In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style='whitegrid', font_scale=1.3)
%matplotlib inline
```

Кейс 1

Задача заключается в работе с данными о трендах на YouTube. В этом нам поможет библиотека `seaborn`.

In [4]:
```
from datetime import datetime as dt
parse = lambda x: dt.strptime(x,'%y.%d.%m')

videos = pd.read_csv("RUvideos_short.csv",
                     parse_dates=['trending_date'],
                     date_parser=parse,
                     dayfirst=True)
videos.head()
```

Out[4]:

	video_id	trending_date	title	channel_title	category_id	publish_time	
0	gDuslQ9avLc	2017-11-14	Захар и Полина учатся экономить	Т—Ж БОГАЧ	22	2017-11-13T09:09:31.000Z	захар и полина|"учимс
1	AOCJIFEA_jE	2017-11-14	Биржа Мемов #29. Большой выпуск	Druzhko Show	22	2017-11-13T17:32:11.000Z	шаги
2	VAWNQDgwwOM	2017-11-14	ХАЙП КЭМП - СВОЙ СРЕДИ ЧУЖИХ	Юлик	24	2017-11-13T16:11:31.000Z	юмор|"комедия"|"влог"
3	gknkFwKQfHg	2017-11-14	Сочная кесадилья с курицей	Hochland	22	2017-11-13T06:51:10.000Z	хохланд|"сыр"|"реце
4	3sYvZcwzZr8	2017-11-14	КЛИПЫ РОДИТЕЛЕЙ НА ШКОЛЬНЫЙ ВЫПУСКНОЙ	Совергон	24	2017-11-13T16:52:36.000Z	Совергон|"Sovergon"|"

В таблице много лишних данных. Оставим следующие столбцы:

- `trending_date` -- дата в формате год-день-месяц;
- `category_id` -- категория видео, названия приведены в файле `RU_category_id.json`;
- `views` -- количество просмотров видео;
- `likes` -- количество лайков;
- `dislikes` -- количество дислайков;
- `comment_count` -- количество комментариев.

Из даты оставим только день. Для этого можно пройтись циклом по всем датам и взять поле `day` у даты. Напечатайте начало таблицы.

In [5]:
```
youtube = videos[
            ['trending_date',
             'category_id',
             'views',
             'likes',
             'dislikes',
             'comment_count']]

