Table of contents

- Описание работы
- Импорт библиотек и данных
- Изучение структуры данных
- Подготовка данных
- Исследовательская часть
  - **Сколько игр выпускалось в разные годы?**
  - **Как менялись продажи по платформам во времени?**
  - **Выбираю актуальные данные для дальнейшего анализа**
  - **Какие платформы лидируют по продажам, растут или падают?**
  - **Глобальным продажи игр в разбивке по платформам**
  - **Взаимосвязь между оценками (игроки и критики) и общими продажами**
  - **Какие жанры игр продаются больше всех? Выделяются ли жанры с высокими и низкими продажами?**
- Портрет пользователя каждого региона
  - **Самые популярные платформы (топ-5)**
  - **Самые популярные жанры (топ-5)**
  - **Влияет ли рейтинг ESRB на продажи в отдельном регионе?**
- Проверка гипотез
  - **Средние пользовательские рейтинги платформ Xbox One и PC одинаковые?**
  - **Средние пользовательские рейтинги жанров Action и Sports разные?**

- Общий вывод

Описание работы

Цель работы состоит в анализе данных о продажах компьютерных игр для определения успешности их выпуска на различных платформах и в разных регионах.

Проблема заключается в том, что необходимо провести исследование и обработку большого объема данных о продажах игр, чтобы выявить закономерности и тенденции, влияющие на успешность игровых платформ и жанров. Это поможет сделать выводы о том, какие платформы и жанры наиболее популярны, а также понять, как отзывы пользователей и критиков влияют на продажи игр.

Данные включают в себя продажи игр, жанры, разбитие по регионам, и другую информацию о каждой игре. Датасет содержит следующие переменные:

- Название игры: название конкретной видеоигры.
- Платформа: игровая платформа, на которой была выпущена игра (например, PlayStation, Xbox, PC и т.д.).
- Год выпуска: год, когда игра была выпущена.
- Жанр: категория игры, определяющая ее основную тематику (например, экшн, стратегия, спорт и т.д.).
- Продажи в Северной Америке: общая сумма продаж игры в регионе Северная Америка (в миллионах долларов).
- Продажи в Европе: общая сумма продаж игры в регионе Европа (в миллионах долларов).
- Продажи в Японии: общая сумма продаж игры в регионе Япония (в миллионах долларов).
- Продажи в других регионах: общая сумма продаж игры в остальных регионах, не включенных в Северную Америку, Европу и Японию (в миллионах долларов).
- Оценка критиков: средняя оценка игры, выставленная критиками (от 0 до 100).
- Оценка пользователей: средняя оценка игры, выставленная пользователями (от 0 до 10).
- Рейтинг ESRB: рейтинг игры, присвоенный Entertainment Software Rating Board (ESRB). Рейтинг указывает на возрастные ограничения и содержательные особенности игры.

Ожидаемые результаты включают выявление наиболее успешных игровых платформ и жанров, а также оценку влияния отзывов критиков и пользователей на продажи игр. Результаты исследования помогут понять

предпочтения игроков и основные факторы, влияющие на коммерческий успех видеоигр.

План работы:

1) Импорт данных, Изучение структуры данных.

1. Осмотреть данные для дальнейшей подготовки

2) Подготовка данных

1. Помимо изменения названий столбцов на нижний регистр, также рекомендуется проверить наличие пробелов или других символов в названиях столбцов, чтобы облегчить доступ к данным при анализе.
2. При замене значений "tbd" в столбце User_Score на NaN, убедитесь, что столбец User_Score имеет числовой тип данных (float) после замены.
3. При замене значений в столбце Year_of_Release на int, проверьте, что не возникло ошибок или пропущенных значений после преобразования.

3) Исследовательская часть

1. Для определения актуального периода можно использовать анализ количества выпущенных игр по годам. Обратите внимание на тренды и периоды, когда количество игр сильно увеличивается или уменьшается.
2. При анализе платформ, выбранных с наибольшими суммарными продажами, рекомендуется также оценить их текущую активность на рынке. Учтите, что новые платформы могут появляться и старые платформы могут исчезать.
3. При анализе влияния отзывов пользователей и критиков на продажи, помимо диаграммы рассеяния и корреляции, можно использовать также дополнительные методы, например, разделение выборки на группы с высокими и низкими оценками для дополнительного сравнительного анализа.

4) Составьте портрет пользователя каждого региона

1. Для анализа различий в долях продаж популярных платформ и жанров в разных регионах, можно использовать не только числовые значения, но и визуализацию данных (например, круговые диаграммы или столбчатые диаграммы).
2. При анализе влияния рейтинга ESRB на продажи в отдельном регионе, помимо описания различий, можно использовать статистические методы (например, t-тест) для проверки гипотезы о наличии статистически значимой связи.

5) Проверьте гипотезы:

1. При формулировании нулевых и альтернативных гипотез, убедитесь, что они ясно отражают сравниваемые параметры и направление ожидаемого эффекта.
2. При выборе критерия для проверки гипотез, обратите внимание на условия применимости выбранного критерия и учтите особенности данных

(например, нормальность распределения).
3. Укажите пороговое значение alpha, которое вы выбрали для проверки гипотез, и обоснуйте свой выбор.

