Predicting IMDb Score

Date - 16/10/2023

Team ID - 3866

Importing Dependencies

In [14]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, mean_absolute_error,mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
import chardet
```

Loading Dataset

In [15]:
```
with open('NetflixOriginals.csv', 'rb') as f:
    result = chardet.detect(f.read())  # or readline if the file is large


dataset = pd.read_csv('NetflixOriginals.csv', encoding=result['encoding'])
```

Data Exploration

In [16]:
```
dataset
```

Out[16]:

	Title	Genre	Premiere	Runtime	IMDB Score	Language
0	Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese
1	Dark Forces	Thriller	August 21, 2020	81	2.6	Spanish
2	The App	Science fiction/Drama	December 26, 2019	79	2.6	Italian
3	The Open House	Horror thriller	January 19, 2018	94	3.2	English
4	Kaali Khuhi	Mystery	October 30, 2020	90	3.4	Hindi
...	...	...	...	...	...	...
579	Taylor Swift: Reputation Stadium Tour	Concert Film	December 31, 2018	125	8.4	English
580	Winter on Fire: Ukraine's Fight for Freedom	Documentary	October 9, 2015	91	8.4	English/Ukranian/Russian
581	Springsteen on Broadway	One-man show	December 16, 2018	153	8.5	English
582	Emicida: AmarElo - It's All For Yesterday	Documentary	December 8, 2020	89	8.6	Portuguese
583	David Attenborough: A Life on Our Planet	Documentary	October 4, 2020	83	9.0	English

584 rows × 6 columns

In [17]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 584 entries, 0 to 583
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Title       584 non-null    object 
 1   Genre       584 non-null    object 
 2   Premiere    584 non-null    object 
 3   Runtime     584 non-null    int64  
 4   IMDB Score  584 non-null    float64
 5   Language    584 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 27.5+ KB
```

In [18]:
```
dataset.describe()
```

Out[18]:

	Runtime	IMDB Score
count	584.000000	584.000000
mean	93.577055	6.271747
std	27.761683	0.979256
min	4.000000	2.500000
25%	86.000000	5.700000
50%	97.000000	6.350000
75%	108.000000	7.000000
max	209.000000	9.000000

