

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
xhamster_data = pd.read_csv("xhamster.csv")
```

In [3]:

```
xhamster_data.head()
```

Out[3]:

	id	upload_date	title	channels	description	nb_views	nb_votes	nb_comr
0	378466	29-06-2010	girl riding black cock	['BBW', 'Black and Ebony', 'Interracial']	Like this vid? Check out my profile page for m...	17262.0	65.0	
1	478576	07-11-2010	masturbation	['Masturbation']	watching a ebony chick cum.	953.0	3.0	
2	287146	12-02-2010	sexy horny booty dance	['Babes', 'Teens', 'Webcams']	Watch as this sexy hot horny babe bounce her n...	6060.0	11.0	
3	378462	29-06-2010	group of young bareback sportsmen d	['Men']	NaN	12742.0	87.0	
4	1583073	18-11-2012	horny latinos double penetrating hot ass in a ...	['Gays']	Three Brazilian latino studs get horny in a st...	32879.0	75.0	

In [4]:



```
xhamster_data.tail()
```

Out[4]:

description	nb_views	nb_votes	nb_comments	runtime	uploader
THIS COLATE CTORY nGETS OPPE...	5842.0	21.0	4.0	269.0	9665e50e0d4683aa88cf78f5c6c095e2064568ea
NaN	64355.0	131.0	7.0	178.0	241d12c4cbd2dc2551199bb767bcc8041854f45c
NaN	26708.0	58.0	7.0	115.0	241d12c4cbd2dc2551199bb767bcc8041854f45c
NaN	17156.0	36.0	5.0	603.0	fd38c6065b9e7f545ec71be0acd6d1340783ec2e
NaN	24503.0	136.0	57.0	98.0	f50680c6576cfa93935a8b49dfa06dc90703192d



In [5]:



```
xhamster_data.shape
```

Out[5]:

```
(786121, 10)
```

In [6]:



```
xhamster_data.columns
```

Out[6]:

```
Index(['id', 'upload_date', 'title', 'channels', 'description', 'nb_views',  
      'nb_votes', 'nb_comments', 'runtime', 'uploader'],  
      dtype='object')
```

In [7]:



```
xhamster_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 786121 entries, 0 to 786120
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   id              786121 non-null  int64
 1   upload_date     785119 non-null  object
 2   title           785329 non-null  object
 3   channels        785119 non-null  object
 4   description     383373 non-null  object
 5   nb_views        785119 non-null  float64
 6   nb_votes        785119 non-null  float64
 7   nb_comments     726301 non-null  float64
 8   runtime         785119 non-null  float64
 9   uploader       753200 non-null  object
dtypes: float64(4), int64(1), object(5)
memory usage: 60.0+ MB
```

In [8]:



```
xhamster_data.describe()
```

Out[8]:

	id	nb_views	nb_votes	nb_comments	runtime
count	7.861210e+05	7.851190e+05	785119.000000	726301.000000	7.851190e+05
mean	9.292871e+05	7.550614e+04	127.861243	11.716061	6.749709e+02
std	5.022363e+05	1.733908e+05	240.880249	14.874591	3.371810e+03
min	3.000000e+00	1.000000e+00	0.000000	1.000000	0.000000e+00
25%	5.098190e+05	5.990000e+03	20.000000	3.000000	1.280000e+02
50%	9.421340e+05	2.299000e+04	56.000000	7.000000	3.590000e+02
75%	1.364660e+06	7.073600e+04	138.000000	15.000000	9.280000e+02
max	1.762464e+06	8.286734e+06	17171.000000	885.000000	2.752990e+06

In [9]:

```
xhamster_data.isnull().sum()
```

Out[9]:

```
id                0
upload_date      1002
title             792
channels         1002
description      402748
nb_views         1002
nb_votes         1002
nb_comments      59820
runtime          1002
uploader        32921
dtype: int64
```

In [10]:

```
xhamster_data = xhamster_data.drop(['id', 'upload_date', 'description',
                                     'nb_comments', 'uploader'], axis = 1)
```

In [11]:

```
xhamster_data.shape
```

Out[11]:

```
(786121, 5)
```

In [12]:

```
xhamster_data.columns
```

Out[12]:

```
Index(['title', 'channels', 'nb_views', 'nb_votes', 'runtime'], dtype='object')
```

In [13]:

```
xhamster_data.dropna(inplace = True)
```

In [14]:

```
xhamster_data.isnull().sum()
```

Out[14]:

```
title          0
channels       0
nb_views       0
nb_votes       0
runtime        0
dtype: int64
```

In [15]:



