Description of the project

You work in the online store "Stremchik", which sells computer games all over the world. Historical data on game sales, user and expert ratings, genres and platforms (for example, Xbox or PlayStation) are available from open sources. You need to identify the patterns that determine the success of the game. This will allow you to bet on a potentially popular product and plan your advertising campaigns.

You have data up to 2016. Let's say it's December 2016 and you're planning a campaign for 2017. We need to work out the principle of working with data. It doesn't matter if you predict 2017 sales based on 2016 data or 2027 based on 2026 data.

Games.csv data description

Name - the name of the game
Platform - platform
Year_of_Release - release year
Genre - game genre
NA_sales - North American sales (millions of dollars)
EU_sales - Sales in Europe (millions of dollars)
JP_sales - Sales in Japan (millions of dollars)
Other_sales - sales in other countries (millions of dollars)
Critic_Score - score of critics (from 0 to 100)
User_Score - user score (from 0 to 10)
Rating is a rating from the ESRB (Entertainment Software Rating Board) organization. This association determines the rating of computer games and assigns them a suitable age category.
Data for 2016 may be incomplete.

Reviewer's comment 2

Getting started is very important. So you explain what it is dedicated to. It would also be good to put the purpose of the work in a separate introduction block. It will be even better if you make a plan for working with hyperlinks. Yes, there is a ToC plugin. But it would be nice to be able to implement an interactive project plan manually. How to implement it - see on
[link] (https://stackoverflow.com/questions/49535664/how-to-hyperlink-in-a-jupyter-notebook/49717704).

Step 2. Open the data file and study the general information

Reviewer's comment

I noticed that the cells in your notebook do not start with 1. Before submitting the work, I recommend restarting your laptop to make sure that all cells are executed correctly.

Reviewer's comment 2
Restarting the project allows you to detect and fix errors hidden in the code.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

from datetime import datetime
from scipy import stats as st

games = pd.read_csv('/datasets/games.csv')
games.head()

games.describe()
games.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16715 entries, 0 to 16714
Data columns (total 11 columns):
Name               16713 non-null object
Platform           16715 non-null object
Year_of_Release    16446 non-null float64
Genre              16713 non-null object
NA_sales           16715 non-null float64
EU_sales           16715 non-null float64
JP_sales           16715 non-null float64
Other_sales        16715 non-null float64
Critic_Score       8137 non-null float64
User_Score         10014 non-null object
Rating             9949 non-null object
dtypes: float64(6), object(5)
memory usage: 1.4+ MB
```

Reviewer's comment
It is correct that you carry out all imports in the first cell of the work. So your colleague who starts the work will be aware of the libraries used in it and can, if necessary, quickly set up the environment.

Step 3. Prepare the data

It is enough just to convert uppercase letters to lowercase

In [2]:
```
games.columns = games.columns.str.lower()
```

Reviewer's comment
This method allows us to automate the process of converting column names to lowercase. This eliminates the possibility of typos.

The rest too, so as not to accidentally get confused

In [3]:
```
games['name'] = games['name'].str.lower()
games['platform'] = games['platform'].str.lower()
games['genre'] = games['genre'].str.lower()
games['rating'] = games['rating'].str.lower()
```

Duplicate check - 0

In [4]:
```
games.duplicated().sum()
```

Out[4]: 0

convert to float format, replacing incorrect values with NaN

In [5]:
```
games['user_score'] = pd.to_numeric(games['user_score'], errors='coerce')
```

Reviewer's comment
Right, tbd is essentially Nan. Great for detecting implicit missing values.

Fill in the gaps in the column with the year of publication with zeros and convert everything to int

In [6]:
```
games['year_of_release'] = games['year_of_release'].fillna(0)
games['year_of_release'] = games['year_of_release'].astype('int')
```

In the column with the names of the games, replace 2 gaps

In [7]:
```
games['name'] = games['name'].fillna('unknown_name')
```

In the column with game genres, replace 2 passes

In [8]:
```
games['genre'] = games['genre'].fillna('unknown_genre')
```

Age rating skips

In [9]:
```
games['rating'] = games['rating'].fillna('unknown_rating')
```

In [10]:
```
games.isnull().sum().sort_values(ascending = False)
```

Out[10]:
```
user_score         9125
critic_score       8578
rating                0
other_sales           0
jp_sales              0
eu_sales              0
na_sales              0
genre                 0
year_of_release       0
platform              0
name                  0
dtype: int64
```

Let's check what unnumbered values are found, otherwise suddenly the audience's assessment - they wrote the number in letters
Basically, gaps are observed in the columns 'critic_score', 'user_score' and 'rating'
Most likely they were pulled up by id of games from another database
It is possible that this data is not available for specific platforms (devices N64, SNES, SAT, 2600, GB, NES, GEN, NG, etc.) or years (too old games)
tbd - means that the rating was under discussion, but, apparently, was never published in the end

Reviewer's comment
Yes, we cannot restore the missing values.
There is not enough data for this. Better to work with less good quality data.
Data errors have been eliminated. The data has been prepared for further analysis.

Count the total sales in all regions and write them down in a separate column

In [11]:
```
games['total_sales'] = games['na_sales'] + games['eu_sales'] + games['jp_sales'] + games['other_sales']
games.head()
```

Out[11]:

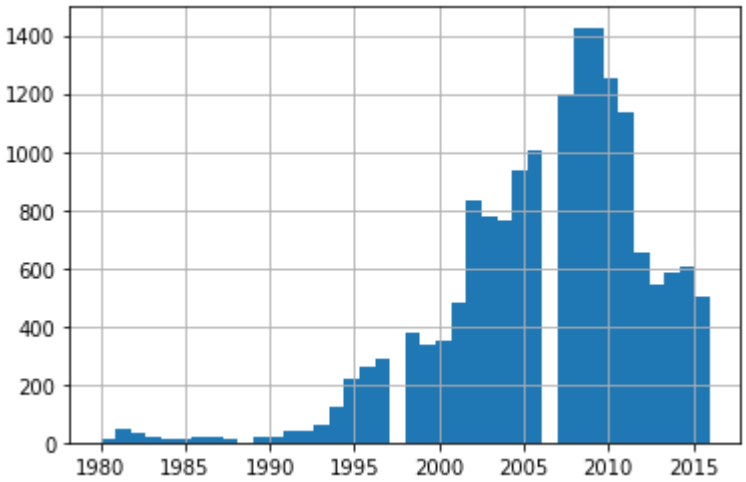
	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales	cr
0	wii sports	wii	2006	sports	41.36	28.96	3.77	8.45	
1	super mario bros.	nes	1985	platform	29.08	3.58	6.81	0.77	
2	mario kart wii	wii	2008	racing	15.68	12.76	3.79	3.29	
3	wii sports resort	wii	2009	sports	15.61	10.93	3.28	2.95	
4	pokemon red/pokemon blue	gb	1996	role-playing	11.27	8.89	10.22	1.00	

Step 4. Conduct exploratory data analysis

Release of games by years

In [12]:
```
games[games['year_of_release'] != 0]['year_of_release'].hist(bins=40)
```

Out[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6363f8b2d0>
```



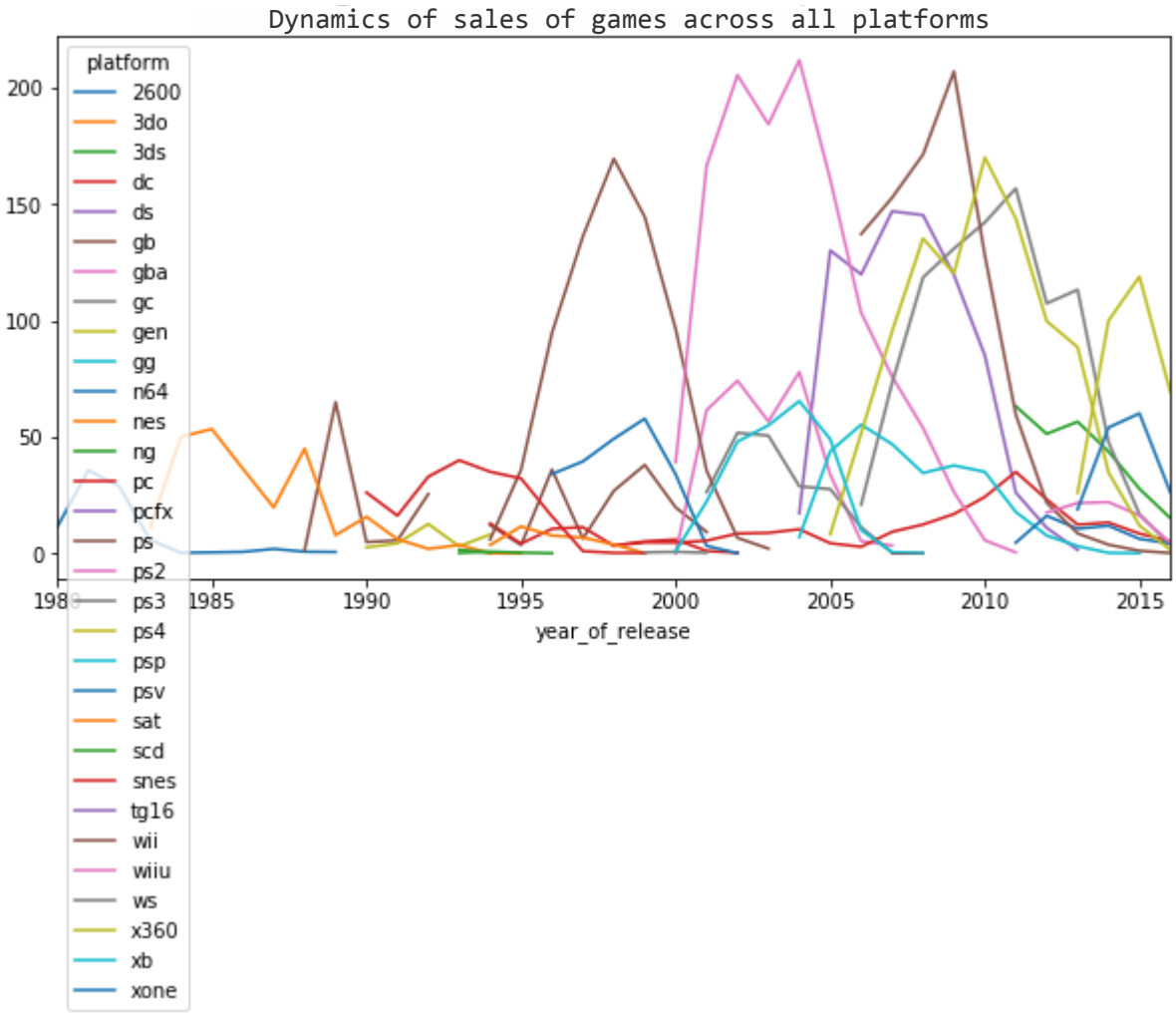
The main peak of game release falls on 2008-2010
Games started publishing in the early 1980s
But it took as much as 15-20 years of technology development to start mass production of games. In my opinion, we can exclude data up to 2000, because the number of games released in those years is insignificant in comparison with further data
The decline is most likely due to the development of mobile devices (smartphones, tablets, etc.) - It became easier for users to play during breaks, in queues, on the way to work, etc.

Reviewer's comment
What do you think is the reason for the decline in the industry of the latest years?

Reviewer's comment 2
I agree with the stated reason for the decline.

