



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)

## РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

по дисциплине Оперативный анализ данных

по теме «Korean drama list»

\_\_\_\_\_

\_\_\_\_\_

Студент ИУ5-54Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

А.А. Свечникова  
(И.О.Фамилия)

Руководитель

\_\_\_\_\_  
(Подпись, дата)

К.Ю. Маслеников  
(И.О.Фамилия)

Консультант

\_\_\_\_\_  
(Подпись, дата)

(И.О.Фамилия)

2022 г.

## **Аннотация**

Данная работа проводится с целью анализа набора данных для выявления в них зависимостей, подтверждающих или опровергающих некоторые гипотезы. Набор данных содержит информацию о телесериалах, выпущенных в Южной Корее. Перед анализом данные будут подвергнуты очистке: устранению ошибок, дубликатов и пропусков. Также данные будут преобразованы для упрощения анализа.

Для обеспечения наглядности результаты анализа будут визуализированы. Для обработки набора данных используются библиотеки pandas, numpy, seaborn и matplotlib языка программирования Python. Обработка производится на платформе Google Colab.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ОСНОВНАЯ ЧАСТЬ.....	5
1 Определение данных.....	5
2 Формулирование гипотез.....	5
3 Получение данных.....	6
4 Загрузка данных.....	6
5 Проверка данных на целостность.....	6
6 Устранение ошибок, пропусков и дубликатов. Преобразование данных.....	8
7 Выбор данных для анализа, агрегирование и исследование данных..	10
8 Анализ с помощью описательной статистики. Визуализация данных и описательной статистики.....	12
9 Формулирование выводов и ограничений.....	19
ЗАКЛЮЧЕНИЕ.....	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	22

## **ВВЕДЕНИЕ**

Цель работы: проанализировать набор данных и выявить в нём зависимости, подтверждающие или опровергающие выдвинутые гипотезы.

Задачи:

- 1) Определить анализируемый набор данных.
- 2) Сформулировать гипотезы о зависимостях, связывающих данные в наборе.
- 3) Освоить инструменты анализа данных языка Python.
- 4) Подготовить данные к анализу.
- 5) Исследовать данные.
- 6) Визуализировать результаты исследований.
- 7) Сформулировать выводы.

## **ОСНОВНАЯ ЧАСТЬ**

### **1 Определение данных**

Набор данных «Korean drama list» содержит сведения о сериалах, выпущенных телекомпаниями Южной Кореи. Исходный набор содержит следующие поля:

- 1) Название сериала.
- 2) Жанры.
- 3) Тэги.
- 4) Число эпизодов.
- 5) Начало показа по телевидению.
- 6) Окончание показа по телевидению.
- 7) Дни, в которые транслировался сериал.
- 8) Телекомпания.
- 9) Продолжительность одного эпизода.
- 10) Оценка по данным портала Mydramalist.
- 11) Число зрителей, поставивших оценки на портале.
- 12) Сравнительный рейтинг данным портала Mydramalist.
- 13) Популярность данным портала Mydramalist.
- 14) Возрастной рейтинг.
- 15) Число зрителей по данным портала Mydramalist.
- 16) Список актёров.
- 17) Платформы, на которых можно посмотреть сериал.
- 18) Рейтинг по данным IMDB.
- 19) Число просмотров по данным IMDB.
- 20) Описание с IMDB.

### **2 Формулирование гипотез**

Были сформулированы следующие гипотезы:

- 1) Большинство сериалов транслируются по выходным.

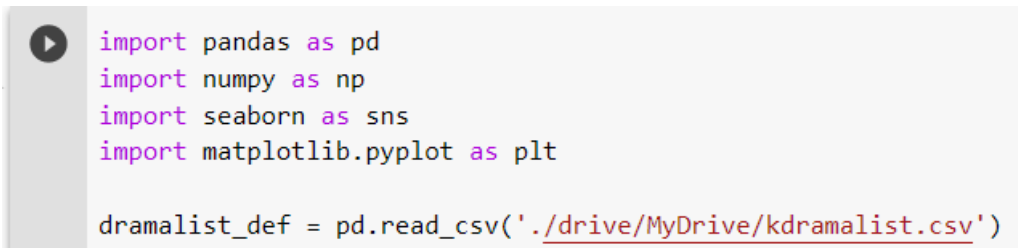
- 2) Чем чаще в течение недели показывают сериал, тем больше у него зрителей.
- 3) Сериалы, выпускаемые крупными телекомпаниями, имеют более высокие рейтинги.
- 4) Более новые сериалы имеют больше зрителей.
- 5) Более новые сериалы имеют более высокие оценки.

### 3 Получение данных

Набор данных был скачан с сайта Kaggle.com в виде файла kdramalist.csv.

### 4 Загрузка данных

Файл, содержащий набор данных, был загружен в Google Colab и импортирован в переменную типа фрейм библиотеки pandas. Фрейм представляет собой таблицу из строк и столбцов.