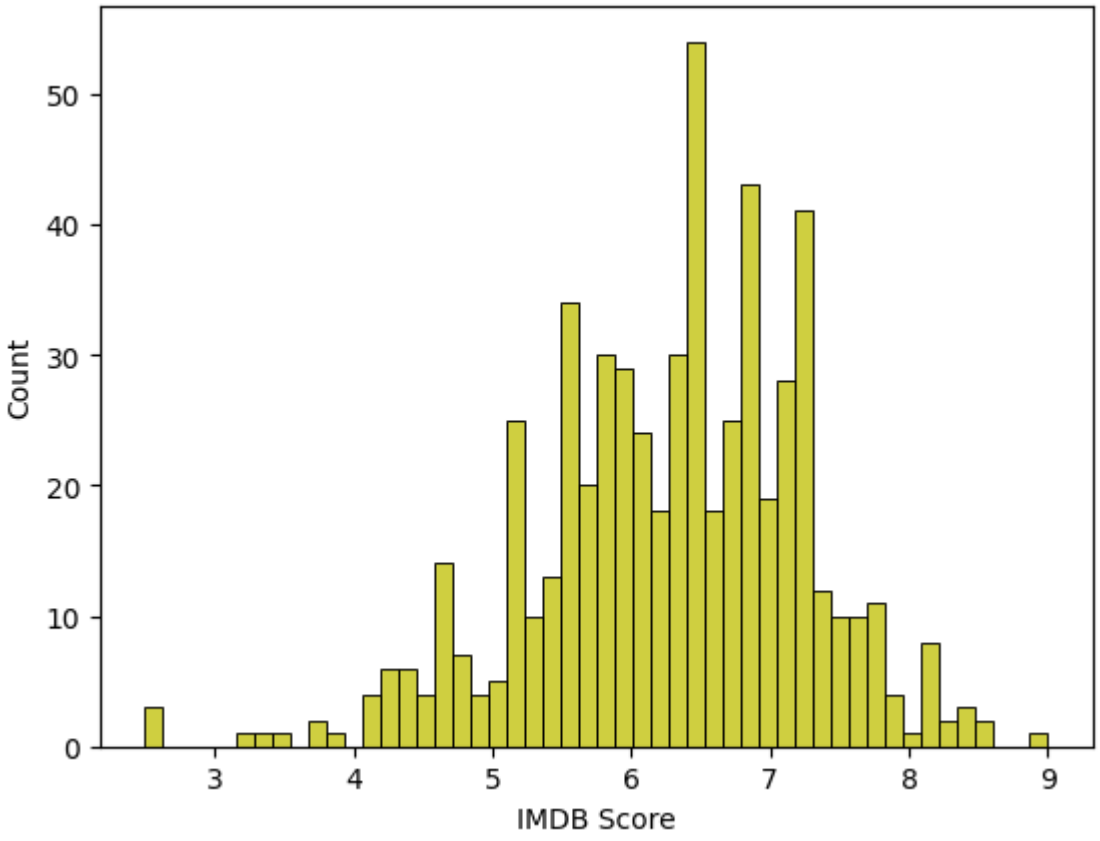
In [19]:
```
dataset.columns
```

Out[19]:
```
Index(['Title', 'Genre', 'Premiere', 'Runtime', 'IMDB Score', 'Language'], dtype='object')
```

Pre-Processing and Visualisation of Data

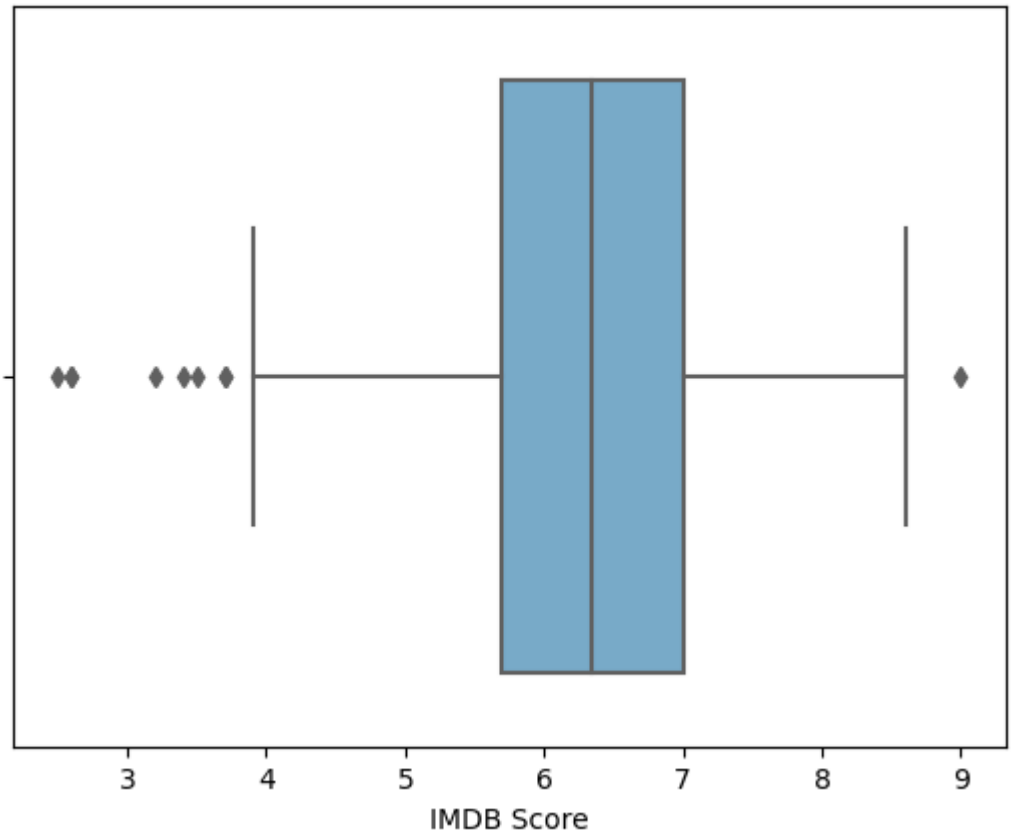
In [23]:
```
sns.histplot(dataset, x='IMDB Score', bins=50, color='y')
```