youtube.trending_date = youtube['trending_date'].dt.day
youtube.head()
```

```
/home/bakhtiyar/anaconda3/lib/python3.7/site-packages/pandas/core/generic.py:5303: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  self[name] = value
```

Out[5]:

	trending_date	category_id	views	likes	dislikes	comment_count
0	14	22	62408	334	190	50
1	14	22	330043	43841	2244	2977
2	14	24	424596	49854	714	2944
3	14	22	112851	3566	122	80
4	14	24	243469	36216	631	1692

2. Некоторая визуализация

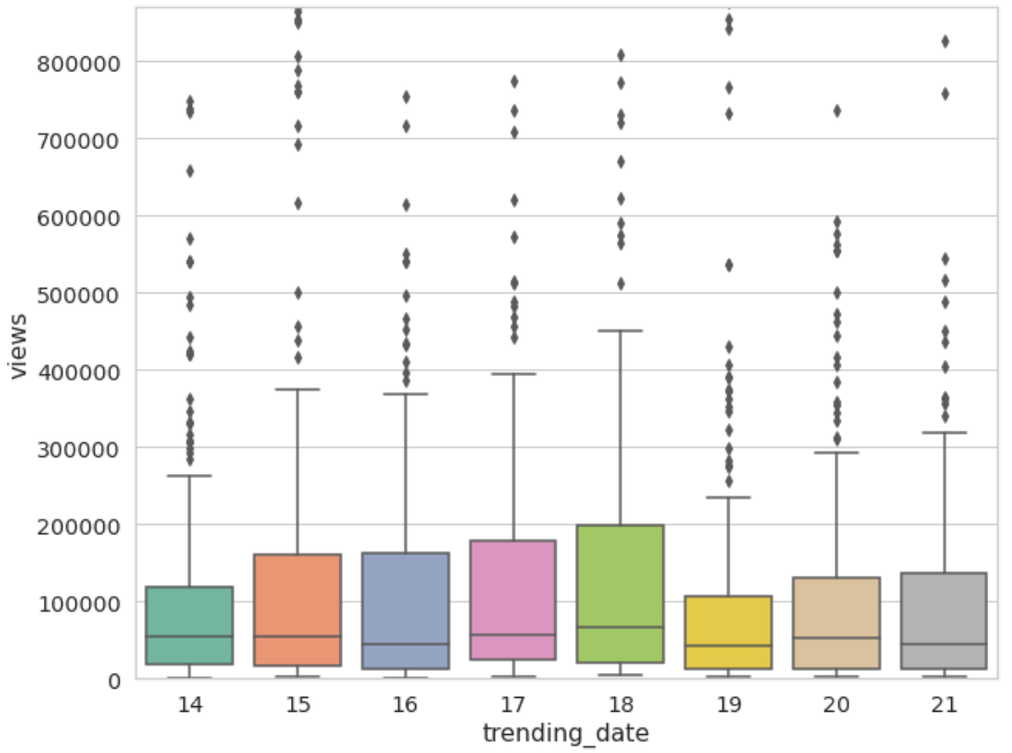
Построим ящики с усами на каждый день по количеству просмотров.

In [7]:
```
(videos['views'].max()+videos['views'].mean())/10
```

Out[7]: 869921.11625

In [8]:
```
plt.figure(figsize=(10, 8))
sns.boxplot(
    data=youtube,
    y='views',
    x='trending_date',
    palette='Set2').set(ylim=(0, 869921))
```

Out[8]: [(0.0, 869921.0)]



Построим jointplot по всем данным для количества просмотров по горизонтальной оси и количества лайков по вертикальной.

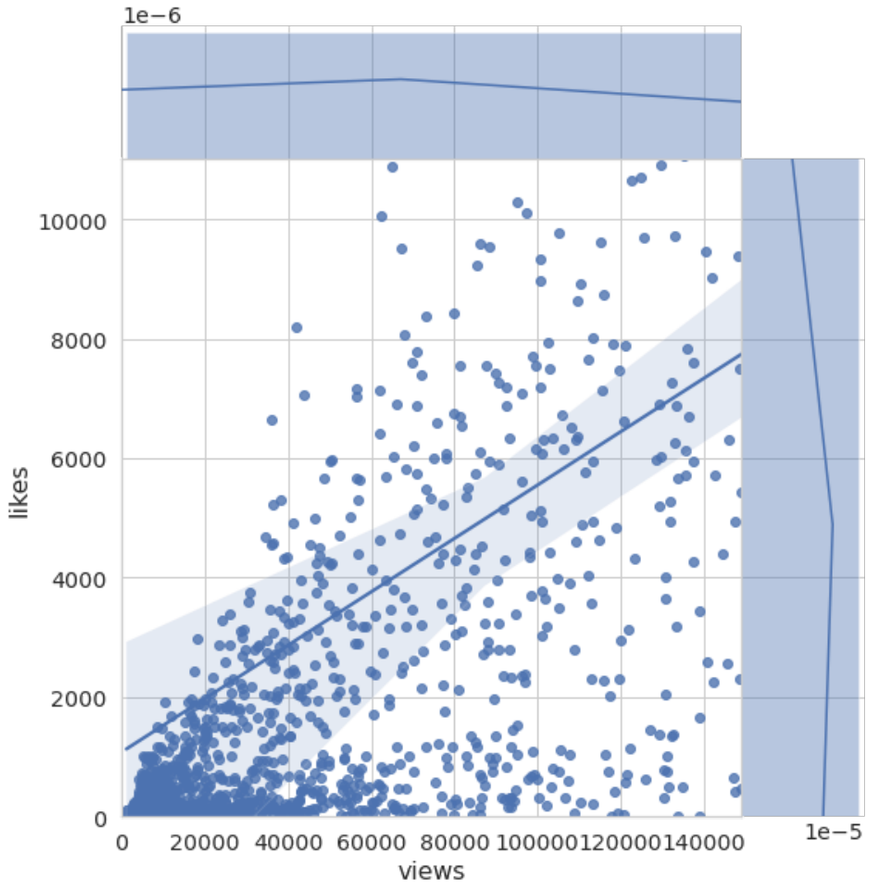
In [9]:
```
videos.describe().transpose()
```

Out[9]:

	count	mean	std	min	25%	50%	75%	max
category_id	1600.0	20.024375	8.475180	1.0	20.00	22.0	25.00	43.0
views	1600.0	165385.162500	442078.329158	1058.0	15342.00	50218.5	148866.00	8533826.0
likes	1600.0	8468.100625	29560.407950	0.0	207.50	1186.5	6020.25	789772.0
dislikes	1600.0	995.541250	4727.892085	0.0	19.75	89.0	404.25	86001.0
comment_count	1600.0	1153.681875	4979.302130	0.0	40.00	188.5	756.50	100581.0

In [10]:
```
sns.jointplot(
    x='views',
    y='likes',
    xlim=(0, 148866),