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

dramalist_def = pd.read_csv('./drive/MyDrive/kdramalist.csv')
```

Рисунок 4.1 – Загрузка данных

### 5 Проверка данных на целостность

Получим описание набора данных. Заметим, что в последних трёх столбцах очень много пропусков. Мы не станем автоматически генерировать пропущенные значения, например, рейтингов, так как наша задача – проанализировать оценки, выставленные реальными зрителями. Строки с пропущенными значениями будут удалены.

```
dramalist_def.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1278 entries, 0 to 1277
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   drama_name                            1278 non-null   object
1   Genres                                1277 non-null   object
2   Tags                                  1278 non-null   object
3   Episodes                              1278 non-null   int64
4   start airing                          1255 non-null   object
5   end airing                            1221 non-null   object
6   Aired On                              1222 non-null   object
7   Original Network                      1250 non-null   object
8   Duration                              1230 non-null   float64
9   score                                 1239 non-null   float64
10  scored by                             1239 non-null   float64
11  Ranked                                1278 non-null   int64
12  Popularity                            1278 non-null   int64
13  Content Rating                        1278 non-null   object
14  Watchers                              1278 non-null   object
15  actors                                1278 non-null   object
16  platforms                             1278 non-null   object
17  imdb_rating                           690 non-null    float64
18  imdb_user_count                       690 non-null    object
19  imdb_description                       668 non-null    object
dtypes: float64(4), int64(3), object(13)
memory usage: 199.8+ KB
```

Рисунок 5.1 – Информация о наборе данных

В некоторых строках в качестве значения указан пустой список, '?' или 'N/A'. Такие строки нужно будет удалить.

dramalist_def.head(10)																				
	drama_name	genres	tags	episodes	start airing	end airing	aired on	original network	duration	score	scored by	ranked	popularity	content rating	watchers	actors	platforms	imdb_rating	imdb_user_count	imdb_description
0	100 Days My Prince	['historical', 'comedy', 'romance', 'drama']	['amnesia', 'hidden identity', 'marriage of co...']	16	10-Sep-18	30-Oct-18	['tuesday', 'monday']	[tvN]	75.0	8.3	21274.0	554	136	15+ - Teens 15 or older	45,852	['Doh Kyung Soo', 'Nam Ji Hyun', 'Kim Seon Ho', ...]	['VieTV', 'Netflix', 'K-DRAMA', 'Apple TV', 'Vi...	7.7	2,188	Upon losing his memory, a crown prince encount...
1	12 Signs of Love	['comedy', 'romance', 'life']	['writer female lead']	16	15-Feb-12	5-Apr-12	['wednesday', 'thursday']	[tvN]	65.0	7.1	893.0	5506	3327	Not Yet Rated	2,543	['Ohn Joo Wan', 'Yoon Jin Seo', 'Lee Yong Woo', ...]	[]	NaN	NaN	NaN
2	12 Years Promise	['food', 'romance', 'drama', 'family']	['unexpected pregnancy', 'miscarriage', 'teena...']	26	22-Mar-14	29-Jun-14	['saturday', 'sunday']	[JTBC]	70.0	7.2	2716.0	5395	1783	15+ - Teens 15 or older	5,435	['Lee So Yeon', 'Namkoong Min', 'Lee Tae im', ...]	['Viki']	7.3	181	A pregnant teen is forced by her family to lea...
3	16 Again	['romance', 'life', 'drama', 'fantasy']	['second chance', 'first love', 'father-son re...']	16	21-Sep-20	10-Nov-20	['monday', 'tuesday']	[JTBC]	70.0	8.7	21409.0	105	125	15+ - Teens 15 or older	47,527	['Kim Ha Neul', 'Yoon Sang Hyun', 'Lee Do Hyun', ...]	['VieTV']	8.2	1,847	A 37-year-old man on the verge of being divorc...
4	365: Repeat the Year	['thriller', 'mystery', 'drama', 'fantasy']	['strong female lead', 'time travel', 'nice ma...']	24	23-Mar-20	28-Apr-20	['monday', 'tuesday']	[MBC]	35.0	8.6	7598.0	250	438	15+ - Teens 15 or older	20,963	['Lee Joon Hyuk', 'Nam Ji Hyun', 'Kim Jee Soo', ...]	['Viki']	8.0	656	A story where ten people get the chance to go ...
5	4 Legendary Villains	['food', 'romance', 'drama', 'family']	['manager supporting character', 'single father...']	40	25-Oct-14	8-Mar-15	['saturday', 'sunday']	[MBC]	64.0	7.7	1151.0	2840	2821	15+ - Teens 15 or older	3,156	['Han Ji Hye', 'Ha Seok Jin', 'Go Doo Shim', ...]	[]	NaN	NaN	NaN
6	5th Republic	['military', 'historical', 'life', 'drama', 'p...	['massacre', 'gwangju uprising', 'death', 'bas...']	41	23-Apr-05	1-Sep-05	['saturday', 'sunday']	[MBC]	65.0	6.9	6.0	76723	99999	15+ - Teens 15 or older	81	['Lee Deok Hwa', 'Seo In Seok', 'Kim Young Ran', ...]	[]	NaN	NaN	NaN
7	7 First Kisses	['comedy', 'romance', 'sci-fi']	['reverse-harem', 'miniseries', 'web series', ...]	8	5-Dec-16	5-Jan-17	['monday', 'thursday']	[Naver TV Cast]	10.0	6.9	18317.0	5949	216	15+ - Teens 15 or older	34,665	['Lee Cho Hee', 'Choi Ji Woo', 'Lee Joon Gi', ...]	['LDF']	NaN	NaN	NaN
8	7th Grade Civil Servant	['action', 'comedy', 'romance']	['national intelligence service', 'rivals to l...']	20	23-Jan-13	28-Mar-13	['wednesday', 'thursday']	[MBC]	65.0	6.9	4686.0	5938	985	15+ - Teens 15 or older	10,347	['Joo Won', 'Choi Kang Hee', 'Kim Min Seop', ...]	[]	6.3	145	A romantic comedy about a spy couple who hides...
9	#Alive	['mystery', 'horror', 'supernatural']	['ghost-seeing male lead', 'ghost', 'investiga...']	8	17-Aug-19	28-Aug-19	['wednesday', 'saturday']	[Naver TV Cast]	10.0	7.4	265.0	68801	6080	15+ - Teens 15 or older	1,025	['Jo Seung Hyun', 'Shin Woo Hee', 'Jeong Seon ...]	['Lululala Story Lab', 'Viki']	NaN	NaN	NaN

Рисунок 5.2 – Первые десять строк набора данных

## 6 Устранение ошибок, пропусков и дубликатов. Преобразование данных

Начнем преобразование данных с того, что выберем столбцы, которые потребуются для анализа, и приведем названия столбцов к единому виду.