6) В общем выводе подведите итоги анализа и ответьте на поставленные в начале работы вопросы. Укажите основные закономерности и тенденции, выявленные в данных, и сделайте релевантные выводы и рекомендации на основе проведенного исследования.

Импорт библиотек и данных

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats as st
import numpy as np

data = pd.read_csv(r"C:\project_data\games\games.csv")
```

Изучение структуры данных

```
# Осмотр
display(data.head(3))
```

```
                Name Platform  Year_of_Release     Genre  NA_sales
EU_sales  \
0         Wii Sports      Wii           2006.0    Sports     41.36
28.96
1  Super Mario Bros.      NES           1985.0  Platform     29.08
3.58
2     Mario Kart Wii      Wii           2008.0    Racing     15.68
12.76

   JP_sales  Other_sales  Critic_Score User_Score Rating
0      3.77         8.45          76.0          8      E
1      6.81         0.77           NaN        NaN    NaN
2      3.79         3.29          82.0        8.3      E
```

```
# Пропуски
print(data.isna().sum())
```

```
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
```

```
# Явные дубликаты
print(data.duplicated().value_counts())
```

```
False    16715
dtype: int64
```

```
# Проверяю уникальные значения в столбцах Platform и Genre (для поиска неявных дубликатов)
print(data['Platform'].unique())
print(data['Genre'].unique())
```

```
['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']
['Sports' 'Platform' 'Racing' 'Role-Playing' 'Puzzle' 'Misc' 'Shooter'
 'Simulation' 'Action' 'Fighting' 'Adventure' 'Strategy' nan]
```

```
# Дубликаты в названиях игр
data['Name'] = data['Name'].str.lower()
print(data.duplicated().value_counts())
```

```
False    16715
dtype: int64
```

Подготовка данных

Названия столбцов нужно привести в нижний регистр

```
data.columns = data.columns.str.lower()
print(data.columns)
```

```
Index(['name', 'platform', 'year_of_release', 'genre', 'na_sales',
'eu_sales',
       'jp_sales', 'other_sales', 'critic_score', 'user_score',
'rating'],
      dtype='object')
```

User_Score надо заменть на float, встречается tbd "to be decided", заменю его на NaN. tbd меняю на отсутствие оценки, так как пока этой оценки нет

```
data['user_score'][data['user_score'] == 'tbd'] = 'nan'
data['user_score'] = data['user_score'].astype('float64')
```

```
C:\Users\Dmitriy\AppData\Local\Temp\ipykernel_7444\624197207.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  data['user_score'][data['user_score'] == 'tbd'] = 'nan'
```

Year_of_Release нужно заменить на int.

```
data['year_of_release'] = data['year_of_release'].astype('Int64')
```

Удалить строки с пропуском в имени и жанре

```
data.dropna(subset=['name', 'genre'], inplace=True)
```

Посчитать суммарные продажи в отдельный столбец

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

Финальная проверка после подготовки

```
display(data.head(5))
```

```
                      name platform  year_of_release         genre
na_sales  \
0                wii sports      Wii             2006        Sports
41.36
1         super mario bros.      NES             1985      Platform
29.08
2            mario kart wii      Wii             2008        Racing
15.68
3         wii sports resort      Wii             2009        Sports
15.61
4  pokemon red/pokemon blue       GB             1996  Role-Playing
11.27

   eu_sales  jp_sales  other_sales  critic_score  user_score rating
all_sales
0     28.96      3.77         8.45          76.0         8.0      E
82.54
1      3.58      6.81         0.77           NaN         NaN    NaN
40.24
2     12.76      3.79         3.29          82.0         8.3      E
35.52
3     10.93      3.28         2.95          80.0         8.0      E
32.77
4      8.89     10.22         1.00           NaN         NaN    NaN
31.38
```

Исследовательская часть

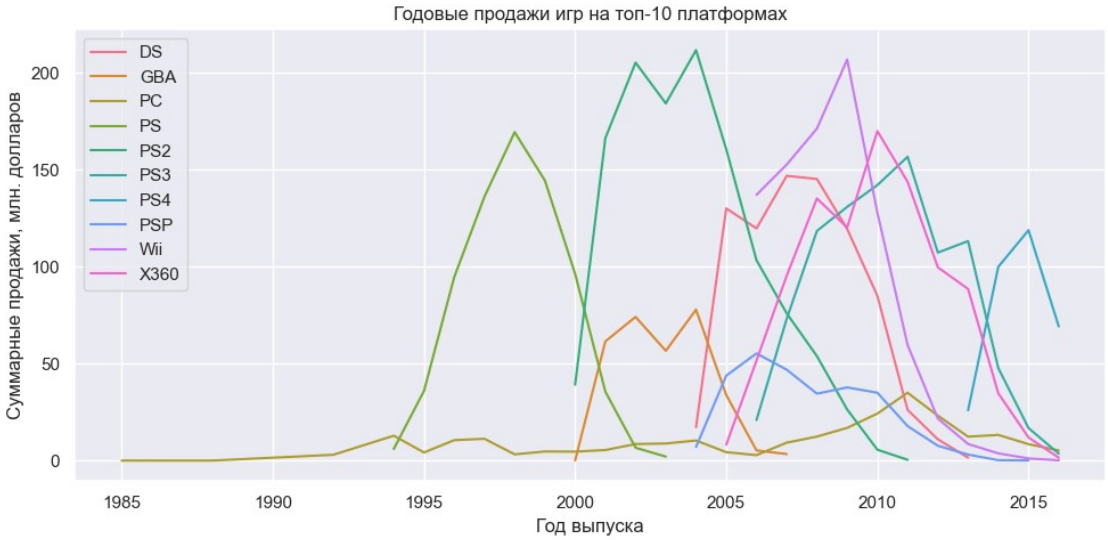
Сколько игр выпускалось в разные годы?

