

Data and Artificial Intelligence

Cyber Shujaa Program

Week 2 Assignment

Data Wrangling

Student Name: Austin Githinji

Student ID: CS-EH03-25417

Introduction

The purpose of this assignment was to gain hands-on experience in data wrangling by using the Netflix dataset from Kaggle. Data wrangling is an essential skill in data science that involves cleaning, structuring, and enriching raw data into a format suitable for analysis and visualization. This project involved working with the Netflix Movies and TV Shows dataset by Shivam Bansal on Kaggle. The dataset includes metadata about Netflix titles such as type, director, cast, country, release year, duration, and genres.

Tasks Completed

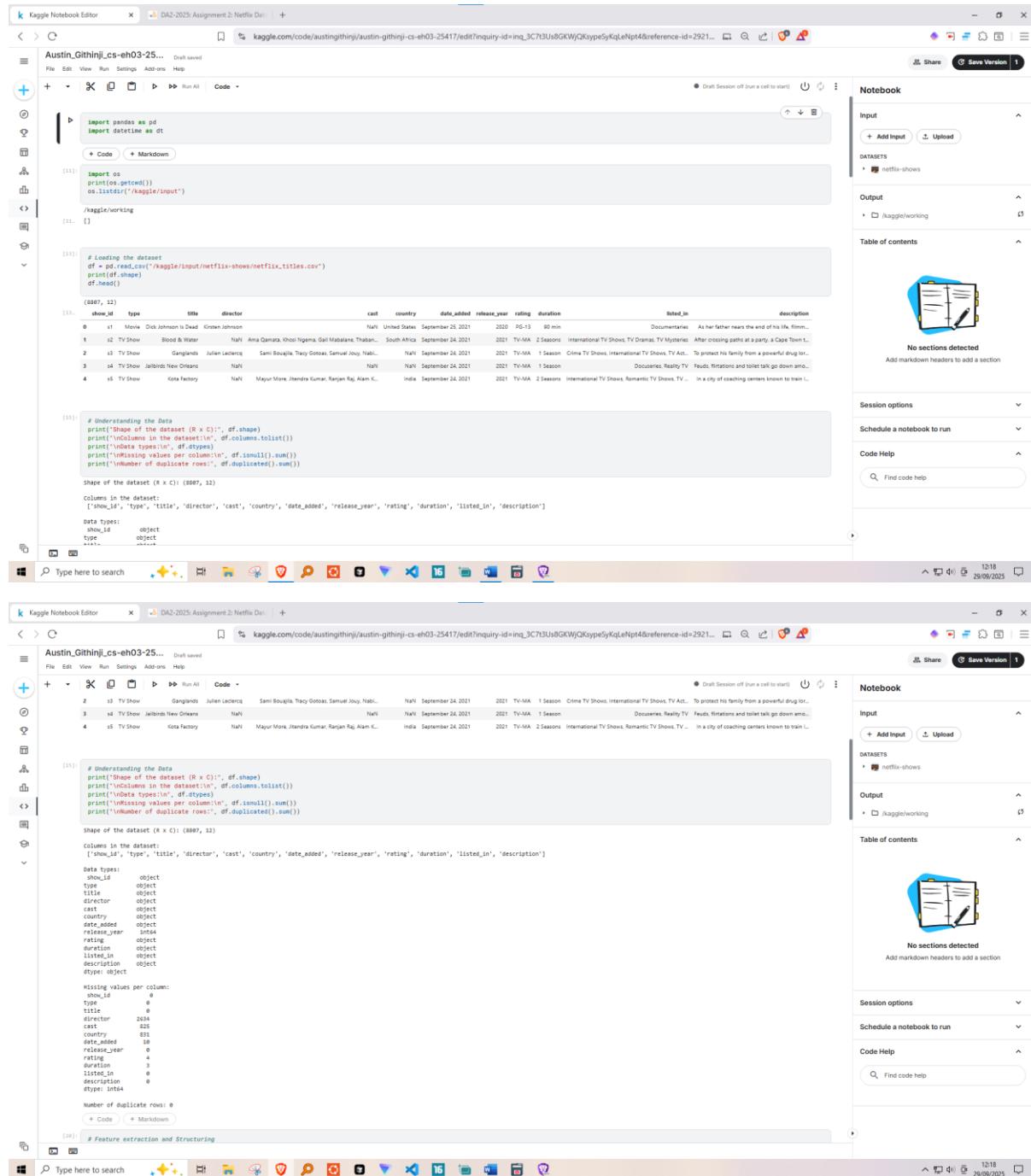
1. Loading and Exploring the Dataset

I loaded the dataset using pandas and checked its structure, shape, and missing values.

Code:

```
import pandas as pd  
  
import datetime as dt  
  
# Load dataset  
  
df = pd.read_csv("/kaggle/input/netflix-shows/netflix_titles.csv")  
  
# Explore structure  
  
print("Shape of the dataset (R x C):", df.shape)  
  
print("\nColumns in the dataset:\n", df.columns.tolist())  
  
print("\nData types:\n", df.dtypes)
```

```
print("\nMissing values per column:\n", df.isnull().sum())
print("\nNumber of duplicate rows:", df.duplicated().sum())
```



The screenshot shows two identical Kaggle Notebooks side-by-side. Both notebooks have the title "Austin_Githinji.cs-eh03-25..." and are running on the URL "kaggle.com/code/austingithinji/austin-githinji.cs-eh03-25417/edit?inquiry-id=ing_3C73Us8GKwQKsype5yKqLeNpt4&reference-id=2921...".

Code:

```
import pandas as pd
import datetime as dt

import os
print(os.getcwd())
os.listdir('/kaggle/input')

#kaggle/working
[]

# Loading the dataset
df = pd.read_csv('/kaggle/input/netflix-shows/netflix_titles.csv')
print(df.shape)
df.head()
```

Output (Top Notebook):

```
(8897, 12)
   show_id  type      title    director      cast    country  date_added  release_year  rating  duration listed_in      description
