

---

---

# Анализ продаж Видеоигр PS4

Артем Резников • 25.10.2023

---



## Информация о датасете:

[Video Games Sales Dataset](#)

### Задача анализа

- Факторы, влияющие на региональные объемы продаж видеоигр для платформы PS4

### Вопросы

- ТОП -10 объемы продаж?
  - Какие самые продаваемые жанры?
  - Влияние возрастного рейтинга?
  - Насколько важно происхождение компании-разработчика?
-

# Атрибуты

---

# Предобработка данных

## Загрузка в Jupiter Notebook

```
import pandas as pd
import numpy as np
import sqlite3
```

+ Код

+ Текст



```
# Загрузка данных
df = pd.read_csv('Video_Games_Sales_as_at_22_Dec_2016.csv', sep=',')
```

```
[ ] df.head()
```

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	

# Предобработка данных

## Проверка дубликатов

```
df.shape

(16719, 16)

[ ] #Проверка дубликатов в данных
df[df.duplicated()]

Name Platform Year_of_Release Genre Publisher NA_Sales EU_Sales JP_Sales Other_Sales Global_Sales Critic_Score
```

## Проверка типов данных

```
df.dtypes

Name object
Platform object
Year_of_Release float64
Genre object
Publisher object
NA_Sales float64
EU_Sales float64
JP_Sales float64
Other_Sales float64
Global_Sales float64
Critic_Score float64
Critic_Count float64
User_Score object
User_Count float64
Developer object
Rating object
dtype: object
```

Вносим изменения в Год выпуска, для правильного отображения даты

Прежде, чем переводить float к int, удаляем NaN значения

```
# Заменяем пропущенные значения в столбце Year_of_Release на ноль
df['Year_of_Release'] = df['Year_of_Release'].fillna(0).astype(int)

# Затем приводим float к int для правильного отображения даты (год)
df['Year_of_Release'] = df['Year_of_Release'].astype(int)
```

# Предобработка данных

## Проверка количества значений в колонках для дальнейшей обработки

df.count()

Name	16717
Platform	16719
Year_of_Release	16450
Genre	16717
Publisher	16665
NA_Sales	16719
EU_Sales	16719
JP_Sales	16719
Other_Sales	16719
Global_Sales	16719
Critic_Score	8137
Critic_Count	8137
User_Score	10015
User_Count	7590
Developer	10096
Rating	9950
dtype:	int64

Year\_of\_Release

2006.0

1985.0

2008.0

2009.0

1996.0

User\_Score

8

NaN

8.3

8

NaN

Critic\_Score

76.0

NaN

82.0

80.0

NaN

# Предобработка данных

## Конвертация значений к удобному формату, удаление бесполезных кортежей

```
[ ] # Меняем str значения на NaN
    # Удаляем строки, при условии, что у нас нет информации об оценках от критиков и юзеров.
    df['User_Score'] = df['User_Score'].replace('tbd', np.nan)
    df = df.dropna(subset=['Critic_Score', 'User_Score'], how='all')

[ ] # Конвертируем Оценки юзеров в 100-бальную систему оценивания.
    df['User_Score'] = df['User_Score'].astype(float).round(1)
    df['User_Score'] = df['User_Score'] * 10

▶ # Конвертируем значения в евро
    columns_to_convert = ['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales']

    for column in columns_to_convert:
        df[column] = df[column].apply(lambda x: round(float(x) / 1.06, 2))
```

# Предобработка данных

## Результат

Мы провели конвертацию валют по текущему курсу в евро (предыдущие данные были в USD).

Год выпуска без плавающей точки.

Оценки критиков и юзеров в одном формате.

Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
2006	Sports	Nintendo	39.02	27.32	3.56
2008	Racing	Nintendo	14.79	12.04	3.58
2009	Sports	Nintendo	14.73	10.31	3.09
2006	Platform	Nintendo	10.64	8.62	6.13
2006	Misc	Nintendo	13.17	8.66	2.76

Critic_Score	Critic_Count	User_Score
76.0	51.0	80.0
82.0	73.0	83.0
80.0	73.0	80.0
89.0	65.0	85.0
58.0	41.0	66.0



# Предобработка данных

## Загрузка в обновленной таблицы базу данных SQL

```
[ ] con = sqlite3.connect('Video_Games_Sales.db', timeout=10)
    cur = con.cursor
```

```
[ ] df.to_sql(con=con, name='Video_Games_Sales.db', index=False)
```

# Анализ данных в SQL

## Региональные продажи ТОП 10

```
WITH RegionalRanks AS (
  SELECT
    Name,
    NA_Sales,
    Genre,
    ROW_NUMBER() OVER (ORDER BY NA_Sales DESC) AS NA_Rank,
    EU_Sales,
    ROW_NUMBER() OVER (ORDER BY EU_Sales DESC) AS EU_Rank,
    JP_Sales,
    ROW_NUMBER() OVER (ORDER BY JP_Sales DESC) AS JP_Rank
  FROM "Video_Games_Sales.db" vgsd
  WHERE Platform IN ('PS4')
)
SELECT
  NA.Name AS NA_Title,
  NA.Genre AS NA_Genre,
  NA.NA_Sales AS NA_Sales,
  EU.Name AS EU_Title,
  EU.Genre AS EU_Genre,
  EU.EU_Sales AS EU_Sales,
  JP.Name AS JP_Title,
  JP.Genre AS JP_Genre,
  JP.JP_Sales AS JP_Sales
FROM RegionalRanks NA
JOIN RegionalRanks EU ON NA.NA_Rank = EU.EU_Rank
JOIN RegionalRanks JP ON NA.NA_Rank = JP.JP_Rank
ORDER BY NA.NA_Rank
LIMIT 10;
```

ABC NA_Title	ABC NA_Genre	123 NA_Sales
Grand Theft Auto V	Action	3.74
Call of Duty: Advanced Warfare	Shooter	2.65
Fallout 4	Role-Playing	2.39
Destiny	Shooter	2.35
NBA 2K16	Sports	2.35
Madden NFL 16	Sports	2.21
Uncharted: The Nathan Drake Collection	Action	1.95
Uncharted 4: A Thief's End	Shooter	1.75
Call of Duty: Ghosts	Shooter	1.68
Call of Duty: Infinite Warfare	Shooter	1.52
ABC JP_Title	ABC JP_Genre	123 JP_Sales
Metal Gear Solid V: The Phantom Pain	Action	0.46
Knack	Platform	0.4
Grand Theft Auto V	Action	0.36
Dark Souls III	Role-Playing	0.32
Bloodborne	Action	0.25
Fallout 4	Role-Playing	0.23
The Witcher 3: Wild Hunt	Role-Playing	0.22
Battlefield 1	Shooter	0.2
Uncharted 4: A Thief's End	Shooter	0.18
Battlefield 4	Shooter	0.16
ABC EU_Title	ABC EU_Genre	123 EU_Sales
Grand Theft Auto V	Action	5.95
FIFA 16	Sports	5.77
FIFA 17	Sports	5.42
FIFA 15	Sports	4.08
Call of Duty: Advanced Warfare	Shooter	3.28
Fallout 4	Role-Playing	3.08
Uncharted 4: A Thief's End	Shooter	2.36
Far Cry 4	Shooter	2.06
Battlefield 1	Shooter	2.03
The Witcher 3: Wild Hunt	Role-Playing	2.01