```
# Данные
pivot_table = data.pivot_table(
    index='year_of_release', values='name', aggfunc='count')
# График
sns.set(style='darkgrid')
plt.figure(figsize=(10, 5))
ax = sns.barplot(x=pivot_table.index, y='name',
                 data=pivot_table, color='steelblue')
plt.title('Количество выпущенных игр по годам')
plt.xlabel('Год выпуска')
plt.ylabel('Количество игр')
ax.set_xticklabels(ax.get_xticklabels(), rotation=90, ha='right')
plt.tight_layout()
plt.show()
```

Как менялись продажи по платформам во времени?

```
# Создаю список топ10-платформ
top_platforms_index = data.pivot_table(index='platform',
values='all_sales', aggfunc='sum').sort_values(
    by='all_sales', ascending=False).head(10).index

# Создание сводной таблицы суммарных продаж по годам и платформам с
фильтрацией датасета по топ10 платформам
top_platforms =
data[data['platform'].isin(top_platforms_index)].pivot_table(
    index='year_of_release', columns='platform', values='all_sales',
aggfunc='sum')

# Создание графика
```

```
sns.set(style='darkgrid')
sns.set_palette('husl', n_colors=len(top_platforms.columns))
plt.figure(figsize=(10, 5))
for platform in top_platforms.columns:
    plt.plot(top_platforms.index, top_platforms[platform],
label=platform)
plt.title('Годовые продажи игр на топ-10 платформах')
plt.legend(labels=top_platforms.columns)
plt.xlabel('Год выпуска')
plt.ylabel('Суммарные продажи, млн. долларов')
plt.tight_layout()
plt.show()
```



Анализирую время жизни платформы для поиска актуального периода

```
# Построение сводной таблицы с началом релиза игр и концом релизов
life_lenght = data.pivot_table(
    index='platform', values='year_of_release', aggfunc=('max',
'min'))

# Длительность релизов в годах
life_lenght['lenght'] = life_lenght['max'] - life_lenght['min']

# Среднее и медиана
print('Медианная продолжительность жизни платформы',
      life_lenght['lenght'].median())
print('Средняя продолжительность жизни платформы',
      round(life_lenght['lenght'].mean(), 1))

# Считаю время жизни платформ из топ10, расчёт такой же как и выше
life_lenght_top =
data[data['platform'].isin(top_platforms_index)].pivot_table(
    index='platform', values='year_of_release', aggfunc=('max',
```

```
'min'))
life_lenght_top['lenght'] = life_lenght_top['max'] -
life_lenght_top['min']
print('Медианная продолжительность жизни платформы из топ-10',
      life_lenght_top['lenght'].median())
print('Средняя продолжительность жизни платформы из топ-10',
      life_lenght_top['lenght'].mean())
```

```
Медианная продолжительность жизни платформы 6.0
Средняя продолжительность жизни платформы 7.6
Медианная продолжительность жизни платформы из топ-10 10.5
Средняя продолжительность жизни платформы из топ-10 13.1
```

Вывод

- Время жизни платформы различно и составляет в среднем 7-13 лет, в медианных значениях 6-10 лет
- Время жизни зависит от успешности платформы в суммарных продажах
- Паттерн продаж в топ-10 схож среди платформ

Выбираю актуальные данные для дальнейшего анализа

Актуальными данными считаю попадающими в период с 2013 год по н.в. (т.е. за три предыдущих года), исходя из времени жизни платформы и динамики общих продаж игр. Рост продаж и падение продаж происходят очень быстро, и актуальность платформы меняется буквально за пару-тройку лет

```
# Новая переменная с актуальными данными
actual_platforms = ['3DS', 'PC', 'PS3', 'PS4', 'PSV', 'WiiU', 'XOne']
data_actual = data[(data['year_of_release'] >= 2013) &
                   (data['platform'].isin(actual_platforms))]
# display(data_actual)
```

Какие платформы лидируют по продажам, растут или падают?