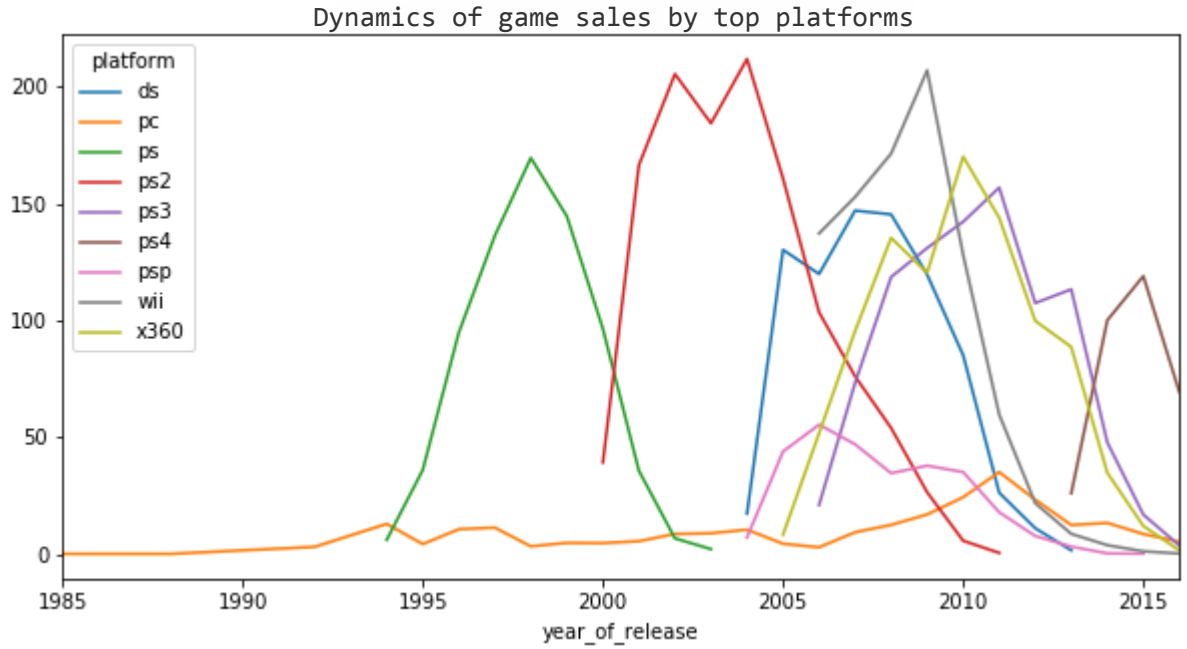
See how sales have changed by platform

In [13]:
```
games.pivot_table(index='year_of_release', columns='platform', values='total_sales', aggfunc = 'sum')
games[games['year_of_release'] != 0].pivot_table(index='year_of_release', columns='platform',
                                    values='total_sales',aggfunc = 'sum').plot(figsize=(10, 5))
plt.title('Dynamics of sales of games across all platforms')
plt.show()
```



Select the platforms with the highest total sales and plot the distribution by year

In [14]:
```
platform_sales = games.pivot_table(index='platform',
                           values='total_sales',
                           aggfunc =
                           'sum').sort_values(by='total_sales',ascending=False)
#print(platform_sales)
platform_sales_top = platform_sales.query('total_sales > 259')
#print(platform_sales_top)

platform_list = ['ps2', 'x360', 'ps3', 'wii', 'ds', 'ps', 'ps4', 'psp', 'pc']

games_top_platform = games.query('platform in @platform_list')

games_top_platform[games_top_platform['year_of_release'] != 0].pivot_table(ind
ex='year_of_release',

columns='platform',

values='total_sales',

aggfunc = 'sum').plot(figsize=(10, 5))
plt.title('Dynamics of game sales by top platforms') plt.show()
```



Find popular platforms in the past that now have sales on zero

ps, ps2, ds, wii, psp

During what characteristic period do new platforms appear and old platforms disappear?

the example of Xbox and PS shows that the average platform relevance cycle is 8-10 years

Let's create a pivot table in the context of platforms and display the maximum and minimum values of the games released on them
Further, we will exclude the threshold values, thus all the "extreme" platforms for which there is no data before 1980, or their cycle has not yet ended in 2016, will disappear.
By the way, the PC platform will disappear immediately

In [15]:
```
plat_year_pivot = games[games['year_of_release'] != 0].pivot_table(index='platform', values='year_of_release', aggfunc = ('max', 'min'))

plat_year_pivot_act = plat_year_pivot.query('(min > 1980) & (max < 2016) & (min != max)')
plat_year_pivot_act['platform_lifetime'] = plat_year_pivot_act['max'] - plat_year_pivot_act['min']
print(plat_year_pivot_act)
print(plat_year_pivot_act['platform_lifetime'].mean())
```

```
           max   min  platform_lifetime
platform
3do       1995  1994                  1
dc        2008  1998                 10
ds        2013  1985                 28
gb        2001  1988                 13
gba       2007  2000                  7
gc        2007  2001                  6
gen       1994  1990                  4
n64       2002  1996                  6
nes       1994  1983                 11
ng        1996  1993                  3
ps        2003  1994                  9
ps2       2011  2000                 11
psp       2015  2004                 11
sat       1999  1994                  5
scd       1994  1993                  1
snes      1999  1990                  9
ws        2001  1999                  2
xb        2008  2000                  8
8.055555555555555
```

There is one particularly "long-playing" console - Nintendo DS (28 years old)

Reviewer's comment

Mean lifetime value platform is given. However, you draw conclusions from the graphs. It is also worth giving the calculation of this value. Consider whether to include all platforms in your calculation. Will there be emissions by the lifespan of the platforms?

Reviewer's comment 2

The calculation is given. It is true that you did not include in the analysis current platforms. The period of their life is still going on. According to DS we have an emission in 1985. In fact, this platform was released in 2004.

Determine the data for which period you need to take in order to exclude a significant distortion of the distribution by platform in 2016

if 1995-2015 is the entire current period, to build a forecast, we will only analyze the data of the decline cycle of the gaming market starting from the peak in 2009, we will take the period 2009-2015

In [16]:
```
years_list = []
for element in range(2009,2016):
    years_list.append(element)
#print(years_list)

games_data = games.query('year_of_release in @years_list')
#print(games_data.head())
```

Reviewer's comment

The current period has been named. It is worth it significantly reduce. Now your period contains several stages of industry development at once: the formation of the market, growth, the peak of 2008 and 2009, as well as the contraction of the market in recent years. Also, most of the platforms are no longer in 2016, they will not help us in building a forecast for 2017. With a decrease in the period, only the latest generations of platforms will be considered, and we will also consider only the currently finite interval of the development of the gaming industry.

Reviewer's comment 2

This selection of up-to-date data will allow us to increase the quality and accuracy of the forecast for 2017.

Next, only work with the data that you have defined. Do not include data from previous years

Which platforms are leading in terms of sales, rising or falling? Choose a few potentially profitable platforms

In [17]:
```
games_data.pivot_table(index = 'platform', values = 'total_sales', aggfunc = 'sum').sort_values(by='total_sales', ascending=False)
```

Out[17]:

platform	total_sales
ps3	715.07
x360	669.18
wii	429.76
ps4	244.89
ds	243.29
3ds	242.67
pc	133.62
xone	133.17
psp	101.83
wiiu	77.59
psv	49.56
ps2	32.49

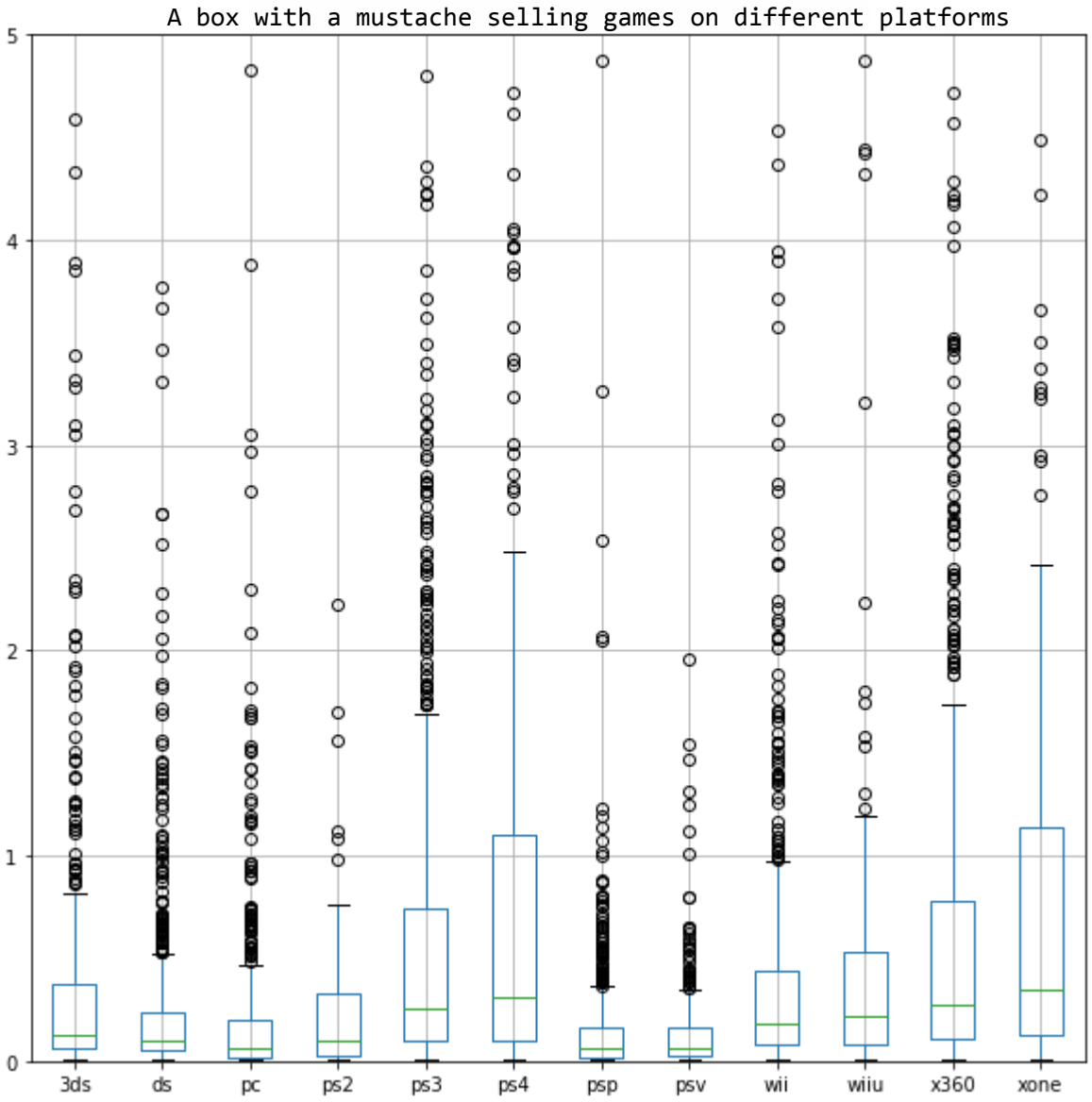
Top sellers for the period under review - PS3, Xbox360 and Wii
However, all of these platforms are already completing their popularity cycle.
PS4 can be considered as potentially profitable, replacing the once popular PS3 and Xbox One, replacing the Xbox 360
sales on PC are not as high as on game consoles, but PC games are relevant at all times

Reviewer's comment

Yes, we can call these platforms promising for 2017.

Plot a box and "mustache" graph of the global sales of each game and breakdown by platform

In [18]:
```
games_data.describe()

pivot_game_sale = games_data.pivot_table(index='name', columns='platform', val
ues='total_sales', aggfunc = 'sum')
#print(pivot_game_sale)

plot = pivot_game_sale.boxplot(figsize=(10, 10)).set_ylim(0, 5)
plt.title('A box with a mustache selling games on different platforms')
plt.show()
#plot = pivot_game_sale.boxplot(figsize=(10, 10))
#plt.show()
```



Reviewer's comment

Warnings are also worth dealing with. Sometimes there are too many of them, so it is important to be able to hide them. The warnings library will help you with this. Try to find a suitable method and remove the warnings.

Is there a big difference in sales? What about average sales across different platforms? Describe the result

There are some "successful" games, sales for which exceed 20, 25 and even 30 million dollars, which differs from the average and goes far beyond the "mustache"
In general, the situation is similar for platforms: the lower mustache rests at 0, and the upper one is in the range of $ 1 million for most of the platforms. There are more successful platforms - PS3, Xbox360 and Wiiu with a top mustache of over $ 1 million; and the PS4 and Xbox One platforms are the next generations of the above platforms, the normal distribution of game sales for which can reach $ 2.5 million.