```
df = dramalist_def[['drama_name', 'Genres', 'Episodes', 'start airing', 'end airing', 'Aired On',
                    'Original Network', 'Duration', 'score', 'scored by', 'actors']]
df.rename(columns = {'drama_name':'title', 'Genres':'genre', 'Episodes':'episodes', 'Aired On':'airing days',
                    'Original Network':'network', 'Duration':'duration', 'actors':'actor'}, inplace = True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1278 entries, 0 to 1277
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   title                  1278 non-null   object
1   genre                  1277 non-null   object
2   episodes               1278 non-null   int64
3   start airing          1255 non-null   object
4   end airing             1221 non-null   object
5   airing days           1222 non-null   object
6   network                1250 non-null   object
7   duration               1230 non-null   float64
8   score                  1239 non-null   float64
9   scored by              1239 non-null   float64
10  actor                  1278 non-null   object
dtypes: float64(3), int64(1), object(7)
memory usage: 110.0+ KB
```

Рисунок 6.1 – Выбор и переименование столбцов

Затем удалим строки с пропущенными значениями. Заметим, что число non-null значений в каждом столбце теперь совпадает с общим числом строк.

```
df.replace('N/A', np.nan, inplace = True)
df.replace('?', np.nan, inplace = True)
df = df.dropna()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1153 entries, 0 to 1277
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   title                  1153 non-null   object
1   genre                  1153 non-null   object
2   episodes               1153 non-null   int64
3   start airing          1153 non-null   object
4   end airing             1153 non-null   object
5   airing days           1153 non-null   object
6   network                1153 non-null   object
7   duration               1153 non-null   float64
8   score                  1153 non-null   float64
9   scored by              1153 non-null   float64
10  actor                  1153 non-null   object
dtypes: float64(3), int64(1), object(7)
memory usage: 108.1+ KB
```

Рисунок 6.2 – Удаление пропущенных значений



df.describe()

	episodes	duration	score	scored by
count	1278.000000	1230.000000	1239.000000	1239.000000
mean	35.031299	55.532520	7.698951	7034.288136
std	41.608015	16.647208	0.597510	12187.126387
min	0.000000	5.000000	4.000000	1.000000
25%	16.000000	40.000000	7.300000	351.000000
50%	16.000000	60.000000	7.700000	1983.000000
75%	32.000000	65.000000	8.100000	7872.500000
max	496.000000	95.000000	10.000000	92027.000000

Рисунок 6.3 – Описание набора данных

Заметим, что минимальное значение столбца scored by, то есть минимальное число зрителей, оценивших сериал, равняется единице. Отберем сериалы с числом зрителей больше 100.

```
[43] df = df[df['scored by'] > 100]
```

Рисунок 6.4 – Фильтрация строк

Некоторые столбцы содержат списки значений, а не одно значение. Это усложнит анализ, поэтому преобразуем столбцы genre, actor, network следующим образом: заменим список значений первым значением в списке. Списки значений в столбце airing days отсортируем, а иначе списки, например, [понедельник, вторник] и [вторник, понедельник] будут считаться различными. Кроме этого, преобразуем столбец scored by к целому числу, так как в нём хранится количество зрителей, оценивших сериал. Из даты начала и окончания трансляции оставим только год, чтобы можно было агрегировать данные по годам.

