```
#Experiment 2
#Name: Yash Shah
#Class: AIA 7
#Roll No: 64
#Enrollment: ADT23SOCB1341
import numpy as np # For numerical operations
import pandas as pd # For data manipulation & analysis
import matplotlib.pyplot as plt  # For data visualization
```

```
df=pd.read_csv("n_movies.csv.zip") # Reads the CSV file into a DataFrame named df
df
```

	title	year	certificate	duration	genre	rating	description	stars	votes
0	Cobra Kai	(2018–)	TV-14	30 min	Action, Comedy, Drama	8.5	Decades after their 1984 All Valley Karate Tou...	['Ralph Macchio, ', 'William Zabka, ', 'Courtn...	177,031
1	The Crown	(2016–)	TV-MA	58 min	Biography, Drama, History	8.7	Follows the political rivalries and romance of...	['Claire Foy, ', 'Olivia Colman, ', 'Imelda St...	199,885
2	Better Call Saul	(2015–2022)	TV-MA	46 min	Crime, Drama	8.9	The trials and tribulations of criminal lawyer...	['Bob Odenkirk, ', 'Rhea Seehorn, ', 'Jonathan...	501,384
3	Devil in Ohio	(2022)	TV-MA	356 min	Drama, Horror, Mystery	5.9	When a psychiatrist shelters a mysterious cult...	['Emily Deschanel, ', 'Sam Jaeger, ', 'Gerardo...	9,773
4	Cyberpunk: Edgerunners	(2022–)	TV-MA	24 min	Animation, Action, Adventure	8.6	A Street Kid trying to survive in a technology...	['Zach Aguilar, ', 'Kenichiro Ohashi, ', 'Emi ...	15,413
...	...	...	...	...	...	...	...	...	...
9952	The Imperfects	(2022–)	TV-MA	45 min	Action, Adventure, Drama	6.3	After an experimental gene therapy turns them ...	['Morgan Taylor Campbell, ', 'Italia Ricci, ',...	3,130
9953	The Walking Dead	(2010–2022)	TV-MA	44 min	Drama, Horror, Thriller	8.1	Sheriff Deputy Rick Grimes wakes up from a com...	['Andrew Lincoln, ', 'Norman Reedus, ', 'Melis...	970,067
9954	The Crown	(2016–)	TV-MA	58 min	Biography, Drama, History	8.7	Follows the political rivalries and romance of...	['Claire Foy, ', 'Olivia Colman, ', 'Imelda St...	199,898
								['Jared Padalecki, '	

Next steps: Generate code with df View recommended plots New interactive sheet

```
df.head()  # Displays first 5 rows of the dataset
```

	title	year	certificate	duration	genre	rating	description	stars	votes
0	Cobra Kai	(2018–)	TV-14	30 min	Action, Comedy, Drama	8.5	Decades after their 1984 All Valley Karate Tou...	['Ralph Macchio, ', 'William Zabka, ', 'Courtn...	177,031
1	The Crown	(2016–)	TV-MA	58 min	Biography, Drama, History	8.7	Follows the political rivalries and romance of...	['Claire Foy, ', 'Olivia Colman, ', 'Imelda St...	199,885
2	Better Call Saul	(2015–2022)	TV-MA	46 min	Crime, Drama	8.9	The trials and tribulations of criminal lawyer...	['Bob Odenkirk, ', 'Rhea Seehorn, ', 'Jonathan...	501,384
3	Devil in Ohio	(2022)	TV-MA	356 min	Drama, Horror, Mystery	5.9	When a psychiatrist shelters a mysterious cult...	['Emily Deschanel, ', 'Sam Jaeger, ', 'Gerardo...	9,773

Next steps: Generate code with df View recommended plots New interactive sheet

```
df.tail() # Displays last 5 rows of the dataset
```

	title	year	certificate	duration	genre	rating	description	stars	votes
9952	The Imperfects	(2022–)	TV-MA	45 min	Action, Adventure, Drama	6.3	After an experimental gene therapy turns them ...	['Morgan Taylor Campbell, ', 'Italia Ricci, ',...	3,130
9953	The Walking Dead	(2010–2022)	TV-MA	44 min	Drama, Horror, Thriller	8.1	Sheriff Deputy Rick Grimes wakes up from a com...	['Andrew Lincoln, ', 'Norman Reedus, ', 'Melis...	970,067
9954	The Crown	(2016–)	TV-MA	58 min	Biography, Drama, History	8.7	Follows the political rivalries and romance of...	['Claire Foy, ', 'Olivia Colman, ', 'Imelda St...	199,898
9955	Supernatural	(2005–2020)	TV-14	44 min	Drama, Fantasy, Horror	8.4	Two brothers follow their father's footsteps a...	['Jared Padalecki, ', 'Jensen Ackles, ', 'Jim ...	439,601

```
df.shape # Shows number of rows & columns (rows, columns)
```

```
(9957, 9)
```

```
df.columns # Lists all column names
```

```
Index(['title', 'year', 'certificate', 'duration', 'genre', 'rating',
       'description', 'stars', 'votes'],
      dtype='object')
```

```
df.info() # Data types, non-null counts, memory usage
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9957 entries, 0 to 9956
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   title        9957 non-null   object 
 1   year         9430 non-null   object 
 2   certificate  6504 non-null   object 
 3   duration     7921 non-null   object 
 4   genre        9884 non-null   object 
 5   rating       8784 non-null   float64
 6   description  9957 non-null   object 
 7   stars        9957 non-null   object 
 8   votes        8784 non-null   object 
dtypes: float64(1), object(8)
memory usage: 700.2+ KB
```

```
df.describe() # Statistical summary of numeric columns
```

	rating
count	8784.000000
mean	6.764515
std	1.214840
min	1.700000
25%	6.100000
50%	6.900000
75%	7.600000
max	9.900000

```
df.isnull().sum().sort_values(ascending=False) # Counts missing values in each column and sorts descending (most missing first)
```

