описание проекта

в данном рассмотрим данные о продажах игр, оценки пользователей и экспертов, жанры и платформы.цель проекта: понять, как рациональней распределить рекламный бюджет.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from scipy import stats as st
import warnings
warnings.filterwarnings("ignore")
```

```
data = pd.read_csv('/datasets/games.csv')
```

```
data
```

	Name	Platform	Year_of_Release	Genre	NA_sales	EU_sales	JP_sales	Other_sales	Critic_Score	User_Score
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	8.45	76.0	8
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	0.77	NaN	NaN
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	3.29	82.0	8.3
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	2.95	80.0	8
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	11.27	8.89	10.22	1.00	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
16710	Samurai Warriors: Sanada Maru	PS3	2016.0	Action	0.00	0.00	0.01	0.00	NaN	NaN
16711	LMA Manager 2007	X360	2006.0	Sports	0.00	0.01	0.00	0.00	NaN	NaN

```
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
```

```
data.columns = [x.lower() for x in data.columns]
```

```
data['name'] = data['name'].str.lower()
```

```
data['genre'] = data['genre'].str.lower()
```

```
data['platform'] = data['platform'].str.lower()
```

```
temp = data.copy()
```

```
temp[temp[['name', 'platform','year_of_release']].duplicated(keep=False)]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-f3b3c717de58> in <cell line: 1>()
----> 1 temp = data.copy()
      2 temp[temp[['name', 'platform','year_of_release']].duplicated(keep=False)]

NameError: name 'data' is not defined
```

SEARCH STACK OVERFLOW

```
data = data.drop_duplicates(subset=['platform', 'name'])
```

```
data.duplicated().sum()
```

```
0
```

```
data = data.astype({'year_of_release':'Int64'})
```

в столбце Year_of_Release заменен тип данных на integer, так как год это целое число

```
data.isna().sum()
```

```
name                  1
platform              0
year_of_release     268
genre                 1
na_sales              0
eu_sales              0
jp_sales              0
other_sales           0
critic_score       8577
user_score         6700
rating             6765
dtype: int64
```

```
data.loc[data['user_score'] == 'tbd', 'user_score'] = None
```

```
data= data.astype({'user_score':'float64'})
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16710 entries, 0 to 16714
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   name             16709 non-null  object 
 1   platform         16710 non-null  object 
 2   year_of_release  16442 non-null  Int64  
 3   genre            16709 non-null  object 
 4   na_sales         16710 non-null  float64
 5   eu_sales         16710 non-null  float64
 6   jp_sales         16710 non-null  float64
 7   other_sales      16710 non-null  float64
 8   critic_score     8133 non-null   float64
 9   user_score       7586 non-null   float64
 10  rating           9945 non-null   object 
dtypes: Int64(1), float64(6), object(4)
memory usage: 1.5+ MB
```

```
data['common_sales'] = data.na_sales + data.eu_sales + data.jp_sales + data.other_sales
```

```
data.head()
```

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	ratin
0	wii sports	wii	2006	sports	41.36	28.96	3.77	8.45	76.0	8.0	
1	super mario bros.	nes	1985	platform	29.08	3.58	6.81	0.77	NaN	NaN	Na
2	mario kart wii	wii	2008	racing	15.68	12.76	3.79	3.29	82.0	8.3	
3	wii sports resort	wii	2009	sports	15.61	10.93	3.28	2.95	80.0	8.0	

```
temp = data.copy()
list_c = ['name', 'platform', 'year_of_release', 'genre', 'critic_score', 'user_score', 'rating']
print(temp.info())
for col_l in list_c:
  print('-'* 25)
  print(col_l, temp[col_l].sort_values().unique())
  print(col_l,': кол-во NaN',temp[col_l].isna().sum(),
        ', процент NaN', round(temp[col_l].isna().mean()*100,2),'%')
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16710 entries, 0 to 16714
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   name             16709 non-null  object 
 1   platform         16710 non-null  object 
 2   year_of_release  16442 non-null  Int64  
 3   genre            16709 non-null  object 
 4   na_sales         16710 non-null  float64
 5   eu_sales         16710 non-null  float64
 6   jp_sales         16710 non-null  float64
 7   other_sales      16710 non-null  float64
 8   critic_score     8133 non-null   float64
 9   user_score       7586 non-null   float64
 10  rating           9945 non-null   object 
 11  common_sales     16710 non-null  float64
dtypes: Int64(1), float64(7), object(4)
memory usage: 1.7+ MB
None
-------------------------
name [' beyblade burst' ' fire emblem fates' " frozen: olaf's quest" ...
 'zyuden sentai kyoryuger: game de gaburincho!!'
 '¡shin chan flipa en colores!' nan]
name : кол-во NaN 1 , процент NaN 0.01 %
-------------------------
platform ['2600' '3do' '3ds' 'dc' 'ds' 'gb' 'gba' 'gc' 'gen' 'gg' 'n64' 'nes' 'ng'
 'pc' 'pcfx' 'ps' 'ps2' 'ps3' 'ps4' 'psp' 'psv' 'sat' 'scd' 'snes' 'tg16'
 'wii' 'wiiu' 'ws' 'x360' 'xb' 'xone']
platform : кол-во NaN 0 , процент NaN 0.0 %
-------------------------
year_of_release <IntegerArray>
[1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992,
 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,
 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, <NA>]
Length: 38, dtype: Int64
year_of_release : кол-во NaN 268 , процент NaN 1.6 %
-------------------------
genre ['action' 'adventure' 'fighting' 'misc' 'platform' 'puzzle' 'racing'
 'role-playing' 'shooter' 'simulation' 'sports' 'strategy' nan]
genre : кол-во NaN 1 , процент NaN 0.01 %
-------------------------
critic_score [13. 17. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34.
 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52.
 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70.
 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88.
 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. nan]
critic_score : кол-во NaN 8577 , процент NaN 51.33 %
-------------------------
user_score [0.  0.2 0.3 0.5 0.6 0.7 0.9 1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.
 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.  3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8
 3.9 4.  4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.  5.1 5.2 5.3 5.4 5.5 5.6
 5.7 5.8 5.9 6.  6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7.  7.1 7.2 7.3 7.4
 7.5 7.6 7.7 7.8 7.9 8.  8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.8 8.9 9.  9.1 9.2
 9.3 9.4 9.5 9.6 9.7 nan]
user_score : кол-во NaN 9124 , процент NaN 54.6 %
-------------------------
rating ['AO' 'E' 'E10+' 'EC' 'K-A' 'M' 'RP' 'T' nan]
```

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

так как пропусков в столбце года образования около 1,5 процентов, то было принято решение удалить эти строки, так как они искажают результат некоторых исследований.Возможно, дубликаты появились из-за того, что продажи в одном из регионов выделилось в отдельную строку. В колонке rating значение tbd было заменено на Nan, так как это аналог пропуска

Шаг 3. Проведение исследовательского анализа данных

