In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from IPython import get_ipython
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
data = pd.read_csv('tumbzilla_labels.csv')
```

In [3]:

```
data.head()
```

Out[3]:

	Unnamed: 0	img_source	length	nb_views	quality	title	
0	0	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	1583	127450	LOW	Tease then suck	http://tl
1	1	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	2501	480620	LOW	Two hot chick on one dick	http://tl
2	2	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	1513	99720	LOW	Chick stretches her pussy and blows for the cam	http://thu
3	3	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	1710	598820	LOW	Fucking my neighbour	http://th
4	4	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	1694	155850	LOW	Small titted whore sucking a cock for lunch	http://thu

In [4]:

```
data.tail()
```

Out[4]:

	Unnamed: 0	img_source	length	nb_views	quality	title	
191527	191527	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	2774	257540	LOW	Czech taxi 39	
191528	191528	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	528	73960	LOW	Casanova Holmes - Quality Vintage 1970s XXX	h
191529	191529	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	672	419960	HD	19 Year Old Blonde Fuck in a threesome	h
191530	191530	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	535	216150	LOW	Virtual Sex - Anya Ivy	
191531	191531	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	460	201700	HD	Brazzers - Ebony teen Peyton Sweet loves white...	

In [5]:

```
data.shape
```

Out[5]:

```
(191532, 10)
```

In [6]:

```
data.columns
```

Out[6]:

```
Index(['Unnamed: 0', 'img_source', 'length', 'nb_views', 'quality', 'title',
       'video_link', 'voting', 'categories', 'tags'],
      dtype='object')
```

In [7]:

```
data = data.drop('Unnamed: 0', axis = 1)
```

In [8]:

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

Out[8]:

```
6
```

In [9]:

```
data = data.drop_duplicates()
```

In [10]:

```
data.isnull().sum()
```

Out[10]:

```
img_source      0
length          0
nb_views        0
quality         0
title           0
video_link      0
voting          0
categories    353
tags          353
dtype: int64
```

In [11]:

```
data = data.dropna()
```

In [12]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 191173 entries, 0 to 191531
Data columns (total 9 columns):
 #   Column      Non-Null Count   Dtype 
---  ------      --------------   ----- 
 0   img_source  191173 non-null  object
 1   length      191173 non-null  int64 
 2   nb_views    191173 non-null  int64 
 3   quality     191173 non-null  object
 4   title       191173 non-null  object
 5   video_link  191173 non-null  object
 6   voting      191173 non-null  int64 
 7   categories  191173 non-null  object
 8   tags        191173 non-null  object
dtypes: int64(3), object(6)
memory usage: 14.6+ MB
```

In [13]:

```
data.describe()
```

Out[13]:

	length	nb_views	voting
count	1.911730e+05	1.911730e+05	191173.000000
mean	8.395711e+02	4.174042e+05	77.478933
std	2.895393e+03	1.037438e+06	6.609911
min	5.000000e+00	1.020000e+03	36.000000
25%	3.750000e+02	8.527000e+04	73.000000
50%	6.010000e+02	1.723500e+05	78.000000
75%	1.065000e+03	3.817400e+05	82.000000
max	1.218521e+06	1.134600e+08	95.000000

In [14]:

```
data.nunique()
```

Out[14]:

```
img_source    191168
length          5054
nb_views       59211
quality            2
title         187859
video_link    191168
voting            43
categories     59116
tags          187958
dtype: int64
```

In [15]:

```
data['quality'].unique()
```

Out[15]:

```
array(['LOW', 'HD'], dtype=object)
```

In [16]:

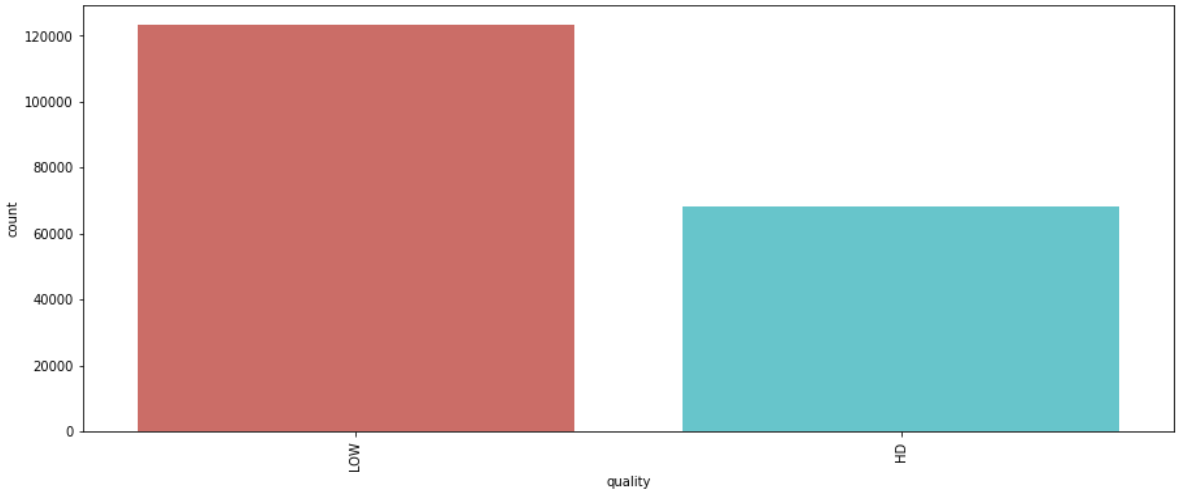
```
data['quality'].value_counts()
```

Out[16]:

```
LOW    123102
HD      68071
Name: quality, dtype: int64
```

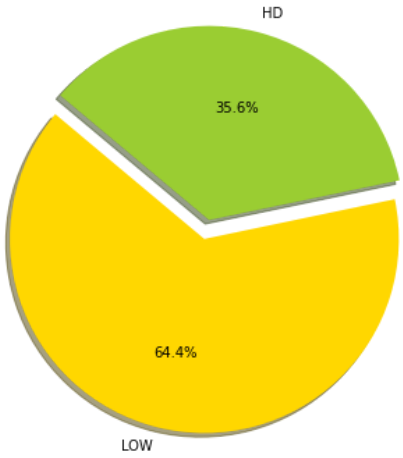
In [17]:

```
plt.figure(figsize=(15,6))
sns.countplot('quality' , data=data, palette='hls')
plt.xticks(rotation = 90)
plt.show()
```



In [18]:

```
plt.figure(figsize=(15,6))
labels = 'LOW', 'HD'
colors = ['gold', 'yellowgreen']
explode = (0.1, 0)

plt.pie(data['quality'].value_counts(), explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=140)

plt.axis('equal')
plt.show()
```



In [19]:

```
data['title'].loc[data['title'].str.contains('سکس نار')]
```

Out[19]:

```
سکس نار          37927
سکس نار          59412
سکس نار         131337
سکس نار         151966
سکس نار         159575
سکس نار         163957
سکس نار         164047
سکس سکس نار     165415
سکس نار         168331
سکس نار         179475
سکس نار         182955
سکس نار         186733
سکس نار         190114
Name: title, dtype: object
```

In [20]:

```
for col in ['title','categories','tags']:
    data[col] = data[col].str.replace('سکس نار','Firefighters') # google translator
    data[col] = data[col].str.replace('\d+','') # unnecessary numbers
    data[col] = data[col].str.replace(r'[_,-]+',' ') # too many underscores and hyphens in
    data[col] = data[col].str.replace(r'\bScene\b','').str.strip() # Scene word appears too
```

In [21]:

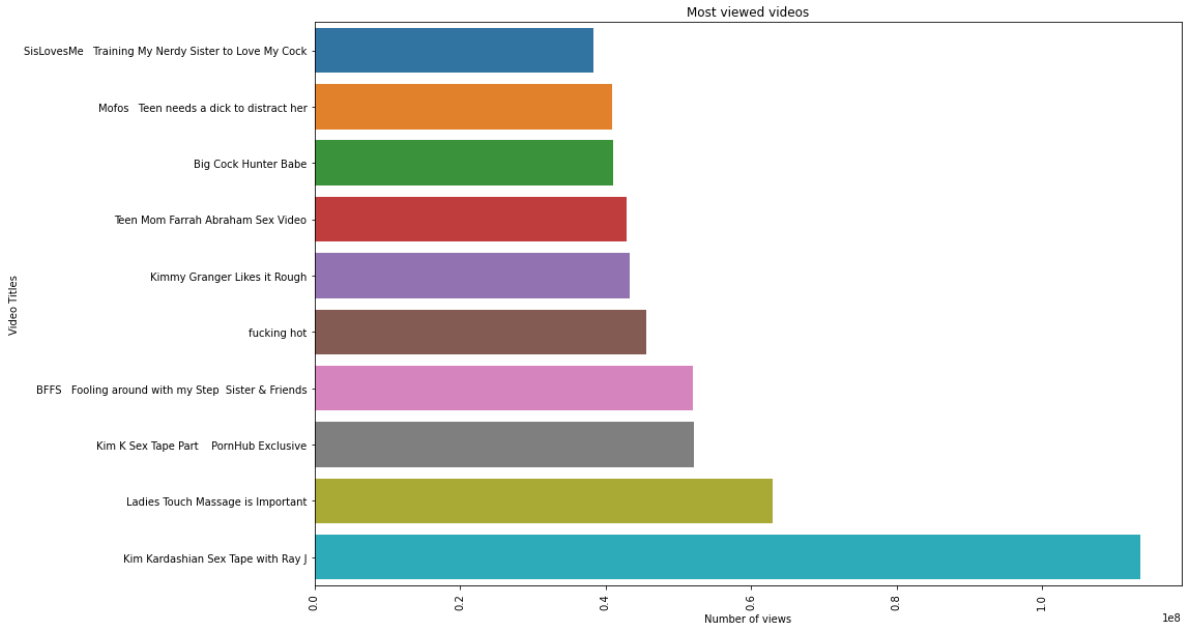
```
data.head()
```

Out[21]:

	img_source	length	nb_views	quality	title	
0	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	1583	127450	LOW	Tease then suck	http://thumbzilla.com
1	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	2501	480620	LOW	Two hot chick on one dick	http://thumbzilla.com
2	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	1513	99720	LOW	Chick stretches her pussy and blows for the cam	http://thumbzilla.com/
3	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	1710	598820	LOW	Fucking my neighbour	http://thumbzilla.com
4	http://cdn-d-img.pornhub.com/m=eafT8daaaa/vide...	1694	155850	LOW	Small titted whore sucking a cock for lunch	http://thumbzilla.com/

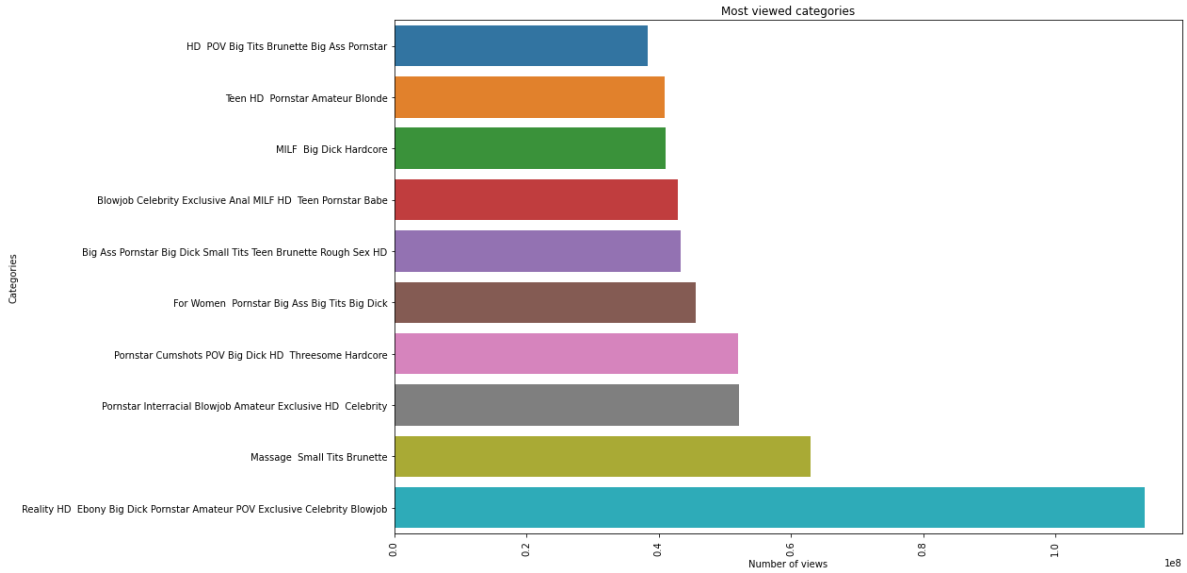
In [23]:

```
most_viewed_title = data[['nb_views','title']].sort_values(by='nb_views', ascending=False).
most_viewed_title.sort_values(by='nb_views',ascending=True, inplace=True)

plt.figure(figsize=(15,10))
ax=sns.barplot(
    y=most_viewed_title['title'],
    x=most_viewed_title['nb_views']
)
plt.xticks(rotation= 90)
plt.ylabel('Video Titles')
plt.xlabel('Number of views')
plt.title('Most viewed videos')
plt.show()
```