It is clearly seen that with the evolution of a specific platform, sales by games are also growing.
For example PS4> PS3> PS2
A XboxOne> Xbox360
Wiiu> Wii

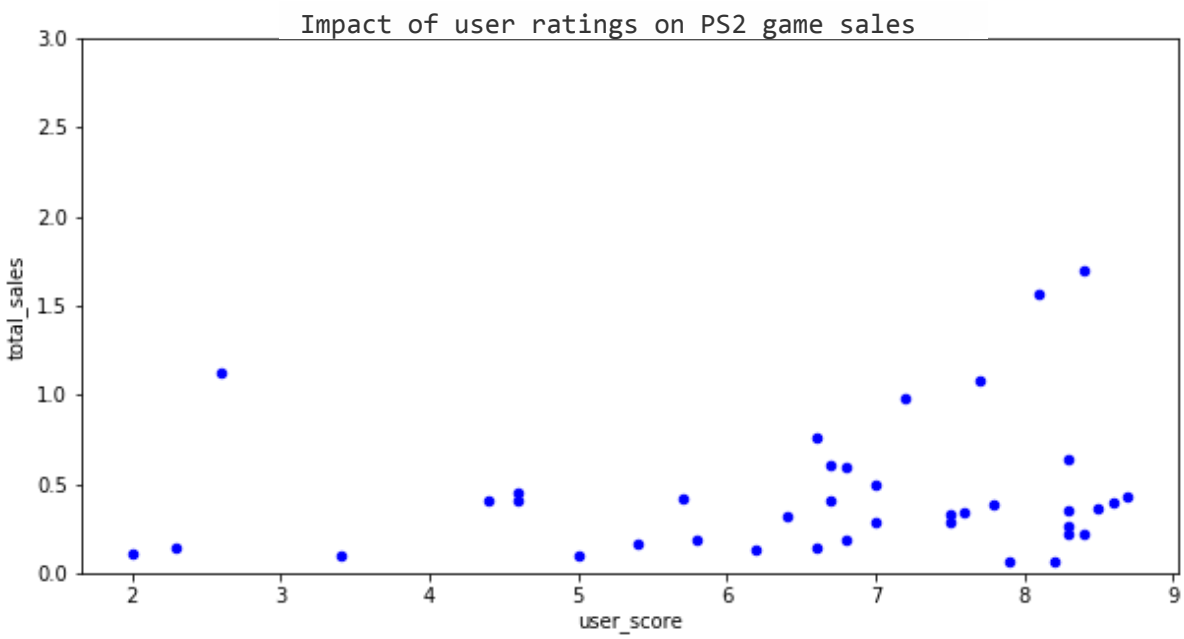
Although we took the data for 2009-2015, the difference in sales can still be affected by an increase in game prices over time, at least due to inflation, as well as an increase in game production costs

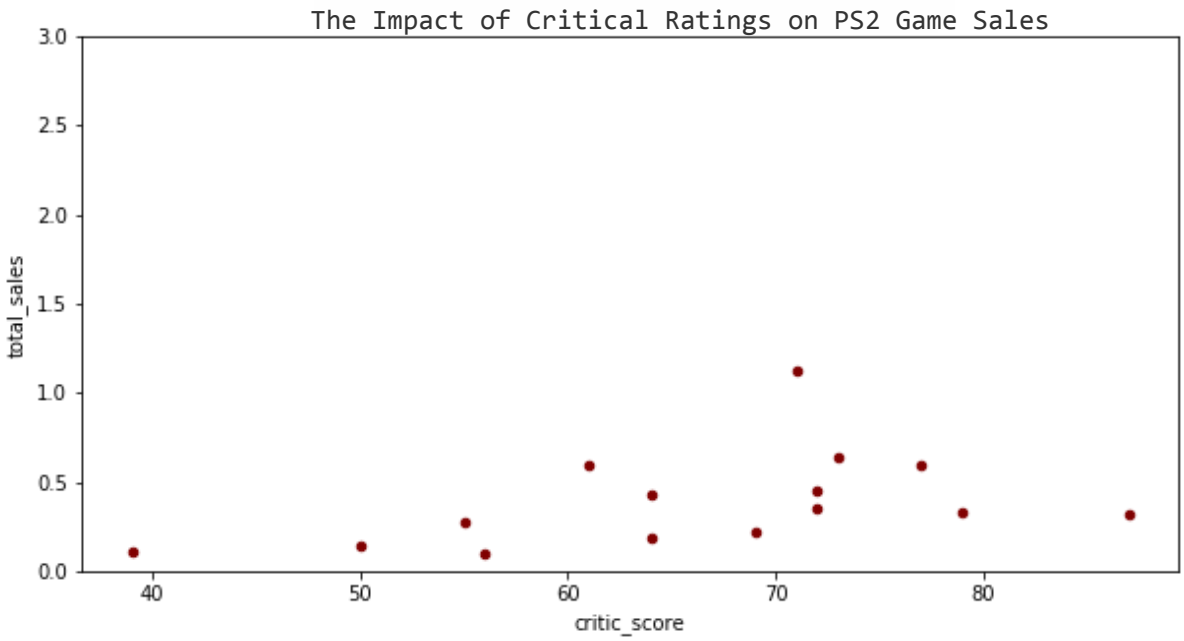
Reviewer's comment

Think about what makes the difference between platforms. Try not only to describe the result, but also to interpret it.

See how user and critical reviews affect sales within one popular platform

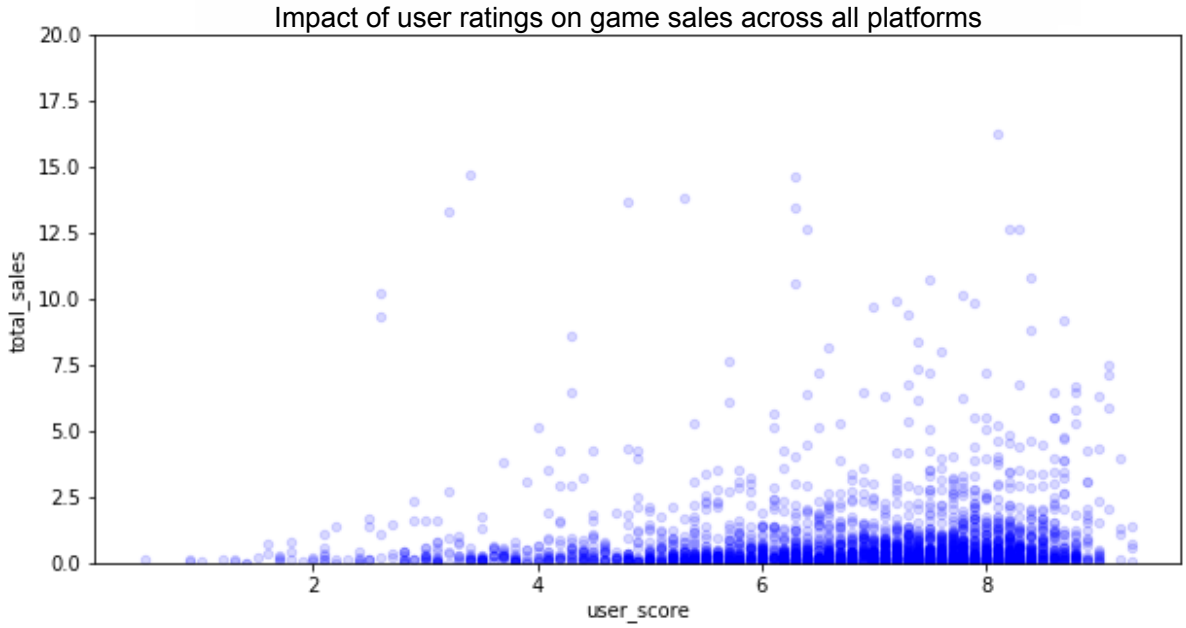
In [19]:
```
games_data_ps2 = games_data.query('platform == "ps2"')
#print(games_data_ps2.head())

games_data_ps2.plot(x='user_score', y='total_sales', kind='scatter', color='bl
ue', alpha = 1, figsize=(10, 5)).set_ylim(0, 3)
plt.title('Impact of user ratings on PS2 game sales')
plt.show()
games_data_ps2.plot(x='critic_score', y='total_sales', kind='scatter', color=
'maroon', alpha = 1, figsize=(10, 5)).set_ylim(0, 3)
plt.title('The Impact of Critical Ratings on PS2 Game Sales')
plt.show()
```

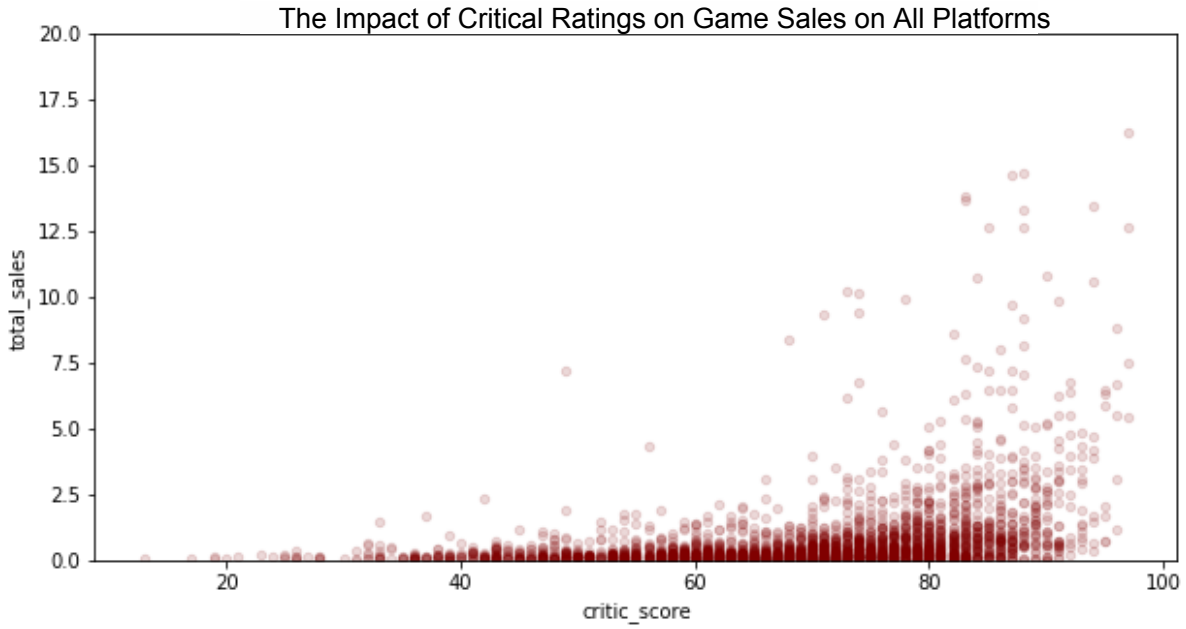




Build a scatterplot and calculate the correlation between reviews and sales

In [20]:
```
corr_data = pd.DataFrame()
corr_data['total_sales'] = games_data['total_sales']
corr_data['user_score'] = games_data['user_score']
corr_data['critic_score'] = games_data['critic_score']
#print(corr_data.head())

corr_data.plot(x='user_score', y='total_sales', kind='scatter', color='blue',
legend=True, alpha = 0.15, figsize=(10, 5)).set_ylim(0, 20)
plt.title('Impact of user ratings on game sales across all platforms')
plt.show()
corr_data.plot(x='critic_score', y='total_sales', kind='scatter', color='maroo
n', legend=True, alpha = 0.15, figsize=(10, 5)).set_ylim(0, 20)
plt.title('The Impact of Critical Ratings on Game Sales on All Platforms  ')
plt.show()

corr_data[['total_sales', 'user_score',
'critic_score']].corr().style.format(" {:.2%}")
```





Out[20]:

	total_sales	user_score	critic_score
total_sales	100.00%	7.69%	30.29%
user_score	7.69%	100.00%	58.14%
critic_score	30.29%	58.14%	100.00%

Formulate conclusions and correlate them with sales of games on other platforms

The diagrams for PS2 and general for all platforms are very similar in general
As a rule, the higher the ratings of the game's critics and users, the higher its sales (not without emissions, of course)

The weakest dependence of sales on user ratings - less than 9%
The dependence of sales on the ratings of critics is higher - about 25%, but it is still considered weak There is a dependence of the ratings of critics and the ratings of users - almost 59%

Reviewer's comment

The result is received. Think about what caused it. It's great that multiple platforms are considered.

Look at the general distribution of games by genre. What about the most profitable genres?

In [21]:
```
games_data.pivot_table(index='genre', values='name', aggfunc = 'count').sort_values(by='name', ascending=False)
```

Out[21]:

genre	name
action	1589
misc	750
sports	710
adventure	647
role-playing	617
shooter	453
simulation	322
racing	297
fighting	236
strategy	221
puzzle	192
platform	167

Most action, misc and sports games

In [22]:
```
games_data.pivot_table(index='genre', values='total_sales', aggfunc = 'sum').sort_values(by='total_sales', ascending=False)
```

Out[22]:

genre	total_sales
action	780.67
shooter	510.81
sports	442.27
role-playing	344.41
misc	307.77
platform	157.02
racing	153.58
fighting	109.02
simulation	102.78
adventure	77.41
strategy	47.00
puzzle	40.38

It is quite expected that sales for these genres will be high.

However, the shooter genre still has high sales, despite the fact that the number of games in this genre is not so high. This indicates the popularity of the shooter genre, more copies are sold

Are high and low selling genres stand out?