    ylim=(0, 11020),
    kind='reg',
    data=youtube,
    height=8,
    space=0)
```

Out[10]: <seaborn.axisgrid.JointGrid at 0x7f3d04603610>



Кейс 2

Netflix за последние 5-10 лет обзавелись большим количеством зрителей. С увеличением числа зрителей увеличилось и разнообразие шоу. Соответственно, перед аналитиками из киноиндустрии встала задача исследования данных с рейтингами различных сериалов.
В данном задании мне предстоит провести визуальный анализ датасета **1000 Netflix Shows** (по состоянию на 11.06.2017) и сделать выводы.

Описание признаков:

- `title` - название шоу.
- `rating` - рейтинг шоу. Например: G, PG, TV-14, TV-MA
- `ratingLevel` - описание рейтинговой группы и особенностей шоу.
- `release year` - год выпуска шоу.
- `user rating score` - оценка пользователей.

In [12]:
```
data = pd.read_csv('netflix_data.csv', encoding='cp437')
del data['ratingDescription'], data['user rating size']
```

Удалим из данных дупликаты. Сколько объектов удалено?

In [13]:
```
size_with_duples = len(data)
data.drop_duplicates(inplace=True)
size_with_duples - len(data)
```

Out[13]: 500

Сколько объектов осталось?

In [14]:
```
len(data)
```

Out[14]: 500

Сколько рейтинговых групп представлено в данных?

In [15]:
```
len(data.rating.drop_duplicates())
```

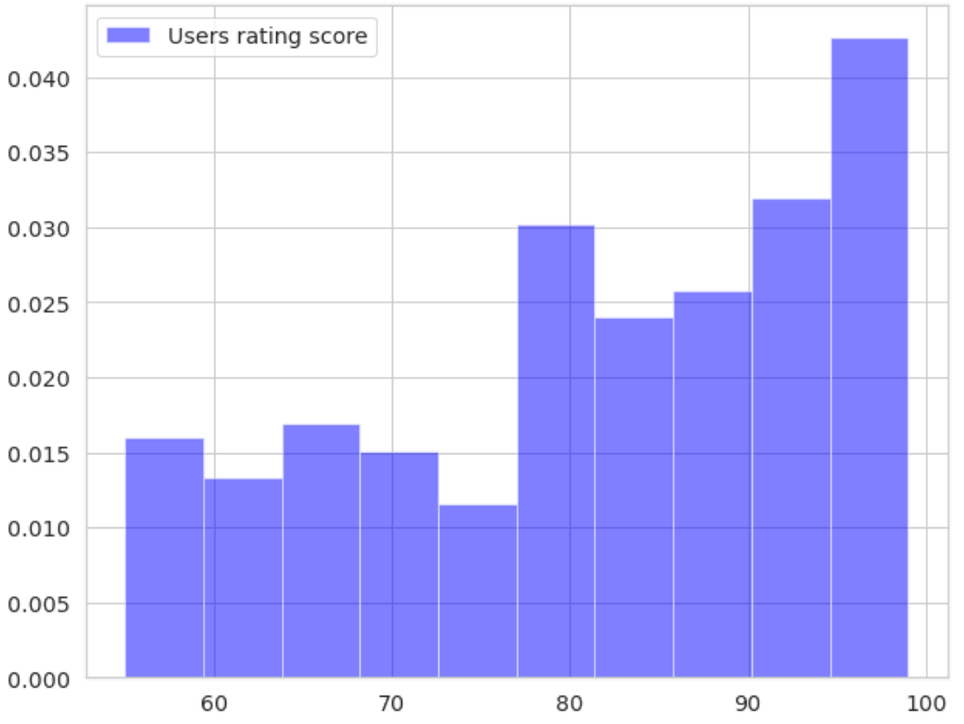
Out[15]: 13

Какие оценки пользователи ставят чаще? Постройте гистограмму оценок.

In [16]:
```
plt.figure(figsize=(10,8))
plt.hist(
    data['user rating score'],
    bins=10,
    density=True,
    alpha=0.5,
    color='blue',
    label='Users rating score')