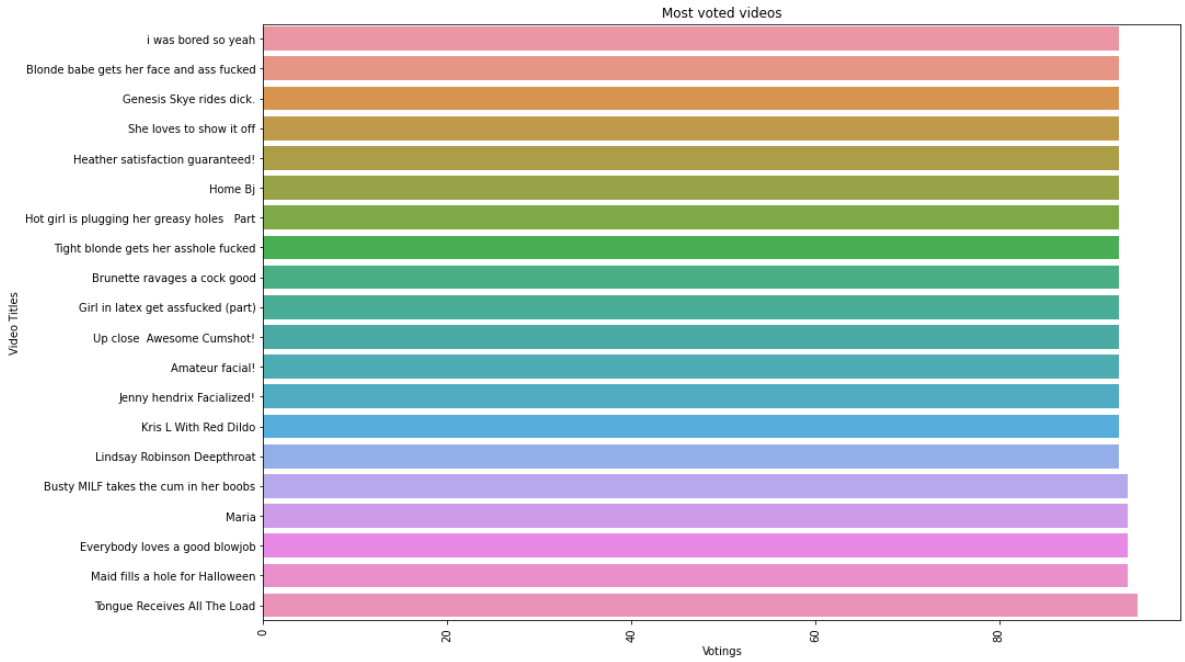
In [24]:

```
most_viewed_cat = data[['nb_views','categories']].sort_values(by='nb_views', ascending=Fals
most_viewed_cat.sort_values(by='nb_views',ascending=True, inplace=True)

plt.figure(figsize=(15,10))
ax=sns.barplot(
    y=most_viewed_cat['categories'],
    x=most_viewed_cat['nb_views']
)
plt.xticks(rotation= 90)
plt.ylabel('Categories')
plt.xlabel('Number of views')
plt.title('Most viewed categories')
plt.show()
```



In [27]:

```
most_voted_cat = data[['voting','title']].sort_values(by='voting', ascending=False).head(20
most_voted_cat.sort_values(by='voting',ascending=True, inplace=True)

plt.figure(figsize=(15,10))
ax=sns.barplot(
    x=most_voted_cat['voting'] ,
    y=most_voted_cat['title']
)
plt.xticks(rotation= 90)
plt.ylabel('Video Titles')
plt.xlabel('Votings')
plt.title('Most voted videos')
plt.show()
```



