

```
import pandas as pd
```

```
df=pd.read_csv("/content/drive/MyDrive/titles.csv")
```

```
df.ndim
```

```
↗ 2
```

```
df.size
```

```
↗ 87750
```


```
df.shape
```

```
↗ (5850, 15)
```


```
df.info()
```

```
↗ <class 'pandas.core.frame.DataFrame'>  
RangeIndex: 5850 entries, 0 to 5849  
Data columns (total 15 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0   id                    5850 non-null   object  
1   title                 5849 non-null   object  
2   type                  5850 non-null   object  
3   description           5832 non-null   object  
4   release_year         5850 non-null   int64  
5   age_certification    3231 non-null   object  
6   runtime              5850 non-null   int64  
7   genres               5850 non-null   object  
8   production_countries 5850 non-null   object  
9   seasons              2106 non-null   float64  
10  imdb_id              5447 non-null   object  
11  imdb_score           5368 non-null   float64  
12  imdb_votes           5352 non-null   float64  
13  tmdb_popularity      5759 non-null   float64  
14  tmdb_score           5539 non-null   float64  
dtypes: float64(5), int64(2), object(8)  
memory usage: 685.7+ KB
```

```
df.head()
```



	id	title	type	description	release_year	age_certification	runtime
0	ts300399	Five Came Back: The Reference Films	SHOW	This collection includes 12 World War II-era p...	1945	TV-MA	5'
1	tm84618	Taxi Driver	MOVIE	A mentally unstable Vietnam War veteran works ...	1976	R	114'
2	tm154986	Deliverance	MOVIE	Intent on seeing the Cahulawassee River before...	1972	R	105'
3	tm127384	Monty Python and the Holy Grail	MOVIE	King Arthur, accompanied by his squire, recrui...	1975	PG	9'
4	tm120801	The Dirty Dozen	MOVIE	12 American military prisoners in World War II...	1967	NaN	150'



Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

Missing values before cleaning :

```
df.isnull()
```



	id	title	type	description	release_year	age_certification	runtime	genre
0	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	True	False	Fals
...
5845	False	False	False	False	False	True	False	Fals
5846	False	False	False	False	False	True	False	Fals
5847	False	False	False	False	False	True	False	Fals
5848	False	False	False	False	False	False	False	Fals
5849	False	False	False	False	False	True	False	Fals

5850 rows x 15 columns



```
df.isnull().sum()
```



	0
id	0
title	1
type	0
description	18
release_year	0
age_certification	2619
runtime	0
genres	0
production_countries	0
seasons	3744
imdb_id	403
imdb_score	482
imdb_votes	498
tmdb_popularity	91
tmdb_score	311

dtype: int64

Removinng duplicates

```
df.drop_duplicates(inplace=True)
```

Converting to lowercase

```
df['production_countries'] = df['production_countries'].str.lower()
```

```
df['genres'] = df['genres'].str.lower()
```

Removing brackets [] and single quotes ' using regex

```
df['genres'] = df['genres'].str.replace("[\[\]']", '', regex=True)
```

```
df['production_countries'] = df['production_countries'].str.replace("[\[\]']", '', regex=T
```

Removing leading and trailing spaces

```
df['genres'] = df['genres'].str.strip()
```

```
df['production_countries'] = df['production_countries'].str.strip()
```

After these operations :

```
df.head(2)
```



	id	title	type	description	release_year	age_certification	runtime
0	ts300399	Five Came Back: The Reference Films	SHOW	This collection includes 12 World War II-era p...	1945	TV-MA	51
1	tm84618	Taxi Driver	MOVIE	A mentally unstable Vietnam War veteran works ...	1976	R	114

Next steps:

[Generate code with df](#)
[View recommended plots](#)
[New interactive sheet](#)

Rename column headers (lowercase, replace spaces with underscores)

```
df.columns = df.columns.str.lower().str.replace(' ', '_')
```

```
df.head(2)
```



	id	title	type	description	release_year	age_certification	runtime
0	ts300399	Five Came Back: The Reference Films	SHOW	This collection includes 12 World War II-era p...	1945	TV-MA	51
1	tm84618	Taxi Driver	MOVIE	A mentally unstable Vietnam War veteran works ...	1976	R	114

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

Fixing data types

```
df['seasons'] = df['seasons'].astype('Int64')
```

Handling missing values


```
df['imdb_score'].fillna(df['imdb_score'].mean(), inplace=True)
```

```
df['title'].fillna("Unknown", inplace=True)
```

```
df['description'].fillna("Unknown", inplace=True)
```

```
df['age_certification'].fillna(df['age_certification'].mode()[0], inplace=True)
```

```
df['seasons'].fillna(df['seasons'].mode()[0], inplace=True)
```




<ipython-input-58-5778eae63dd5>:1: FutureWarning: A value is trying to be set on a copy of an object. This behavior will change in pandas 3.0. This inplace method will never work because the operation will not be performed on the original object.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({c

```
df['seasons'].fillna(df['seasons'].mode()[0], inplace=True)
```

```
num_cols = ['imdb_votes', 'tmdb_popularity', 'tmdb_score']
for col in num_cols:
    df[col].fillna(df[col].mean(), inplace=True)
```




<ipython-input-62-28b46205f59d>:3: FutureWarning: A value is trying to be set on a copy of an object. This behavior will change in pandas 3.0. This inplace method will never work because the operation will not be performed on the original object.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({c

```
df[col].fillna(df[col].mean(), inplace=True)
```

```
df['imdb_id'].fillna(df['imdb_id'].mode()[0], inplace=True)
```




<ipython-input-63-e79e403f3d2c>:1: FutureWarning: A value is trying to be set on a copy. The behavior will change in pandas 3.0. This inplace method will never work because th

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({c

```
df['imdb_id'].fillna(df['imdb_id'].mode()[0], inplace=True)
```

After filling missing values

```
df.isnull().sum()
```



	0
id	0
title	0
type	0
description	0
release_year	0
age_certification	0
runtime	0
genres	0
production_countries	0
seasons	0
imdb_id	0
imdb_score	0
imdb_votes	0