```
data.pivot_table(index='year_of_release', values='common_sales', aggfunc='sum').plot(kind='bar', figsize=(10,5))
plt.ylabel('продажи')
plt.xlabel('год выпуска игр')
plt.title('количество продаж за каждый год');
```


```
data.pivot_table(index='platform',values='common_sales').sort_values('common_sales',ascending=False).head(10)
```

platform	common_sales
gb	2.622990
nes	2.561735
gen	1.050000
snes	0.836987
ps4	0.801378
x360	0.779846
2600	0.745517
ps3	0.713663
wii	0.692986
n64	0.689905

```
data.loc[(data.platform == 'ps4')|\
         (data.platform == 'gen')|\
         (data.platform == 'x360')|\
         (data.platform == 'ps3')|\
         (data.platform == 'xone')|\
         (data.platform == 'wii')]\
    .pivot_table(index='year_of_release',values='common_sales', columns='platform')\
    .sort_values('year_of_release').plot(kind='bar')
plt.title('количество продаж за каждый год на наиболее популярных платформах')
plt.xlabel('год')
plt.ylabel('продажи')
;
```

```
''
```

популярность платформы живет около 5 лет, поэтому будем считать актуальным период с 2013 года

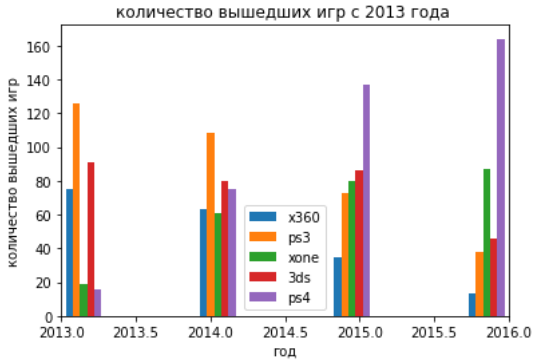
```
data_2013 = data[data['year_of_release'] >= 2013]
```

```
top10_platform = data_2013.pivot_table(index='platform', values='common_sales',aggfunc='sum').sort_values('common_sales', asce
plt.title('количество продаж на каждой платформе с 2013 года')
plt.xlabel('платформа')
plt.ylabel('количество продаж');
```

лидируют по продажам платформы xbox one, ps4 и игровая консоль 3 ds

```
x360 = data_2013.loc[(data.platform=='x360'), 'year_of_release']
ps3 = data_2013.loc[(data.platform=='ps3'), 'year_of_release']
wii = data_2013.loc[(data.platform=='xone'), 'year_of_release']
ds = data_2013.loc[(data.platform=='3ds'), 'year_of_release']
ps4 = data_2013.loc[(data.platform=='ps4'), 'year_of_release']
```

```
plt.hist((x360,ps3, wii, ds, ps4), label=['x360','ps3','xone','3ds','ps4'])
plt.xlim(2004)
plt.legend()
plt.title('количество вышедших игр с 2013 года ')
plt.xlabel('год')
plt.ylabel('количество вышедших игр')
plt.xlim(2013,2016)
plt.show()
```



платформа ps3 была наиболее папулярной в 2013 году, но затем количество вышедших игр снижается, после выхода ей на смену ps4 резко увеличивается количество игр на эту платформу

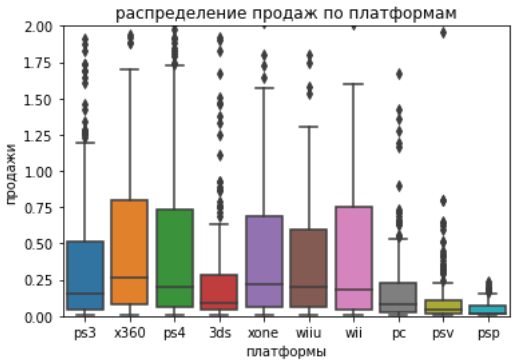
далее рассмотрим продажи игр на каждой платформе

```
boxplot_data = data_2013\
    .loc[(data.platform == '3ds')|\
         (data.platform == 'ps3')|\
         (data.platform == 'ps4')|\
         (data.platform == 'x360')|\
         (data.platform == 'xone')|\
         (data.platform == 'wiiu')|\
         (data.platform == 'pc')|\
         (data.platform == 'psv')|\
         (data.platform == 'psp')|\
         (data.platform == 'wii')]
```

```
boxplot_data.year_of_release.min()
```

```
2013
```

```
sns.boxplot(x='platform',y='common_sales',data=boxplot_data)
plt.ylim(0,2)
plt.title('распределение продаж по платформам')
plt.ylabel('продажи')
plt.xlabel('платформы');
```



самые высокие средние продажи на платформе ps4, самые низкие на 3 ds

выявление зависимостей отзывов на продажи

