🎥 Netflix Dashboard Tutorial: Interactive Storytelling with Panel and hvPlot

◇ Overview of Visualization Types

My dashboard features the following visualizations:

1. **Bar Plot: Rating Distribution**
    a. Type: Horizontal Bar Plot
    b. Description: Shows how frequently each rating type (e.g., TV-MA, PG) occurs across the selected content type.
    c. Purpose: Helps identify which ratings dominate for Movies vs. TV Shows.
2. **Line Plot: Titles Released Over Time**
    a. Type: Line Plot
    b. Description: Tracks the number of titles released each year.
    c. Purpose: Helps understand trends in Netflix content expansion.
3. **Bar Plot: Top 10 Genres**
    a. Type: Horizontal Bar Plot
    b. Description: Displays the 10 most common genres.
    c. Purpose: Provides insight into Netflix's content diversity and audience preferences.
4. **Scatter Plot: Duration vs. Date Added**
    a. Type: Scatter Plot with color encoding
    b. Description: Shows the duration of content over time and maps release year with a color gradient.
    c. Purpose: Visualizes temporal shifts in content length and release trends.

🧮 Complementary Strengths

- **Bar charts** provide categorical insights (ratings, genres).
- **Line charts** show trends over time (e.g., content release volume).
- **Scatter plots** allow multivariate exploration (duration, date added, release year).

These charts complement one another by providing both categorical and temporal exploration paths. The scatter plot in particular brings depth via color encoding and interactivity.

Dashboard Considerations

- **Interactivity**: All charts respond to the content type (Movie or TV Show) selected via a widget.
- **Linked Filters**: Users can explore content characteristics by toggling between types.
- **Responsiveness**: Panel allows layout organization for small and large screens.

◇ Panel & hvPlot

- **Panel**:
  - Created by: HoloViz
  - License: Open Source (BSD)
  - Description: A dashboarding library designed for Jupyter, Python scripts, or standalone apps.
  - Installation:

```
pip install panel
```

- **hvPlot**:
  - Built on top of HoloViews
  - Offers a high-level API for plotting directly from pandas DataFrames.
  - Installation:

```
pip install hvplot
```

Framework Approach

- **Declarative**: Users specify *what* to display, not how.
- **Jupyter-Friendly**: Fully integrates with Jupyter Notebooks.

- **Interactive Widgets**: Panel supports seamless widget use (`@pn.depends`).
- **Limitations**:
  - Limited low-level styling (vs. Matplotlib)
  - Browser-based: Not ideal for static exports

Why This Framework?

- No Matplotlib or Altair allowed: Panel + hvPlot is a powerful alternative.
- Easy to use for beginners.
- Perfect for combining interactive visualizations with narrative writing in Jupyter.
- Compatible with Hugging Face Spaces and other deployment platforms.
- That's what I learn from this class.

◇ Dataset Used

Dataset: `netflix_titles.csv`

- Source: Kaggle Netflix Dataset
- Size: ~6,000 rows
- Key columns: `type`, `title`, `rating`, `duration`, `date_added`, `release_year`, `listed_in`

☑ Cleaning Steps

```
import pandas as pd

# Load dataset
df = pd.read_csv("netflix_titles.csv")

# Clean duration column
df['duration_clean'] = pd.to_numeric(df['duration'].str.replace(' min', ''),
errors='coerce')

# Parse date_added
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')
df['year_added'] = df['date_added'].dt.year

# Extract genres
df['genres'] = df['listed_in'].str.split(', ')
```

Building Dashboard Components

```
import panel as pn
import hvplot.pandas
pn.extension()

# Content type selector
type_selector = pn.widgets.RadioButtonGroup(
    name='Type', options=['Movie', 'TV Show'], button_type='success')

@pn.depends(type_selector)
def filter_data(content_type):
    return df[df['type'] == content_type]
```

1. Rating Distribution

```
@pn.depends(type_selector)
def rating_plot(content_type):
    data = filter_data(content_type)
    return data['rating'].value_counts().sort_values().hvplot.barh(
        title='🎟️ Rating Distribution', xlabel='Count', ylabel='Rating',
        height=300, width=400, color='tomato')
```

2. Titles Released Over Time

```
@pn.depends(type_selector)
def yearly_plot(content_type):
    data = filter_data(content_type)
    return data['release_year'].value_counts().sort_index().hvplot.line(
        title='📅 Titles Released Over Time', xlabel='Year', ylabel='Number of
Titles',
        line_width=3, color='dodgerblue', height=300, width=500)
```

3. Top Genres

```
@pn.depends(type_selector)
def genre_plot(content_type):
    data = filter_data(content_type)
    return
data['genres'].explode().value_counts().head(10).sort_values().hvplot.barh(
        title='🎬 Top Genres', xlabel='Count', ylabel='Genre',
        color='mediumseagreen', height=300, width=400)
```

4. Duration vs. Date Added

```
@pn.depends(type_selector)
def scatter_plot(content_type):
    data = filter_data(content_type)
    return data.hvplot.scatter(
        x='date_added', y='duration_clean', c='release_year', cmap='plasma',
        title='⏱ Duration vs. Date Added', xlabel='Date Added',
ylabel='Duration',
        height=300, width=500, size=5, alpha=0.6)
```

Layout

```
dashboard = pn.Column(
    "# 🎥 Netflix Content Explorer",
    "Use the toggle to explore visualizations by content type.",
    type_selector,
    pn.Row(rating_plot, genre_plot),
    pn.Row(yearly_plot, scatter_plot),
)
dashboard.servable()
```

Conclusion

This demonstrates how Panel and hvPlot can be used to create a rich, interactive data story. The coordinated visualizations enable users to filter and interpret patterns in Netflix content over time. With only a few lines of code, we built a fully interactive dashboard deployable on web platforms.

Deployment (Hugging Face)

```
panel serve app.py --autoreload --shoW
```

Requirements.txt

```
panel
hvplot
pandas
```