```

from datetime import date
def get_first(x):
    lst = eval(x)
    if len(lst) == 0:
        return np.nan
    else:
        return lst[0]

def sort_days(x):
    lst = eval(x)
    if len(lst) == 0:
        return np.nan
    else:
        return sorted(lst)

def get_year(x):
    return datetime.strptime(x, '%d-%b-%y').year

df['genre'] = df['genre'].apply(get_first)
df['network'] = df['network'].apply(get_first)
df['actor'] = df['actor'].apply(get_first)
df['airing days'] = df['airing days'].apply(sort_days)
df['scored by'] = df['scored by'].apply(int)
df['start airing'] = df['start airing'].apply(get_year)
df['end airing'] = df['end airing'].apply(get_year)

```

Рисунок 6.5 – Трансформация значений столбцов

df.head()

	title	genre	episodes	start airing	end airing	airing days	network	duration	score	scored by	actor
0	100 Days My Prince	historical	16	2018	2018	[monday, tuesday]	tvN	75.0	8.3	21274	Doh Kyung Soo
1	12 Signs of Love	comedy	16	2012	2012	[thursday, wednesday]	tvN	65.0	7.1	893	Ohn Joo Wan
2	12 Years Promise	food	26	2014	2014	[saturday, sunday]	jTBC	70.0	7.2	2716	Lee So Yeon
3	18 Again	romance	16	2020	2020	[monday, tuesday]	jTBC	70.0	8.7	21409	Kim Ha Neul
4	365: Repeat the Year	thriller	24	2020	2020	[monday, tuesday]	MBC	35.0	8.6	7598	Lee Joon Hyuk

Рисунок 6.6 – Новый вид набора данных

## 7 Выбор данных для анализа, агрегирование и исследование данных

Создадим новые столбцы в наборе. Подсчитаем, как часто показывали сериал в течение недели, и показывали ли его в выходные. Эти сведения пригодятся нам при проверке гипотез.

```
def count_days(l):
    return len(l)

def is_weekend(l):
    # l = eval(x)
    return (('saturday' in l) or ('sunday' in l))

df['days count'] = df['airing days'].apply(count_days)
df['in weekend'] = df['airing days'].apply(is_weekend)
```

```
df.head(5)
```

	title	genre	episodes	start airing	end airing	airing days	network	duration	score	scored by	actor	days count	in weekend
0	100 Days My Prince	historical	16	2018	2018	[monday, tuesday]	tvN	75.0	8.3	21274	Doh Kyung Soo	2	False
1	12 Signs of Love	comedy	16	2012	2012	[thursday, wednesday]	tvN	65.0	7.1	893	Ohn Joo Wan	2	False
2	12 Years Promise	food	26	2014	2014	[saturday, sunday]	jTBC	70.0	7.2	2716	Lee So Yeon	2	True
3	18 Again	romance	16	2020	2020	[monday, tuesday]	jTBC	70.0	8.7	21409	Kim Ha Neul	2	False
4	365: Repeat the Year	thriller	24	2020	2020	[monday, tuesday]	MBC	35.0	8.6	7598	Lee Joon Hyuk	2	False

Рисунок 7.1 – Создание новых столбцов

Построим матрицу корреляций для числовых характеристик набора данных.

```
plt.figure(figsize=(16, 6))
heatmap = sns.heatmap(df.corr(), vmin=-1, vmax=1, annot=True, cmap='BrBG')
heatmap.set_title('Матрица корреляций', fontdict={'fontsize':12}, pad=12);
```

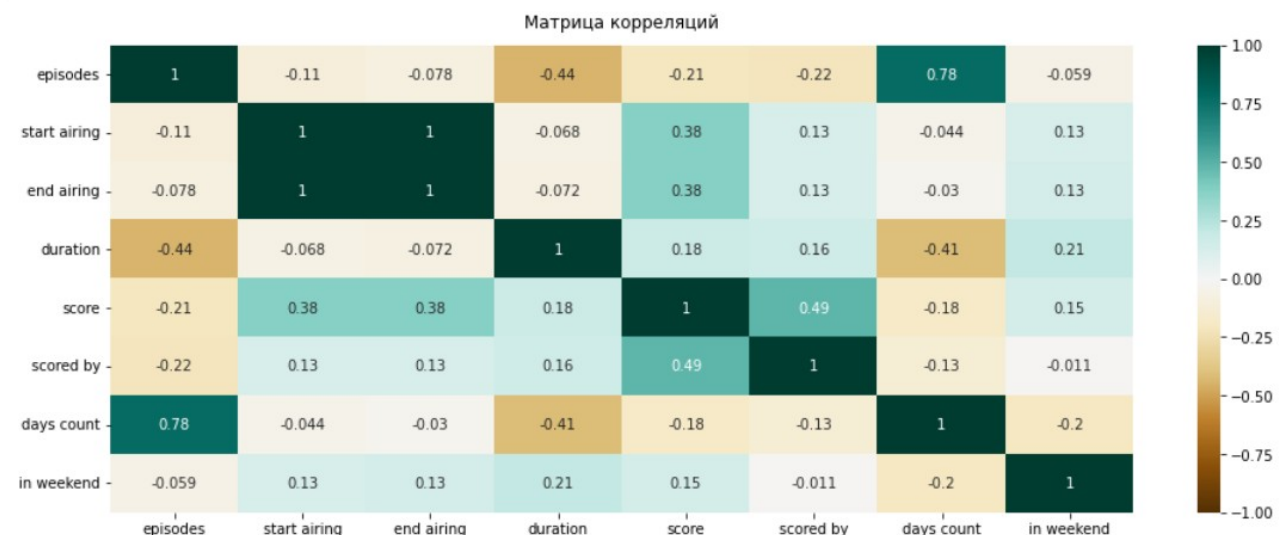


Рисунок 7.2 – Матрица корреляций

Корреляция значений столбцов start airing и end airing равна единице. Это очевидно, ведь событие не может кончиться раньше, чем началось. Среди остальных параметров наибольшую, и притом положительную, корреляцию имеют значения столбцов days count и episodes, то есть чем больше серий в сериале, тем чаще его показывают в течение недели. В целом можно сказать, что параметры не сильно связаны, так как в матрице много значений, близких к нулю.

Подсчитаем, сколько сериалов сняла каждая телекомпания, то есть сколько раз то или иное значение встречается в столбце network. Отобразим десять

самых крупных телекомпаний на круговой диаграмме. Эти данные пригодятся для проверки гипотезы.