Out[23]:
```
<Axes: xlabel='IMDB Score', ylabel='Count'>
```



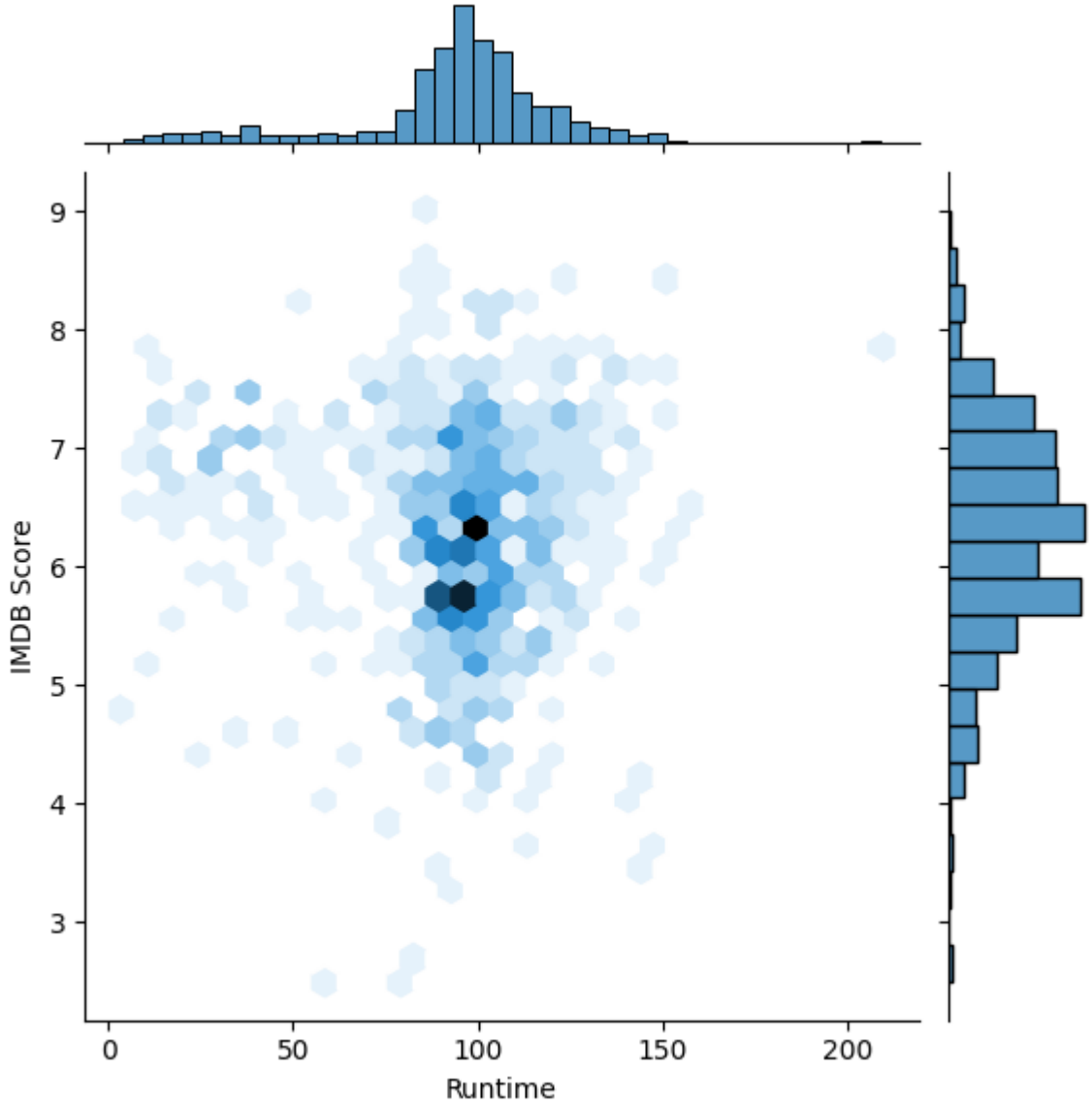
In [24]:
```
sns.boxplot(dataset, x='IMDB Score',  palette='Blues')
```