```
data_2013.corr()
```

	year_of_release	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	common_sales
year_of_release	1.000000	-0.124551	-0.087983	-0.074142	-0.090225	0.064322	0.039318	-0.117878
na_sales	-0.124551	1.000000	0.769995	0.264513	0.817367	0.301130	-0.020010	0.922617
eu_sales	-0.087983	0.769995	1.000000	0.244616	0.934796	0.280785	-0.027040	0.928031
jp_sales	-0.074142	0.264513	0.244616	1.000000	0.195192	0.134143	0.194025	0.434394
other_sales	-0.090225	0.817367	0.934796	0.195192	1.000000	0.275289	-0.011500	0.921370
critic_score	0.064322	0.301130	0.280785	0.134143	0.275289	1.000000	0.502221	0.313700
user_score	0.039318	-0.020010	-0.027040	0.194025	-0.011500	0.502221	1.000000	-0.002608
common_sales	-0.117878	0.922617	0.928031	0.434394	0.921370	0.313700	-0.002608	1.000000

корреляция между продажами и оценкой критиков 0.32

```
plt.scatter(data_2013 ['critic_score'], data_2013 ['common_sales'])
plt.ylim(0,8)
plt.xlabel('оценки критиков')
plt.ylabel('продажи')
plt.title('зависимость оценок критиков на продажи');
```

зависимость между оценкой эксперта и продажами экспоненциальная

теперь рассмотрим зависимость внутри одной платформы

```
data_2013.loc[data_2013.platform == 'ps4'].corr()
```

	year_of_release	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	common_sales
year_of_release	1.000000	-0.248961	-0.208306	-0.060993	-0.234796	-0.021142	0.152447	-0.235032
na_sales	-0.248961	1.000000	0.785362	0.472981	0.944259	0.415008	-0.020933	0.928160
eu_sales	-0.208306	0.785362	1.000000	0.464563	0.944698	0.346720	-0.048925	0.958157
jp_sales	-0.060993	0.472981	0.464563	1.000000	0.496467	0.322358	0.171332	0.527129
other_sales	-0.234796	0.944259	0.944698	0.496467	1.000000	0.409191	-0.035639	0.998051
critic_score	-0.021142	0.415008	0.346720	0.322358	0.409191	1.000000	0.557654	0.406568
user_score	0.152447	-0.020933	-0.048925	0.171332	-0.035639	0.557654	1.000000	-0.031957
common_sales	-0.235032	0.928160	0.958157	0.527129	0.998051	0.406568	-0.031957	1.000000

коэффициент корреляции между оценкой критика и продажами на платформе ps4 равен 0.4

```
plt.scatter(data_2013.loc[data_2013.platform == 'ps4','critic_score'], data_2013.loc[data_2013.platform == 'ps4','common_sales

plt.xlabel('оценка критиков')
plt.ylabel('продажи')
plt.title('зависимость оценок критиков на продажи на платформе ps4');
```

у платформы PS4 зависимость между оценкой эксперта и продажами экспоненциальная. с увеличением оценки возрастают и продажи

```
plt.scatter(data_2013.loc[data_2013.platform == 'ps4','user_score'], data_2013.loc[data_2013.platform == 'ps4','common_sales']

plt.xlabel('оценка пользователей')
plt.ylabel('продажи')
plt.title('зависимость оценок пользователей на продажи на платформе ps4');
```

коэффициент корреляции около 0, поэтому оценка пользователей не влияет на продажи

```
data_2013.loc[data_2013.platform == 'ps3'].corr()
```

	year_of_release	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	common_sales
year_of_release	1.000000	-0.217596	-0.167604	-0.195894	-0.181897	-0.167495	-0.270341	-0.201274
na_sales	-0.217596	1.000000	0.874896	0.439867	0.932098	0.335205	-0.013560	0.954921
eu_sales	-0.167604	0.874896	1.000000	0.443809	0.975743	0.309561	-0.022848	0.974740
jp_sales	-0.195894	0.439867	0.443809	1.000000	0.459609	0.302327	0.244048	0.516258
other_sales	-0.181897	0.932098	0.975743	0.459609	1.000000	0.315748	0.004633	0.989812
critic_score	-0.167495	0.335205	0.309561	0.302327	0.315748	1.000000	0.599920	0.334285

коэффициент корреляции между оценкой критика и продажами на платформе ps3 равен 0.33

common_sales	-0.201274	0.954921	0.974740	0.516258	0.989812	0.334285	0.002394	1.000000

```
plt.scatter(data_2013.loc[data_2013.platform == 'ps3','critic_score'], data_2013.loc[data_2013.platform == 'ps3','common_sales

plt.xlabel('оценка критиков')
plt.ylabel('продажи')
plt.title('зависимость оценок критиков на продажи на платформк ps3');
```

оценок недостаточно для определение зависомости

```
plt.scatter(data_2013.loc[data_2013.platform == 'ps3','user_score'], data_2013.loc[data_2013.platform == 'ps3','common_sales']

plt.xlabel('оценка пользователей')
plt.ylabel('продажи')
plt.title('зависимость оценок пользователей на продажи на платформе ps3');
```

оценка пользователей не влияет на продажи ps3

```
data_2013.loc[data_2013.platform == 'xone'].corr()
```

коэффициент корреляции между оценкой критика и продажами на платформе xone равен 0.42

```
plt.scatter(data_2013.loc[data_2013.platform == 'xone','critic_score'], data_2013.loc[data_2013.platform == 'xone','common_sal

plt.xlabel('оценка критиков')
plt.ylabel('продажи')
plt.title('зависимость оценок критиков на продажи на платформк xone');
```