plt.legend()
plt.show()
```

```
/home/bakhtiyar/anaconda3/lib/python3.7/site-packages/numpy/lib/histograms.py:839: Runtim
eWarning: invalid value encountered in greater_equal
  keep = (tmp_a >= first_edge)
/home/bakhtiyar/anaconda3/lib/python3.7/site-packages/numpy/lib/histograms.py:840: Runtim
eWarning: invalid value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```



In [295]:
```
data['user rating score'].mean()
```

Out[295]:
```
81.3984375
```

Вывод: средний зрительский балл - 81, но по гистограмме видно, что большинство оценок не менее 90. Продукты компании высоко ценятся у зрителей.

Выведим основную информацию об оценках пользователей: среднее, стандартное отклонение, минимум, максимум, медиана.

In [303]:
```
data['user rating score'].describe()
```

Out[303]:
```
count    256.000000
mean      81.398438
std       12.730904
min       55.000000
25%       71.000000
50%       83.500000
75%       93.000000
max       99.000000
Name: user rating score, dtype: float64
```

Еомментарий: Медиана и среднее отличаются. Медиана и среднее могут значительно отличаться, т.к. медиана более устайяива к выбросам.

В какие годы были запущены шоу, представленные в датасете?

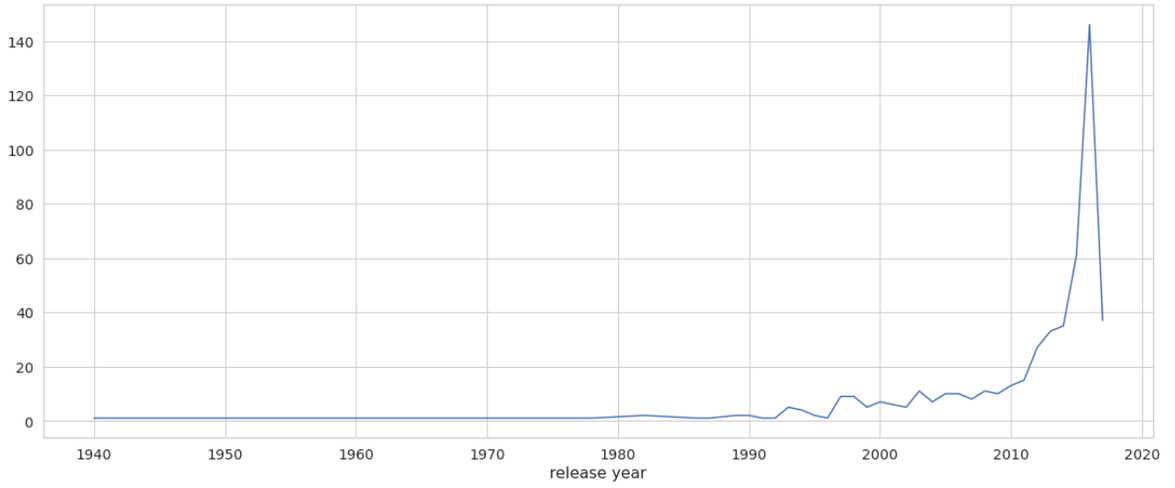
In [314]:
```
sorted(data['release year'].drop_duplicates())
```

Out[314]:
```
[1940,
 1976,
 1978,
 1982,
 1986,
 1987,
 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,
 2017]
```

Построим график, показывающий распределение количества запущенных шоу в зависимости от года. Наблюдается ли рост? Есть ли выбросы?

