_sales
M	444.13
E	405.75
T	229.07
E10+	205.37
EC	1.32

Name: na_sales, dtype: float64

Отсортируем сводную таблицу `ESRB_regions` по убыванию продаж в Европе и выведем на экран первые 5 штук.

```
In [56]: top5_ESRB_eu = ESRB_regions.sort_values(by='eu_sales', ascending=False)['eu_sales'].head()
top5_ESRB_eu
```

```
Out [56]:
```

rating	eu_sales
M	329.57
E	261.79
T	154.32
E10+	123.52
RP	0.07

Name: eu_sales, dtype: float64

Отсортируем сводную таблицу `ESRB_regions` по убыванию продаж в Японии и выведем на экран первые 5 штук.

```
In [57]: top5_ESRB_jp = ESRB_regions.sort_values(by='jp_sales', ascending=False)['jp_sales'].head()
top5_ESRB_jp
```

```
Out [57]:
```

rating	jp_sales
E	61.49
T	50.76
M	36.29
E10+	20.82
EC	0.00

Name: jp_sales, dtype: float64

Отсортируем сводную таблицу `ESRB_regions` по убыванию продаж в Японии и выведем на экран первые 5 штук.

```
In [58]: plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
plt.bar(top5_ESRB_na.index, top5_ESRB_na)
plt.title('Рейтинги Северной Америки');
plt.subplot(1, 3, 2)
plt.bar(top5_ESRB_eu.index, top5_ESRB_eu)
plt.title('Рейтинги Европы');
plt.subplot(1, 3, 3)
plt.bar(top5_ESRB_jp.index, top5_ESRB_jp)
plt.title('Рейтинги Японии');
```


Вывод: в Европе и Северной Америке популярнее жанры «Mature» — «Для взрослых» и «Everyone» — «Для всех». В Японии предпочитают «Everyone» — «Для всех» и «Teen» — «Для подростков».

Вывод

В Северной Америке популярнее Xbox, в Европе — PlayStation, в Японии — более экзотические приставки, такие как Nintendo.

Жанровые предпочтения игроков из Северной Америки и Европы схожи, это — экшен игры, спортивные симуляторы и шутеры. Но в Японии популярнее всего ролевые игры.

В Европе и Северной Америке популярны жанры «Mature» — «Для взрослых» и «Everyone» — «Для всех». В Японии предпочитают «Everyone» — «Для всех» и «Teen» — «Для подростков».

Шаг 5. Проверьте гипотезы

Сформулируем следующие гипотезы.

Нулевая гипотеза: средние пользовательские рейтинги платформ Xbox One и PC одинаковые.

Альтернативная гипотеза: средние пользовательские рейтинги платформ Xbox One и PC различаются.

Выборы независимы, поскольку мы измеряем рейтинги двух разных игровых платформ.

Выберем пользовательские рейтинги на платформе Xbox One методом `query()`, удалив пустые значения методом `dropna()`.

```
In [59]: XOne_user_scores = games2.query('platform == "XOne"')['user_score'].dropna()
XOne_user_scores

Out[59]: 99      8.1
165      7.9
179      5.4
242      6.4
270      6.2
...
16630     8.2
16643     6.6
16645     7.2
16660     6.7
16672     9.5
Name: user_score, Length: 246, dtype: float64
```

Выберем пользовательские рейтинги на платформе PC методом `query()`, удалив пустые значения методом `dropna()`.

```
In [60]: PC_user_scores = games2.query('platform == "PC"')['user_score'].dropna()
PC_user_scores

Out[60]: 85      7.6
192      4.0
218      8.2
313      8.1
458      3.9
...
16690     6.7
16692     7.6
16696     5.8
16702     7.2
16705     5.8
Name: user_score, Length: 583, dtype: float64
```

Найдём дисперсии пользовательских рейтингов платформ Xbox One и PC соответственно.

```
In [61]: np.var(XOne_user_scores)
```

```
Out[61]: 1.8590752858748099
```

```
In [62]: np.var(PC_user_scores)
```

```
Out[62]: 2.275273221551742
```

Проверим гипотезу о равенстве среднего двух генеральных совокупностей методом `scipy.stats.ttest_ind()`. Дисперсии пользовательских рейтингов платформ Xbox One и PC различаются, поэтому примем параметр `equal_var = False`.

```
In [63]: alpha = 0.05 # пороговое значение

results = st.ttest_ind(
    XOne_user_scores,
    PC_user_scores,
    equal_var = False
)

print('p-значение:', results.pvalue)

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

p-значение: 0.5742999617022789
Не получилось отвергнуть нулевую гипотезу
```

Вывод: средние пользовательские рейтинги платформ Xbox One и PC одинаковые.

