

# Live Viewers of Top Kid T.V Analysis Using Python

## Overview

This Jupyter notebook presents an analysis of the top episodes of kids' TV programming based on the number of live viewers. The dataset includes information from three major networks: Disney, Nickelodeon, and Cartoon Network. Each row in the dataset represents a specific episode title, original movie, or event, along with the corresponding series or movie name.



## Import Library

```
In [1]: import pandas as pd
```

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

```
C:\Users\Syed Arif\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.25.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")
```

## Uploading Csv file

```
In [3]: df = pd.read_excel(r"C:\Users\Syed Arif\Desktop\Kids Show most Viewed .xlsx")
```

## Data Preprocessing

### .head()

head is used show to the By default = 5 rows in the dataset

```
In [4]: df.head()
```

Out[4]:

	Channel	Episode title	Series	Total Live Viewers	Release Year
0	Cartoon Network	Magic Ball of Buu	Dragon Ball Z	3649000	2001
1	Cartoon Network	Gather for the Tournament	Dragon Ball Z	3624000	2001
2	Cartoon Network	The Losses Begin	Dragon Ball Z	3510000	2001
3	Cartoon Network	Vegeta's Pride	Dragon Ball Z	3389000	2001
4	Cartoon Network	The Terror of Majin Buu	Dragon Ball Z	3349000	2001

### .tail()

tail is used to show rows by Descending order

```
In [5]: df.tail()
```

Out[5]:

	Channel	Episode title	Series	Total Live Viewers	Release Year
55	Nickelodeon	iSaved Your Life	iCarly	11164000	2010
56	Nickelodeon	iGot A Hot Room	iCarly	7740000	2010
57	Nickelodeon	Original Event	Kids' Choice Awards 2010	7639000	2010
58	Nickelodeon	Original Movie	Fred: The Movie	7597000	2010
59	Nickelodeon	iPsycho	iCarly	7518000	2010

### .shape

It show the total no of rows & Column in the dataset

```
In [6]: df.shape
```

Out[6]: (60, 5)

## .Columns

It show the no of each Column

```
In [7]: df.columns
```

```
Out[7]: Index(['Channel', 'Episode title', 'Series', 'Total Live Viewers',  
              'Release Year'],  
             dtype='object')
```

## .dtypes

This Attribute show the data type of each column

```
In [8]: df.dtypes
```

```
Out[8]: Channel                object  
        Episode title         object  
        Series                object  
        Total Live Viewers    int64  
        Release Year          int64  
        dtype: object
```

## .unique()

In a column, It show the unique value of specific column.

```
In [9]: df["Episode title"].unique()
```

```
Out[9]: array(['Magic Ball of Buu', 'Gather for the Tournament',  
              'The Losses Begin', 'Vegeta's Pride', 'The Terror of Majin Buu',  
              'Meal Time', 'The Last of Mercenary Tao', 'Race Against Time',  
              'Ambush', 'Rising Malevolence', 'The Last Episode, Really!',  
              'The Mystery Begins', 'Alien Swarm', 'Million Dollar Babies',  
              'Curse of the Lake Monster', 'Original Movie', 'Burning Low',  
              'Finn the Human/Jake the Dog', 'Day of the Great Devourer',  
              'Wake Up', 'Country Cousins', 'On the Road Again?',  
              "That's So Suite Life of Hannah Montana", 'Rollercoaster',  
              'Me and Mr. Jonas and Mr. Jonas and Mr. Jonas',  
              'New Kid in The Town', 'Crazy 10-Minute Sale',  
              'Wizards on Deck with Hannah Montana', 'He Could Be The One',  
              'Lost At Sea', 'Paint by Committee', 'I'll Always Remember You',  
              'Walk a Mile in My Pants', 'Wizards vs. Werewolves', 'Start It Up',  
              'Twister: Part 3', 'Wherever I Go',  
              'Who Will Be the Family Wizard', 'Special Delivery', 'Star Wars',  
              'All Grown Up', 'The Sponge Who Could Fly',  
              'Have You Seen This Snail?', 'Duces and Dragons',  
              'Atlantis SquarePantis', 'Fairly OddBaby',  
              'What Ever Happened to SpongeBob?', 'iGo to Japan',  
              'Original Event', 'Iquit iCarly', 'iFight Shelby Marx',  
              'Truth or Square', 'iThink They Kissed', 'iSaved Your Life',  
              'iGot A Hot Room', 'iPsycho'], dtype=object)
```