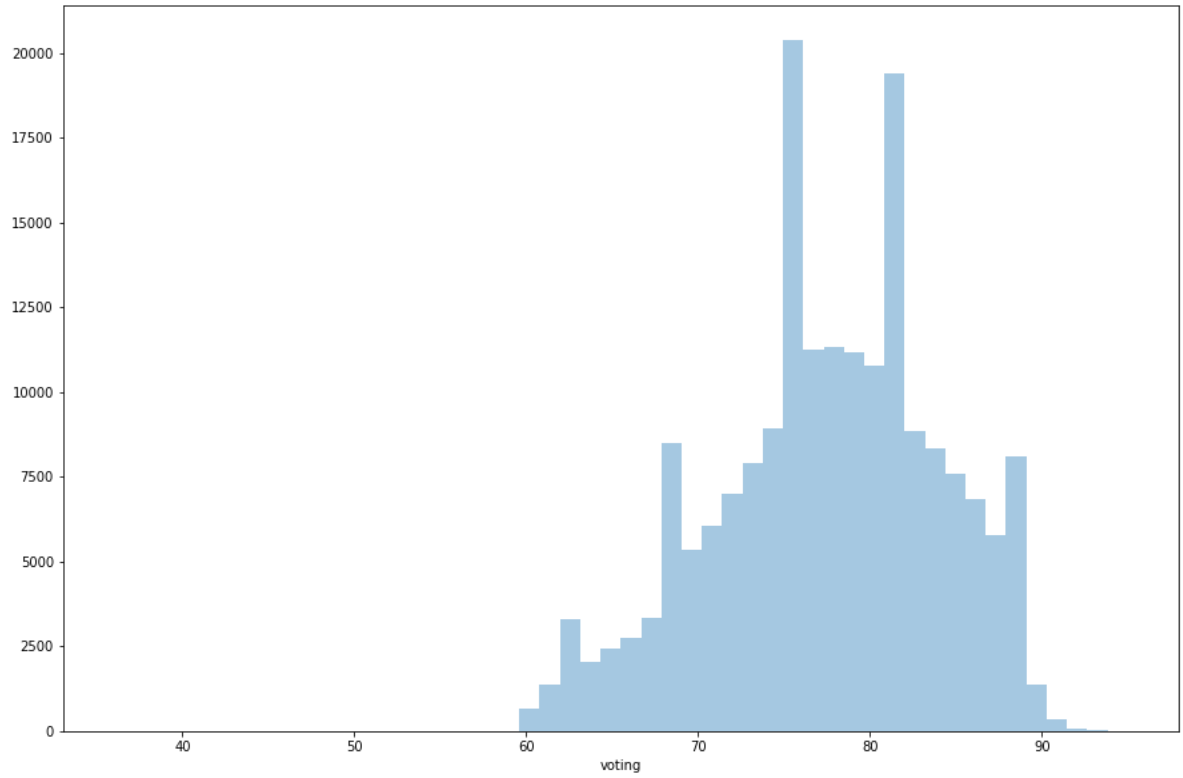
In [28]:

```
print(data['voting'].min())
print(data['voting'].mean())
print(data['voting'].max())
```

```
36
77.47893269447046
95
```

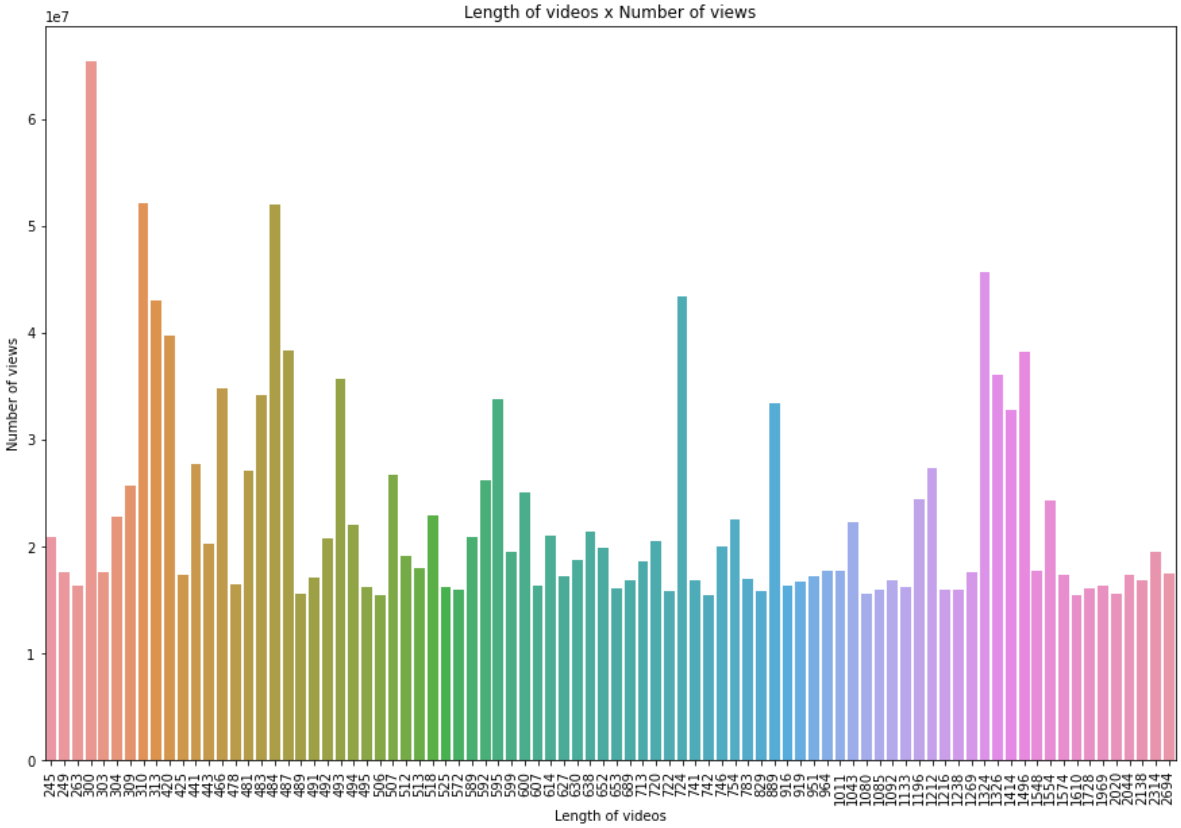
In [29]:

```
plt.figure(figsize=(15,10))
sns.distplot(data['voting'],kde = False)
plt.show()
```