Outsiders in sales - puzzle and strategy games

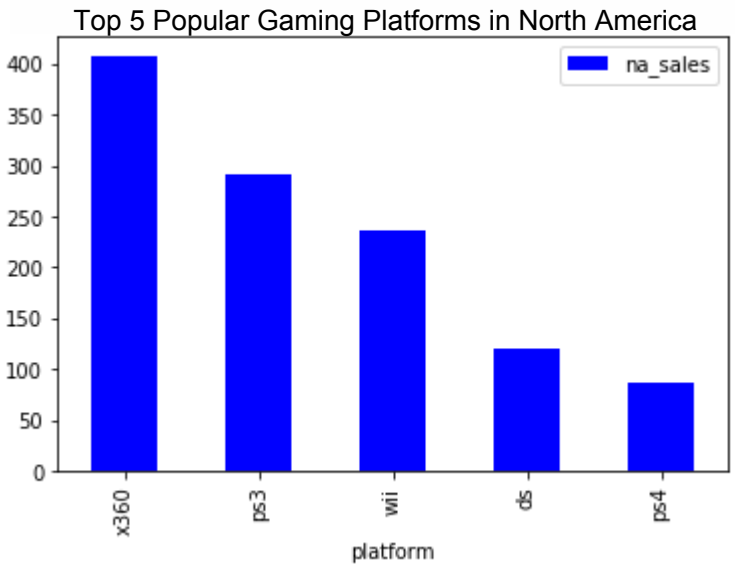
Reviewer's comment

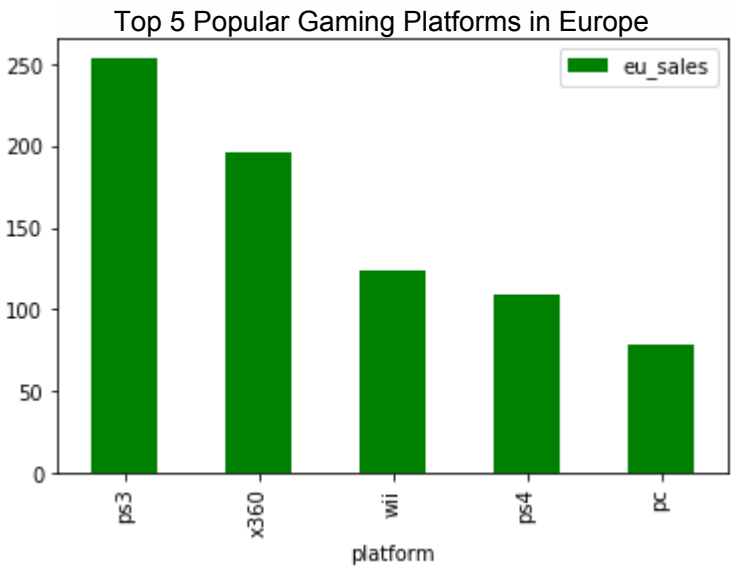
An analysis of the popularity of genres has been carried out. However, do not forget that the production of games in the genres of Action or Shooter is much more expensive than the production of Puzzle games.

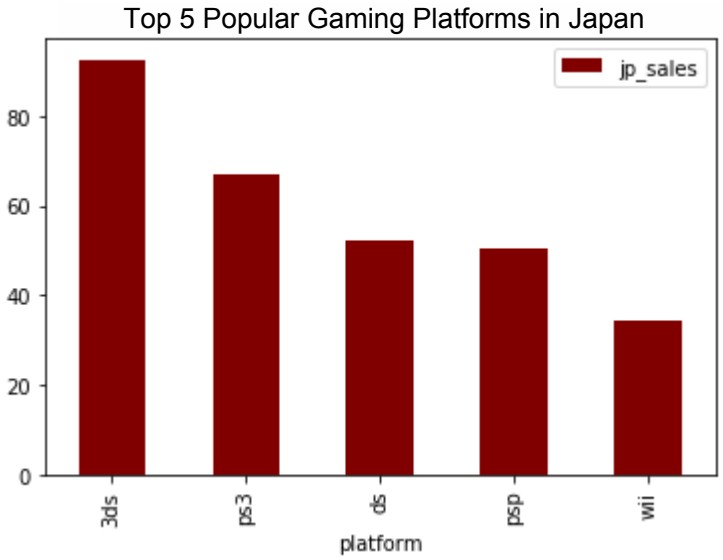
Step 5. Make a portrait of the user of each region

Define for each user region (NA, EU, JP):
Most popular platforms (top 5). Describe the differences in sales shares

In [23]:
```
games_data.groupby(by='platform').agg({'na_sales':'sum'}).sort_values(by='na_s
ales', ascending=False).head(5).plot(kind='bar', color='blue', legend=True)
plt.title('Top 5 Popular Gaming Platforms in North America')
plt.show()
games_data.groupby(by='platform').agg({'eu_sales':'sum'}).sort_values(by='eu_s
ales', ascending=False).head(5).plot(kind='bar', color='green', legend=True)
plt.title('Top 5 Popular Gaming Platforms in Europe')
plt.show()
games_data.groupby(by='platform').agg({'jp_sales':'sum'}).sort_values(by='jp_s
ales', ascending=False).head(5).plot(kind='bar', color='maroon', legend=True)
plt.title('Top 5 Popular Gaming Platforms in Japan')
plt.show()
```


Top 5 Popular Gaming Platforms in North America
na_sales
400
350
300
250
200
150
100
50
0
x360
ps3
wii
ds
ps4
platform

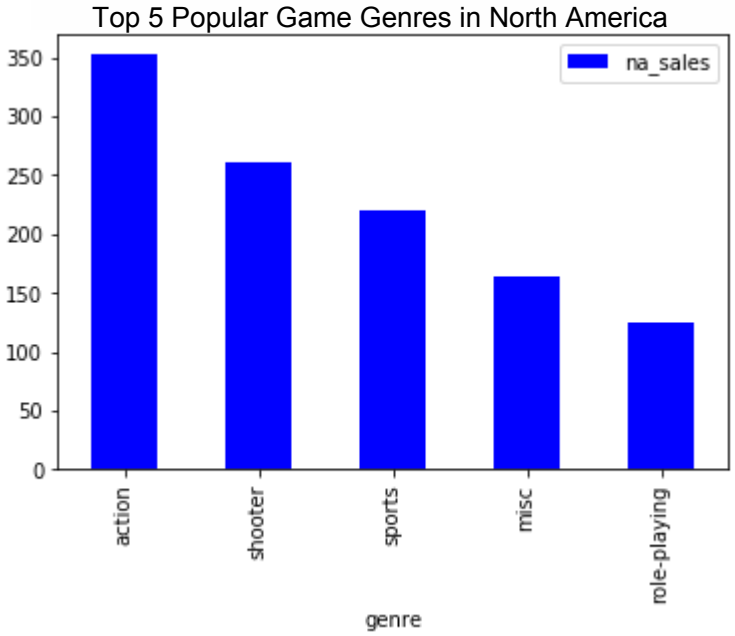

Top 5 Popular Gaming Platforms in Europe
eu_sales
250
200
150
100
50
0
ps3
x360
wii
ps4
pc
platform

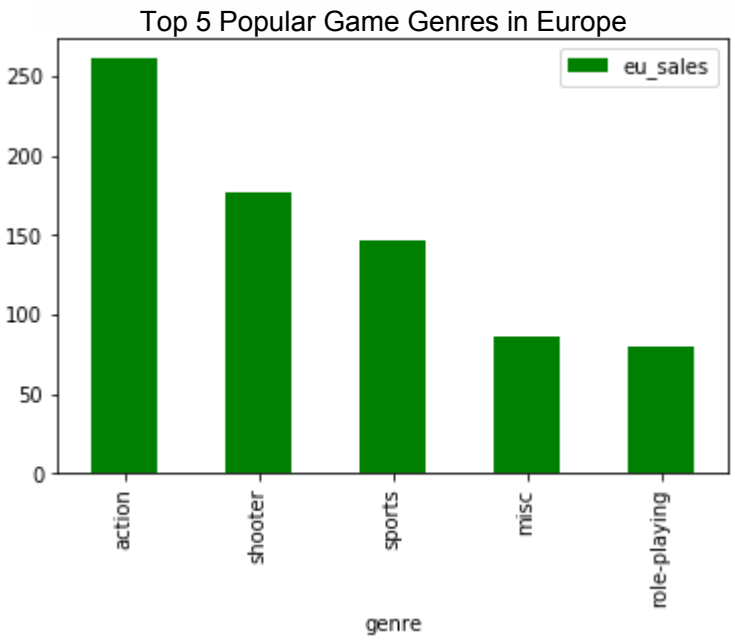

Top 5 Popular Gaming Platforms in Japan
jp_sales
80
60
40
20
0
3ds
ps3
ds
psp
wii
platform

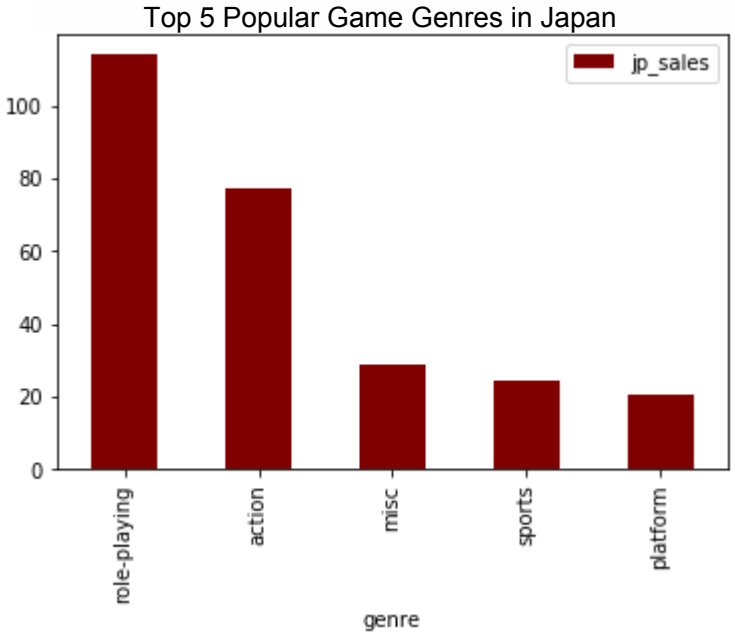
North American share of sales is the largest, in the observed period players preferred the Xbox 360 and PS3
The share of sales in the European market is almost 2 times less than the North American one, the top in popularity is the same platforms, only in reverse order (PS3 and Xbox 360)
In Japan, the market is even smaller, and the most popular console is Nintendo 3DS, released in 2011 Most likely because the Japanese prefer portable consoles to play on the go or on the go

Most popular genres (top 5). Explain the difference

In [24]:
```
games_data.groupby(by='genre').agg({'na_sales':'sum'}).sort_values(by='na_sales', ascending=False).head(5).plot(kind='bar', color='blue', legend=True)
plt.title('Top 5 Popular Game Genres in North America')
plt.show()
games_data.groupby(by='genre').agg({'eu_sales':'sum'}).sort_values(by='eu_sales', ascending=False).head(5).plot(kind='bar', color='green', legend=True)
plt.title('Top 5 Popular Game Genres in Europe')
plt.show()
games_data.groupby(by='genre').agg({'jp_sales':'sum'}).sort_values(by='jp_sales', ascending=False).head(5).plot(kind='bar', color='maroon', legend=True)
plt.title('Top 5 Popular Game Genres in Japan')
plt.show()
```


Top 5 Popular Game Genres in North America
na_sales
350
300
250
200
150
100
50
0
action
shooter
sports
misc
role-playing
genre


Top 5 Popular Game Genres in Europe
eu_sales
250
200
150
100
50
0
action
shooter
sports
misc
role-playing
genre



Action and shooter genres are preferred in North America and Europe

Whereas in Japan, RPGs are the most popular (well, in other words, they are not there like in the rest of the world)

Firstly, the Japanese gaming market is one of the most ancient, so the average age of players can often reach 40-50 years.

Secondly, as can be seen from the conclusions above, these same users have been playing on Nintendo since the late 80s and continue to play on Nintendo of the new generation.

Third, the mentality of the Japanese is very different from that of an American or a European.

They have their own completely unique culture, with their own traditions and characteristics. Take at least the same cult of anime, manga and comics, slot machines and other Japanese "jokes".