0       1  Movie  Dick Johnson is Dead  Kristen Johnson    NaN  United States  2020-09-25  2020-09-13  90 min  Documentaries  As his father nears the end of his life, film...
1       1  TV Show  Blood & Water  Gail O'Grady, Gail MacBane, Thalassia...  NaN  South Africa  2021-09-24  2021-09-14  130 min  Crime TV Shows, International TV Shows, TV Dramas, TV Mysteries  After crossing paths at a party, a Cape Town...
2       2  TV Show  Ganglands  Julian Lachowicz  Sami Bouajila, Tracy Gohaus, Samuel Jouy, Nabi...  NaN  South Africa  2021-09-24  2021-09-14  130 min  Crime TV Shows, International TV Shows, TV Dramas, TV Mysteries  To protect his family from a powerful drug lord, a...
3       3  TV Show  Jallikud New Orleans  NaN  NaN  September 24, 2021  2021-09-14  130 min  Documentaries, Reality TV  Feuds, filtrations and toilet talk go down amo...
4       4  TV Show  Kota Factory  NaN  Mayur More, Jitendra Kumar, Ranjan Raj, Alen K...  India  September 24, 2021  2021-09-14  130 min  International TV Shows, Romantic TV Shows, TV ...  In a city of coaching centers known to train...
```

Output (Bottom Notebook):

```
# Understanding the Data
print("Shape of the dataset (R x C):", df.shape)
print("Columns in the dataset:\n", df.columns.tolist())
print("Data types:\n", df.dtypes)
print("Missing values per column:\n", df.isnull().sum())
print("Number of duplicate rows:", df.duplicated().sum())

Shape of the dataset (R x C): (8897, 12)
Columns in the dataset:
['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added', 'release_year', 'rating', 'duration', 'listed_in', 'description']
Data types:
show_id    object
type      object
title     object
director  object
cast      object
country   object
date_added  int64
release_year  int64
rating    object
duration   int64
listed_in  object
description  object
dtype: object

Missing values per column:
show_id      0
type        0
title       0
director    2634
cast        825
country     831
date_added   0
release_year  0
rating       4
duration     3
listed_in    0
description   0
dtype: int64

Number of duplicate rows: 0
```

