



▶ Play

Instructor: Dr. Tran Thanh Tung
Data Science & Data Visualization

NETFLIX

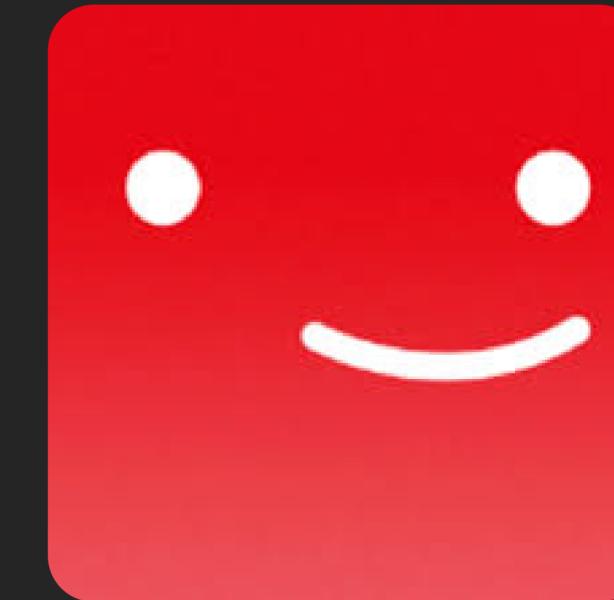
Who's Watching?



Danh Đức



Long Lê



Nguyễn Hiệu



Ngọc Sang



Thanh Quân

Start

NETFLIX CONTENT



01

INTRODUCTION

02

TASK TIMELINE

03

METHODOLOGY

04

RESULT

05

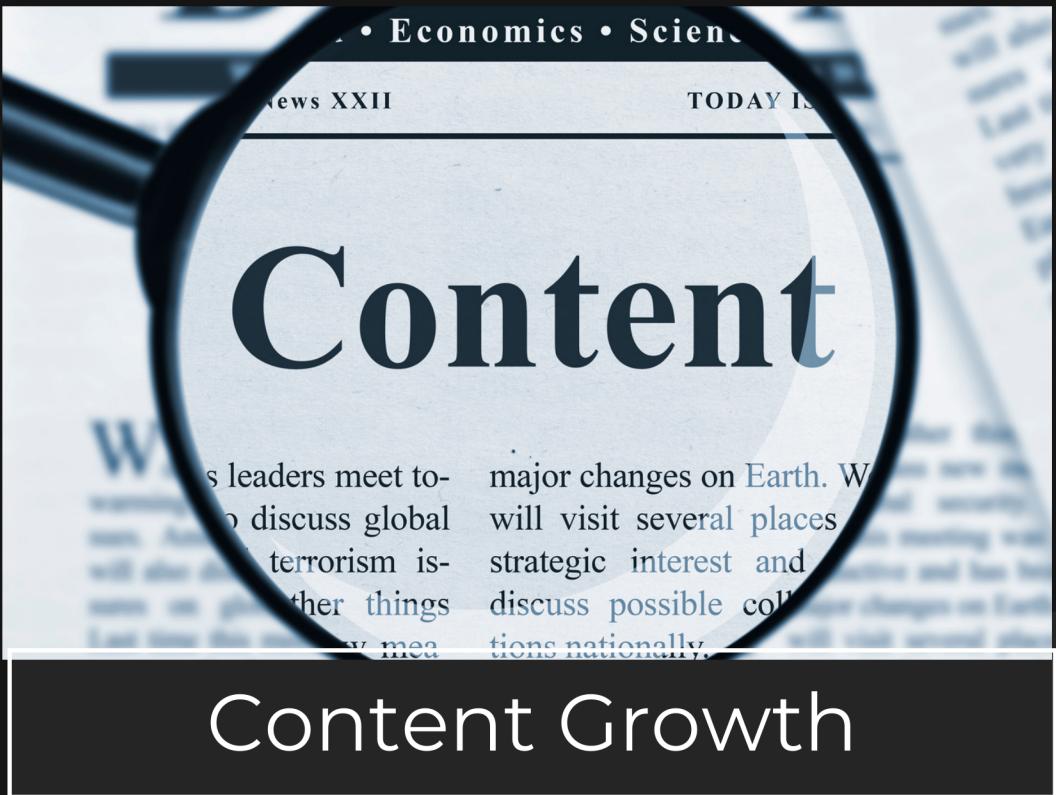
CONCLUSION

06

REFERENCES

I. INTRODUCTION

Scene Overview



This scene visualizes the increase in Netflix's content over time, showing a timeline on the x-axis and the count of Movies and TV Shows on the y-axis.