## .nunique()

It will show the total no of unique value from whole data frame

```
In [10]: df.nunique()
```

```
Out[10]: Channel          3  
         Episode title    56  
         Series          32  
         Total Live Viewers 56  
         Release Year     12  
         dtype: int64
```

## .describe()

It show the Count, mean , median etc

```
In [11]: df.describe()
```

```
Out[11]:
```

	Total Live Viewers	Release Year
count	6.000000e+01	60.000000
mean	6.591817e+06	2007.866667
std	2.355390e+06	3.306217
min	3.283000e+06	2001.000000
25%	3.971250e+06	2007.000000
50%	7.210000e+06	2009.000000
75%	7.771500e+06	2010.000000
max	1.191300e+07	2014.000000

## .value\_counts

It Shows all the unique values with their count

```
In [12]: df["Channel"].value_counts()
```

```
Out[12]: Cartoon Network    20  
Disney                    20  
Nickelodeon                20  
Name: Channel, dtype: int64
```

## .isnull()

It shows the how many null values

```
In [13]: df.isnull()
```

Out[13]:

	Channel	Episode title	Series	Total Live Viewers	Release Year
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False
10	False	False	False	False	False
11	False	False	False	False	False
12	False	False	False	False	False
13	False	False	False	False	False
14	False	False	False	False	False
15	False	False	False	False	False
16	False	False	False	False	False
17	False	False	False	False	False
18	False	False	False	False	False
19	False	False	False	False	False
20	False	False	False	False	False
21	False	False	False	False	False
22	False	False	False	False	False
23	False	False	False	False	False
24	False	False	False	False	False
25	False	False	False	False	False
26	False	False	False	False	False
27	False	False	False	False	False
28	False	False	False	False	False
29	False	False	False	False	False
30	False	False	False	False	False
31	False	False	False	False	False
32	False	False	False	False	False
33	False	False	False	False	False
34	False	False	False	False	False
35	False	False	False	False	False

	Channel	Episode title	Series	Total Live Viewers	Release Year
36	False	False	False	False	False
37	False	False	False	False	False
38	False	False	False	False	False
39	False	False	False	False	False
40	False	False	False	False	False
41	False	False	False	False	False
42	False	False	False	False	False
43	False	False	False	False	False
44	False	False	False	False	False
45	False	False	False	False	False
46	False	False	False	False	False
47	False	False	False	False	False
48	False	False	False	False	False
49	False	False	False	False	False
50	False	False	False	False	False
51	False	False	False	False	False
52	False	False	False	False	False
53	False	False	False	False	False
54	False	False	False	False	False
55	False	False	False	False	False
56	False	False	False	False	False
57	False	False	False	False	False
58	False	False	False	False	False
59	False	False	False	False	False

**.info()**

To Show Data type of each column



```
In [14]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Channel                60 non-null    object
 1   Episode title          60 non-null    object
 2   Series                 60 non-null    object
 3   Total Live Viewers     60 non-null    int64
 4   Release Year           60 non-null    int64
dtypes: int64(2), object(3)
memory usage: 2.5+ KB
```