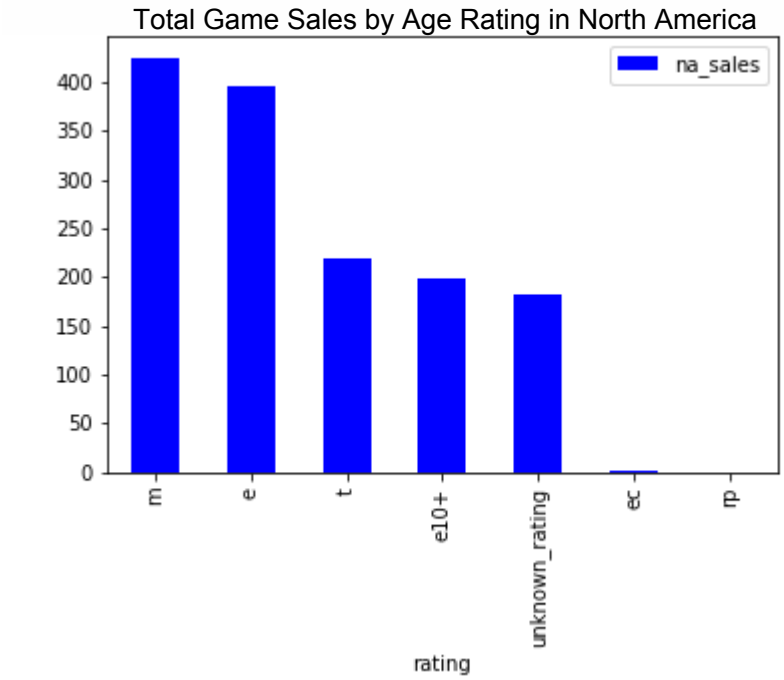
The Japanese are more peaceful and prefer harmony in everything. Therefore, it is noticeable that shooters are not at all interesting to them, unlike users of other considered markets.

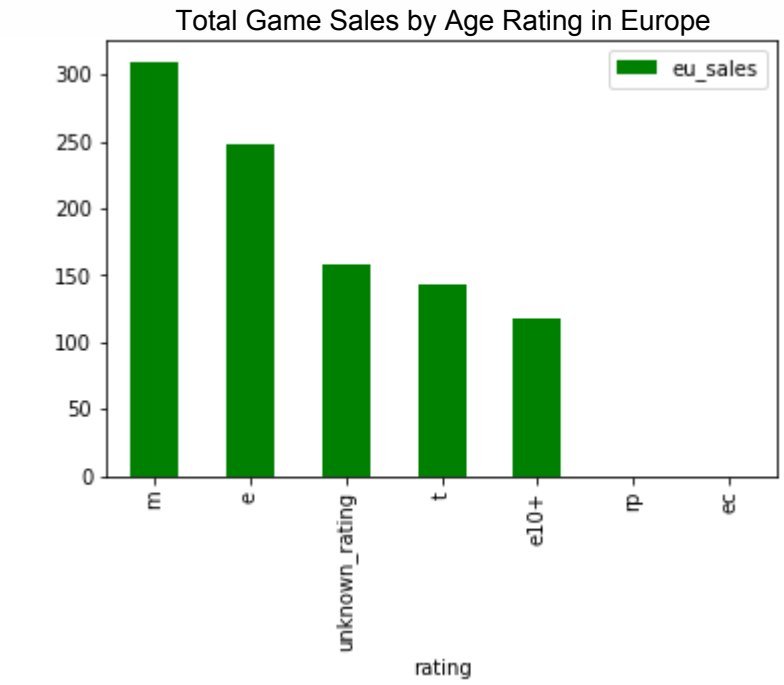
Reviewer's comment

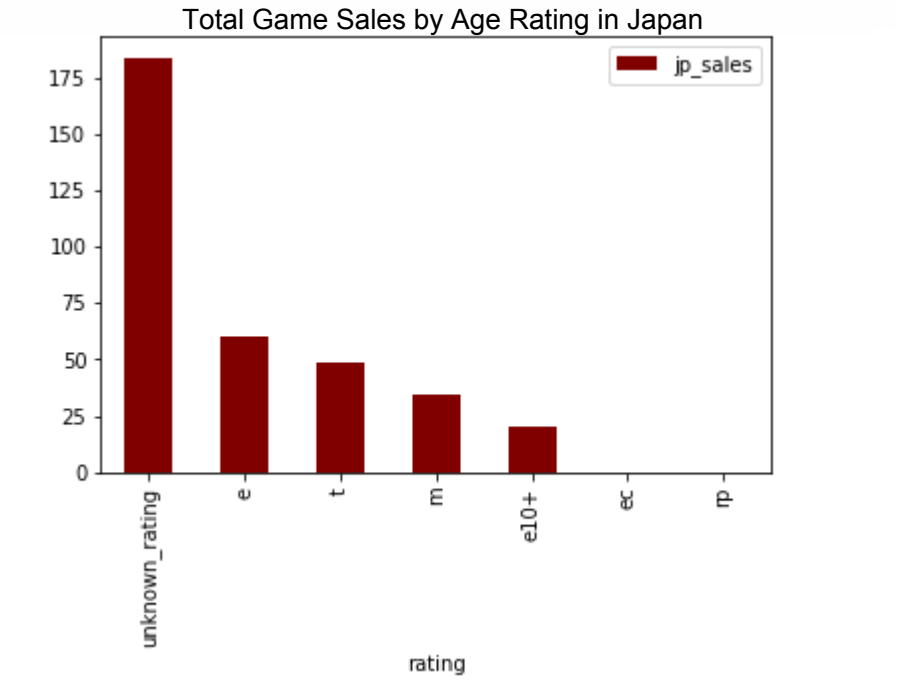
What makes the Japanese game market so different from others?

Does the ESRB rating affect sales in a particular region?

In [25]:
```
games_data.groupby(by='rating').agg({'na_sales':'sum'}).sort_values(by='na_sales', ascending=False).plot(kind='bar', color='blue', legend=True)
plt.title('Total Game Sales by Age Rating in North America')
plt.show()
games_data.groupby(by='rating').agg({'eu_sales':'sum'}).sort_values(by='eu_sales', ascending=False).plot(kind='bar', color='green', legend=True)
plt.title('Total Game Sales by Age Rating in Europe')
plt.show()
games_data.groupby(by='rating').agg({'jp_sales':'sum'}).sort_values(by='jp_sales', ascending=False).plot(kind='bar', color='maroon', legend=True)
plt.title('Total Game Sales by Age Rating in Japan')
plt.show()
```


Total Game Sales by Age Rating in North America
na_sales
400
350
300
250
200
150
100
50
0
m
e
t
e10+
unknown_rating
ec
rp
rating


Total Game Sales by Age Rating in Europe
eu_sales
300
250
200
150
100
50
0
m
e
unknown_rating
t
e10+
rp
ec
rating



Reviewer's comment
Cool, you noticed an important feature of the data - most of the games in the Japanese region are unrated. Why do you think we get this result? Unloading is not to blame)

In America and Europe, the situation is about the same, the most popular games are in category E (for everyone), then - games with an unspecified rating

In Japan, on the contrary, in 1st place - the rating is not specified

It is possible that the most popular games for the Japanese market are produced by Japan itself, and are not particularly popular in the rest of the world, therefore they do not have an international rating.

Reviewer's comment
A portrait of a typical user for each region is obtained. All necessary graphs are provided. It's great that the individual characteristics of each region are noted.

Step 6. Conduct a study of statistical indicators
How do user ratings and critics ratings change across genres?

In [26]:
```
user_genre_pivot = games_data.pivot_table(index='genre',values='user_score', aggfunc = 'mean')
print('Average user ratings by game genre')
print(user_genre_pivot.sort_values(by='user_score', ascending=False))
```

```
Average user ratings by game genre
              user_score
genre
role-playing    7.311296
platform        7.225000
puzzle          7.145000
adventure       7.110169
fighting        6.969032
action          6.835253
strategy        6.752475
misc            6.692105
shooter         6.533521
racing          6.459880
simulation      6.325510
sports          6.173829
```

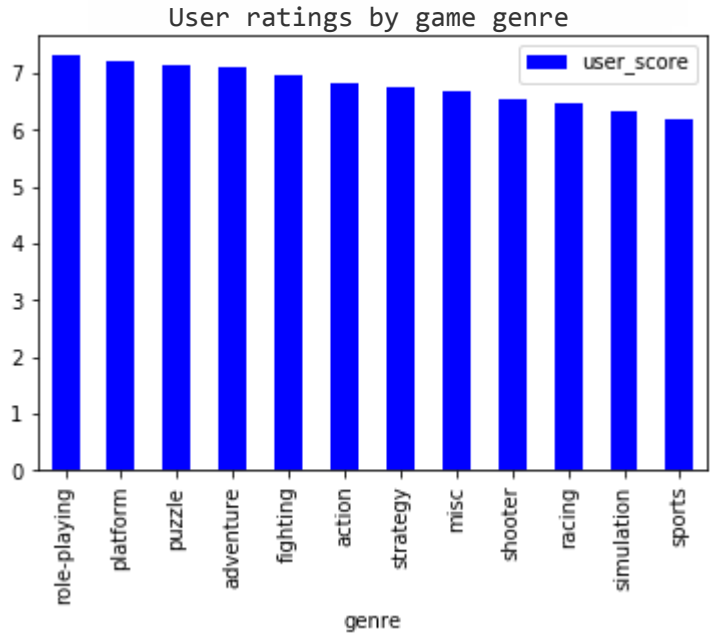
The average user rating for all genres is in the range of 6.2-7.3 (1-10)

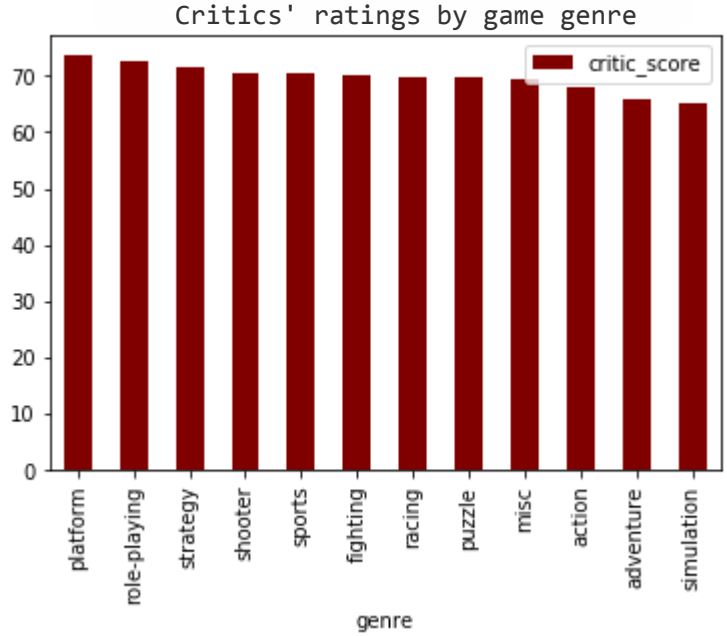
In [27]:
```
critic_genre_pivot = games_data.pivot_table(index='genre',values='critic_score', aggfunc = 'mean')
print('Average Critic Ratings by Game Genre')
print(critic_genre_pivot.sort_values(by='critic_score', ascending=False))
```

```
Average Critic Ratings by Game Genre
              critic_score
genre
platform         73.560748
role-playing     72.434483
strategy         71.585106
shooter          70.597598
sports           70.465190
fighting         70.096552
racing           69.833333
puzzle           69.563636
misc             69.227273
action           67.901985
adventure        65.951220
simulation       65.041667
```

The average critics' rating for all genres is in the range of 65-73 (1-100)
What indicates the similarity of the ratings data

In [28]:
```
games_data.groupby(by='genre').agg({'user_score':'mean'}).sort_values(by='user
_score', ascending=False).plot(kind='bar', color='blue', legend=True)
plt.title('User ratings by game genre')
plt.show()

games_data.groupby(by='genre').agg({'critic_score':'mean'}).sort_values(by='cr
itic_score', ascending=False).plot(kind='bar', color='maroon', legend=True)
plt.title('Critics' ratings by game genre')
plt.show()
```





Users are more willing to give high scores to the RPG, Platform, Puzzle genres
Critics are more willing to give high scores for almost the same genres - Platform, RPG, Strategy