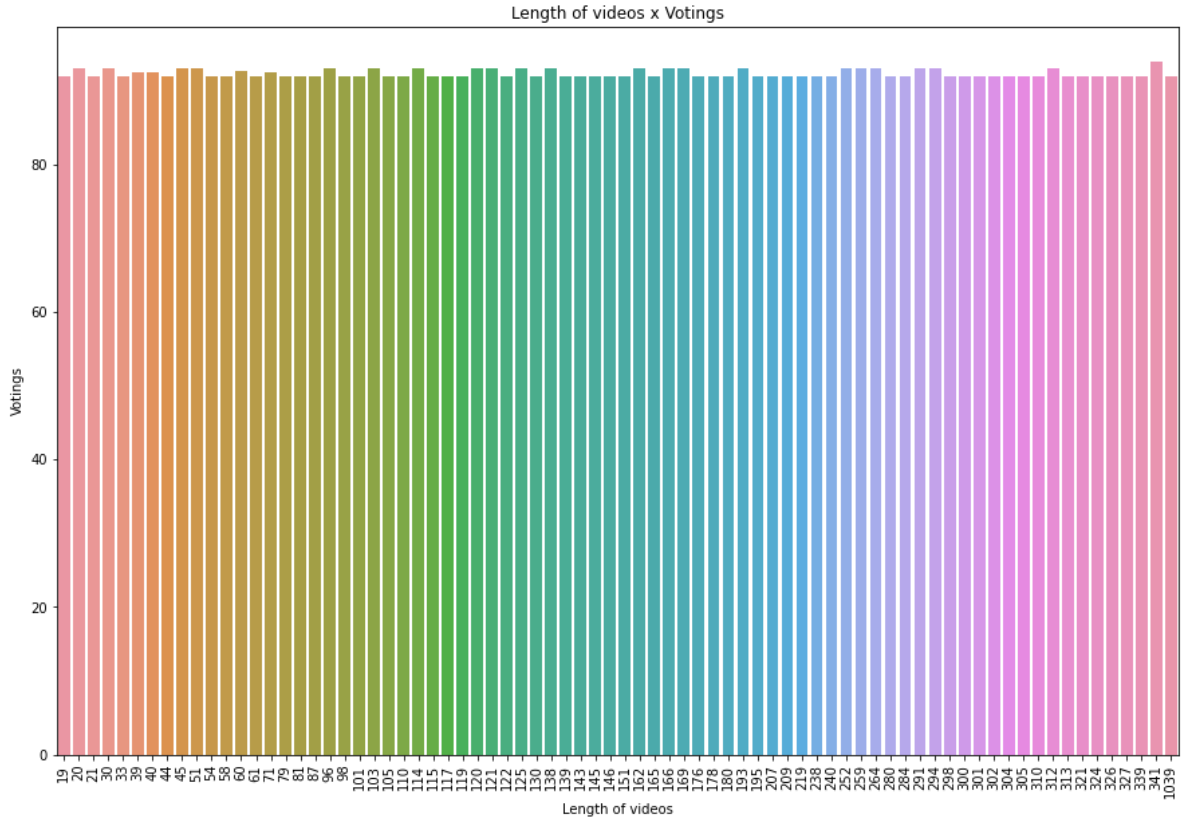
In [34]:

```
data2 = data[['nb_views','length']].sort_values(by='nb_views', ascending=False).head(100)

plt.figure(figsize=(15,10))
ax=sns.barplot(
    y=data2['nb_views'],
    x=data2['length'],
    ci = None
)
plt.xticks(rotation= 90)
plt.ylabel('Number of views')
plt.xlabel('Length of videos')
plt.title('Length of videos x Number of views')
plt.show()
```