```
nts = df['network'].value_counts().head(10)
plt.figure(figsize=(8, 8))
plt.pie(nts.values, labels=nts.index, colors = sns.color_palette('pastel'), autopct='%d%%')
plt.title('Количество сериалов у телекомпаний')
```

Text(0.5, 1.0, 'Количество сериалов у телекомпаний')

Количество сериалов у телекомпаний

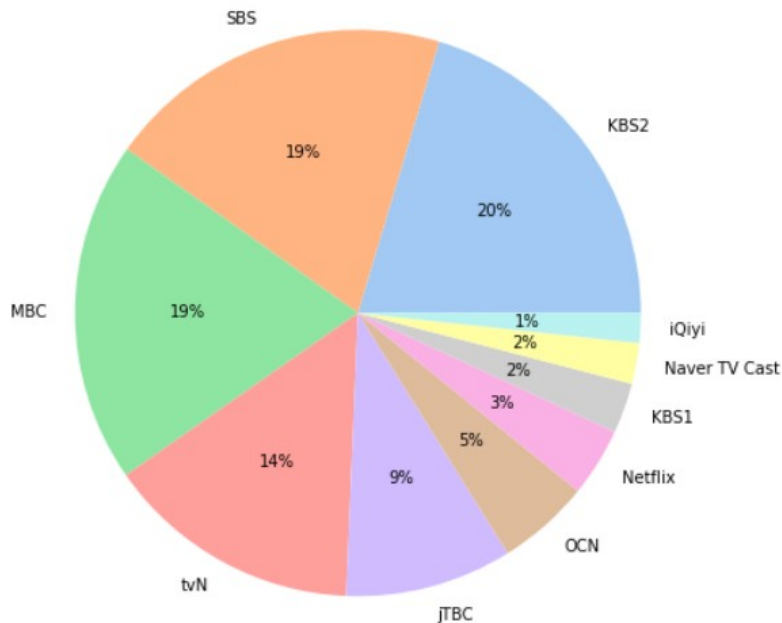


Рисунок 7.3 – Десять крупнейших телекомпаний

Лидером является телекомпания KBS2.

## 8 Анализ с помощью описательной статистики. Визуализация данных и описательной статистики

Перейдём к проверке гипотез. Проверим гипотезу «Чем чаще показывают сериал, тем больше у него зрителей». Для этого построим график зависимости значений столбца scored by от значений столбца days count.

```
plt.figure(figsize=(10, 6))
plt.xlabel('число дней показа')
plt.ylabel(['число зрителей'])
sns.lineplot(data = df, x = 'days count', y = 'scored by', estimator = np.average)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f3e51db5eb0>