	0
certificate	3453
duration	2036
votes	1173
rating	1173
year	527
genre	73
title	0
description	0
stars	0

dtype: int64

```
df = df.drop(columns=['certificate', 'description']) # Drops columns with many null values
```

```
df = df.dropna() # Removes rows containing any null values
```

```
print(df.isnull().sum()) # Verifies no missing values remain
```

```
title       0
year        0
duration    0
genre       0
rating      0
stars       0
votes       0
dtype: int64
```

```
X = df.drop(columns=['rating'])   # Features: all columns except 'horror'
y = df['rating']                  # Target: the 'horror' column to predict
```

```
from sklearn.model_selection import train_test_split

# 80% training, 20% testing
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

print("Training Data Shape:", X_train.shape)
print("Testing Data Shape:", X_test.shape)
```

```
Training Data Shape: (6138, 6)
Testing Data Shape: (1535, 6)
```

Now, let's create a 'decade' column from the 'year' column to analyze the average rating by decade.

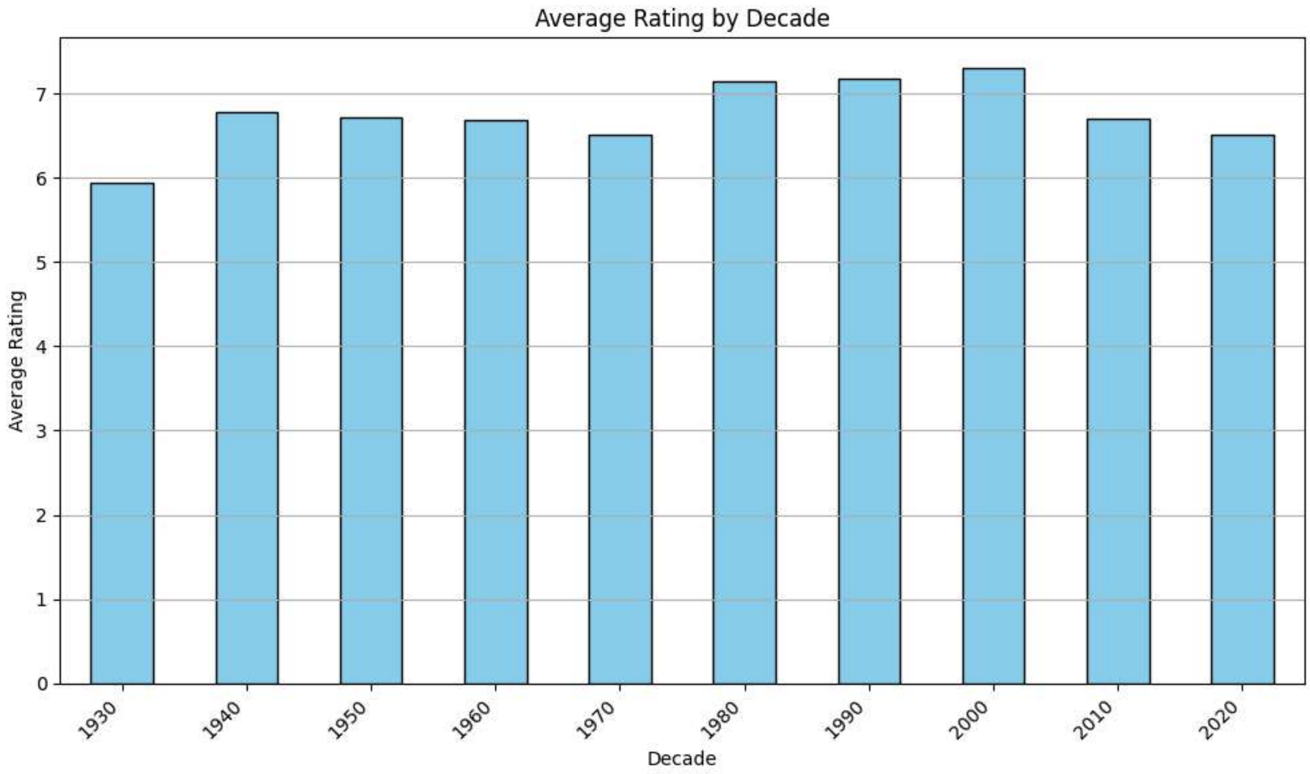
```
# Extract the decade from the 'year' column
# We need to clean up the 'year' column first to extract the year
df['year'] = df['year'].str.extract(r'(\d{4})').astype(float)
df['decade'] = (df['year'] // 10 * 10).astype(int)

# Calculate the average rating per decade
avg_rating_decade = df.groupby('decade')['rating'].mean()

# Display the average rating per decade
display(avg_rating_decade)
```

	rating
decade	
1930	5.933333
1940	6.781818
1950	6.707407
1960	6.681818
1970	6.506667
1980	7.145946

```
Finally, let's visualize the average rating by decade using a bar plot
```

Finally, let's visualize the average rating by decade using a bar plot.

```
plt.figure(figsize=(10, 6))
avg_rating_decade.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title('Average Rating by Decade')
plt.xlabel('Decade')
plt.ylabel('Average Rating')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```


Average Rating by Decade
Average Rating
Decade