```
# Сумма продаж за актуальный период
print('Сумма продаж за актуальный период')
display(data_actual.pivot_table(index='platform', values='all_sales',
        aggfunc='sum').sort_values(by='all_sales', ascending=False))

# Сводная таблица по продажам по годам/платформе за актуальный период
print('Сводная таблица по продажам по годам/платформе за актуальный
период')
top_platforms_actual = data_actual.pivot_table(
    index='year_of_release', columns='platform', values='all_sales',
aggfunc='sum')
display(top_platforms_actual)
```

```
Сумма продаж за актуальный период
```

```
          all_sales
platform
PS4          314.14
PS3          181.43
XOne         159.32
3DS          143.25
WiiU          64.63
PC            39.43
PSV           32.99
```

```
Сводная таблица по продажам по годам/платформе за актуальный период
```

platform	3DS	PC	PS3	PS4	PSV	WiiU	XOne
year_of_release							
2013	56.57	12.38	113.25	25.99	10.59	21.65	18.96
2014	43.76	13.28	47.76	100.00	11.90	22.03	54.07
2015	27.78	8.52	16.82	118.90	6.25	16.35	60.14
2016	15.14	5.25	3.60	69.25	4.25	4.60	26.15

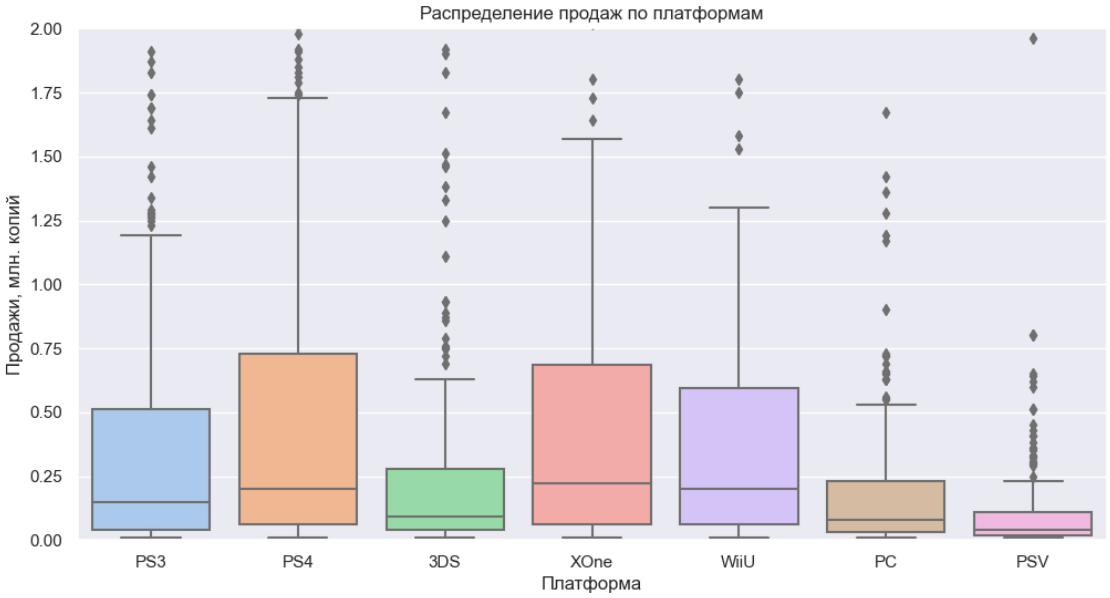
Вывод

Самая актуальная и прибыльная (только по сумме продаж, цена и прибыль с продаж в датасете не указана) на 2016 год является **PS4**. С большим отрывом за ней идёт **XOne и 3DS**. Все остальные платформы сильно уступают по сумме продаж.

Общее количество продаж для каждой платформы падает. В 2015 году был рост продаж. Возможно данные за 2016 год не полные, поэтому видим спад.

Глобальным продажи игр в разбивке по платформам

```
plt.figure(figsize=(12, 6))
plt.ylim(0, 2)
sns.set_palette('pastel')
sns.boxplot(data=data_actual, x='platform', y='all_sales')
plt.title('Распределение продаж по платформам')
plt.xlabel('Платформа')
plt.ylabel('Продажи, млн. копий')
plt.show()
```



Вывод

Для актуальных платформ и уже утрачивающих актуальность сходная картина по продажам - 75% игр едва ли дотягивают до 750 тыс копий. Для 3DS, PC, PSV 75% игр продано в тираже до 250 тысяч копий.

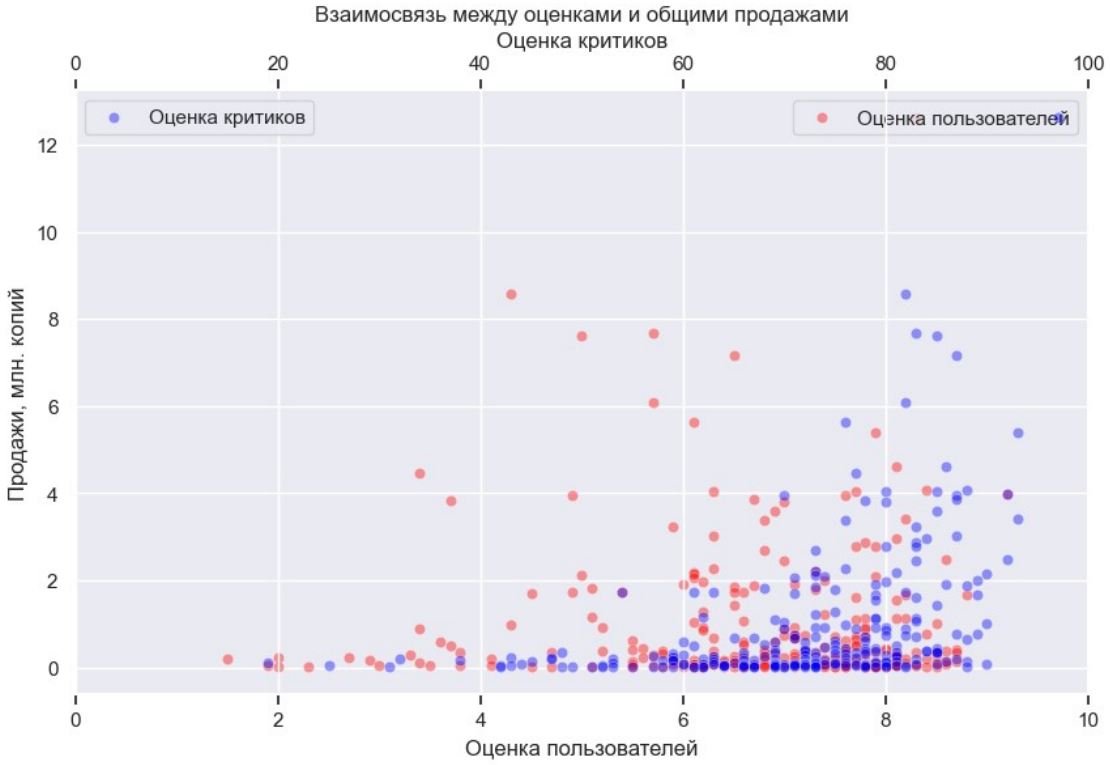
Медиана (очень усредненно) для многих платформ находится в районе 200 тысяч копий проданных игр

Однако для каждой актуальной платформы есть немного крайне успешных игр, проданных от миллиона копий и до 20+ миллионов.

Взаимосвязь между оценками (игроки и критики) и общими продажами