## Is there any Null value present in any Column ? Show with heatmap

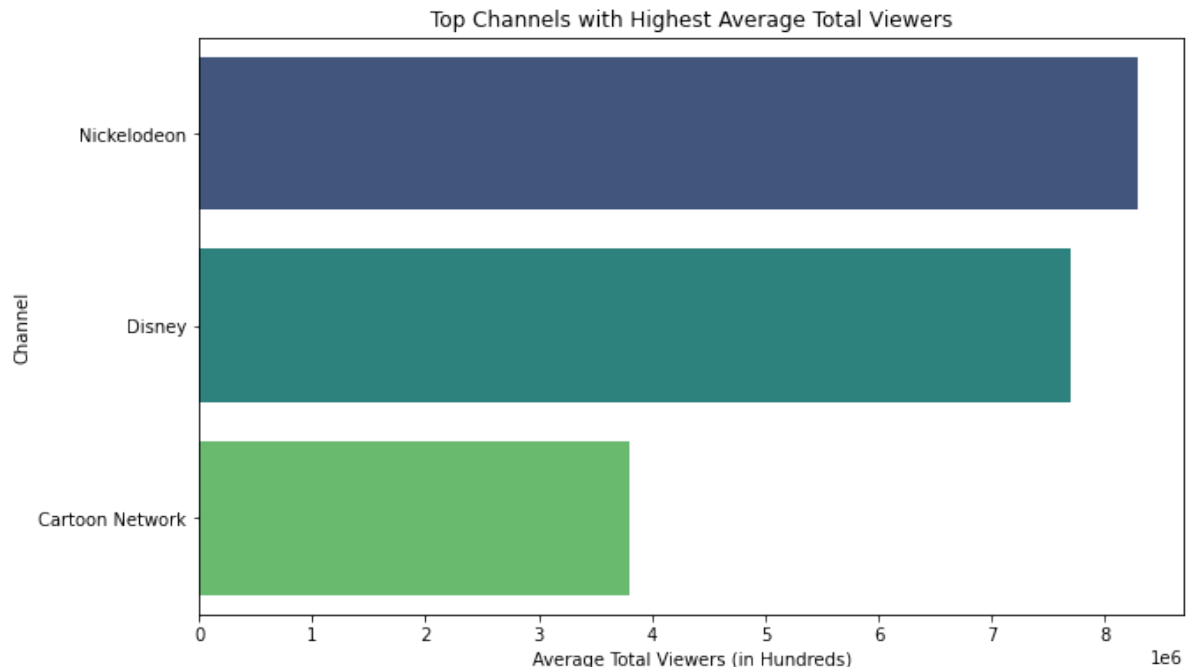
```
In [15]: df.isnull().sum()
```

```
Out[15]: Channel                0
         Episode title          0
         Series                 0
         Total Live Viewers     0
         Release Year           0
         dtype: int64
```

```
In [16]: # Convert 'Total viewers' to hundreds format
         #df['Total Live Viewers'] = df['Total Live Viewers'] // 100
```

## Question 1: Top channels with highest average total viewers

```
In [17]: top_channels = df.groupby('Channel')['Total Live Viewers'].mean().sort_values(
plt.figure(figsize=(10, 6))
sns.barplot(x=top_channels.values, y=top_channels.index, palette="viridis")
plt.xlabel("Average Total Viewers (in Hundreds)")
plt.ylabel("Channel")
plt.title("Top Channels with Highest Average Total Viewers")
plt.show()
```



## Question 2: Episode or series with highest total viewers

```
In [18]: highest_viewers_episode = df[df['Total Live Viewers'] == df['Total Live Viewers'].max()
print("Episode or series with highest total viewers:\n", highest_viewers_episode)
```

Episode or series with highest total viewers:

	Channel	Episode title	Series	Total Live Viewers	Release Year
40	Nickelodeon	All Grown Up	Rugrats	11913000	2001

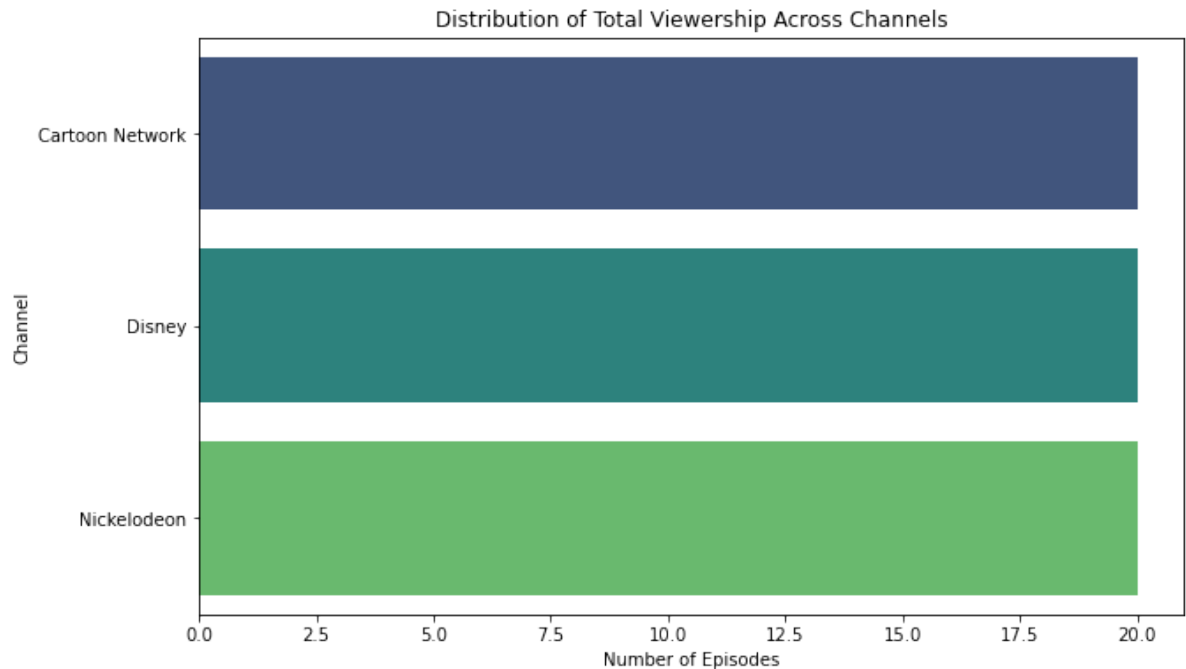
## Question 3: Correlation between release year and total viewership

```
In [19]: correlation = df['Release Year'].corr(df['Total Live Viewers'])
correlation
```