```
xhamster_data.shape
```

Out[15]:

```
(784328, 5)
```

In [16]:



```
xhamster_data['channels'].unique()
```

Out[16]:

```
array(["['BBW', 'Black and Ebony', 'Interracial']", "['Masturbation']",
      "['Babes', 'Teens', 'Webcams']", ...,
      "['Black and Ebony', 'Tits', 'Group Sex', 'Big Boobs', 'MILFs']",
      "['Anal', 'Bisexuals', 'Indian']",
      "['British', 'Squirting', 'Webcams']"], dtype=object)
```

In [17]:



```
xhamster_data['channels'].value_counts()
```

Out[17]:

['Men']	110374
['Gays', 'Men']	19735
['Amateur']	18558
['Shemales']	15040
['Gays']	14837
...	
['Amateur', 'Matures', 'Old+Young', 'Strapon']	1
['Latin ', 'Lingerie', 'Spanking']	1
['Cumshots', 'Face Sitting', 'Lingerie', 'Teens']	1
['Asian', 'Lesbians', 'Masturbation', 'Japanese']	1
['British', 'Squirting', 'Webcams']	1

Name: channels, Length: 54583, dtype: int64

In [22]:



```
xhamster_channels = xhamster_data['channels'].value_counts()
```

In [24]:



```
xhamster_channels.head()
```

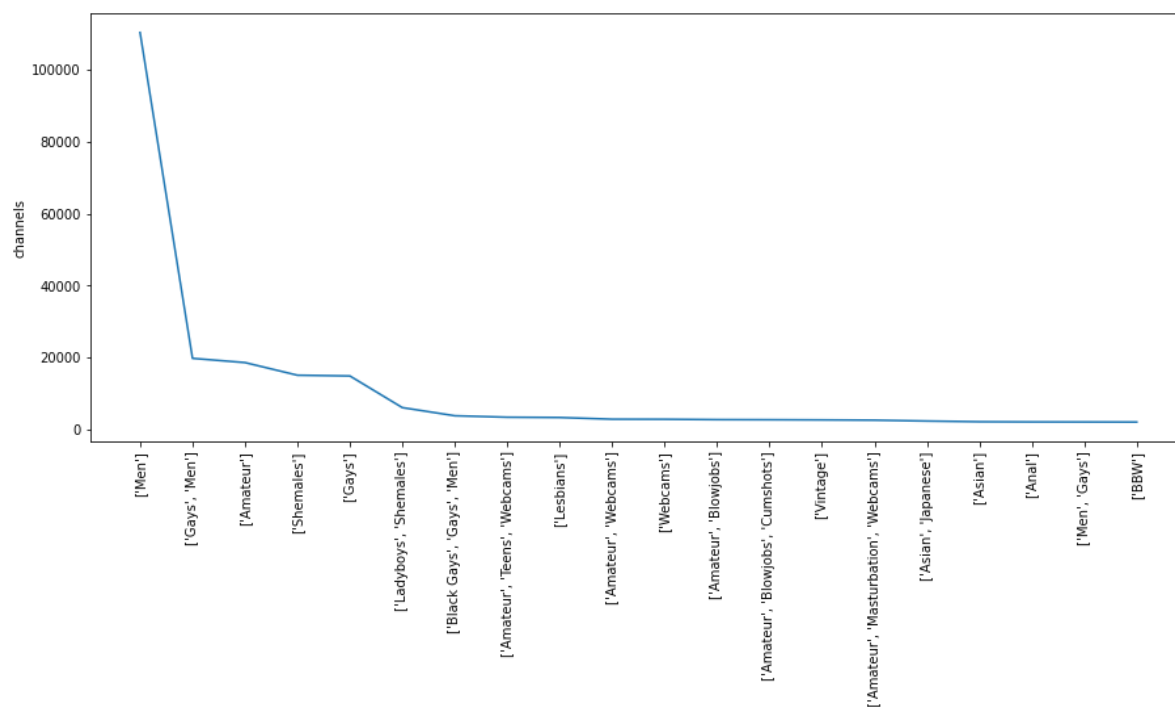
Out[24]:

['Men']	110374
['Gays', 'Men']	19735
['Amateur']	18558
['Shemales']	15040
['Gays']	14837

Name: channels, dtype: int64

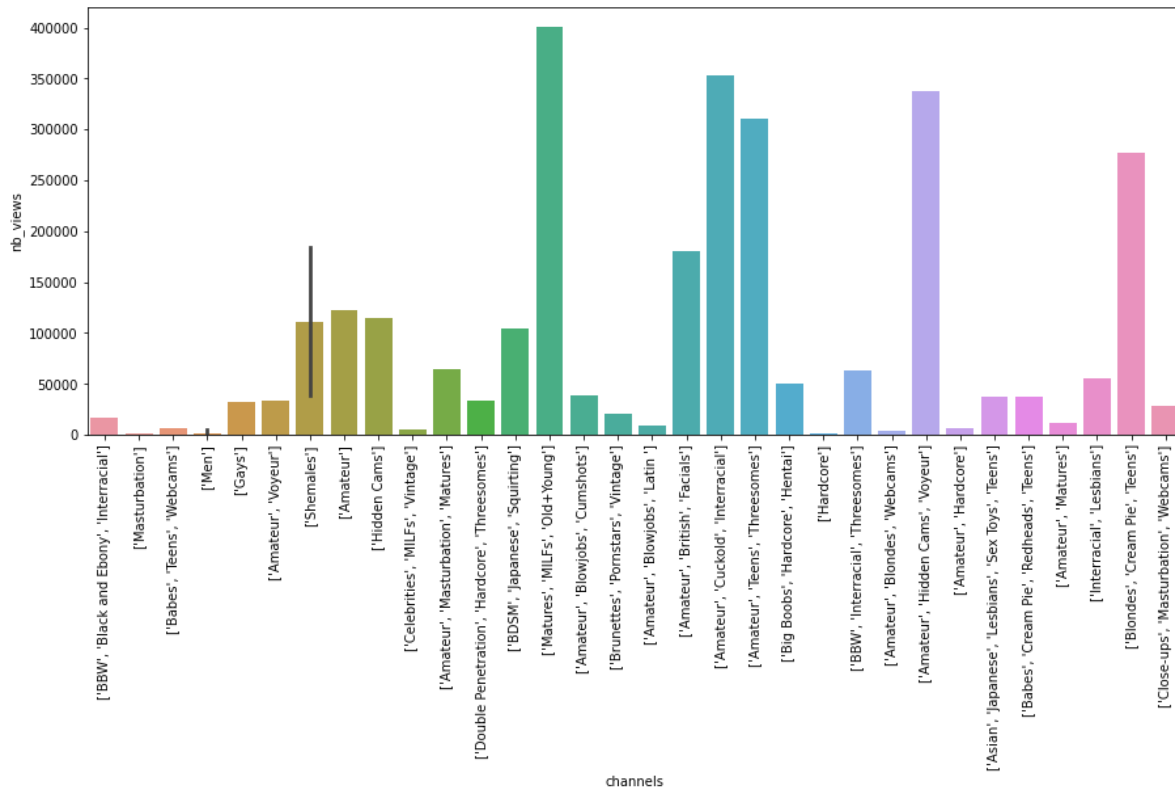
In [29]:

```
plt.figure(figsize=(15,6))  
sns.lineplot(data = xhamster_channels.head(20))  
plt.xticks(rotation = 90)  
plt.show()
```



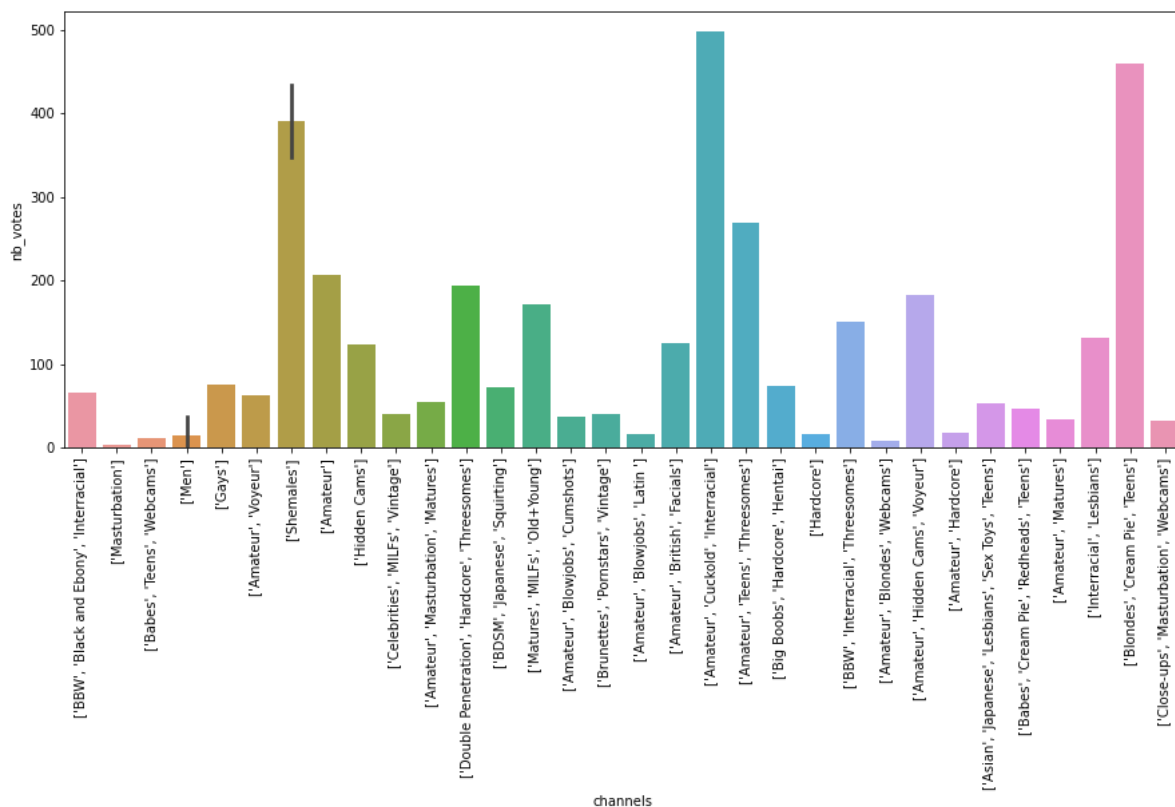
In [32]:

```
plt.figure(figsize=(15,6))
sns.barplot(x = 'channels', y = 'nb_views', data = xhamster_data.head(40))
plt.xticks(rotation = 90)
plt.show()
```



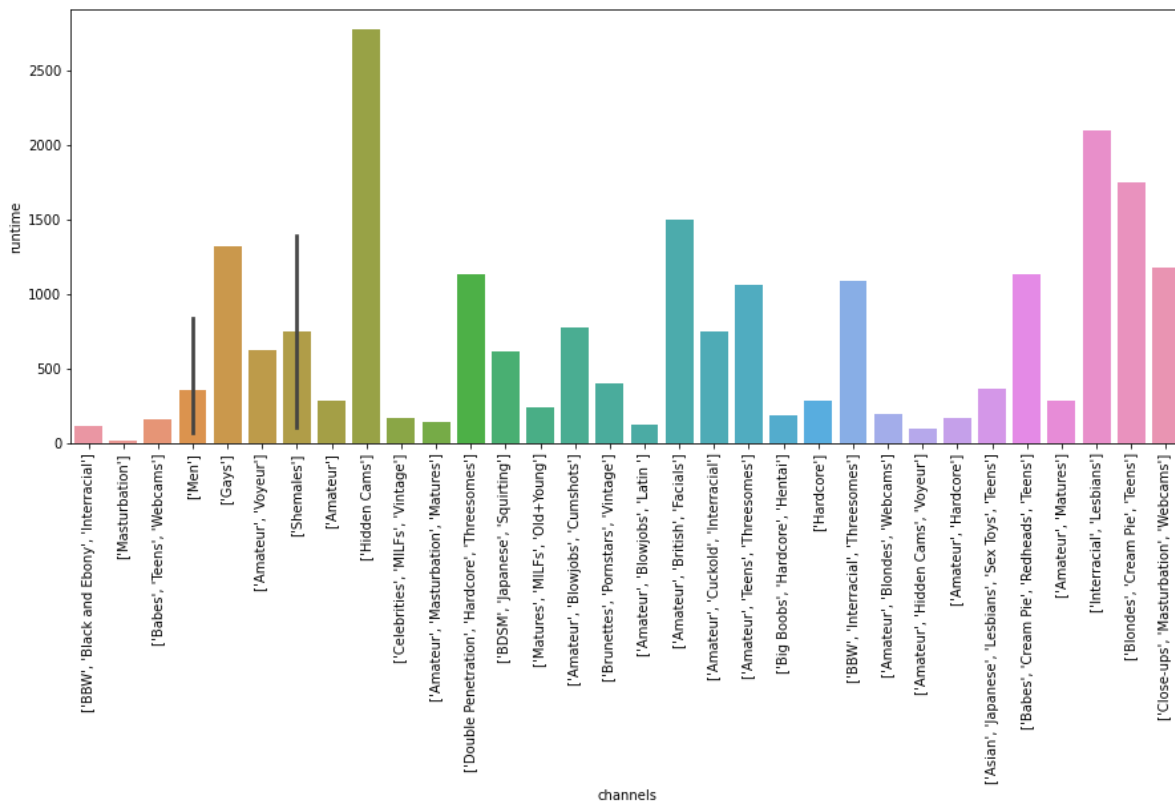
In [33]:

```
plt.figure(figsize=(15,6))
sns.barplot(x = 'channels', y = 'nb_votes', data = xhamster_data.head(40))
plt.xticks(rotation = 90)
plt.show()
```



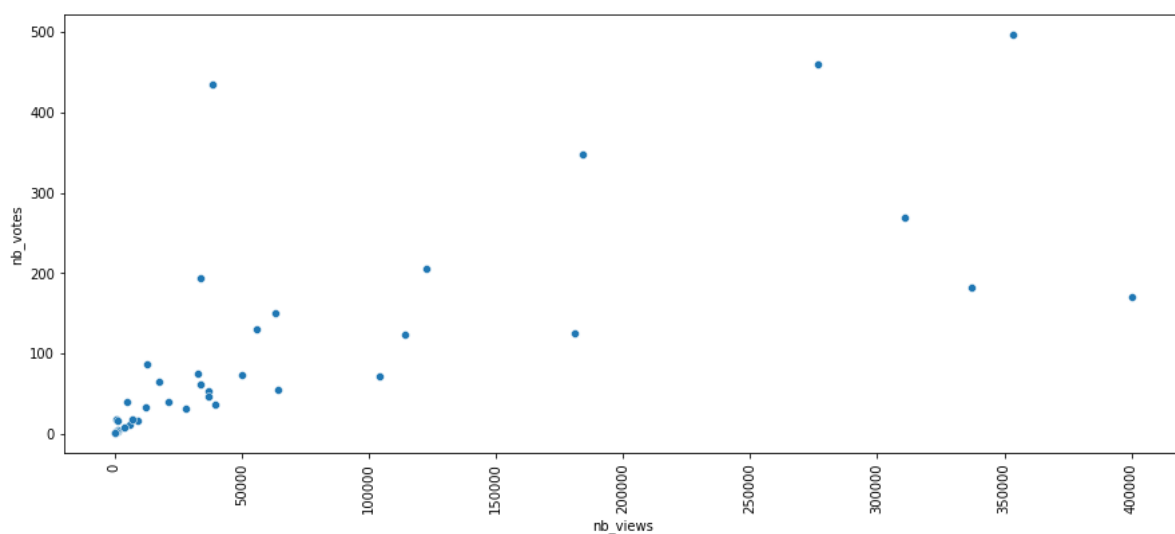
In [34]:

```
plt.figure(figsize=(15,6))
sns.barplot(x = 'channels', y = 'runtime', data = xhamster_data.head(40))
plt.xticks(rotation = 90)
plt.show()
```



In [35]:

```
plt.figure(figsize=(15,6))
sns.scatterplot(x = 'nb_views', y = 'nb_votes', data = xhamster_data.head(40))
plt.xticks(rotation = 90)
plt.show()
```



In [36]:

```
from sklearn import preprocessing
```

In [38]:

```
label_encoder = preprocessing.LabelEncoder()
```

In [39]:

```
xhamster_data['channels'] = label_encoder.fit_transform(xhamster_data['channels'])
```

In [40]:

```
xhamster_data.head()
```

Out[40]:

	title	channels	nb_views	nb_votes	runtime
0	girl riding black cock	21066	17262.0	65.0	120.0
1	masturbation	52799	953.0	3.0	15.0
2	sexy horny booty dance	26948	6060.0	11.0	163.0
3	group of young bareback sportsmen d	53178	12742.0	87.0	1980.0
4	horny latinos double penetrating hot ass in a ...	46749	32879.0	75.0	1318.0

In [41]:

```
x = xhamster_data.drop(['title', 'nb_views'], axis = 1)  
y = xhamster_data['nb_views']
```

In [42]:

```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.33)
```

In [45]:

```
from sklearn.linear_model import LinearRegression
```

In [46]:

```
model = LinearRegression()  
model.fit(X_train, y_train)
```

Out[46]:

```
LinearRegression()
```

In [47]:



```
y_pred = model.predict(X_test)
```

In [48]:



```
print("Training Accuracy :", model.score(X_train, y_train))  
print("Testing Accuracy :", model.score(X_test, y_test))
```

Training Accuracy : 0.7362646718372391

Testing Accuracy : 0.7381458835107014