2. Data Discovery

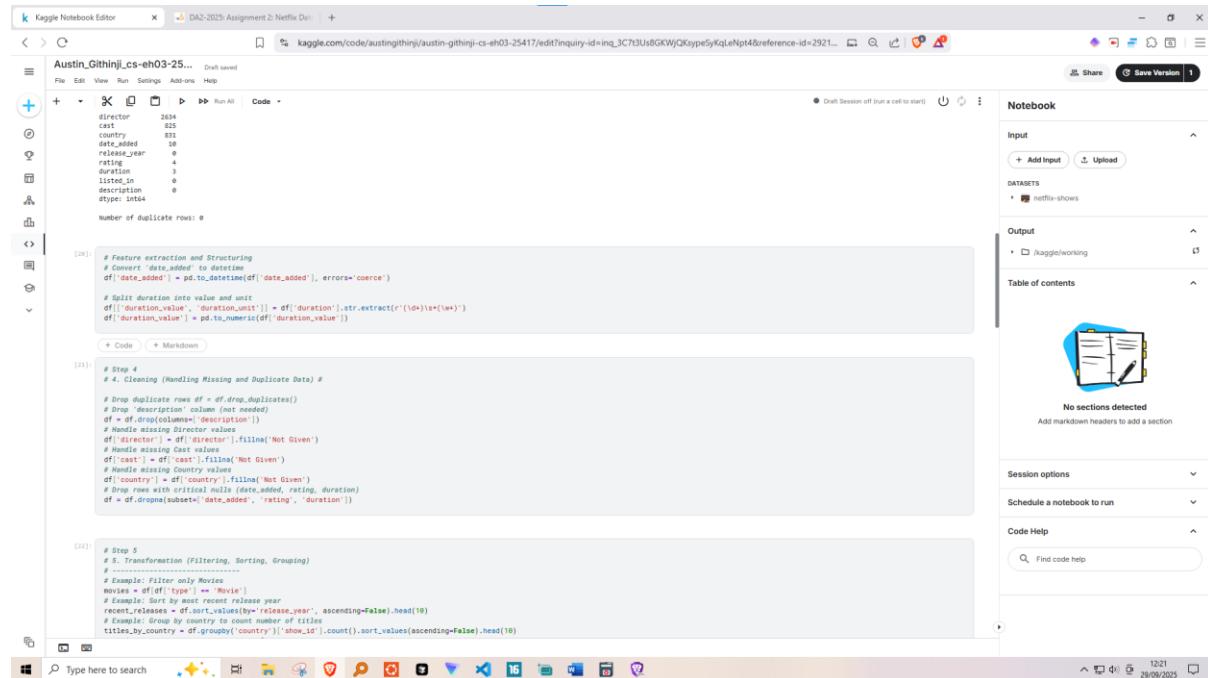
I checked data types, identified missing values, and looked for duplicates and inconsistencies.