Calculate the mean, variance and standard deviation

In [29]:
```
games_data_genre_score = pd.DataFrame()
games_data_genre_score['genre'] = games_data['genre']
games_data_genre_score['critic_score'] = games_data['critic_score']
games_data_genre_score['user_score'] = games_data['user_score']

genre_list = ['action', 'adventure', 'fighting', 'misc', 'platform', 'puzzle',
'racing', 'role-playing', 'shooter', 'simulation', 'sports', 'strategy']
for genre in genre_list:
    variance_estimate =
np.var(games_data_genre_score[games_data_genre_score[ 'genre'] == genre],
ddof=1)

print('Dispersions by genre:', genre)
print(variance_estimate)
print('Standard deviation by genre:', genre)
print(np.sqrt(variance_estimate))
print('-----------------------')
```

```
Dispersions by genre: action
critic_score    188.620195
user_score        1.812026
dtype: float64
Standard deviation by genre: action
critic_score    13.733907
user_score       1.346115

dtype: float64 -----------------------
Dispersions by genre: adventure
critic_score    228.030388
user_score        2.697161
dtype: float64
Standard deviation by genre: adventure
critic_score    15.100675
user_score       1.642303

dtype: float64 -----------------------
Dispersions by genre: fighting
critic_score    204.65728
user_score        1.82332
dtype: float64
Standard deviation by genre: fighting
critic_score    14.305848
user_score       1.350304
dtype: float64
-----------------------
Dispersions by genre: misc
critic_score    170.907476
user_score        2.366128
dtype: float64
Standard deviation by genre: misc
critic_score    13.073159
user_score       1.538222

dtype: float64 -----------------------
Dispersions by genre: platform
critic_score    191.022218
user_score        2.314505
dtype: float64
Standard deviation by genre: platform
critic_score    13.821079
user_score       1.521350
dtype: float64
-----------------------
Dispersions by genre: puzzle
critic_score    119.546801
user_score        2.045103
dtype: float64
Standard deviation by genre: puzzle
critic_score    10.933746
user_score       1.430071
dtype: float64
-----------------------
Dispersions by genre: racing
critic_score    199.900200
user_score        2.296754
```

```
dtype: float64
Standard deviation by genre: racing
critic_score    14.138607
user_score       1.515505

dtype: float64 -----------------------
Dispersions by genre: role-playing
critic_score    152.156592
user_score        1.417805
dtype: float64
Standard deviation by genre: role-playing
critic_score    12.335177
user_score       1.190716

dtype: float64 -----------------------
Dispersions by genre: shooter
critic_score    224.337585
user_score        2.404834
dtype: float64
Standard deviation by genre: shooter
critic_score    14.977903
user_score       1.550753

dtype: float64 -----------------------
Dispersions by genre: simulation
critic_score    165.050877
user_score        3.357384
dtype: float64
Standard deviation by genre: simulation
critic_score    12.847213
user_score       1.832317

dtype: float64 -----------------------
Dispersions by genre: sports
critic_score    226.154340
user_score        2.712877
dtype: float64
Standard deviation by genre: sports
critic_score    15.038429
user_score       1.647081

dtype: float64 -----------------------
Dispersions by genre: strategy
critic_score    150.976550
user_score        2.646919
dtype: float64
Standard deviation by genre: strategy
critic_score    12.287252
user_score       1.626935
dtype: float64
-----------------------
```

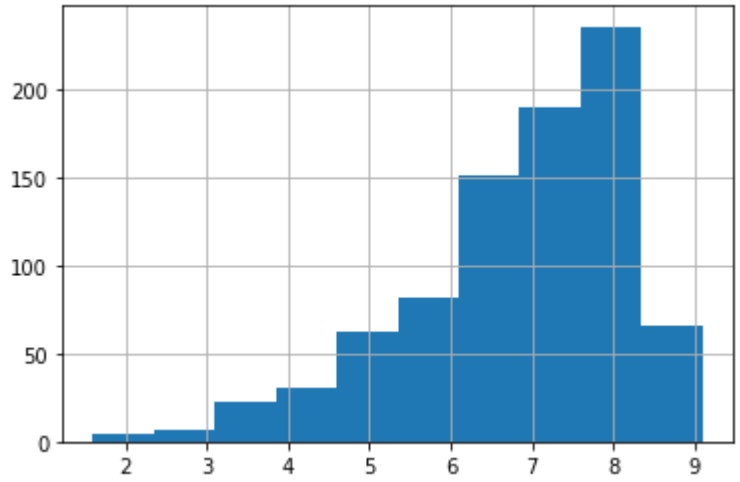
Overall, Critical Review Standard Deviation across all genres is around 10-15 points
Minimum values for RPG and Puzzle genres
Maximum in Adventure and Sports genres

The standard deviation of user reviews across all genres is also about the same - about 1.5
The minimum outstanding value for the RPG genre, minimum distribution spread

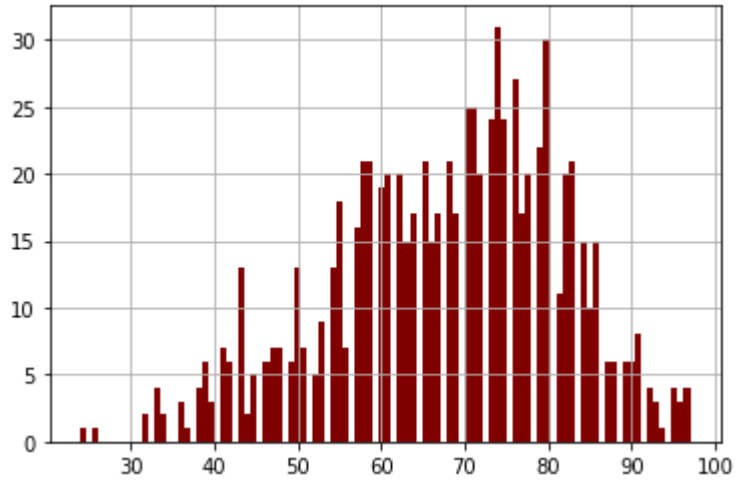
Plot histograms. Describe the distribution

In [30]:
```
for genre in genre_list:
    print('Density of distribution of user ratings by genre:', genre)
    games_data[games_data['genre'] == genre]['user_score'].hist(bins=10)
    plt.show()
    print('-------------------------------------------------------')
    print('Density of distribution of critics' ratings by genre:', genre)
    games_data[games_data['genre'] == genre]['critic_score'].hist(bins=100,
    color='maroon')
    plt.show()
    print('-------------------------------------------------------')
```

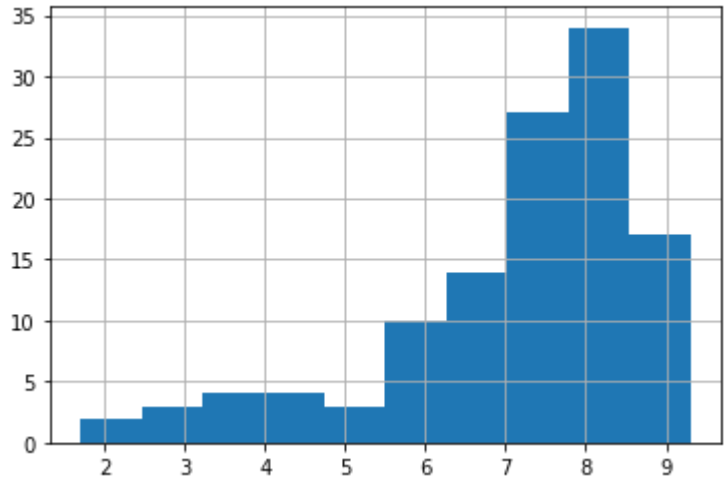
Density of distribution of user ratings by genre: action



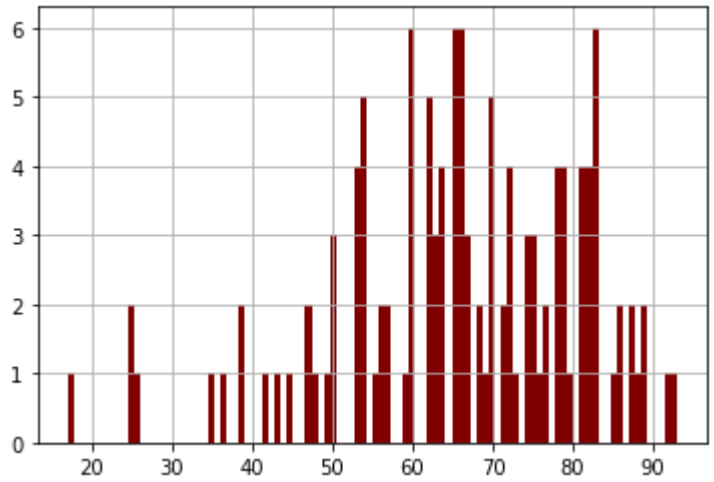
Density of distribution of critics' ratings by genre: action



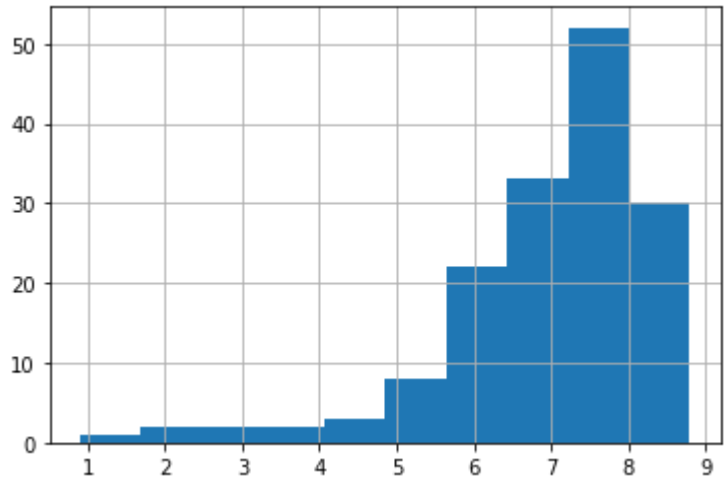
Density of distribution of user ratings by genre: adventure



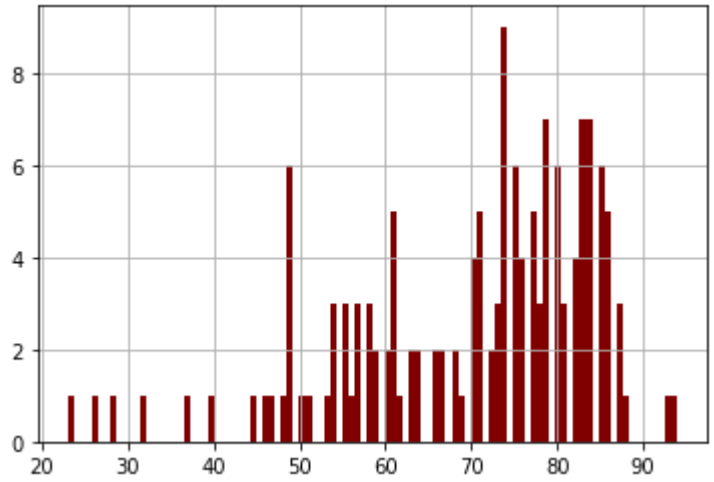
Density of distribution of critics' ratings by genre: adventure



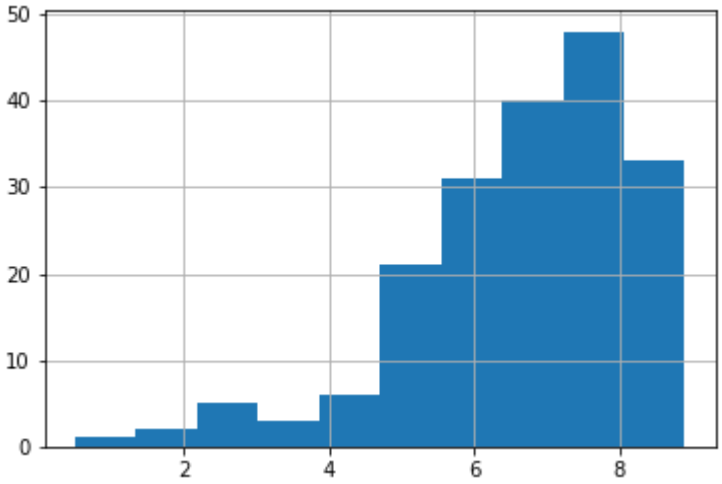
Density of distribution of user ratings by genre: fighting