Out[24]:
```
<Axes: xlabel='IMDB Score'>
```



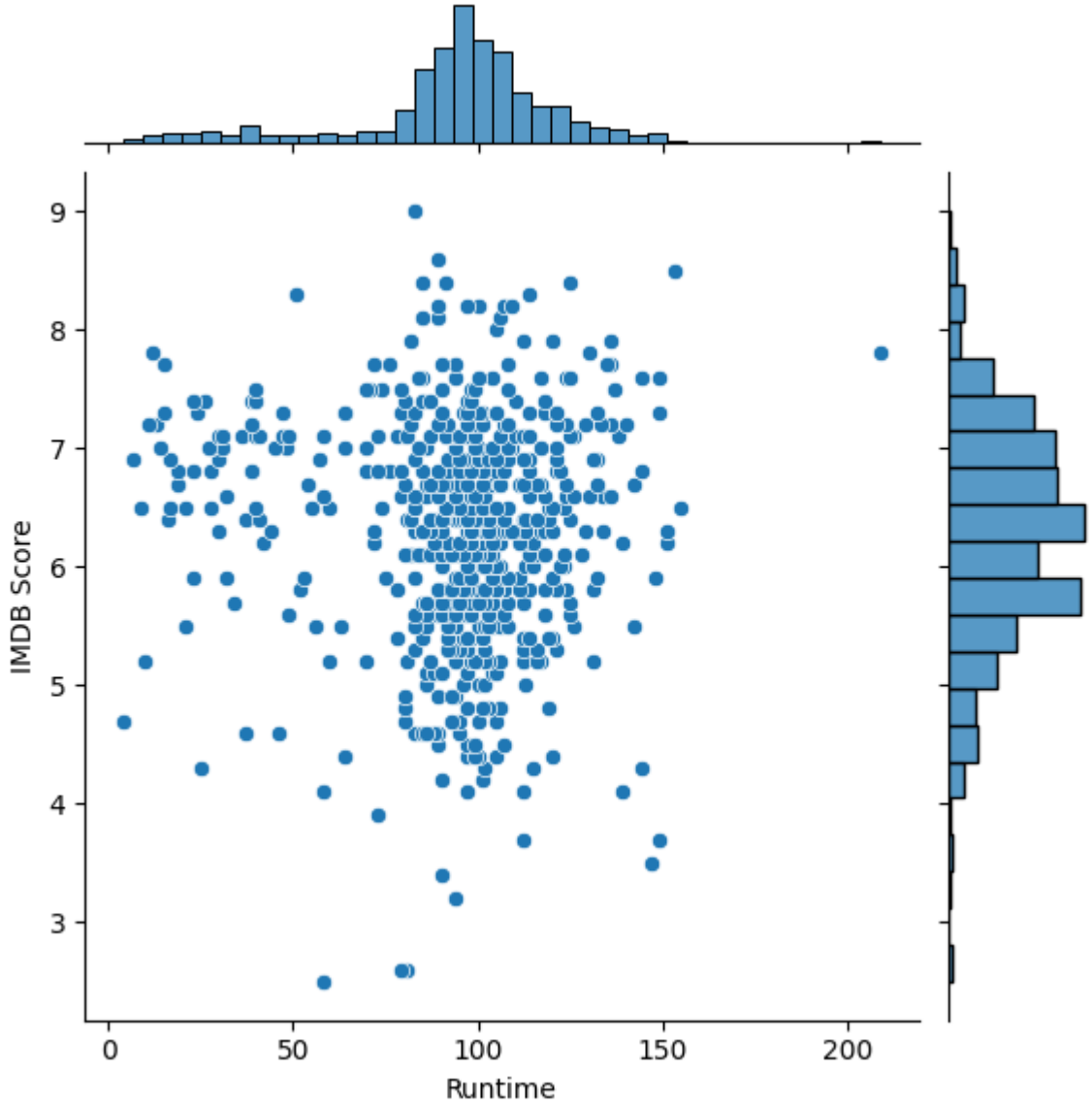
In [25]:
```
sns.jointplot(dataset, x='Runtime', y='IMDB Score', kind='hex')
```