Code:

```
df.info()
```

`df.isnull().sum()`

`df.duplicated().sum()`



The screenshot shows a Kaggle Notebook Editor window. The notebook title is "Austin_Githinji.cs-eh03-25...". The code cell contains several steps of data processing:

```

Austin_Githinji.cs-eh03-25...
File Edit View Run Settings Add-ons Help
Austin_Githinji.cs-eh03-25... Draft saved
+ - X Run All Code
Number of duplicate rows: 0

[1]: # Feature extraction and coercing
# Convert date_added to datetime
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')

# Split duration into value and unit
df[['duration_value', 'duration_unit']] = df['duration'].str.extract(r'(\d+)\s*(\w+)')
df['duration_value'] = pd.to_numeric(df['duration_value'])

# Step 4
# 4. Cleaning (Handling Missing and Duplicate Data)
# Drop duplicates rows if c of drop_duplicates()
# Drop description column not needed
df = df.drop(columns=['description'])

# Handle missing Director values
df['director'] = df['director'].fillna('Not Given')
# Handle missing cast values
df['cast'] = df['cast'].fillna('Not Given')
# Handle missing Country values
df['country'] = df['country'].fillna('Not Given')
# Drop rows with critical nulls (date_added, rating, duration)
df = df.dropna(subsets=['date_added', 'rating', 'duration'])

# Step 5
# 5. Transformation (Filtering, Sorting, Grouping)
# -----
# Example: Filter only Movies
movies = df[df['type'] == 'Movie']
# Example: Sort by most recent release year
recent_releases = df.sort_values(by='release_year', ascending=False).head(10)
# Example: Group by country to count number of titles
titles_by_country = df.groupby('country')['show_id'].count().sort_values(ascending=False).head(10)

```

The notebook sidebar includes sections for Input, Datasets (netflix-shows), Output, and Session options.

3. Structuring the Data

I converted `date_added` to datetime format and extracted numeric values and units from the `duration` column.

Code:

```

# Convert 'date_added' to datetime

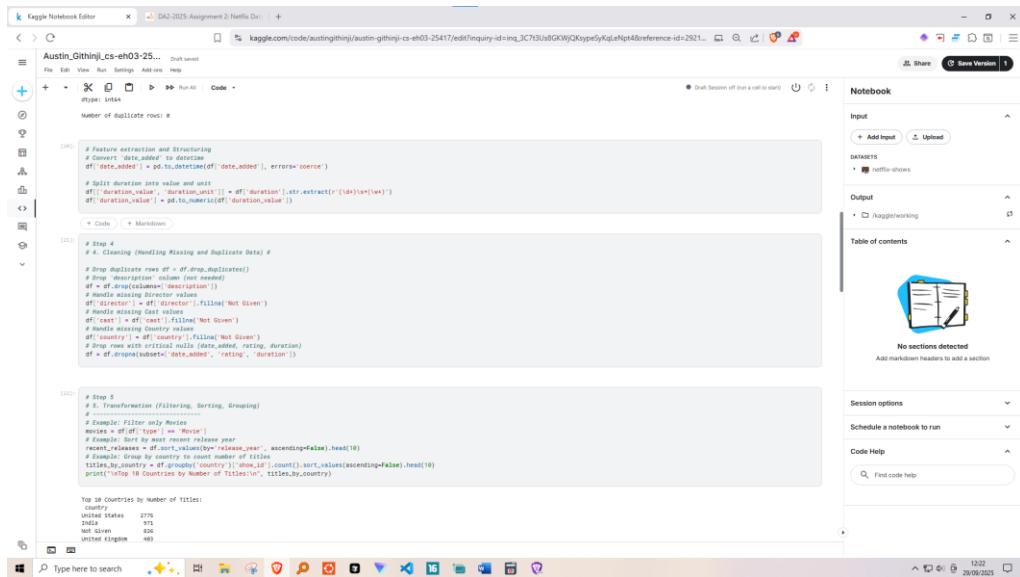
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')

# Split duration into value and unit

df[['duration_value', 'duration_unit']] = df['duration'].str.extract(r'(\d+)\s*(\w+)')

df['duration_value'] = pd.to_numeric(df['duration_value'])