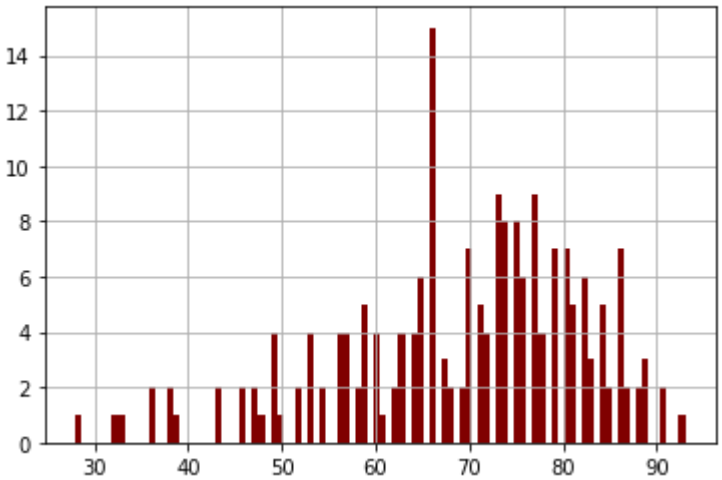
Density of distribution of critics' ratings by genre: fighting



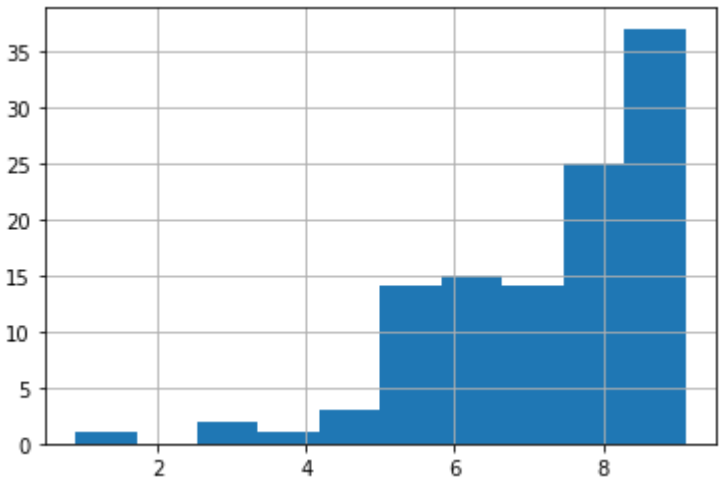
Density of distribution of user ratings by genre: misc



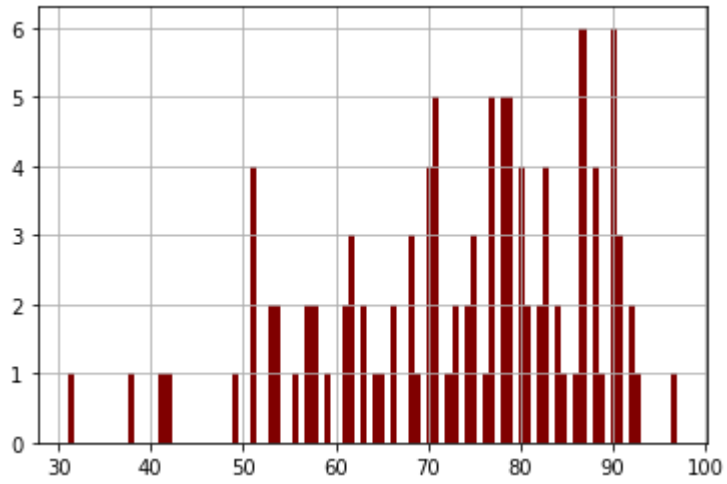
Density of distribution of critics' ratings by genre: misc



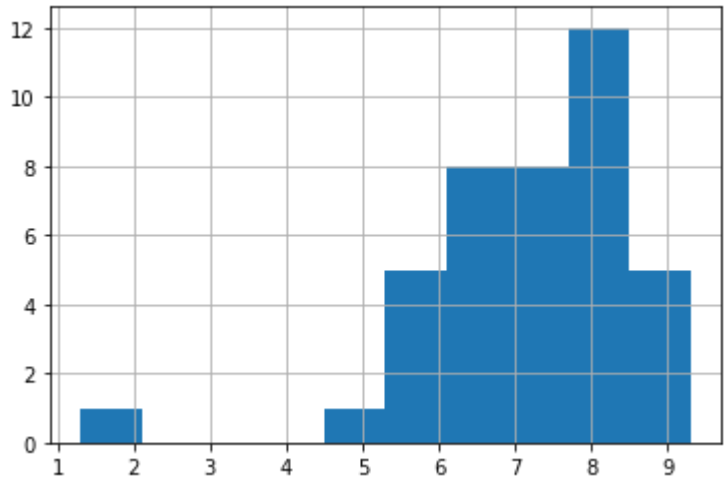
Density of distribution of user ratings by genre: platform



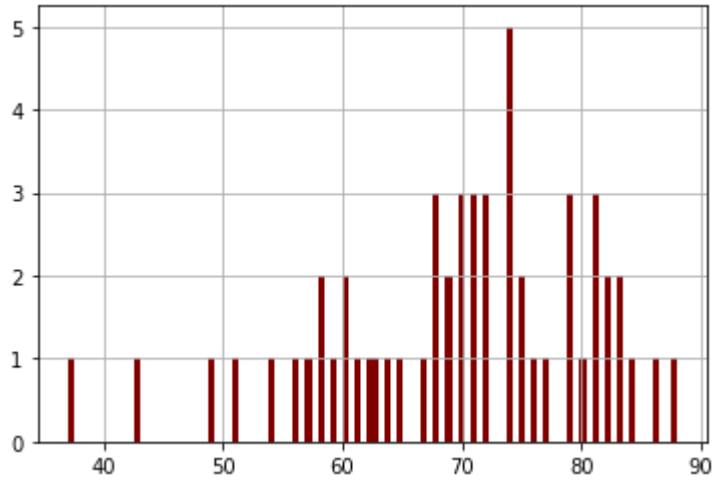
Density of distribution of critics' ratings by genre: platform



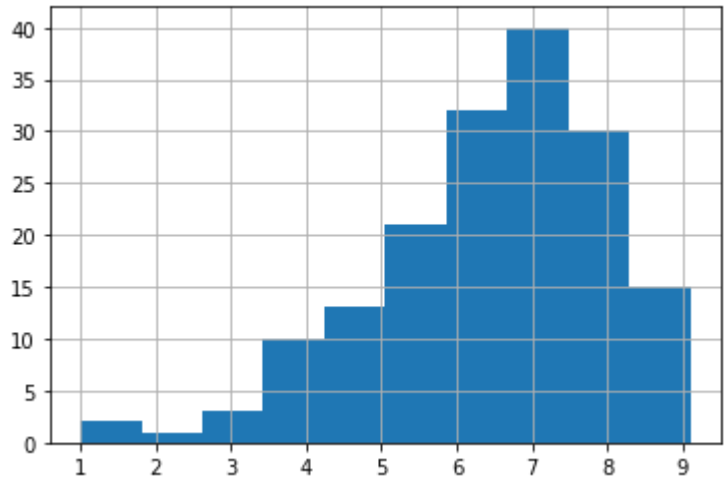
Density of distribution of user ratings by genre: puzzle



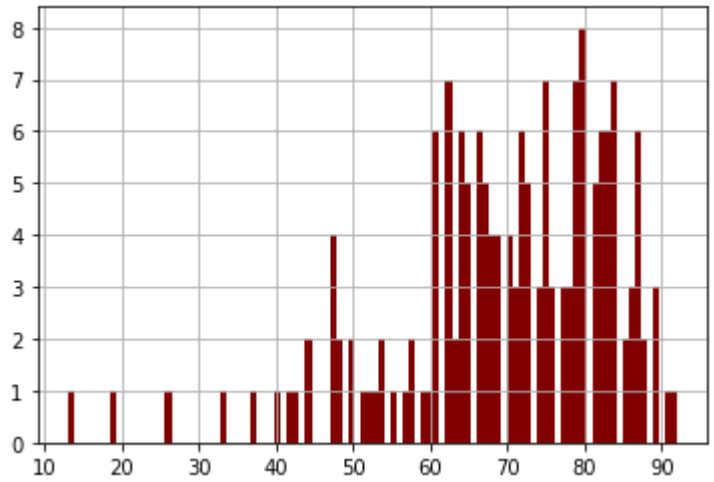
Density of distribution of critics' ratings by genre: puzzle



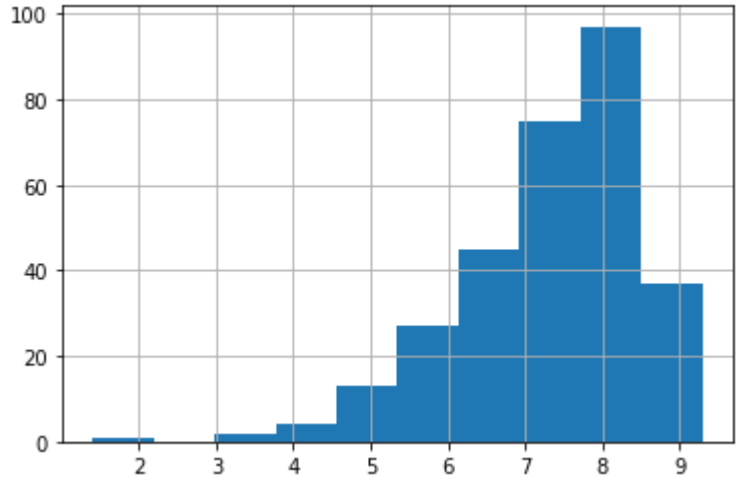
Density of distribution of user ratings by genre: racing



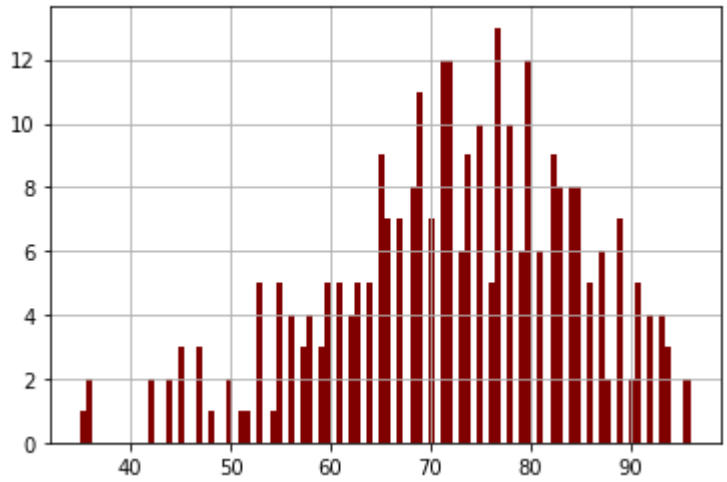
Density of distribution of critics' ratings by genre: racing



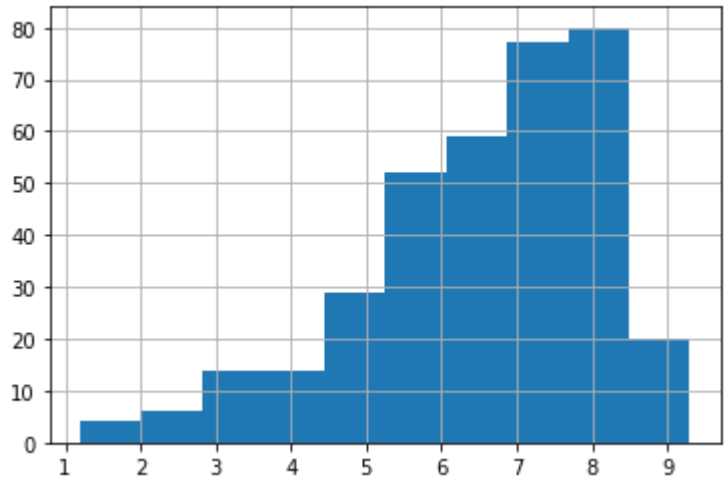
Density of distribution of user ratings by genre: role-playing



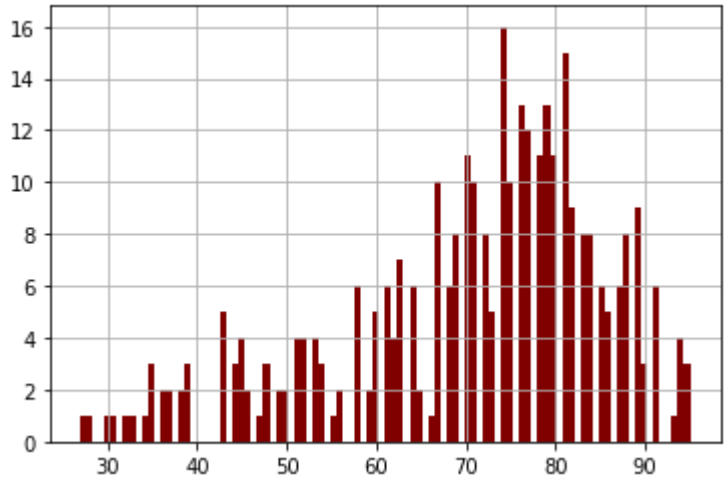
Density of distribution of critics' ratings by genre: role-playing