Out[25]:
```
<seaborn.axisgrid.JointGrid at 0x1c319bf9750>
```



In [26]:
```
sns.jointplot(dataset, x='Runtime', y='IMDB Score')
```

Out[26]:
```
<seaborn.axisgrid.JointGrid at 0x1c319ea6b10>
```

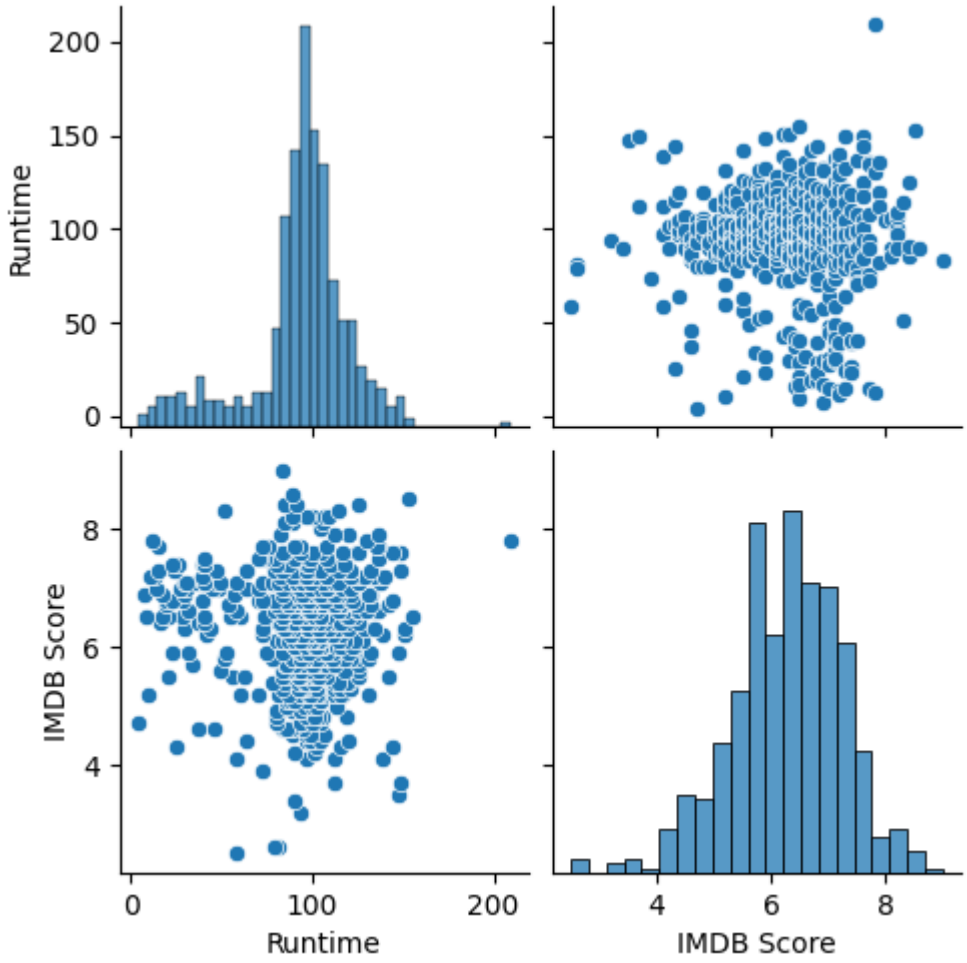


In [27]:
```
plt.figure(figsize=(12,8))
sns.pairplot(dataset)