```



The screenshot shows a Jupyter Notebook interface within a browser window. The notebook has two cells visible:

```

In [1]: # Feature extraction and Structuring
# Convert 'date_added' to datetime
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')

# Split duration into value and unit
df['duration'] = df['duration'].str.extract(r'(\d+)(\w*)')
df['duration_value'] = pd.to_numeric(df['duration'].str.extract(r'(\d+)'))
```

```

In [2]: # Step 4
# 4. Cleaning (Handling Missing and Duplicate Data)
# Drop duplicate rows of df.drop_duplicates()
# Drop columns that are not needed
df = df.drop(columns=['description'])

# Handle missing Director values
df['director'] = df['director'].fillna('Not Given')

# Handle missing Cast values
df['cast'] = df['cast'].fillna('Not Given')

# Handle missing Country values
df['country'] = df['country'].fillna('Not Given')

# Drop rows with critical nulls
df = df.dropna(subset=['date_added', 'rating', 'duration'])

```

The notebook sidebar shows the following sections:

- Notebook**: Input, Add Input, Upload, Datasets (netflix-shows), Output (kaggle/working), Table of contents.
- Session options**: Schedule a notebook to run.
- Code Help**: Find code help.

The status bar at the bottom right shows the time as 12:22 and the date as 28/09/2025.

4. Cleaning the Data

I removed duplicate rows, dropped unnecessary columns, and handled missing values.

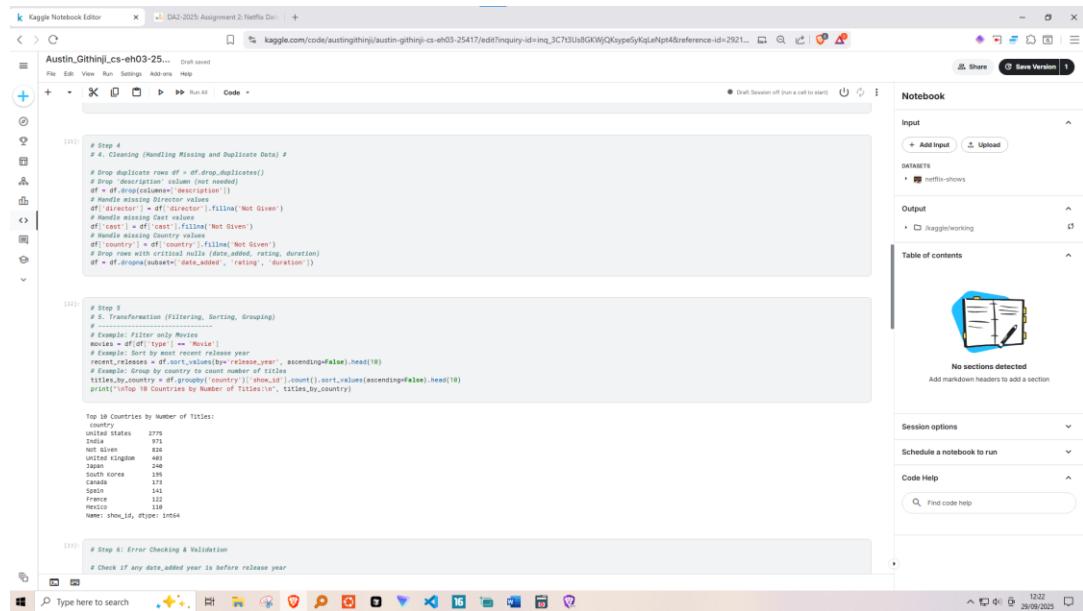
Code:

```
# Drop duplicates
df = df.drop_duplicates()

# Drop 'description' column
df = df.drop(columns=['description'])

# Fill missing values
df['director'] = df['director'].fillna('Not Given')
df['cast'] = df['cast'].fillna('Not Given')
df['country'] = df['country'].fillna('Not Given')

# Drop rows with critical nulls
df = df.dropna(subset=['date_added', 'rating', 'duration'])
```



```

# Step 4
# 4. Cleaning (Handling Missing and Duplicates Data)
# Drop duplicate rows if there are any duplicates
df = df.drop_duplicates()
# Drop missing Director values
df['director'].fillna('Not Given')
# Handle missing Cast values
df['cast'] = df['cast'].fillna('Not Given')
# Handle missing Country values
df['country'] = df['country'].fillna('Not Given')
# Drop rows with critical nulls (data_added, rating, duration)
df = df.dropna(subset=[ 'data_added', 'rating', 'duration'])
# Step 5
# 5. Transformation (Filtering, Sorting, Grouping)
# Example: Filter only Movies
movies = df[df['type'] == 'Movie']
# Example: Sort by most recent release year
recent_releases = df.sort_values(by='release_year', ascending=False).head(10)
# Example: Group by country to count number of titles
titles_by_country = df.groupby('country')['show_id'].count().sort_values(ascending=False).head(10)
print("Top 10 Countries by Number of Titles:")
print(titles_by_country)

# Step 6: Error Checking & Validation
# Check if any date_added year is before release year