```
plt.scatter(data_2013.loc[data_2013.platform == 'xone','user_score'], data_2013.loc[data_2013.platform == 'xone','common_sales

plt.xlabel('оценка пользователей')
plt.ylabel('продажи')
plt.title('зависимость оценок пользователей на продажи на платформе xone');
```

коэффициент корреляции около 0, поэтому оценка пользователей не влияет на продажи

у платформы xone зависимость между оценкой эксперта и продажами экспоненциальная. с увеличением оценки возрастают и продажи

```
data_2013.loc[data_2013.platform == 'x360'].corr()
```

	year_of_release	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	common_sales
year_of_release	1.000000	-0.218293	-0.203537	-0.285360	-0.218199	-0.245439	-0.248736	-0.220495
na_sales	-0.218293	1.000000	0.866574	0.634340	0.985236	0.342724	-0.012298	0.984299
eu_sales	-0.203537	0.866574	1.000000	0.612002	0.934769	0.336418	-0.009435	0.941008
jp_sales	-0.285360	0.634340	0.612002	1.000000	0.641529	0.290613	0.112592	0.648860
other_sales	-0.218199	0.985236	0.934769	0.641529	1.000000	0.349204	-0.018868	0.998640
critic_score	-0.245439	0.342724	0.336418	0.290613	0.349204	1.000000	0.520946	0.350345
user_score	-0.248736	-0.012298	-0.009435	0.112592	-0.018868	0.520946	1.000000	-0.011742
common_sales	-0.220495	0.984299	0.941008	0.648860	0.998640	0.350345	-0.011742	1.000000

коэффициент корреляции с x360 равен 0.35

```
plt.scatter(data_2013.loc[data_2013.platform == 'x360','critic_score'], data_2013.loc[data_2013.platform == 'x360','common_sal

plt.xlabel('оценка критиков')
plt.ylabel('продажи')
plt.title('зависимость оценок критиков на продажи на платформк x360');
```

оценок недостаточно для определение зависомости

```
plt.scatter(data_2013.loc[data_2013.platform == 'x360','user_score'], data_2013.loc[data_2013.platform == 'x360','common_sales

plt.xlabel('оценка пользователей')
plt.ylabel('продажи')
plt.title('зависимость оценок пользователей на продажи на платформе x360');
```

оценка пользователей не влияет на продажи x360

```
data_2013.loc[data_2013.platform == '3ds'].corr()
```

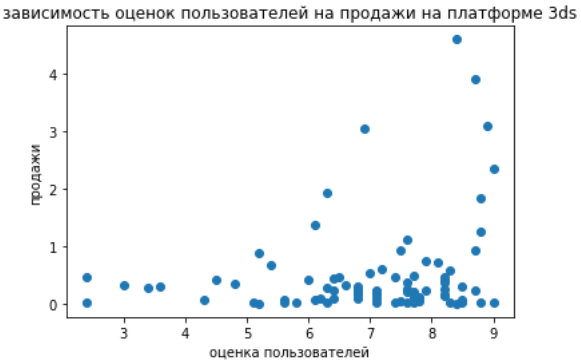
	year_of_release	na_sales	eu_sales	jp_sales	other_sales	critic_score	user_score	common_sales
year_of_release	1.000000	-0.075933	-0.115177	-0.063487	-0.083695	0.166254	0.240047	-0.090086
na_sales	-0.075933	1.000000	0.931893	0.641878	0.993708	0.369653	0.241036	0.938867
eu_sales	-0.115177	0.931893	1.000000	0.633982	0.958274	0.268851	0.114930	0.927821
jp_sales	-0.063487	0.641878	0.633982	1.000000	0.645306	0.301810	0.259370	0.854173
other_sales	-0.083695	0.993708	0.958274	0.645306	1.000000	0.354365	0.209418	0.945649
critic_score	0.166254	0.369653	0.268851	0.301810	0.354365	1.000000	0.769536	0.357057
user_score	0.240047	0.241036	0.114930	0.259370	0.209418	0.769536	1.000000	0.241504
common_sales	-0.090086	0.938867	0.927821	0.854173	0.945649	0.357057	0.241504	1.000000

коэффициент корреляции между оценкой критика и продажами на платформе 3ds равен 0.31

```
plt.scatter(data_2013.loc[data_2013.platform == '3ds','critic_score'], data_2013.loc[data_2013.platform == '3ds','common_sales

plt.xlabel('оценка критиков')
plt.ylabel('продажи')
plt.title('зависимость оценок критиков на продажи на платформк 3ds');
```

оценок недостаточно для определение зависомости

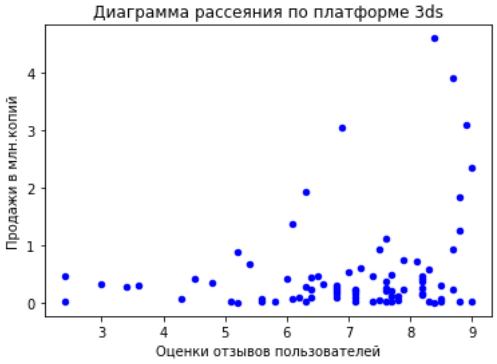
```
plt.scatter(data_2013.loc[data_2013.platform == '3ds','user_score'], data_2013.loc[data_2013.platform == '3ds','common_sales']

plt.xlabel('оценка пользователей')
plt.ylabel('продажи')
plt.title('зависимость оценок пользователей на продажи на платформе 3ds');
```



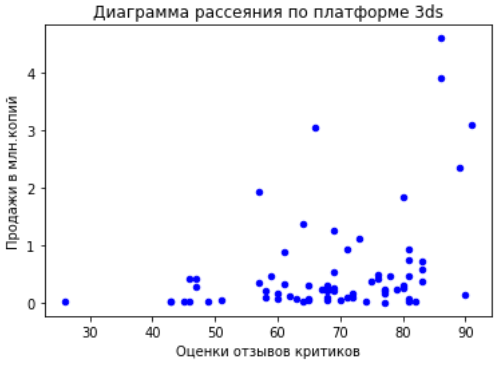
оценок недостаточно для определение зависомости