```

```
C:\Users\shaba\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qb
z5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\seaborn\axisgri
d.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

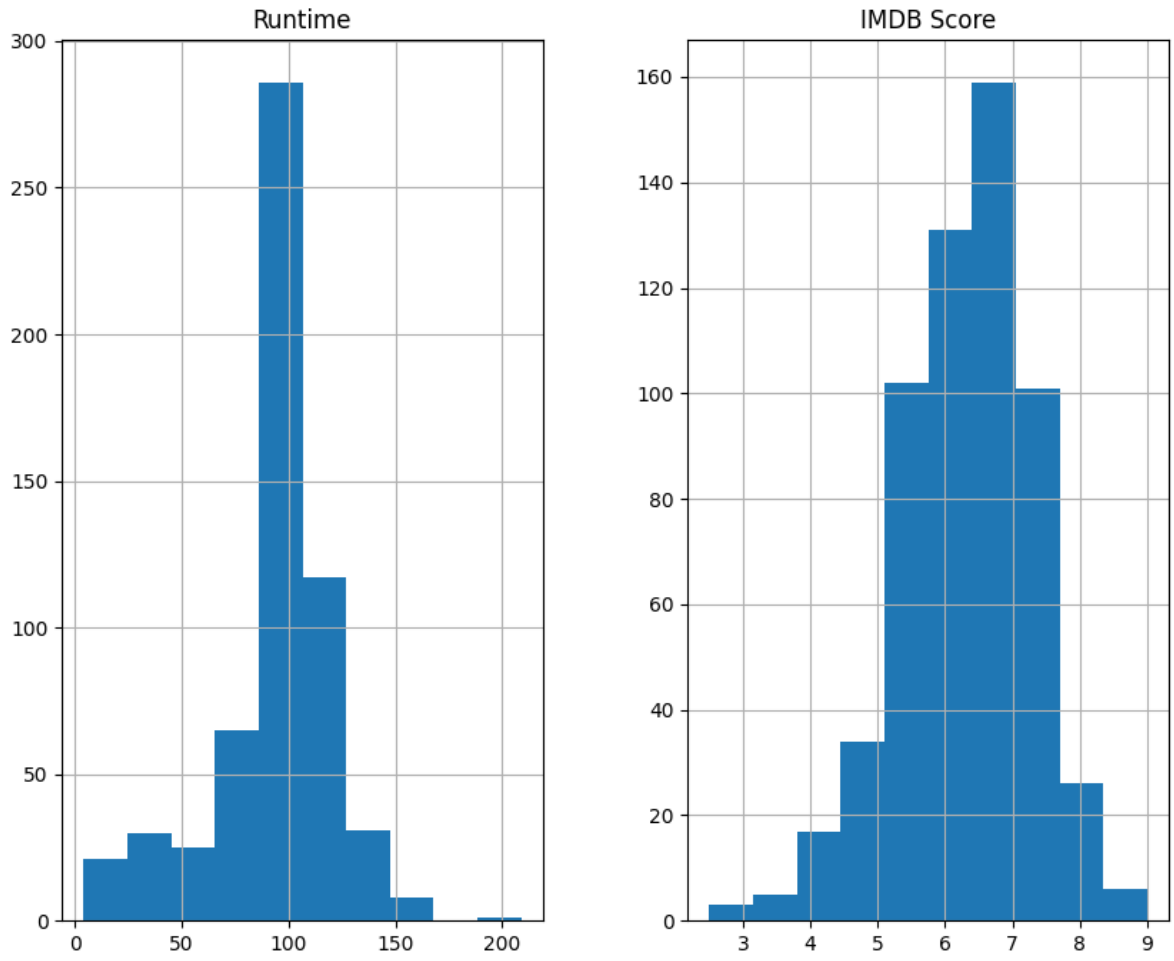
Out[27]:
```
<seaborn.axisgrid.PairGrid at 0x1c319e80790>