# Анализ данных в SQL

## Региональные продажи ТОП 10

```
WITH RegionalRanks AS (  
  SELECT  
    Name,  
    NA_Sales,  
    Genre,  
    ROW_NUMBER() OVER (ORDER BY NA_Sales DESC) AS NA_Rank,  
    EU_Sales,  
    ROW_NUMBER() OVER (ORDER BY EU_Sales DESC) AS EU_Rank,  
    JP_Sales,  
    ROW_NUMBER() OVER (ORDER BY JP_Sales DESC) AS JP_Rank  
  FROM "Video_Games_Sales.db" vgsd  
  WHERE Platform IN ('PS4')  
)  
SELECT  
  NA.Name AS NA_Title,  
  NA.Genre AS NA_Genre,  
  NA.NA_Sales AS NA_Sales,  
  EU.Name AS EU_Title,  
  EU.Genre AS EU_Genre,  
  EU.EU_Sales AS EU_Sales,  
  JP.Name AS JP_Title,  
  JP.Genre AS JP_Genre,  
  JP.JP_Sales AS JP_Sales  
FROM RegionalRanks NA  
JOIN RegionalRanks EU ON NA.NA_Rank = EU.EU_Rank  
JOIN RegionalRanks JP ON NA.NA_Rank = JP.JP_Rank  
ORDER BY NA.NA_Rank  
LIMIT 10;
```

RBC NA_Title	RBC NA_Genre	123 NA_Sales
Grand Theft Auto V	Action	3.74
Call of Duty: Advanced Warfare	Shooter	2.65
Fallout 4	Role-Playing	2.39
Destiny	Shooter	2.35
NBA 2K16	Sports	2.35
Madden NFL 16	Sports	2.21
Uncharted: The Nathan Drake Collection	Action	1.95
Uncharted 4: A Thief's End	Shooter	1.75
Call of Duty: Ghosts	Shooter	1.68
Call of Duty: Infinite Warfare	Shooter	1.52
RBC JP_Title	RBC JP_Genre	123 JP_Sales
Metal Gear Solid V: The Phantom Pain	Action	0.46
Knack	Platform	0.4
Grand Theft Auto V	Action	0.36
Dark Souls III	Role-Playing	0.32
Bloodborne	Action	0.25
Fallout 4	Role-Playing	0.23
The Witcher 3: Wild Hunt	Role-Playing	0.22
Battlefield 1	Shooter	0.2
Uncharted 4: A Thief's End	Shooter	0.18
Battlefield 4	Shooter	0.16
RBC EU_Title	RBC EU_Genre	123 EU_Sales
Grand Theft Auto V	Action	5.95
FIFA 16	Sports	5.77
FIFA 17	Sports	5.42
FIFA 15	Sports	4.08
Call of Duty: Advanced Warfare	Shooter	3.28
Fallout 4	Role-Playing	3.08
Uncharted 4: A Thief's End	Shooter	2.36
Far Cry 4	Shooter	2.06
Battlefield 1	Shooter	2.03
The Witcher 3: Wild Hunt	Role-Playing	2.01

## Продажи ТОП 10 по регионам

- Во всех регионах предпочтения по Action и Shooter похожи (раскрученные бренды, AAA игры)
- Следует обратить внимание, что спортивные игры отличаются в топе:  
 NA - регби и баскетбол  
 EU - футбол

---

# Анализ данных в SQL

---

## Самые продаваемые жанры

```
SELECT
    Genre,
    SUM(NA_Sales) AS Total_NA_Sales,
    SUM(EU_Sales) AS Total_EU_Sales,
    SUM(JP_Sales) AS Total_JP_Sales
FROM "Video_Games_Sales.db" vgsd
WHERE Platform = 'PS4'
GROUP BY Genre
ORDER BY SUM(NA_Sales) DESC;
```

---

RBC Genre ▼	123 Total_NA_Sales ▼	123 Total_EU_Sales ▼	123 Total_JP_Sales ▼
Action	25.01	33.79	3.28
Shooter	21.69	27.35	2.1
Sports	18.64	24.41	0.78
Role-Playing	5.73	7.55	1.52
Fighting	2.69	2.25	0.52
Racing	1.95	5.29	0.2
Platform	1.85	2.68	0.43
Misc	1.2	1.08	0.04
Adventure	1	1.53	0.14
Strategy	0.16	0.21	0.02
Simulation	0.11	0.44	0.06
Puzzle	0	0.02	0

## Самые продаваемые жанры

- Самые популярные жанры это Action, Shooter и Sports. В регионах NA, EU они зарабатывают больше всего денег. В Японии, на третьем месте - RPG.