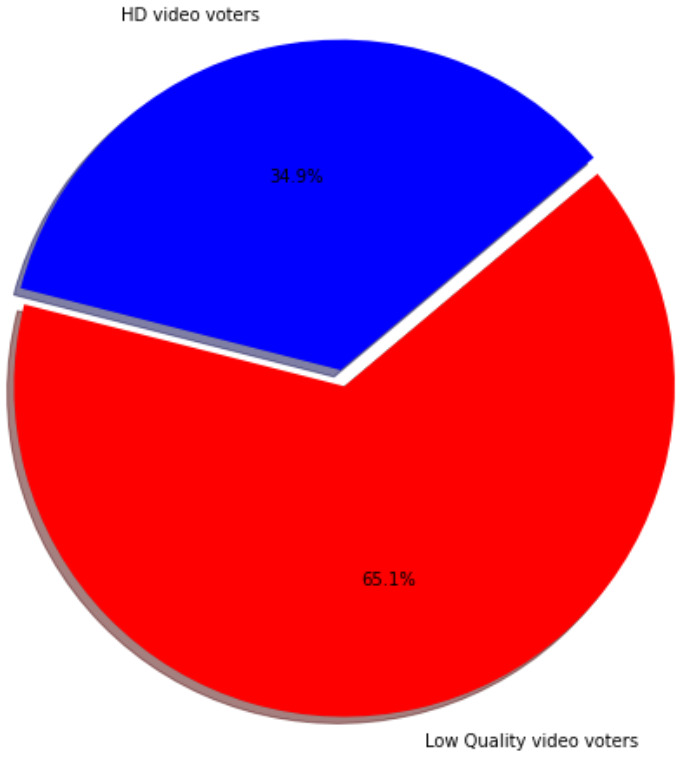
In [35]:

```
data2 = data[['voting','length']].sort_values(by='voting', ascending=False).head(100)

plt.figure(figsize=(15,10))
ax=sns.barplot(
    y=data2['voting'],
    x=data2['length'],
    ci=None
)
plt.xticks(rotation= 90)
plt.ylabel('Votings')
plt.xlabel('Length of videos')
plt.title('Length of videos x Votings')
plt.show()
```



In [37]:

```
plt.figure(figsize=(12,8))
labels = ['HD video voters','Low Quality video voters']
sizes = [
    np.array(data['voting'].loc[data['quality']=='HD']).sum(),
    np.array(data['voting'].loc[data['quality']=='LOW']).sum()
]
colors = ['blue', 'red']
explode = (0.05, 0)

plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=40)

plt.axis('equal')
plt.show()
```

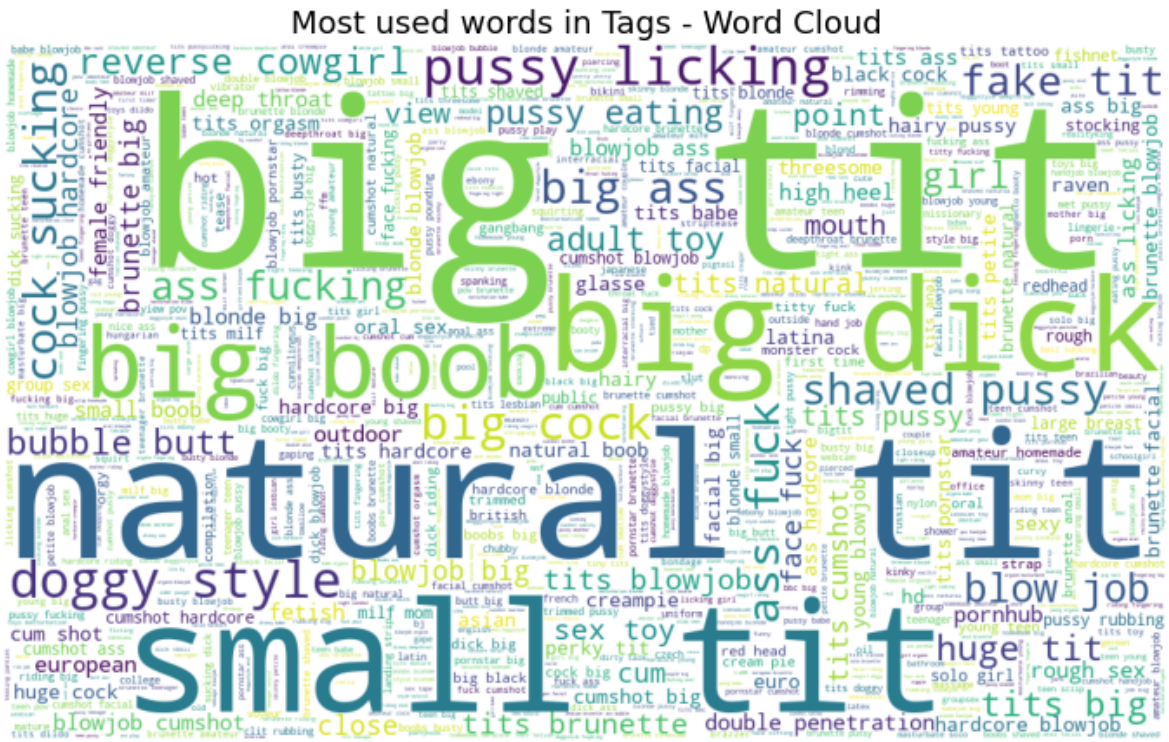


In [38]:

```
from wordcloud import WordCloud, STOPWORDS
from nltk.corpus import stopwords
```