```
import matplotlib.pyplot as plt
df_sc, y = data_2013.copy(), 'common_sales'
for platform, games_on_pl in df_sc.groupby('platform'):
    print('='*60)
    print('Расчет по Платформе',platform)
    # Считаем сколько в колонке не пустых отзывов
    not_user = len(games_on_pl[games_on_pl['user_score'].notna() == True])
    not_critic = len(games_on_pl[games_on_pl['critic_score'].notna() == True])
    sum_not = 3 # Задаем количество не пустых значений для вывода диаграммы и расчета корреляции
    if not_user > sum_not:
        games_on_pl.plot(kind='scatter', x='user_score', y=y, color='b')
        display(games_on_pl[['user_score', y]].corr()[y])
        plt.xlabel('Оценки отзывов пользователей')
        plt.ylabel('Продажи в млн.копий')
        plt.title('Диаграмма рассеяния по платформе '+platform)
        plt.show()
    else:
        print('Для платформы',platform, 'не хватает данных для построения диаграммы и расчета корреляции отзывов пользователей
    if not_critic > sum_not:
        games_on_pl.plot(kind='scatter', x='critic_score', y=y, color='b')
        display(games_on_pl[['critic_score', y]].corr()[y])
        plt.xlabel('Оценки отзывов критиков')
        plt.ylabel('Продажи в млн.копий')
        plt.title('Диаграмма рассеяния по платформе '+platform)
        plt.show()
    else:
        print('Для платформы',platform, 'не хватает данных для построения диаграммы и расчета корреляции отзывов критиков')
```

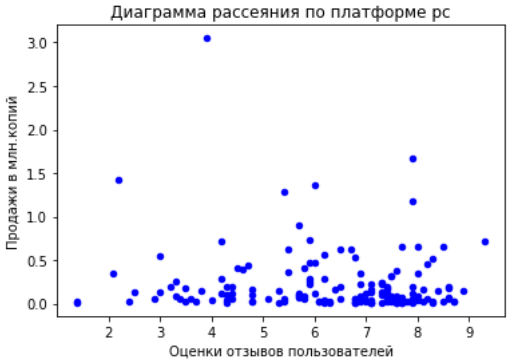
```
============================================================
Расчет по Платформе 3ds
user_score      0.241504
common_sales    1.000000
Name: common_sales, dtype: float64
```



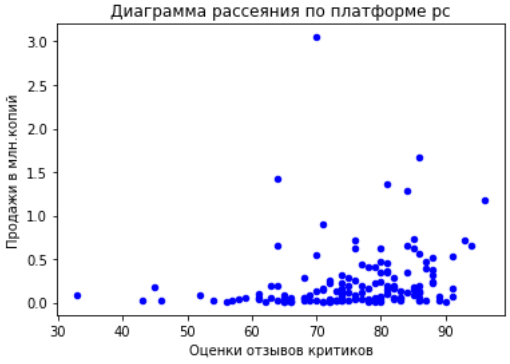

```
critic_score    0.357057
common_sales    1.000000
Name: common_sales, dtype: float64
```




```
============================================================
Расчет по Платформе ds
Для платформы ds не хватает данных для построения диаграммы и расчета корреляции отзывов пользователей
Для платформы ds не хватает данных для построения диаграммы и расчета корреляции отзывов критиков
============================================================
Расчет по Платформе pc
user_score     -0.093842
common_sales    1.000000
Name: common_sales, dtype: float64
```




```
critic_score    0.19603
common_sales    1.00000
Name: common_sales, dtype: float64
```




```
============================================================
Расчет по Платформе ps3
user_score      0.002394
common_sales    1.000000
Name: common_sales, dtype: float64
```

платформы ps3, x360, 3ds в актульном периоде имеют немного продаж, поэтому определить зависимость между оценкой и продажами не получится, а платформы ps4 xone имеют экспоненциальную зависимость

- общее распределение игр по жанрам

```
data_2013.pivot_table(index='genre', aggfunc='median').sort_values('common_sales', ascending=False)
```

genre	common_sales	critic_score	eu_sales	jp_sales	na_sales	other_sales	user_score	year_of_release
shooter	0.450	76.0	0.190	0.00	0.200	0.050	6.55	2014.0
sports	0.240	77.0	0.050	0.00	0.080	0.020	5.50	2014.5
platform	0.225	77.0	0.080	0.00	0.090	0.025	7.10	2014.0
role-playing	0.125	74.0	0.010	0.05	0.020	0.010	7.60	2014.0
fighting	0.125	72.0	0.020	0.03	0.045	0.010	7.50	2014.0
racing	0.120	74.0	0.060	0.00	0.030	0.010	6.20	2014.0
action	0.110	73.0	0.020	0.01	0.020	0.010	7.10	2015.0
simulation	0.100	69.5	0.035	0.00	0.000	0.005	6.80	2015.0
misc	0.100	75.0	0.010	0.02	0.010	0.000	7.00	2014.0
strategy	0.080	79.0	0.025	0.00	0.000	0.000	7.10	2015.0
puzzle	0.060	71.0	0.000	0.02	0.000	0.000	7.50	2014.0
adventure	0.030	72.0	0.000	0.01	0.000	0.000	7.50	2014.0

```
user score     -0.031957
```

в жанрах шутеры и спортивные продажи самые высокие, самые низкие-приключенческий жанр и пазлы.

Диаграмма рассеяния по платформе ps4

Шаг 4. Составьте портрет пользователя каждого региона