Out[19]: 0.10726222451782469

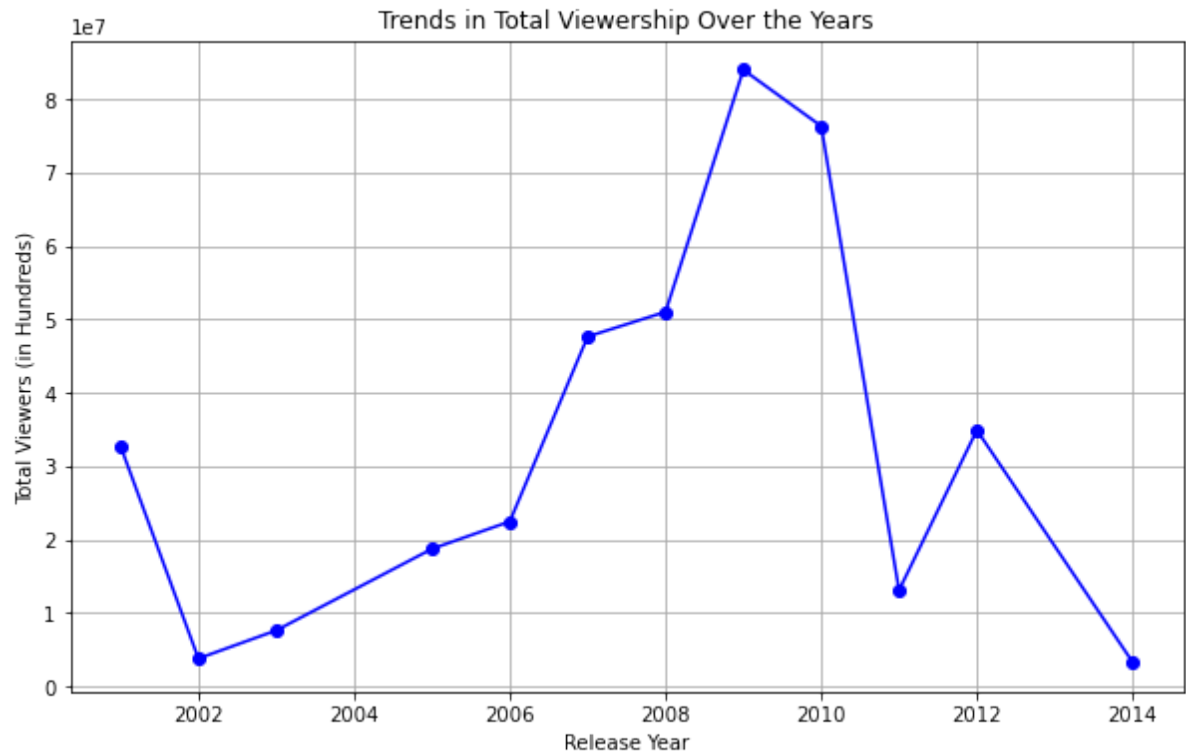
## Question 4: Distribution of total viewership across channels

```
In [20]: channel_distribution = df['Channel'].value_counts()
plt.figure(figsize=(10, 6))
sns.countplot(y="Channel", data=df, palette="viridis")
plt.xlabel("Number of Episodes")
plt.ylabel("Channel")
plt.title("Distribution of Total Viewership Across Channels")
plt.show()
```