```
# Создание отдельных осей для каждого графика
fig, ax1 = plt.subplots(figsize=(10, 6))
ax2 = ax1.twiny()

# График для оценок пользователей
sns.scatterplot(data=data_actual.loc[data['platform'] == 'PS4'],
x='user_score', y='all_sales',
                alpha=0.4, ax=ax1, color='red', label='Оценка
пользователей')

# График для оценок критиков
sns.scatterplot(data=data_actual.loc[data['platform'] == 'PS4'],
x='critic_score',
                y='all_sales', alpha=0.4, ax=ax2, color='blue',
label='Оценка критиков')

ax1.set_title(
    'Взаимосвязь между оценками и общими продажами')
```

```
ax1.set_xlabel('Оценка пользователей')
ax2.set_xlabel('Оценка критиков')
ax1.set_ylabel('Продажи, млн. копий')
ax1.set_xlim(0, 10)
ax2.set_xlim(0, 100)
handles, labels = ax1.get_legend_handles_labels()
ax1.legend(handles, labels, loc='upper right')
plt.show()
```




```
# Корреляция между оценками пользователей и общими продажами
user_score_corr = data_actual.loc[data['platform'] == 'PS4']
['all_sales'].corr(
    data_actual.loc[data['platform'] == 'PS4']['user_score'])
print('Корреляция между оценками пользователей и общими продажами:',
user_score_corr)

# Корреляция между оценками критиков и общими продажами
critic_score_corr = data_actual.loc[data['platform'] == 'PS4']
['all_sales'].corr(
    data_actual.loc[data['platform'] == 'PS4']['critic_score'])
print('Корреляция между оценками критиков и общими продажами:',
critic_score_corr)
```

```
Корреляция между оценками пользователей и общими продажами: -
0.03195711020455643
Корреляция между оценками критиков и общими продажами:
0.40656790206178123
```

Вывод

Корреляции между оценками пользователей и общими продажами для платформы PS4 нет. Существуте слабая корреляция между оценками критиков и общими продажами для платформы PS4. Оценка критиков более подходящая для оценки продаж.

Соотнесите выводы с продажами игр на других платформах.

```
# Расчёт корреляции для продаж/оценок на других платформах

for i in data_actual['platform'].unique():
    corr = data_actual[data['platform'] == i]['all_sales'].corr(
        data_actual[data['platform'] == i]['user_score'])
    print(
        f'Корреляция между оценками пользователей и общими продажами
для платформы {i} {corr:.1f}')

for i in data_actual['platform'].unique():
    corr = data_actual[data['platform'] == i]['all_sales'].corr(
        data_actual[data['platform'] == i]['critic_score'])
    print(
        f'Корреляция между оценками критиков и общими продажами для
платформы {i} {corr:.1f}')
```

```
Корреляция между оценками пользователей и общими продажами для
платформы PS3 0.0
Корреляция между оценками пользователей и общими продажами для
платформы PS4 -0.0
Корреляция между оценками пользователей и общими продажами для
платформы 3DS 0.2
Корреляция между оценками пользователей и общими продажами для
платформы XOne -0.1
Корреляция между оценками пользователей и общими продажами для
платформы WiiU 0.4
Корреляция между оценками пользователей и общими продажами для
платформы PC -0.1
Корреляция между оценками пользователей и общими продажами для
платформы PSV 0.0
Корреляция между оценками критиков и общими продажами для платформы
PS3 0.3
Корреляция между оценками критиков и общими продажами для платформы
PS4 0.4

C:\Users\Dmitriy\AppData\Local\Temp\ipykernel_7444\1149844352.py:4:
UserWarning: Boolean Series key will be reindexed to match DataFrame
index.
  corr = data_actual[data['platform'] == i]['all_sales'].corr(
C:\Users\Dmitriy\AppData\Local\Temp\ipykernel_7444\1149844352.py:5:
UserWarning: Boolean Series key will be reindexed to match DataFrame
index.
```