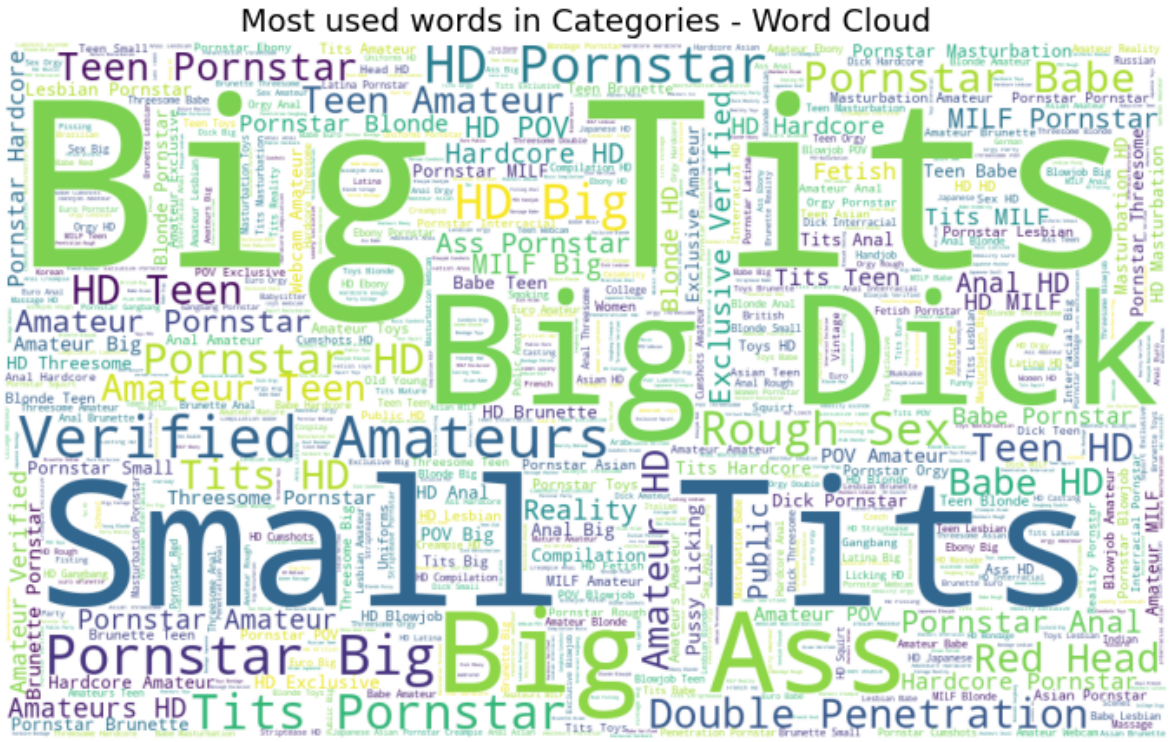
In [39]:

```
stopwords = set(STOPWORDS)
wordcloud = WordCloud(
                            background_color='white',
                            stopwords=stopwords,
                            max_words=1500,
                            max_font_size=200,
                            width=1000, height=600,
                            random_state=69,
                           ).generate(" ".join(data['tags'].astype(str)))

fig = plt.figure(figsize = (12,12))
plt.imshow(wordcloud)
plt.title("Most used words in Tags - Word Cloud", fontsize=18)
plt.axis('off')
plt.show()
```



In [40]:

```
wordcloud = WordCloud(
                            background_color='white',
                            stopwords=stopwords,
                            max_words=1500,
                            max_font_size=200,
                            width=1000, height=600,
                            random_state=69,
                           ).generate(" ".join(data['categories'].astype(str)))

fig = plt.figure(figsize = (12,12))
plt.imshow(wordcloud)
plt.title("Most used words in Categories - Word Cloud", fontsize=18)
plt.axis('off')
plt.show()
```



In [41]:

```
wordcloud = WordCloud(
                            background_color='white',
                            stopwords=stopwords,
                            max_words=1500,
                            max_font_size=200,
                            width=1000, height=600,
                            random_state=69,
                           ).generate(" ".join(data['title'].astype(str)))

fig = plt.figure(figsize = (12,12))
plt.imshow(wordcloud)
plt.title("Most used words in Titles - Word Cloud", fontsize=18)
plt.axis('off')
plt.show()
```