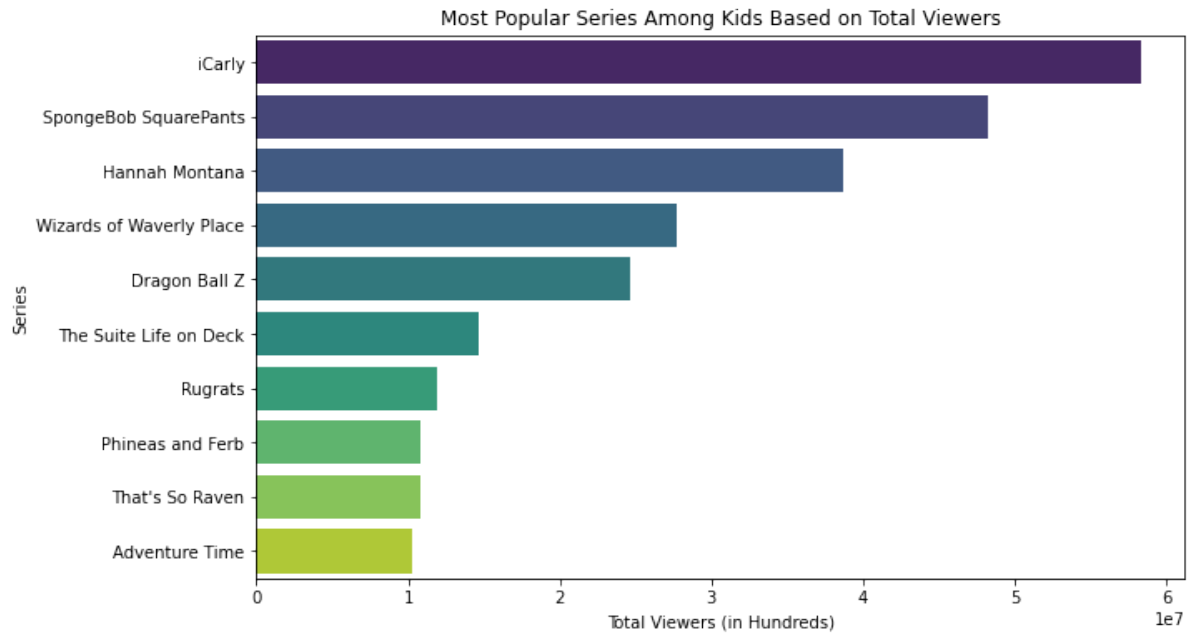
## Question 5: Trends in total viewership over the years

```
In [21]: total_viewers_trends = df.groupby('Release Year')['Total Live Viewers'].sum()
plt.figure(figsize=(10, 6))
total_viewers_trends.plot(marker='o', color='b')
plt.xlabel("Release Year")
plt.ylabel("Total Viewers (in Hundreds)")
plt.title("Trends in Total Viewership Over the Years")
plt.grid(True)
plt.show()
```



**Question 6: Most popular series among kids based on total viewers**

```
In [22]: popular_series = df.groupby('Series')['Total Live Viewers'].sum().sort_values(
plt.figure(figsize=(10, 6))
sns.barplot(x=popular_series.values, y=popular_series.index, palette="viridis")
plt.xlabel("Total Viewers (in Hundreds)")
plt.ylabel("Series")
plt.title("Most Popular Series Among Kids Based on Total Viewers")
plt.show()
```



## Question 7: Seasonal pattern in total viewership

```
In [23]: seasonal_pattern = df.groupby(df['Release Year'] // 10 * 10)['Total Live Viewers']
plt.figure(figsize=(10, 6))
seasonal_pattern.plot(marker='o', color='g')
plt.xlabel("Release Decade")
plt.ylabel("Average Total Viewers (in Hundreds)")
plt.title("Seasonal Pattern in Total Viewership")
plt.grid(True)
plt.show()
```