Сформулируем следующие гипотезы.

Нулевая гипотеза: средние пользовательские рейтинги жанров Action и Sports одинаковые.

Альтернативная гипотеза: средние пользовательские рейтинги жанров Action и Sports разные.

Выборы независимы, поскольку мы измеряем рейтинги двух разных жанров.

Выберем пользовательские рейтинги жанров Action методом `query()`, удалив пустые значения методом `dropna()`.

```
In [64]: action_user_scores = games2.query('genre == "Action"')['user_score'].dropna()
action_user_scores

Out[64]: 16      8.2
23      8.1
42      8.3
46      8.7
81      6.6
...
16680     7.2
16691     5.7
16692     7.6
16699     6.7
16710     7.4
Name: user_score, Length: 1660, dtype: float64
```

Выберем пользовательские рейтинги жанров Sports методом `query()`, удалив пустые значения методом `dropna()`.

```
In [65]: sports_user_scores = games2.query('genre == "Sports"')['user_score'].dropna()
sports_user_scores

Out[65]: 3      8.0
15      7.4
77      4.3
94      5.0
119      4.2
...
16638     5.3
16639     7.1
16643     6.6
16688     4.8
16700     6.0
Name: user_score, Length: 716, dtype: float64
```

Найдём дисперсии пользовательских рейтингов жанров Action и Sports соответственно.

```
In [66]: np.var(action_user_scores)
```

```
Out[66]: 1.9941436928436638
```

```
In [67]: np.var(sports_user_scores)
```

```
Out[67]: 2.717323956025093
```

Проверим гипотезу о равенстве среднего двух генеральных совокупностей методом `scipy.stats.ttest_ind()`. Дисперсии пользовательских рейтингов жанров Action и Sports различаются, поэтому примем параметр `equal_var = False`.

```
In [68]: alpha = 0.05 # пороговое значение

results = st.ttest_ind(
    action_user_scores,
    sports_user_scores,
    equal_var = False
)

print('p-значение:', results.pvalue)

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

p-значение: 4.432399268923206e-12
Отвергаем нулевую гипотезу
```

Вывод: средние пользовательские рейтинги жанров Action и Sports разные.

Вывод

Средние пользовательские рейтинги платформ Xbox One и PC различаются.

Средние пользовательские рейтинги жанров Action и Sports разные.

Шаг 6. Напишите общий вывод

Для заполнения пропусков в столбцах можно копировать значения из открытых источников средством автоматизации AutoIt.

Возможные причины пропусков — ошибки названий для серий игр или для малоизвестность тайтлов.

Пропуски в столбцах `critic_score`, `rating` оставим без изменения, поскольку удаление соответствующих строк может негативно сказаться на результатах исследования.

Аббревиатура tbd (To Be Determined — будет определено или To Be Decided — будет решено) в столбцах с оценками пользователей означает, что пользовательские оценки отсутствуют или их недостаточно, чтобы выставить оценку игре.

Максимальное количество игр выходило в период с 2005 по 2010 год. Примем актуальный период с 2000 по 2016 год.

Платформы с наибольшими суммарными продажами — мертвые платформы. Средний жизненный цикл платформы — 8.5 лет.

Самые прибыльные платформы на 2017 год — PS4 и Xbox One. Продажи игр на 3DS и WiiU падают. PC — самая стабильная платформа.

Рынок игр для всех платформ неоднороден. Есть очевидные хиты продаж и множество проходных игр.

Продажи мало зависят от отзывов пользователей и критиков.

Лучше всего продаются экшен игры, шутеры и спортивные симуляторы. Хуже всего — приключения, стратегии и пазлы.

В Северной Америке популярнее Xbox, в Европе — PlayStation, в Японии — более экзотические приставки, такие как Nintendo.

Жанровые предпочтения игроков из Северной Америки и Европы схожи, это — экшен игры, спортивные симуляторы и шутеры. Но в Японии популярнее всего ролевые игры.

В Европе и Северной Америке популярны жанры «Mature» — «Для взрослых» и «Everyone» — «Для всех». В Японии предпочитают «Everyone» — «Для всех» и «Teen» — «Для подростков».

Средние пользовательские рейтинги платформ Xbox One и PC одинаковые.

Средние пользовательские рейтинги жанров Action и Sports разные.

Общий вывод: наиболее популярны экшен игры, спортивные симуляторы и шутеры на платформах PS4, Xbox One и PC.