```
data_2013.pivot_table(index='platform', values=['na_sales','jp_sales', 'eu_sales','common_sales']).sort_values('common_sales',
```

platform	common_sales	eu_sales	jp_sales	na_sales
ps4	0.801378	0.359923	0.040714	0.277398
x360	0.735484	0.228602	0.002742	0.439032
xone	0.645020	0.208866	0.001377	0.377004
wii	0.593913	0.257826	0.002174	0.285217
wiiu	0.562000	0.172609	0.094609	0.254000

```
na_platform = data_2013.pivot_table(index='platform', values='na_sales', aggfunc='sum').sort_values('na_sales', ascending=Fals
```

```
na_platform_top5 = na_platform[:5]

na_platform_top5_2 = na_platform[5:]
na_platform_top5_2 = pd.DataFrame([na_platform_top5_2.sum()], index=["Other"])
na_platform_totals = na_platform_top5.append(na_platform_top5_2)
```

20 30 40 50 60 70 80 90 100

```
jp_platform = data_2013.pivot_table(index='platform', values='jp_sales', aggfunc='sum').sort_values('jp_sales', ascending=Fals
```

Расчет по Платформе psp

```
jp_platform_top5 = jp_platform[:5]

jp_platform_top5_2 = jp_platform[5:]
jp_platform_top5_2 = pd.DataFrame([jp_platform_top5_2.sum()], index=["Other"])
jp_platform_totals = jp_platform_top5.append(jp_platform_top5_2)
```

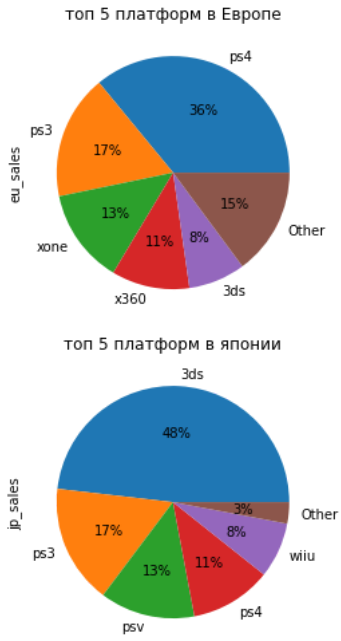
Name: common_sales, dtype: float64

```
eu_platform = data_2013.pivot_table(index='platform', values='eu_sales', aggfunc='sum').sort_values('eu_sales', ascending=Fals
```

0.7

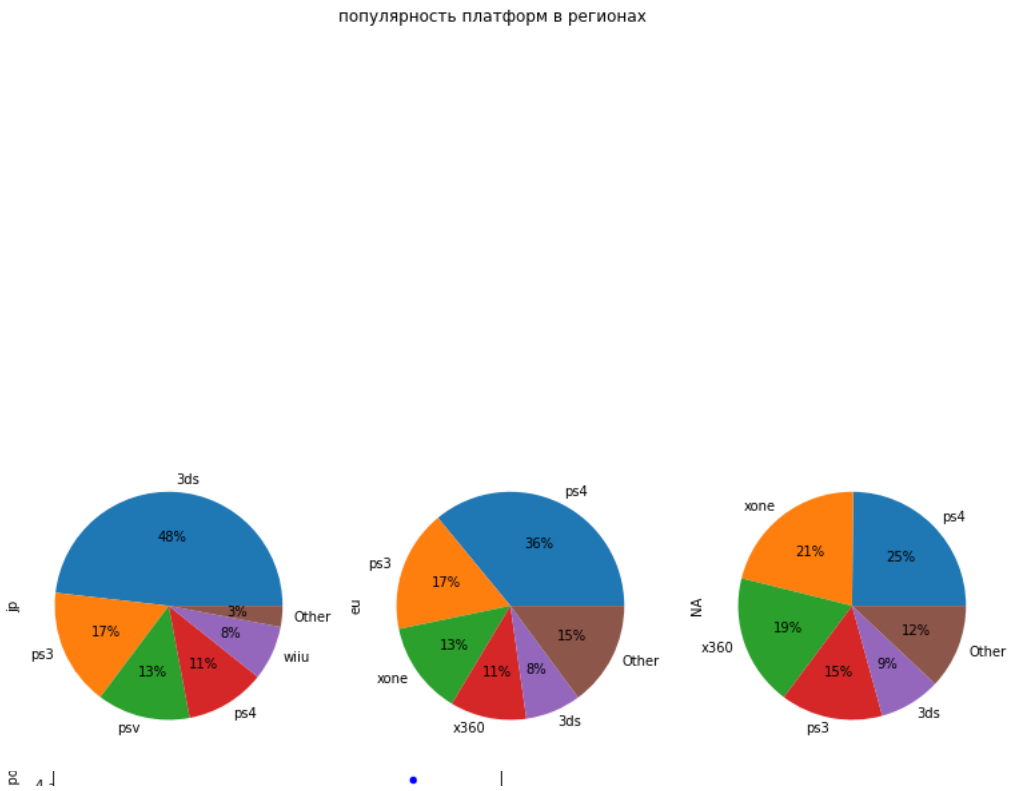
```
eu_platform_top5 = eu_platform[:5]

eu_platform_top5_2 = eu_platform[5:]
eu_platform_top5_2 = pd.DataFrame([eu_platform_top5_2.sum()], index=["Other"])
eu_platform_totals = eu_platform_top5.append(eu_platform_top5_2)
```

```
eu_platform_totals.plot('platform','eu_sales',kind='pie', autopct='%1.0f%%', legend=False, title='топ 5 платформ в Европе')

jp_platform_totals.plot('platform','jp_sales',kind='pie', autopct='%1.0f%%', legend=False, title='топ 5 платформ в японии')

na_platform_totals.plot('platform','na_sales',kind='pie', autopct='%1.0f%%', legend=False, title='топ 5 платформ в северной ам

plt.show()
```




```
fig, axes = plt.subplots(nrows = 1, ncols = 3, figsize = (13,13));

jp_platform_totals.plot(
    kind='pie',
    title = 'популярность платформ в регионах ',
    ylabel = 'jp',
    autopct='%1.0f%%',
    legend=False,
    ax = axes[0],
    subplots=True );

eu_platform_totals.plot(
    kind='pie',
    ylabel = 'eu',
    autopct='%1.0f%%',
    legend=False,
    ax = axes[1],
    subplots=True);

na_platform_totals.plot(
    kind='pie',
    title = 'популярность платформ в регионах ',
    ylabel = 'NA',
    autopct='%1.0f%%',
    legend=False,
    ax = axes[2],
```