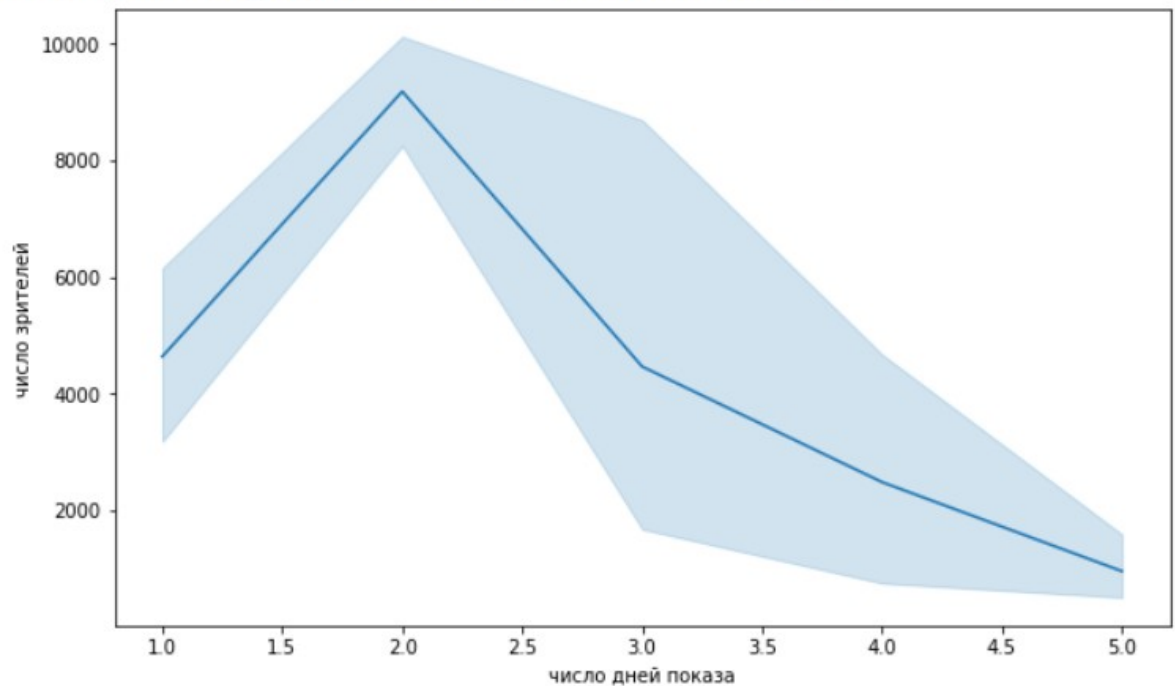


Рисунок 8.1 – Зависимость числа зрителей от числа дней показа

Между этими двумя значениями не наблюдается устойчивой зависимости. Этот вывод также можно было сделать на основе матрицы корреляций, так как корреляция этих параметров составляет только -0,13. Следовательно, данная гипотеза не подтвердилась.

Теперь проверим гипотезу «Больше всего сериалов показывают по выходным».

```

▶ dub = df.pivot_table(index = 'in weekend', values = 'title', aggfunc = 'count')
plt.figure(figsize = (10, 6))
sns.barplot(x = dub.index, y = dub['title'])
plt.title('Число сериалов в будни и выходные')
plt.xlabel('в эфире в выходной день')
plt.ylabel('количество')

```

```

☞ Text(0, 0.5, 'количество')

```



Рисунок 8.2 – Число сериалов в будни и выходные

Эта гипотеза также не подтвердилась. На диаграмме ясно видно, что только по будням показывают гораздо больше сериалов.

Проверим гипотезу «Сериалы, выпускаемые крупными телекомпаниями, имеют более высокие рейтинги». Создадим таблицу со средним рейтингом сериалов каждой телекомпании и отобразим на диаграмме десять телекомпаний с самыми высокими рейтингами.

```
sdf = df[['network', 'score']].groupby('network').mean().sort_values(by='score',
                                                                    ascending = False).head(10)

plt.figure(figsize=(6,6))
plt.title('Телекомпании с высокими рейтингами')
plt.xlabel('рейтинг')
plt.ylabel('телекомпания')
sns.barplot(y = sdf.index, x = sdf['score'], palette='pastel')
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f3e4c5a29d0>

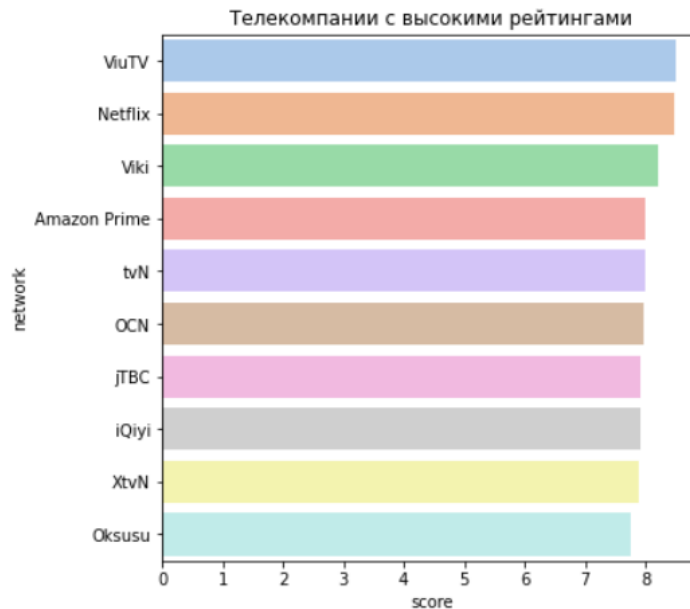


Рисунок 8.3 – Диаграмма со средними рейтингами по компаниям

Видно, что многие крупные телекомпании, отраженные на рисунке 7.3, не вошли в полученную подборку. Следовательно, эта гипотеза тоже не верна. Это может быть вызвано тем, что крупная компания может меньше заботиться о качестве выпускаемых продуктов, тогда как небольшие компании стремятся улучшить качество, чтобы точно привлечь зрителей.

Прежде чем проверять гипотезы, связанные с датой выхода сериала, выясним общий характер этих данных в наборе.

```
yrs = df['start airing'].value_counts()
plt.figure(figsize=(10, 6))
sns.lineplot(x = yrs.index, y = yrs.values)
plt.title('Количество сериалов в год')
plt.xlabel('Год')
plt.ylabel('Число сериалов')
```

```
Text(0, 0.5, 'Число сериалов')
```



Рисунок 8.4 – Диаграмма, отражающая количество сериалов, вышедших в каждом году

```
yrs.head(3)
```

```
2018    100
2019     94
2020     92
Name: start airing, dtype: int64
```

Рисунок 8.5 – Годы с наибольшим числом сериалов

Как видно, датасет содержит в основном данные о сериалах, выпущенных с 2011 по 2020 годы, причем наибольшее число было выпущено в 2018 году.

Проверим гипотезу «Более новые сериалы имеют большее число зрителей».