```

5. Transformation and Enrichment

I applied filtering, sorting, and grouping to gain insights.

Code:

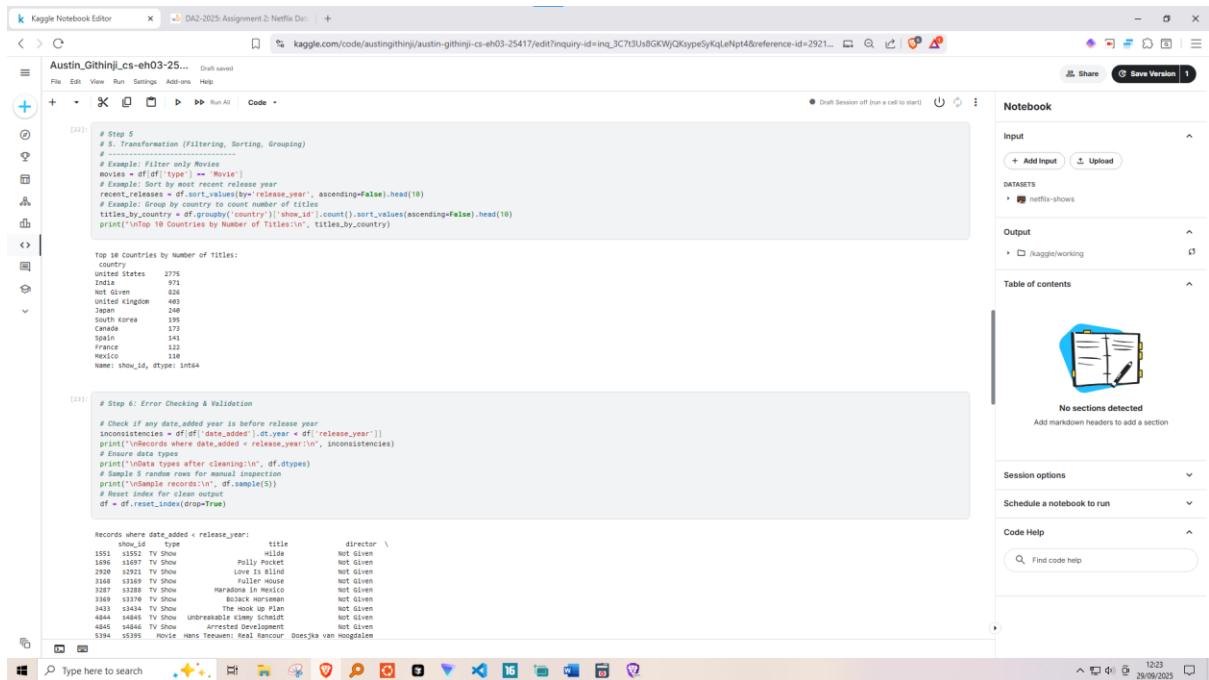
```

# Filter only Movies
movies = df[df['type'] == 'Movie']

# Sort by most recent release year
recent_releases = df.sort_values(by='release_year', ascending=False).head(10)

# Group by country and count number of titles
titles_by_country = df.groupby('country')['show_id'].count().sort_values(ascending=False).head(10)
print(titles_by_country)