```
    subplots=True);
plt.show()
```



самая популярная платформа в Японии 3ds, она занимает почти половину всех вышедших игр.В европе и северно америке -PS4

```
na_genre = data_2013.pivot_table(index='genre', values='na_sales', aggfunc='sum').sort_values('na_sales', ascending=False)
na_genre_top5 = na_genre[:5]

na_genre_top5_2 = na_genre[5:]
na_genre_top5_2 = pd.DataFrame([na_genre_top5_2.sum()], index=["Other"])
na_genre_totals = na_genre_top5.append(na_genre_top5_2)
```

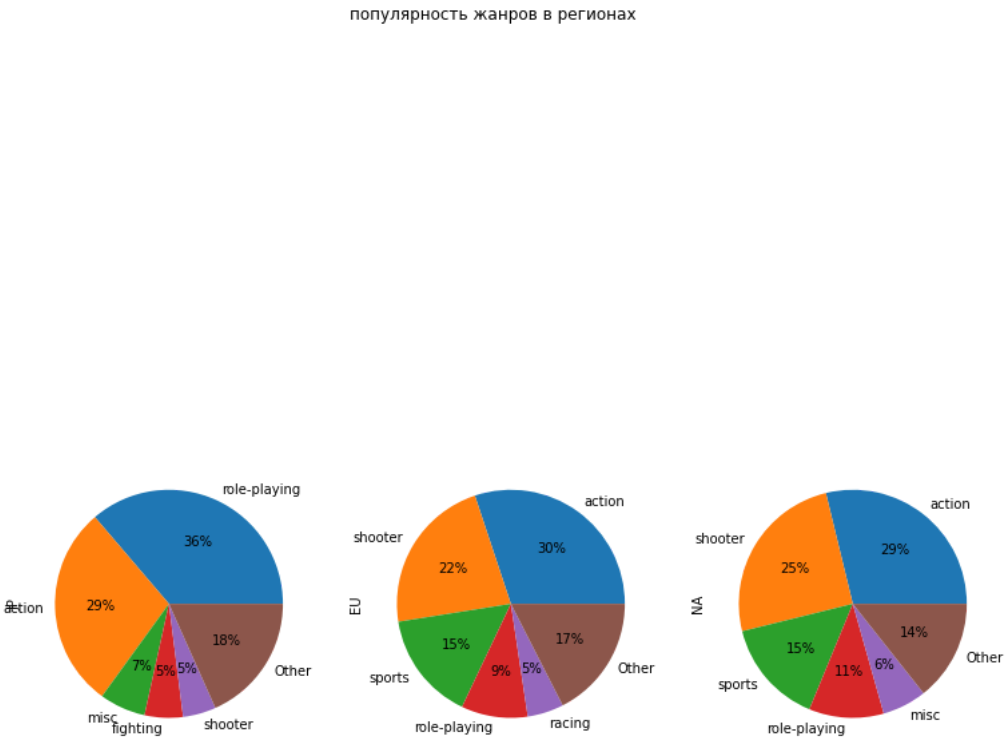
Диаграмма рассеяния по платформе xone

```
jp_genre = data_2013.pivot_table(index='genre', values='jp_sales', aggfunc='sum').sort_values('jp_sales', ascending=False)
jp_genre_top5 = jp_genre[:5]

jp_genre_top5_2 = jp_genre[5:]
jp_genre_top5_2 = pd.DataFrame([jp_genre_top5_2.sum()], index=["Other"])
jp_genre_totals = jp_genre_top5.append(jp_genre_top5_2)
```

```
eu_genre = data_2013.pivot_table(index='genre', values='eu_sales', aggfunc='sum').sort_values('eu_sales', ascending=False)
eu_genre_top5 = eu_genre[:5]

eu_genre_top5_2 = eu_genre[5:]
eu_genre_top5_2 = pd.DataFrame([eu_genre_top5_2.sum()], index=["Other"])
eu_genre_totals = eu_genre_top5.append(eu_genre_top5_2)
```

```
common_sales    1.000000
fig, axes = plt.subplots(nrows = 1, ncols = 3, figsize = (13,13));

jp_genre_totals.plot(
    kind='pie',
    title = 'популярность платформ в регионах ',
    ylabel = 'JP',
    autopct='%1.0f%%',
    legend=False,
    ax = axes[0],
    subplots=True );

eu_genre_totals.plot(
    kind='pie',
    ylabel = 'EU',
    autopct='%1.0f%%',
    legend=False,
    ax = axes[1],
    subplots=True);

na_genre_totals.plot(
    kind='pie',
    title = 'популярность жанров в регионах ',
```

```
    ylabel = 'NA',
    autopct='%1.0f%%',
    legend=False,
    ax = axes[2],
    subplots=True);
plt.show()
```



в японии предпочитают ролевые игры. в европе-экшн, а северной америке-шутеры

```
data_2013.rating =  data_2013.rating.fillna('no rating')
```

```
data_2013.pivot_table(index='rating',values=['eu_sales', 'jp_sales','na_sales','common_sales'], aggfunc='sum')\
            .sort_values('common_sales', ascending=False)
```

rating	common_sales	eu_sales	jp_sales	na_sales
M	371.68	145.32	14.11	165.21
no rating	276.84	78.91	85.05	89.42
E	200.16	83.36	15.14	79.05
T	126.62	41.95	20.59	49.79
E10+	115.39	42.69	5.89	54.24