In [17]:
```
plt.figure(figsize=(20, 8))
data.groupby('release year').count().title.plot()
```

Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3d04ddf590>
```



Вывод: количество запущенных шоу начало возрастать в середине 1990-го десятилетия, отмечаю взрывной рос в середине 2010-го десятилетия. Скорее всего рост наблюдается благодаря переходу на стриминговый сервис.

Сравним среднюю оценку пользователей в 2016 со средней оценкой в 2017. Можно ли сделать вывод, что 2017 год успешнее для Netflix? ("Успешнее" значит, что пользователи в среднем ставили более высокие оценки) Ответить на этот вопрос мне поможет график, который я построил выше.

In [18]:
```
print('2016: ', data.groupby('release year').mean().loc[2016])
print('2017: ',data.groupby('release year').mean().loc[2017])
```

```
2016:  user rating score    84.313953
Name: 2016, dtype: float64
2017:  user rating score    88.125
Name: 2017, dtype: float64
```

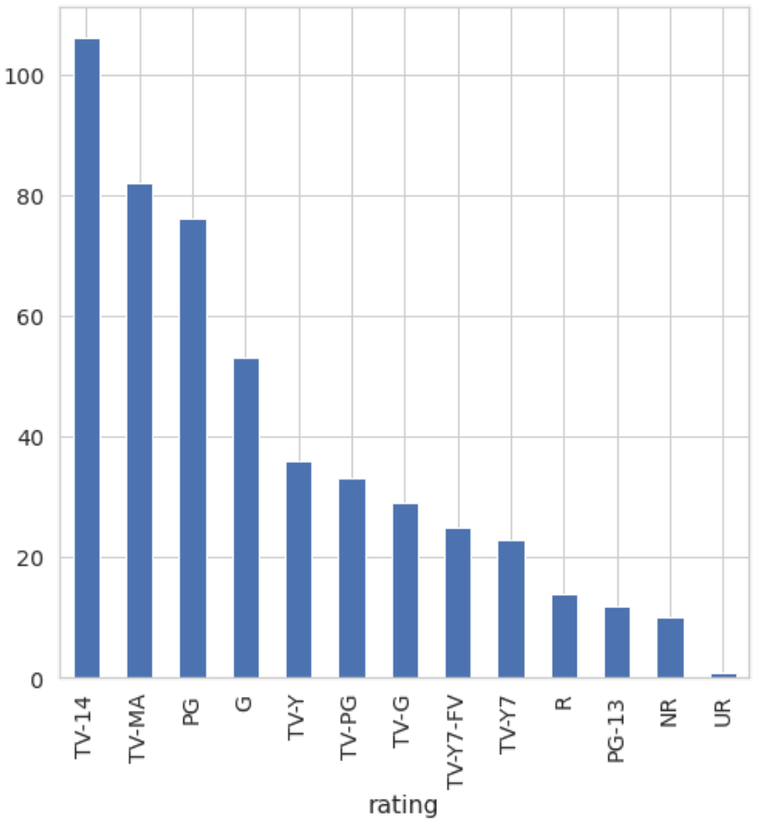
In [19]:
```
data.groupby('release year').count().title.tail()
```

Out[19]:
```
release year
2013     33
2014     35
2015     61
2016    146
2017     37
Name: title, dtype: int64
```

Вывод: 2017 выдался более успешным для компании. Однако мы можем наблюдать резкое адение количества выпускаемых шоу..

In [20]:
```
data.groupby('rating').size().sort_values(
    ascending=False).plot(
    kind='bar',
    figsize=(8, 8),
    label='rating')
```

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3d029d2650>



Вывод: видим, что большинчтво шой категории TV-14. Наиболее выпускаемыми являются шоу для подростков.

Составим топ-13 самых высоко оцененных шоу. Выберем из данного топа шоу, которое наиболее нравится. Обозначим это шоу N. Ответим на следующие вопросы:

- Какое шоу является худшим по оценкам в рейтинговой группе, к которой принадлежит N?
- Сколько шоу было выпущено в одном году с N?
- Насколько бы изменилась средняя оценка шоу, выпущенных в одном году с N, если бы Netflix не запустили шоу N?

In [21]:
```
data.sort_values('user rating score', ascending=False).head(13)
```

Out[21]:

	title	rating	ratingLevel	release year	user rating score
41	13 Reasons Why	TV-MA	For mature audiences. May not be suitable for...	2017	99.0
350	Lost	TV-14	Parents strongly cautioned. May be unsuitable ...	2010	98.0
10	Once Upon a Time	TV-PG	Parental guidance suggested. May not be suitab...	2016	98.0
64	Friends	TV-14	Parents strongly cautioned. May be unsuitable ...	2003	98.0
72	Orange Is the New Black	TV-MA	For mature audiences. May not be suitable for...	2016	98.0
27	The Flash	TV-PG	Parental guidance suggested. May not be suitab...	2016	98.0
25	Marvel's Iron Fist	TV-MA	NaN	2017	98.0
88	Finding Dory	PG	mild thematic elements	2016	98.0
62	Family Guy	TV-MA	For mature audiences. May not be suitable for...	2015	98.0
63	Criminal Minds	TV-14	Parents strongly cautioned. May be unsuitable ...	2016	98.0
3	Prison Break	TV-14	Parents strongly cautioned. May be unsuitable ...	2008	98.0
2	Grey's Anatomy	TV-14	Parents strongly cautioned. May be unsuitable ...	2016	98.0
8	The Walking Dead	TV-MA	For mature audiences. May not be suitable for...	2015	98.0

Взял "Family Guy", категория - "TV-MA", год выпуска - 2015, средняя оценка - 98.0

In [22]:
```
print('Худшее по оценкам шоу, категории TV-MA - ',
      data[data.rating == 'TV-MA'].sort_values(
          by='user rating score').dropna().head(1).title)
```