```
plt.figure(figsize=(8, 8))
plt.title('Число зрителей по годам')
plt.xlabel('Начало трансляции')
plt.ylabel('Число зрителей')
sns.scatterplot(data=df, x='start airing', y='scored by')
```

Рисунок 8.6 – Построение диаграммы зависимости числа зрителей от года выпуска

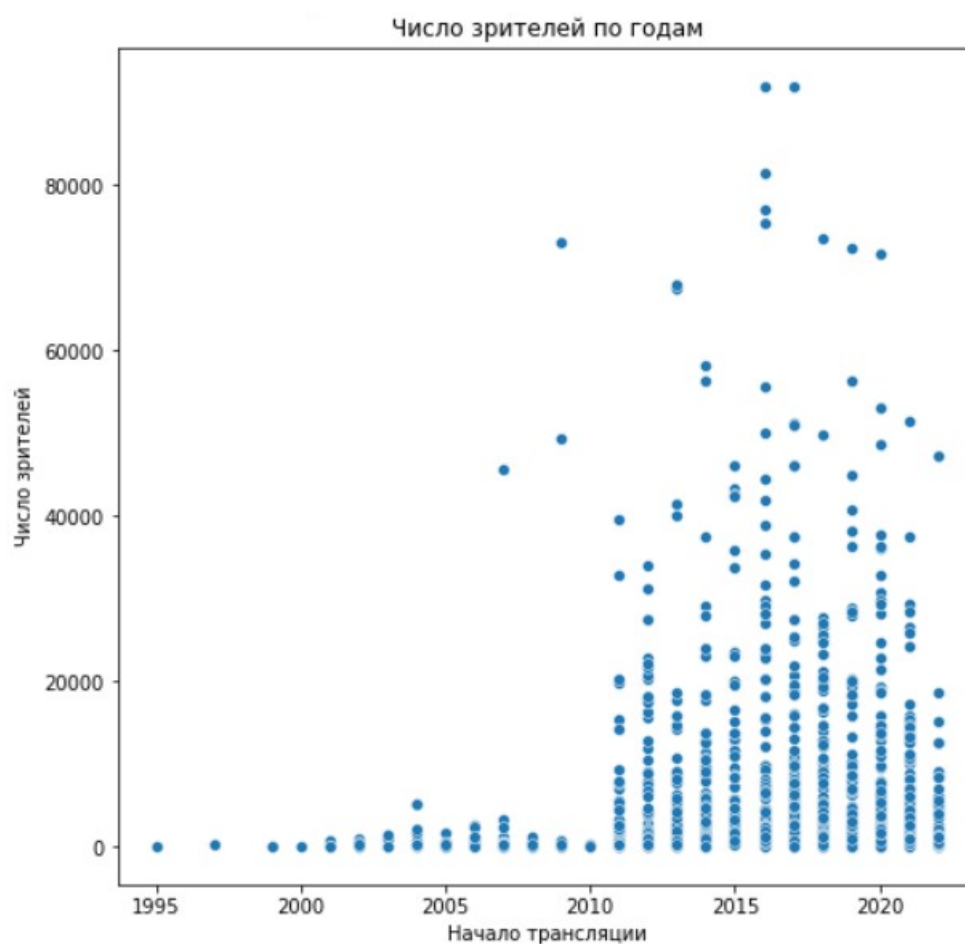


Рисунок 8.7 – Диаграмма зависимости числа зрителей от года выпуска

```
dub = df.pivot_table(index = 'start airing', values = 'scored by', aggfunc = 'mean')
plt.figure(figsize = (6, 6))
plt.title('Число зрителей по годам')
plt.xlabel('Начало трансляции')
plt.ylabel('Число зрителей')
sns.scatterplot(data=dub, x=dub.index, y='scored by')
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f3e4b434ca0>