```
  data_actual[data['platform'] == i]['user_score'])
C:\Users\Dmitriy\AppData\Local\Temp\ipykernel_7444\1149844352.py:10:
UserWarning: Boolean Series key will be reindexed to match DataFrame
index.
  corr = data_actual[data['platform'] == i]['all_sales'].corr(
C:\Users\Dmitriy\AppData\Local\Temp\ipykernel_7444\1149844352.py:11:
UserWarning: Boolean Series key will be reindexed to match DataFrame
index.
  data_actual[data['platform'] == i]['critic_score'])

Корреляция между оценками критиков и общими продажами для платформы
3DS 0.4
Корреляция между оценками критиков и общими продажами для платформы
XOne 0.4
Корреляция между оценками критиков и общими продажами для платформы
WiiU 0.4
Корреляция между оценками критиков и общими продажами для платформы PC
0.2
Корреляция между оценками критиков и общими продажами для платформы
PSV 0.3
```

Вывод

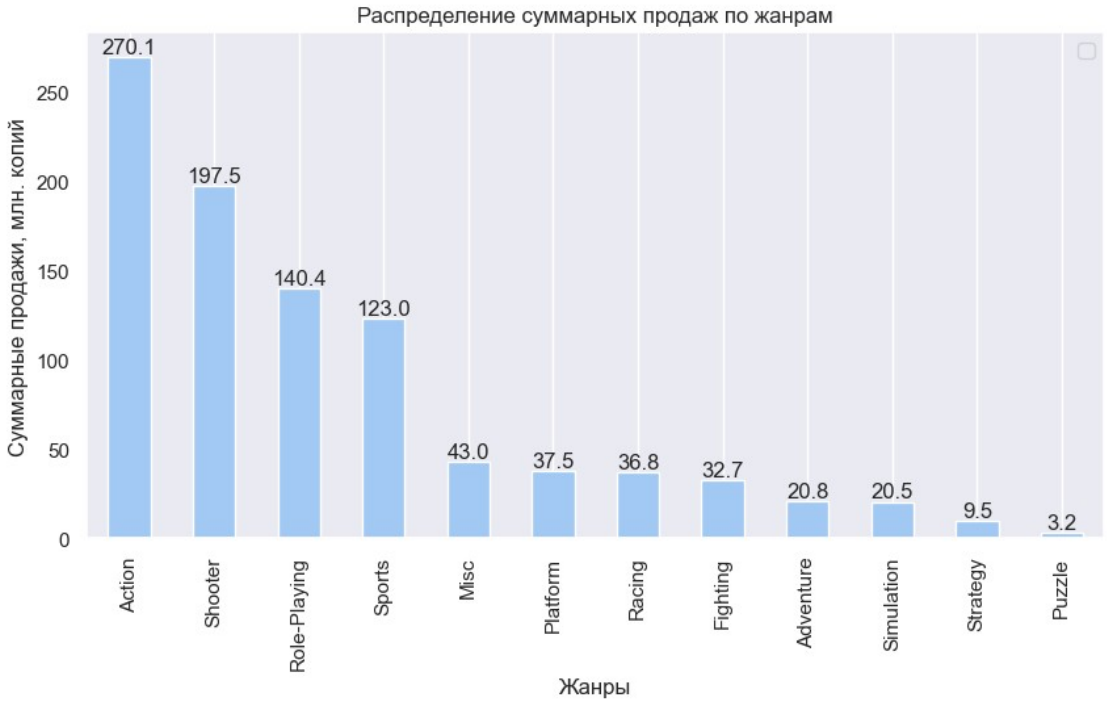
В целом данные для корреляции по всем актуальным платформам соотносятся с оценкой пользователей/продажами, кроме платформы WiiU.

Для каждой платформы существует слабая корреляция между оценками критиков и продажами. Для некоторых платформ она выше, для некоторых ниже. Самая слабая корреляция для оценок игр на PC.

Какие жанры игр продаются больше всех? Выделяются ли жанры с высокими и низкими продажами?

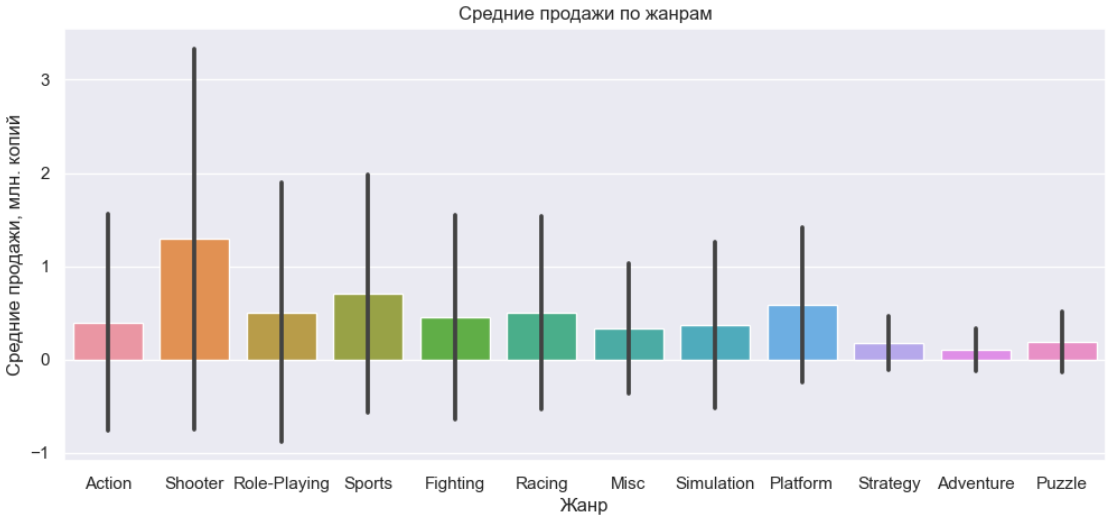
```
# Создание сводной таблицы суммарных продаж по жанрам
genre_sales = data_actual.pivot_table(
    index='genre', values='all_sales', aggfunc='sum')
# Сортировка по убыванию
genre_sales_sorted = genre_sales.sort_values(by='all_sales',
ascending=False)