---

## Анализ данных в SQL

---

Влияние оценок критиков и  
юзеров на глобальные продажи

```
SELECT
    Genre,
    Global_Sales,
    ROUND(AVG(Critic_Score)) AS Average_Critic_Score,
    ROUND(AVG(User_Score)) AS Average_User_Score
FROM "Video_Games_Sales.db" vgsd
WHERE Platform = 'PS4'
GROUP BY Genre
ORDER BY Global_Sales DESC;
```

---

ABC Genre	123 Global_Sales	123 Average_Critic_Score	123 Average_User_Score
Action	11.9	72	69
Sports	8.08	68	58
Shooter	7.23	75	67
Role-Playing	6.75	73	75
Fighting	2.62	73	72
Racing	1.95	72	67
Platform	1.8	72	60
Adventure	1.57	73	77
Misc	0.81	71	63
Simulation	0.39	73	79
Strategy	0.2	70	63
Puzzle	0.03	88	84

## Влияние оценок критиков и юзеров на глобальные продажи

- Чаще всего оценки критиков выше, чем у остальных игроков. Так как оценки критиков появляются раньше, нежели других. Это может напрямую влиять на рост продаж.



# Анализ данных в SQL

---

Какие жанры самые  
продаваемые по годам

```
SELECT
    Year_of_Release,
    Genre,
    SUM(NA_Sales) AS Total_NA_Sales,
    SUM(EU_Sales) AS Total_EU_Sales,
    SUM(JP_Sales) AS Total_JP_Sales,
    Global_Sales
FROM "Video_Games_Sales.db" vgsd
WHERE Platform = 'PS4'
GROUP BY Year_of_Release
ORDER BY Year_of_Release;
```

---

## Какие жанры самые продаваемые по годам

- Виден рост интереса к спортивным играм.

123 Year_of_Release ▼	ABC Genre ▼	123 Total_NA_Sales ▼	123 Total_EU_Sales ▼	123 Total_JP_Sales ▼	123 Global_Sales ▼
2,013	Shooter	9.13	10.15	0.9	3.61
2,014	Action	26.23	36.12	1.96	11.9
2,015	Sports	27.21	33.96	3.76	8.08
2,016	Sports	17.46	26.37	2.47	7.16

---

# Анализ данных в SQL

## Влияние возрастного рейтинга

```
SELECT
    Rating,
    COUNT(Rating) AS Total_Games,
    ROUND(SUM(NA_Sales), 2) AS NA_Sales,
    ROUND(SUM(EU_Sales), 2) AS EU_Sales,
    ROUND(SUM(JP_Sales), 2) AS JP_Sales
FROM "Video_Games_Sales.db" vgsd
WHERE Platform = 'PS4'
GROUP BY Rating
ORDER BY NA_Sales DESC;
```

---

ABC Rating	123 Total_Games	123 NA_Sales	123 EU_Sales	123 JP_Sales
M	75	39.93	54.19	4.93
E	52	15.78	26.01	0.8
T	82	15.47	17.34	2.54
E10+	39	8.54	8.32	0.65

## Влияние возрастного рейтинга

- Лучше всего продаются игры с Взрослым возрастным рейтингом. Из-за того, что во многих странах регуляция продажи игр отсутствует - подростки в т.ч активно покупают игры M.

---

# Выводы

---

- 
- Рост интереса в спортивных играх.
  - При создании спортивного проекта следует учитывать интерес к видам спорта у региона основной аудитории.
  - Критики обладают тенденцией к завышению оценок и управлением общественным мнением.
  - М рейтинг не отпугивает, а наоборот привлекает игроков.
  - Наибольших продаж достигают AAA-игры.
  - Такие регионы, как NA и EU имеют схожие предпочтения, тогда как у JP в топ попадают отличающиеся жанры и игры. Это говорит о том, что стоит рассматривать Восточный рынок отдельно от Европейского и Американского.
-