```
Худшее по оценкам шоу, категории TV-MA -  380    Bitten
Name: title, dtype: object
```

In [23]:
```
print("Шоу выпущенных в одном году с Family Guy -",
      len(data[data['release year'] == 2015]))
```

```
Шоу выпущенных в одном году с Family Guy - 61
```

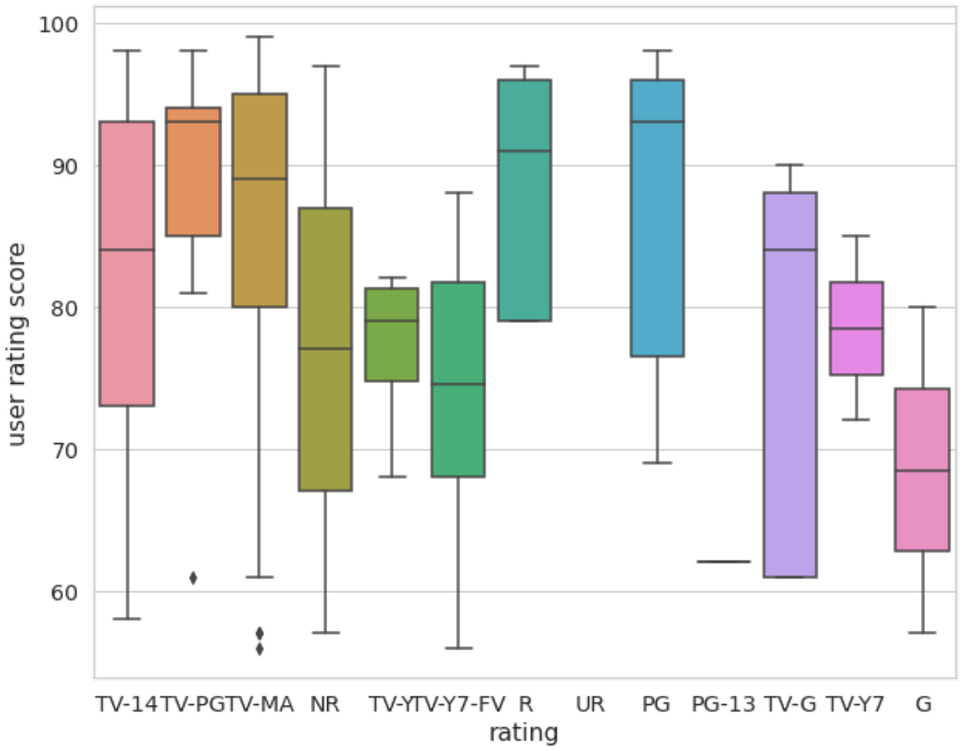
In [24]:
```
ans = data[data['release year'] == 2015].set_index('title').drop(
    ['Family Guy']).mean()['user rating score']
print("Средний рейтинг шоу 2015 года, без Family Guy -", int(ans))
```