```
data_2013.rating.unique()
```

```
array(['M', 'no rating', 'E', 'T', 'E10+'], dtype=object)
```

самые высокие суммарные продажи у игр для лиц старше 17 лет, самые низкие-для лиц старше 10 лет

```
temp = data.copy()
print(temp.rating.isna().sum(), temp.rating.isna().sum()/len(temp))
temp.rating.value_counts()
```

```
6676 0.40605802566753846
E       3920
T       2903
M       1536
E10+    1393
EC         8
K-A        3
RP         1
```

```
AO          1
Name: rating, dtype: int64
```

Шаг 5. Проверьте гипотезы

Средние пользовательские рейтинги платформ Xbox One и PC одинаковые;

H0: Средние пользовательские рейтинги платформ Xbox One и PC одинаковые

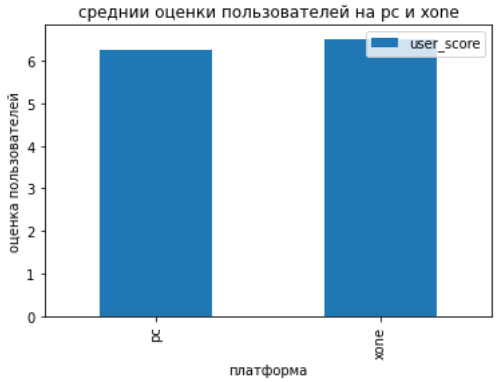
H1: Средние пользовательские рейтинги платформ Xbox One и PC различны

```
Xone_PC_user_score = data_2013.loc[(data.platform == 'xone') | (data.platform == 'pc')]\
                               .groupby('platform').agg({'user_score':'mean'})
```

```
Xone_PC_user_score
```

platform	user_score
pc	6.269677
xone	6.521429

```
Xone_PC_user_score.plot(kind='bar')
plt.xlabel('платформа')
plt.ylabel('оценка пользователей')
plt.title('среднии оценки пользователей на pc и xone');
```




```
xone = data_2013[data.platform == 'xone'].user_score.dropna().reset_index(drop=True)
pc = data_2013[data.platform == 'pc'].user_score.dropna().reset_index(drop=True)
```

```
results = st.ttest_ind(xone, pc )
print('p-значение:', results.pvalue)
alpha = 0.01
```

```
p-значение: 0.14012658403611647
```

```
if results.pvalue < alpha:

    print('Отвергаем нулевую гипотезу')
else:
    print('Не получилось отвергнуть нулевую гипотезу')
```

```
Не получилось отвергнуть нулевую гипотезу
```

вывод: Средние пользовательские рейтинги платформ Xbox One и PC различны. значение p-value оказалось меньше 1% уровня значимости.Следовательно, нулевая гипотеза отвергнута

H0: Среднее пользовательского рейтинга жанрв Action равно среднему пользовательского рейтинга жанрв Sports

H1: Среднее пользовательского рейтинга жанрв Action не равно среднему пользовательского рейтинга жанрв Sports

```
action_score = data_2013[data.genre == 'action'].user_score.dropna()
```

```
sports_score = data_2013[data.genre == 'sports'].user_score.dropna()


results = st.ttest_ind(action_score, sports_score )
print('p-значение:', results.pvalue)
alpha = 0.05
```

```
p-значение: 1.0517832389140023e-27
```

```
if results.pvalue < alpha:

    print('Отвергаем нулевую гипотезу')
else:
    print('Не получилось отвергнуть нулевую гипотезу')
```

```
Отвергаем нулевую гипотезу
```

вывод:Средние пользовательские рейтинги жанров Action и Sports разные, значение p-value оказалось выше 5% уровня значимости

были проведены тесты о равенстве средних двух генеральных совокупностей.

необходимо было провести двусторонний t-тест, формулировкой нулевой гипотезы стало выражение о равенсте двух средних значений генераальной совокупности, альтернативная гипотеза-обратная нулевой

Шаг 6. Напишите общий вывод

В работе были рассмотрены данные по выпуску игр на различные платформы.

Были заменены названия столбцов, посчитаны суммарные продажи, постреено распределение по годам,на основе которого выбран актульный период,в этом периоде были рассмотрены продажи на популярные платформы, рассмотрены зависимости отзывов критиков на продажи, рассмотрено количество продаж в зависимости от жанров, рассмотрены самые популярные платформы и жанры в кажом регионе,были проверены две гипотезы о равенстве средних пользовательских рейтингов платформ Xbox One и PC и средних пользовательских рейтингов жанров Action и Sports.

наибольшее количество игр было выпущено в 2010 году, затем их количество начало уменьшаться.Популярность платформы живет около 5 лет, поэтому будем считать актуальным период с 2013 года. Лидируют по продажам платформы xbox one, ps4 и игровая консоль 3ds. Платформа ps3 была наиболее папулярной в 2013 году, но затем количество вышедших игр снижается, после выхода ей на смену ps4 резко увеличивается количество игр на эту платформу. Самые высокие средние продажи на платформе ps4, самые низкие на 3ds. В жанрах шутеры и спортивные продажи самые высокие, самые низкие-приключенческий жанр и пазлы. Самая популярная платформа в Японии 3ds, она занимает почти половину всех вышедших игр.В европе и северно америке -PS4. В японии предпочитают ролевые игры, в европе-экшн, а северной америке-шутеры.Самые высокие суммарные продажи у игр для лиц старше 17 лет, самые низкие-для лиц старше 10 лет

вывод: наиболее целеосознанно вкладывать рекламный бюджет в игры в европе и северной америке на платформе ps4 в жанре шутеры и экшн для игр, предназначенных для лиц старше 17 лет