# Создание графика
fig, ax = plt.subplots(figsize=(10, 5))
genre_sales_sorted.plot(kind='bar', ax=ax)

# Добавление подписей значений на столбцах
for i, value in enumerate(genre_sales_sorted['all_sales']):
    ax.text(i, value, f'{value:.1f}', ha='center', va='bottom')

# Настройка оформления графика
ax.set_title(
    'Распределение суммарных продаж по жанрам')
```

```
ax.set_xlabel('Жанры')
ax.set_ylabel('Суммарные продажи, млн. копий')
ax.legend('')
ax.grid(axis='y')

plt.show()
```




```
# График средних продаж с ошибками
fig, ax = plt.subplots(figsize=(12, 5))
sns.set(style='darkgrid')
sns.barplot(data=data_actual, x='genre', y='all_sales', errorbar='sd',
ax=ax)
ax.set_title('Средние продажи по жанрам')
ax.set_xlabel('Жанр')
ax.set_ylabel('Средние продажи, млн. копий')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Вывод

Сказать о прибыльность сложно, так как не указана цена игр и цена разработки/маркетинга и прочего. Самые частопродающиеся жанры - Action, Shooter, Role-Playing, Sports. Самые малопродаваемые - Strategy, Puzzle.

В среднем по продажам игр на каждый жанр, наиболее продающиеся: Shooter (в среднем 1.29 млн копий), но с высоким стандартным отклонением; Sports (в среднем 0.7 млн копий), но с высоким стандартным отклонением.

В целом для каждого жанра характерно высокое стандартное отклонение, прогнозировать продажи по жанру - гадание

Портрет пользователя каждого региона

Определите для пользователя каждого региона (NA, EU, JP)

Самые популярные платформы (топ-5)

```
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))

region_title = ['Доля продаж в северной Америке',
                'Доля продаж в Европе', 'Доля продаж в Японии']

# Построение графиков для каждого региона
for i, region in enumerate(['na_sales', 'eu_sales', 'jp_sales']):
    # Получение данных по продажам и платформам
    sales_by_platform = (data_actual
                         .pivot_table(index='platform', values=region,
aggfunc='sum')
                         .sort_values(by=region, ascending=False)
                         .head(5))
```

```
    ax = axes[i]
    ax.pie(sales_by_platform[region],
           labels=sales_by_platform.index, autopct='%1.1f%%')
    ax.set_title(region_title[i])

plt.tight_layout()
plt.show()
```

Вывод

Продажи на платформах сильно зависят от региона. В целом регионы EU и NA схожи по продажам по платформам.

1. Для североамериаканского рынка наиболее востребованы (за четыре прошедших года) **PS4, XOne (занимают больше половины рынка)**.
2. Для европейского рынка наиболее востребованы (за четыре прошедших года) **PS4 (почти половина рынка), на втором месте PS3**.
3. Для японского рынка наиболее востребованы (за четыре прошедших года) **3DS (половина рынка), остальные платформы в равной степени делят рынок**.

Самые популярные жанры (топ-5)

```
# Строю диаграммы pie для каждого региона по продажам по жанрам
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))

region_title = ['Доля продаж в северной Америке',
                'Доля продаж в Европе', 'Доля продаж в Японии']

# Построение графиков для каждого региона
for i, region in enumerate(['na_sales', 'eu_sales', 'jp_sales']):
    # Получение данных по продажам и платформам
    sales_by_platform = (data_actual
                         .pivot_table(index='genre', values=region,
aggfunc='sum')
                         .sort_values(by=region, ascending=False)
```

```
                            .head(5))
    ax = axes[i]
    ax.pie(sales_by_platform[region],
           labels=sales_by_platform.index, autopct='%1.1f%%')
    ax.set_title(region_title[i])

plt.tight_layout()
plt.show()
```

Вывод

Продажи игр различных жанров сильно зависят от региона. Однако регионы EU и NA очень схожи.

1. Для североамериаканского рынка наиболее востребованы жанры (за четыре прошедших года) **Action, shooter, sports**.
2. Для европейского рынка наиболее востребованы жанры (за четыре прошедших года) **Action, shooter, sports**.
3. Для японского рынка наиболее востребованы жанры (за четыре прошедших года) **Action и Role-playing**.