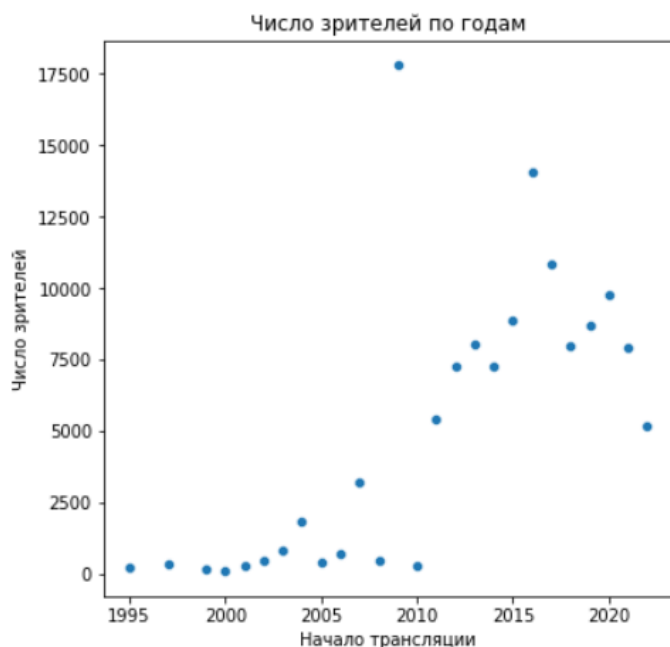


Рисунок 8.8 – Диаграмма зависимости среднего числа зрителей от года выпуска

Действительно, с течением времени как общее, так и среднее число зрителей за год в целом увеличивается. Гипотеза подтвердилась. Скорее всего, это вызвано тем, что большинство пользователей портала Mydramalist, с которого взяты эти данные, предпочитают смотреть современные сериалы. Спад числа зрителей, заметный после 2020 года, может быть связан с тем, что в связи с пандемией стало выходить меньше сериалов.

Проверим гипотезу «Более новые сериалы имеют более высокие рейтинги».

```
plt.figure(figsize = (10, 6))
plt.title('Зависимость рейтинга от года выпуска')
plt.xlabel('Год выпуска')
plt.ylabel('Средний рейтинг')
sns.lineplot(data = df, x = 'start airing', y = 'score', estimator=np.average, ci = None)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f3e480a3940>



Рисунок 8.9 – График зависимости среднего рейтинга от года выпуска

Гипотезу можно считать подтвержденной, так как в целом есть тенденция к увеличению среднего рейтинга с течением времени. Кроме того, в матрице корреляций можно увидеть, что значения score и start airing имеют не очень большую, но все же положительную корреляцию. Необычно выглядит увеличение рейтинга в 1995 году. Выведем сериалы, вышедшие в этом году.

```
df[df['start airing'] == 1995]
```

	title	genre	episodes	start airing	end airing	airing days	network	duration	score	scored by	actor	days count	in weekend
882	Sandglass	action	24	1995	1995	[friday, monday, thursday, tuesday, wednesday]	SBS	51.0	8.1	215	Go Hyun Jung	5	False

Рисунок 8.10 – Сериалы, вышедшие в 1995

Как видно, в этом году вышел только один сериал, который был высоко оценен. Следовательно, соответствующее высокое значение рейтинга можно считать всплеском, не оказывающим серьезного влияния на результаты анализа.

## 9 Формулирование выводов и ограничений

Получены следующие результаты проверки гипотез:

- 1) Большинство сериалов транслируются по выходным – не подтвердилось
- 2) Чем чаще в течение недели показывают сериал, тем больше у него зрителей – не подтвердилось
- 3) Сериалы, выпускаемые крупными телекомпаниями, имеют более высокие рейтинги – не подтвердилось
- 4) Более новые сериалы имеют больше зрителей - подтвердилось
- 5) Более новые сериалы имеют более высокие оценки – подтвердилось

Выяснилось, что исходный набор данных содержал пропущенные значения и малоинформативные строки (например, сериалы, оцененные только одним зрителем). Для повышения качества анализа эти строки были удалены.

В основном набор данных содержал сведения о сериалах, выпущенных после 2010 года. Следовательно, сделанные по результатам анализа выводы могут хуже отражать реальную картину для сериалов, выпущенных до 2010 года.

## **ЗАКЛЮЧЕНИЕ**

В ходе научно-исследовательской работы был проведен анализ набора данных «Korean drama list», содержащий сведения о телесериалах. Были исследованы предпочтения зрителей и деятельность телекомпаний. Результаты анализа позволяют оценить общие тенденции развития этой индустрии и восприятие ее публикой. Кроме того, в результате анализа были проверены выдвинутые перед началом работы гипотезы.

Для повышения качества анализа данные были предварительно очищены и трансформированы. Это также позволило уменьшить объем обрабатываемых данных и, следовательно, уменьшить время и ресурсы компьютера, затрачиваемые на вычисления.

Анализ проводился с помощью библиотек pandas, matplotlib и seaborn языка программирования Python. Была изучена документация к этим библиотекам и выбраны наиболее подходящие инструменты. Кроме того, были получены навыки работы с Jupyter Notebook и Google Colab.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Анализ данных и процессов: учеб. пособие / А. А. Барсегян, М. С. Куприянов, И. И. Холод, М. Д. Тесс, С. И. Елизаров. – 3-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2009. - 512 с.
2. Официальная документация Pandas: [Электронный ресурс]. // URL: [pandas - Python Data Analysis Library \(pydata.org\)](https://pandas.pydata.org/) (Дата обращения: 28.12.2022)
3. Официальная документация Seaborn: [Электронный ресурс]. // URL: [seaborn: statistical data visualization — seaborn 0.12.1 documentation \(pydata.org\)](https://seaborn.pydata.org/) (Дата обращения: 28.12.2022)
4. Официальная документация Matplotlib: [Электронный ресурс]. // URL: [Matplotlib — Visualization with Python](https://matplotlib.org/) (Дата обращения 28.12.2022)
5. Pandas. Работа с данными / М. И. Абдрахманов. – 2-е изд. – 170 с.