```



The screenshot shows a Jupyter Notebook interface within a browser window. The notebook contains two code cells:

```

# Step 5: Transformation (Filtering, Sorting, Grouping)
# Example: Filter only Movies
movies = df[df['type'] == 'Movie']
# Example: Sort by most recent release year
recent_movies = movies.sort_values('release_year', ascending=False).head(10)
# Example: Group by country to count number of titles
titles_by_country = df.groupby('country')['show_id'].count().sort_values(ascending=False).head(10)
print("\nTop 10 Countries by Number of Titles:\n", titles_by_country)

top_10_countries = titles_by_country.head(10)
print(top_10_countries)

# Step 6: Error Checking & Validation
# Check if any date_added year is before release year
inconsistencies = df[df['date_added'] < df['release_year']]
print("Records where date_added < release_year:\n", inconsistencies)
# Ensure data types
print("Data types after cleaning:\n", df.dtypes)
# Sample 5 random records for manual inspection
print("Sampled records:\n", df.sample(5))
# Reset index for clean output
df = df.reset_index(drop=True)

```

The second code cell displays a table of records where the date added is earlier than the release year:

	show_id	type	title	director
1551	31552	TV Show	Polly Pocket	Not Given
1696	11697	TV Show	Lover Is Land	Not Given
2392	31369	TV Show	Fuller House	Not Given
3168	31369	TV Show	Miranda In Mexico	Not Given
3287	32326	TV Show	Kids in the Hall	Not Given
3303	32326	TV Show	Rock of Ages	Not Given
3433	33434	TV Show	The Hook Up Plan	Not Given
4844	54845	TV Show	Unbreakable Kimmy Schmidt	Not Given
4845	54845	TV Show	Arrested Development	Not Given
5394	53395	Movie	Hans Telemann: Real Encour	Doesja van Hoogdalem

6. Validation

I checked for inconsistencies, confirmed data types, and sampled random rows.

Code:

```
# Check logical consistency
```

```
inconsistencies = df[df['date_added'].dt.year < df['release_year']]
```

```
print(inconsistencies)
```

```
# Check data types
```

```
print(df.dtypes)
```

```
# Sample 5 random rows
```

```
print(df.sample(5))
```

```
# Reset index
```

```
df = df.reset_index(drop=True)
```

Kaggle Notebook Editor DA2-2025: Assignment 2: Netflix Data + kaggle.com/code/austingithinji/austin-githinji-cs-eh03-25417/edit?inquiry_id=inq_3C73Us8GKwQKkpe5yKqLeNpt4&reference_id=2921...

Austin_Githinji.cs-eh03-25417 Draft saved

File Edit View Run Settings Addons Help

+ Run All Code

Step 6: Error Checking & Validation

```
# Check if any date_added year is before release year
inconsistencies = df[(df['date_added'].dt.year < df['release_year'])]
print("Inconsistencies where date_added < release_year:\n", inconsistencies)

# Ensure data types
print("\nData types after cleaning:\n", df.dtypes)
# Sample 5 rows for visual inspection
print("\nSample records:\n", df.sample(5))

# Reset index for clean output
df = df.reset_index(drop=True)
```

Records where date_added < release_year:

id	title	director
1551	Polly	Hilde
1696	Polly Pocket	Not Given
2920	Love Is Blind	Not Given
3168	Fuller House	Not Given
3287	Mercion in Mexico	Not Given
3303	Reunited	Not Given
3433	The Work Up Plan	Not Given
4844	Unbreakable Kimmy Schmidt	Not Given
4845	Arrested Development	Not Given
5395	Movie Hans Treenut: Real Ranchur Doesjka van Hoogdalem	Not Given
5654	Senses	Not Given
5657	Tokyo Ghoul	Not Given
7863	Incoming	Eric Jargosa
7111	Jack Taylor	Stuart Orme

cast:

id	name	character
1551	Bella Ramsey	Meera
1696	Emily Tennant	Shannon Choi-Kent
2920	Nick Lachey	Vanessa Lachey
3168	Candace Cameron Bure	Bebe Sweetly
3287	Reunited	Not Given
3303	Will Arnett	Aaron Paul
3433	Amy Sedaris	Alison B...
3435	Paula Malcomson	Izta Hervort
3436	Elle Varner	Jade
4844	Jason Bateman	Portia de Rossi
4845	Arrested Development	Not Given
5395	Tuppence Middleton	Brian J.
5654	Marcel Wensselaar	Paul Freeman
5657	Irwan Khan	Irwan Khan
7863	Scott Adkins	Aaron McCusker
7111	Stuart Orme	Not Given

country data_added release_year rating:

id	country	data_added	release_year	rating
1551	United Kingdom, Canada, United States	2020-12-14	2021	TV-Y7
1696	Canada, United States, Ireland	2020-11-15	2021	TV-Y
2920	United States	2020-12-01	2020	TV-PG
3168	United States	2019-12-06	2020	TV-PG
3287	Argentina, United States, Mexico	2019-11-13	2020	TV-MA

7. Exporting the Dataset

Finally, I exported the cleaned dataset as a CSV.

Code:

```
df.to_csv('/kaggle/working/cleaned.netflix.csv', index=False)
```

```
print("Cleaned dataset exported successfully!")
```

Kaggle Notebook Editor DA2-2025: Assignment 2: Netflix Data + kaggle.com/code/austingithinji/austin-githinji-cs-eh03-25417/edit?inquiry_id=inq_3C73UsBGKwQKkypeSyKqLeNpt4&reference_id=2921...

Austin_Githinji.cs-eh03-25417 Draft saved

File Edit View Run Settings Add-ons Help

Run All Code

Draft Session off (run a cell to start)

Notebook

Input

+ Add Input Upload

DATASETS

netflix-shows

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Session options

Schedule a notebook to run

Code Help

Find code help

```

Sample Records:
show_id type title \
3892 33093 TV Show Street Food
3562 52609 Movie Jerry Before Seinfeld
4540 52610 Movie Surge Yang Tai Chin
3621 33622 Movie Scaredall
6140 56401 Movie American Experience: The Island Murder

director cast \
3892 Not Given Not Given
3248 Michael J. Fox, Griffin Jerry Seinfeld
4168 Kunz Age, Fedi Muril, Laudy Cynthia Bella, Kaline Shah, ...
3621 Billy Corben Not Given
6140 Mark Samitier Blair Brown

listed_in release_year rating duration \
3892 United States 2019-04-26 2018 TV-MA 1 Season
3248 United States 2017-09-19 2017 TV-14 62 min
4168 United States 2019-04-26 2018 TV-14 105 min
3621 United States 2019-04-06 2018 TV-MA 105 min
6140 United States 2019-05-01 2018 TV-14 53 min

duration_unit \
3892 Season
3248 Month
4168 Min
3621 Min
6140 Min

[24]: # Step 7 Publish (Save Cleaned Dataset)
# df.to_csv('kaggle/working/cleaned.netflix.csv', index=False)
print('UnCleaned dataset exported successfully!')
```

Cleaned dataset exported successfully!

Conclusion

This project covered the full data wrangling workflow: loading, exploring, cleaning, structuring, transforming, validating, and exporting. I learned how to handle missing values, remove duplicates, split and reformat columns, and apply filtering, sorting, and grouping.

Kaggle Notebook Link

<https://www.kaggle.com/code/austingithinji/austin-githinji-cs-eh03-25417>

Conclusion

This week I gained a good grounding on the introductory concepts relating to data science and artificial intelligence. I am getting a better understanding that I can build on as we work on more advanced concepts in later weeks. I have posted my writeup on my blog and I look forward to building a portfolio that I can showcase on my CV as I look for jobs in Data and AI.