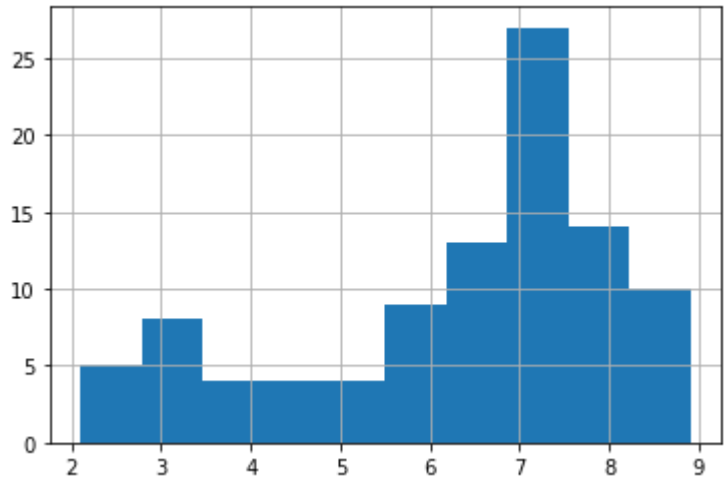
Density of distribution of user ratings by genre: shooter



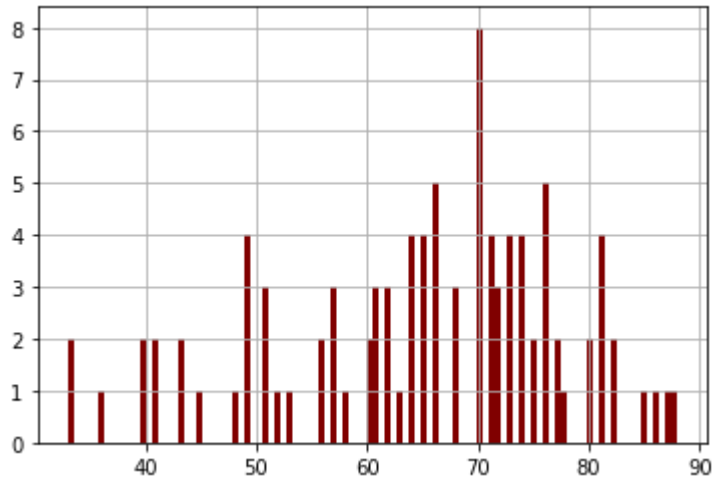
Density of distribution of critics' ratings by genre: shooter



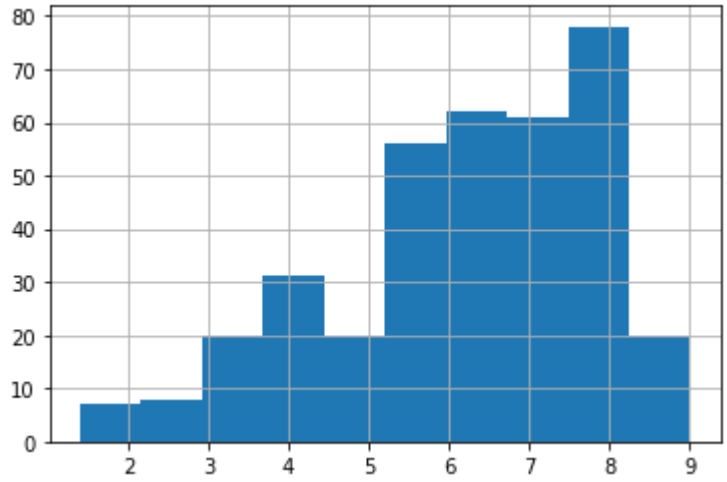
Density of distribution of user ratings by genre: simulation



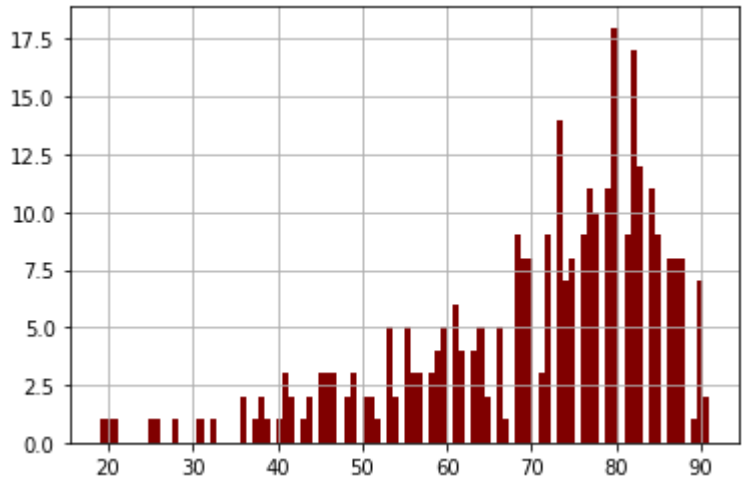
Density of distribution of critics' ratings by genre: simulation



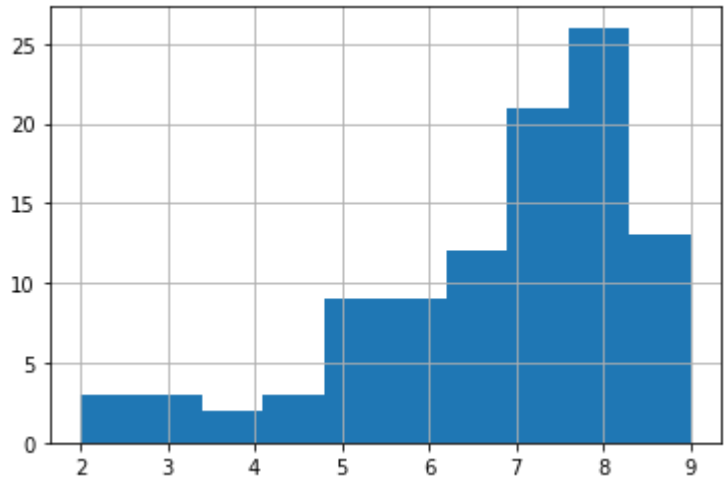
Density of distribution of user ratings by genre: sports



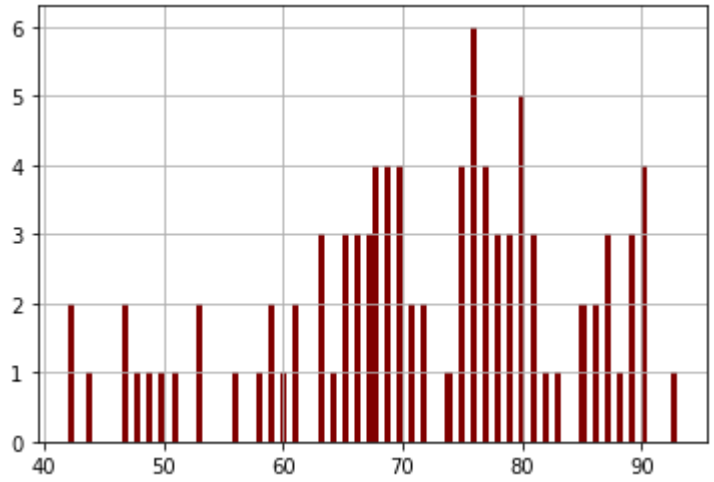
Density of distribution of critics' ratings by genre: sports



Density of distribution of user ratings by genre: strategy



Density of distribution of critics' ratings by genre: strategy



The histograms by both users and critics across all genres are similar in that they are negatively skewed due to the average rating above 5 out of 10.

Commentary by the stat reviewer

Indicators are calculated. Besides them, they ask us also construct and histograms based on reviews of critics and users for all considered genres. Please complete this section.

Reviewer's comment 2

The required graphs are given. What do you think, Why do we get left-skewed distributions for most genres?

Step 7. Test hypotheses

The average user ratings for the Xbox One and PC platforms are the same:
H0: Average user ratings for Xbox One and PC platforms are the same
H1: Average user ratings for Xbox One and PC platforms vary

Set your own threshold value alpha:
0.05 is the standard value for this kind of research (not too hard)

In [31]:
```
alpha = 0.05

xbox_one = games_data[(games_data['platform'] == 'xone') & (games_data['user_score'] > 0)]['user_score']
pc = games_data[(games_data['platform'] == 'pc') & (games_data['user_score'] > 0)]['user_score']

results = st.ttest_ind(xbox_one, pc)

print('p-value:', results.pvalue)

if (results.pvalue < alpha):
    print("Rejecting the null hypothesis")
else:
    print("Failed to reject the null hypothesis")
```

```
p-value: 0.1708077691700714
Failed to reject the null hypothesis
```

This means that we could not reject the hypothesis that the average user ratings for the Xbox One and PC platforms are the same.
Yet this does not mean that we have proven that these ratings are the same.
It's just that our sample looks like these ratings are similar.
It often happens that on one console the game turns out to be successful without serious bugs, while on another platform the main bugs may not be fixed for years
But, apparently, the quality of games on PC and Xbox One is about the same

Reviewer's commentary

The hypotheses are formulated and verified correctly.

The average user ratings of the Action and Sports genres are different:

H0: Average user ratings for Action and Sports genres are the same
H1: Average user ratings for Action and Sports are different

In [32]:
```
alpha = 0.05

action = games_data[(games_data['genre'] == 'action') & (games_data['user_score'] > 0)]['user_score']
sports = games_data[(games_data['genre'] == 'sports') & (games_data['user_score'] > 0)]['user_score']

results = st.ttest_ind(action, sports)

print('p-value:', results.pvalue)

if (results.pvalue < alpha):
    print("Rejecting the null hypothesis")
else:
    print("Failed to reject the null hypothesis")
```

```
p-value: 4.707667517418828e-13
Rejecting the null hypothesis
```

We reject the hypothesis that the average user ratings for the Action and Sports genres are the same. In our sample, it turns out that users, on average, give different ratings to games in the Action and Sports genres.

Reviewer's comment

The second hypothesis was also tested without complaints.

How did you formulate the null and alternative hypotheses:

H0 - always for equality, or for the absence of any changes

H1 - alternative (opposite)

What criterion was used to test hypotheses and why:

Student's t-test, because we are working with a sample, not a general population

Conclusions:

Games began publishing in the early 1980s.

But it took as long as 15-20 years of technology development to start mass production of games. The main peak in the release of games falls on 2008-2010.

This was followed by a decline in the gaming industry associated with the development of smartphones and tablets, which is why many users switched to mobile devices.

On average, the platform relevance cycle is 8-10 years.

Then the relevance of the platform disappears, or the next generation of the platform comes out.

There is one special console that remained relevant for an abnormally long period - for 28 years Nintendo did not let go of the players, mostly Japanese.

To build forecasts for 2017, we took data for the period of the industry downturn in 2009-2015.

It is noticeable that with the evolution of a specific platform, sales by games, as well as the cost of producing games, grow.

Most of the games were released in the Action, Miscellaneous and Sports genres.

In different markets, user preferences regarding gaming platforms and genres may differ.

For example, as already mentioned, the Japanese prefer Nintendo and RPGs the most.

In general, the Japanese market is very different from other markets, mainly due to all the early development since the 80s and the completely unique mentality of the Japanese.

Reviewer's comment

The final conclusion is the main result of your work. It is worth writing it in detail based on the results of the work done. It can be given values obtained during operation. You can also describe everything that was done in the work.

Reviewer's comment 2

Conclusions are well described. Answers to the main issues of the project. In the conclusions, you can give the previously obtained values. You can also describe everything that was done in the course of the work. It will be even better if you provide recommendations for the company on further actions.

Reviewer's comment

If you like the visualization topic, you can explore the methods of the seaborn library. It allows you to build pretty presentable graphs. You have completed all the work points, well done! There are few criticisms. However, it is important to work with them. There are also a fair number of yellow comments that are worth correcting. I think you can handle this quickly. Looking forward to your work :)

Reviewer's comment 2

Blots corrected and now the job is done OK. You've got a great project, well done. Congratulations on your completed project. I hope it was interesting and informative. Thanks for leaving comments on the fixes. Good luck in the future :)