Влияет ли рейтинг ESRB на продажи в отдельном регионе?

```
# Строю диаграммы pie для каждого региона по продажам по рейтингу
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))
region_title = ['Доля продаж в северной Америке',
                'Доля продаж в Европе', 'Доля продаж в Японии']

# Построение графиков для каждого региона
for i, region in enumerate(['na_sales', 'eu_sales', 'jp_sales']):
    # Получение данных по продажам и платформам
    sales_by_platform = (data_actual
                            .pivot_table(index='rating', values=region,
aggfunc='sum')
                            .sort_values(by=region, ascending=False)
                            .head(5))
    ax = axes[i]
    ax.pie(sales_by_platform[region],
           labels=sales_by_platform.index, autopct='%1.1f%%')
    ax.set_title(region_title[i])
```

```
plt.tight_layout()
plt.show()
```

Вывод

Основную долю рынка EU и NA занимают игры с рейтингом M. На втором месте игры с рейтингом E. Примерно четверть рынка занимают игры с рейтингом T и E10+.

Рынок видеоигр Японии отличается. В Основном продаются игры рейтинга T и E, игры рейтинга M занимают только четверть рынка.

Проверка гипотез

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

Средние пользовательские рейтинги платформ Xbox One и PC одинаковые?

```
# Задаю значения для выборок
sample_1 = data_actual[data_actual['platform']
                       == 'XOne'].dropna(subset=['user_score'])
sample_2 = data_actual[data_actual['platform']
                       == 'PC'].dropna(subset=['user_score'])

# Уровень значимости
alpha = 0.05

# T-test для двух независимых выборок
results = st.ttest_ind(sample_1['user_score'], sample_2['user_score'])
# print(results)
print('p-значение: ', results.pvalue)

# Вывод результата
```

```
if results.pvalue < alpha:
    print('Отвергаем нулевую теорию')
else:
    print('Не отвергаем нулевую теорию')
```

```
p-значение:  0.14012658403611647
Не отвергаем нулевую теорию
```

Вывод

- Нулевая гипотеза - Средние пользовательские рейтинги платформ Xbox One и PC одинаковые
- Альтернативная - Средние пользовательские рейтинги платформ Xbox One и PC не одинаковые
- Метод проверки - t-test для двух независимых выборок. Критический уровень статистической значимости - 0.05.

Результат: Средние пользовательские рейтинги платформ Xbox One и PC одинаковые, значимого различия нет.

Средние пользовательские рейтинги жанров Action и Sports разные?

```
# Задаю значения для выборок
sample_1 = data_actual[data_actual['genre']
                       == 'Action'].dropna(subset=['user_score'])
sample_2 = data_actual[data_actual['genre']
                       == 'Sports'].dropna(subset=['user_score'])

# Уровень значимости
alpha = 0.05

# T-test для двух независимых выборок
results = st.ttest_ind(sample_1['user_score'], sample_2['user_score'])
print(results)
print('p-значение: ', results.pvalue)

# Вывод результата
if results.pvalue < alpha:
    print('Отвергаем нулевую теорию')
else:
    print('Не отвергаем нулевую теорию')
```

```
Ttest_indResult(statistic=9.773362788475188,
pvalue=1.1721377725302828e-20)
p-значение:  1.1721377725302828e-20
Отвергаем нулевую теорию
```

Вывод

- Нулевая гипотеза - средние пользовательские рейтинги жанров Action и Sports одинаковые

- Альтернативная - средние пользовательские рейтинги жанров Action и Sports не одинаковые
- Метод проверки - t-test для двух независимых выборок. Критический уровень статистической значимости - 0.05.

Результат: средние пользовательские рейтинги жанров Action и Sports не одинаковые, значимое различие есть.

Общий вывод

1. Рынок видоигр очень динамичный, платформы приходят и уходят, срок жизни около 6 лет, но первый и последний год из них это резкий подьем и резкое падение продаж. Стабильно платформы держаться около 4 лет. Поэтому при создании игр очень важно искать актуальную сейчас и в будущем платформу.
2. Регионы EU и NA похожи по потреблению игр, однако JP регион сильно другой, предпочитая дургие платформы, игры, жанры и аудитория (судя по рейтингу) другая, более молодая вероятно.
3. Жанры игр сильно отличаются по продажам. Action/Shooter на первом месте по сумме продаж. Паззлы и стратегии продаются на порядки меньше.
4. Какая платформа лучше по мнению пользователей, Xbox или PS? Статистически достовернного отличия нет. Хотя игры на PS продаются лучше чем на XBox.
5. Про прибыльность игр/платформы/жанров невозможно ничего сказать не зная затраты на разработку/маркетинг. Хотя есть множество игр которые продались миллионами копий, но были ли они прибыльными? Так же невозможно оценить прибыль от платформы. Однако если взять только общие продажи, то выпуск игры на PS будет более востребованным.
6. Жанры отличаются друг от друга не только количеством продаж, но и средней оценкой.
7. Очень интересная корреляция между рейтингом от пользователей и общими продажами. Её нет. Значит этот рейтинг для прогнозирования продаж не так и важен, в отличии от рейтинга критиков.