```
Средний рейтинг шоу 2015 года, без Family Guy - 82
```

Ответим на следующие вопросы при помощи `boxplot` :

- Какую рейтинговую группу зрители оценивали выше всего в последние пять лет?
- Как менялись оценки пользователей с течением времени? Построим `boxplot` для каждого десятилетия.

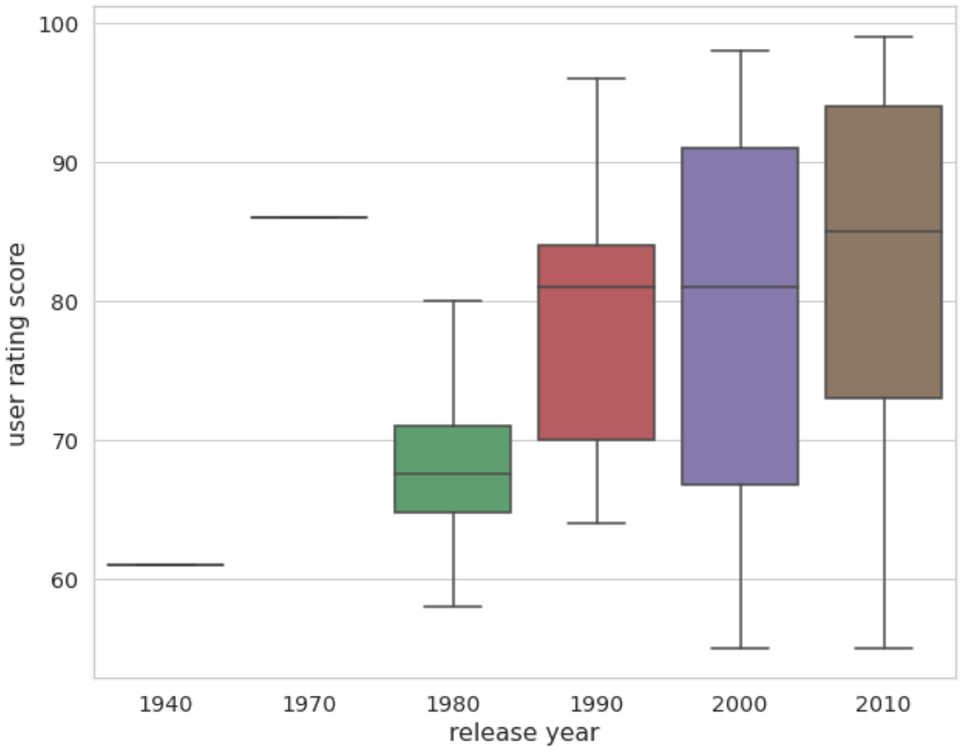
In [25]:
```
plt.figure(figsize=(10,8))
sns.boxplot(
    x='rating',
    y='user rating score',
    data=data[data['release year'] > 2012]);
```



Вывод: За последние 5 лет (начал с 2012, т.к. последние данные датируются 2017) наивысший рейтинг имели шоу категорий TV-PG, PG и R.

In [26]:
```
tmp = data
tmp['release year'] = data['release year'] - data['release year'] % 10
plt.figure(figsize=(10,8))
sns.boxplot(
    x='release year',
    y='user rating score',
    data=tmp)
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3d02977e50>



Вывод: Видим, что начиная с 1980-ых рейтинги постоянно растут, а также можно отметить снижение скорости роста рейтингов.