<Figure size 1200x800 with 0 Axes>
```



In [28]:
```
dataset.hist(figsize=(10,8))
```

Out[28]:
```
array([[<Axes: title={'center': 'Runtime'}>,
        <Axes: title={'center': 'IMDB Score'}>]], dtype=object)
```



Visualising Correlation

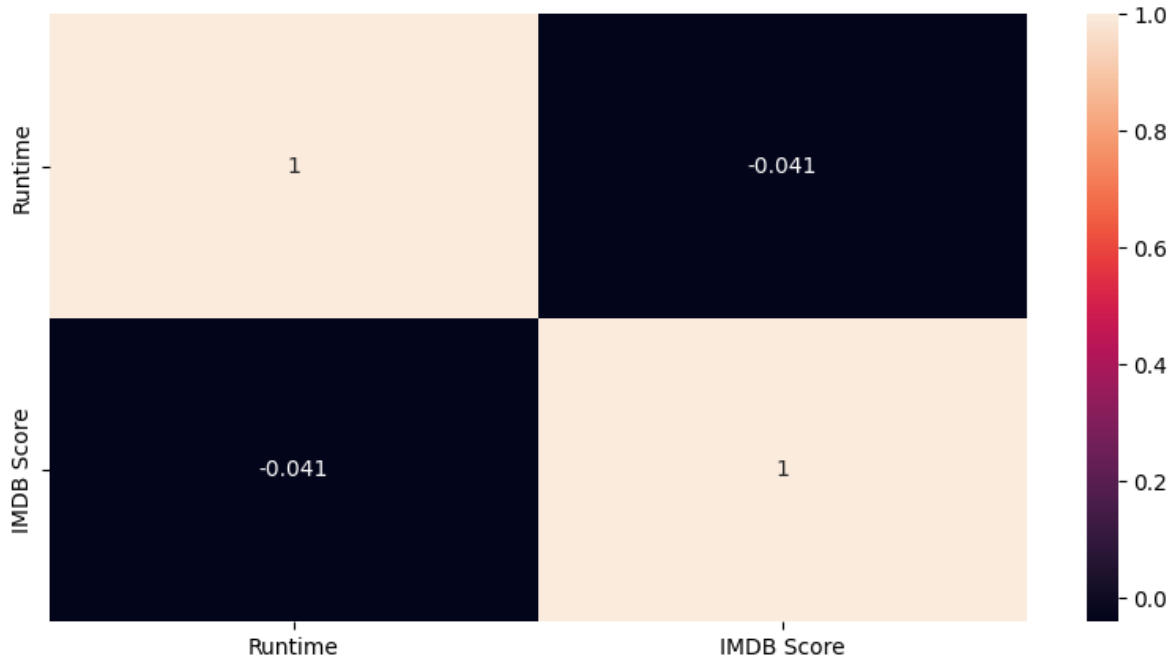
In [54]:
```
dataset[['Runtime','IMDB Score']].corr()
```

Out[54]:

	Runtime	IMDB Score
Runtime	1.000000	-0.040896
IMDB Score	-0.040896	1.000000

In [55]:
```
plt.figure(figsize=(10,5))
sns.heatmap(dataset[['Runtime','IMDB Score']].corr(), annot=True)
```

Out[55]: <Axes: >



Thank You

Type *Markdown* and LaTeX: α^2