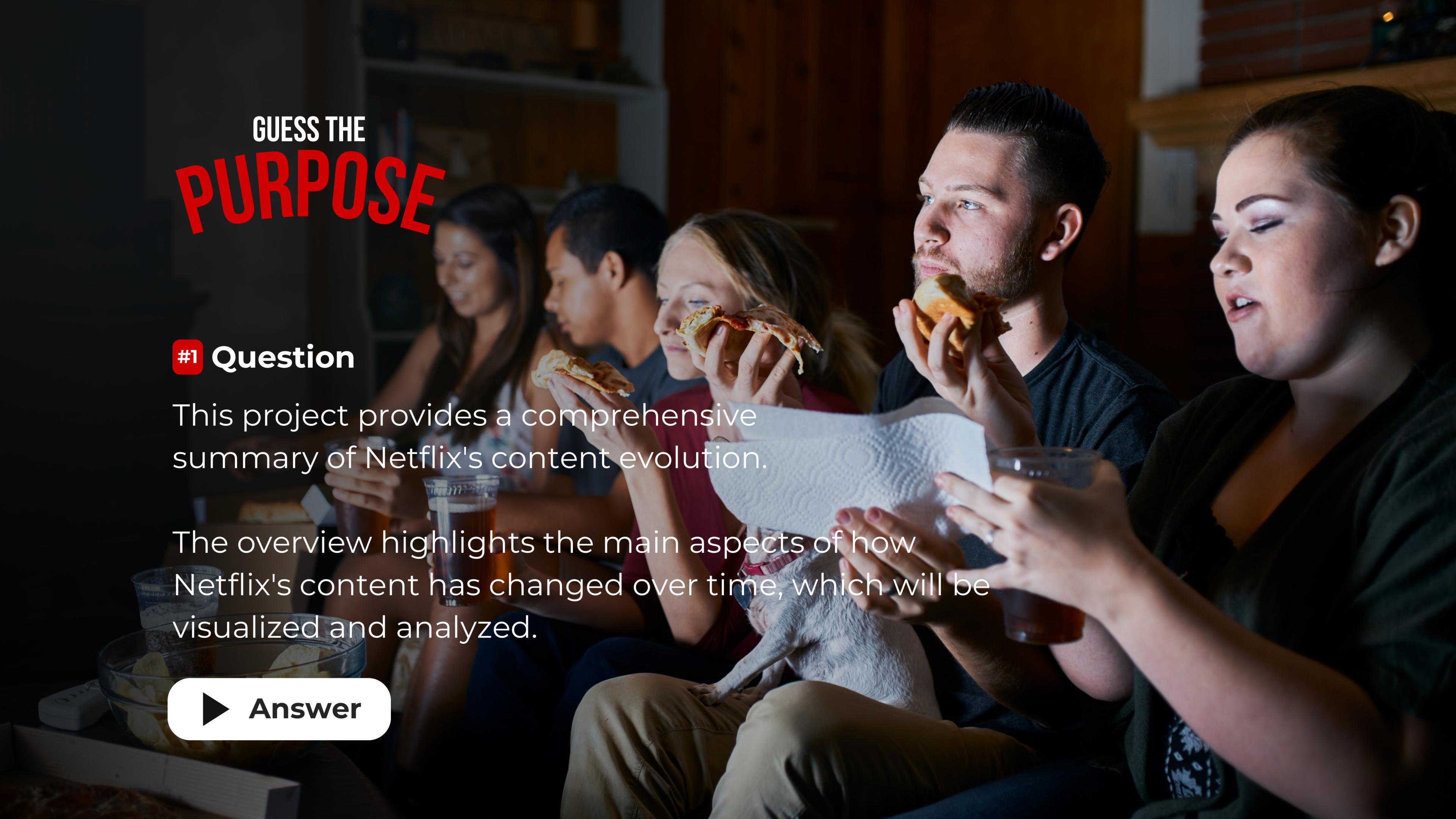
A Closer Look

This scene highlights the countries that produce the most content for Netflix, emphasizing the United States and India.



Shifting the Focus

This scene examines the evolution of Netflix's content distribution based on age ratings.

A photograph of a group of five people (three men and two women) sitting on a couch in a dimly lit room, watching television. They are all holding pizzas and glasses of beer. A small dog is sitting on the lap of the person on the right. The scene suggests a casual, social gathering.

GUESS THE **PURPOSE**

#1 Question

This project provides a comprehensive summary of Netflix's content evolution.

The overview highlights the main aspects of how Netflix's content has changed over time, which will be visualized and analyzed.



Answer

GUESS THE **GOAL**

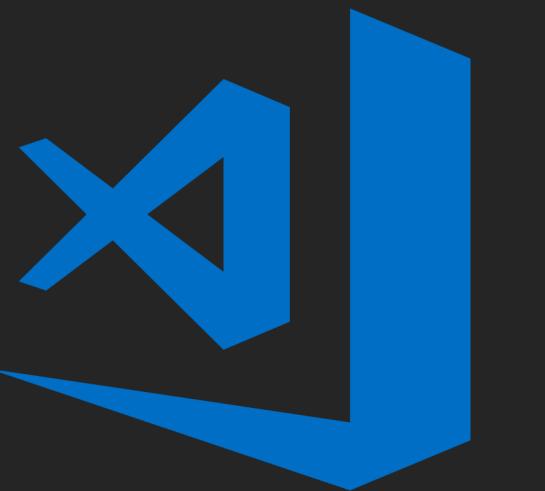
A photograph of a group of five people sitting on a couch in a dimly lit room, watching television. They are all holding slices of pizza and looking towards the screen. A white paper towel is held up in front of the text box.

Process data to identify necessary attributes for visualization.

Manage and adjust charts using D3.js libraries for better alignment with potential future alterations.

GUESS THE

TECHNIQUE & TOOLS USED



2. TIMELINE



Phase	Week
Planning	1
Data Pulling	2
Preprocessing Data	3
Conceptual Design	5-8
Implementing Design	8-13
Presentation	14

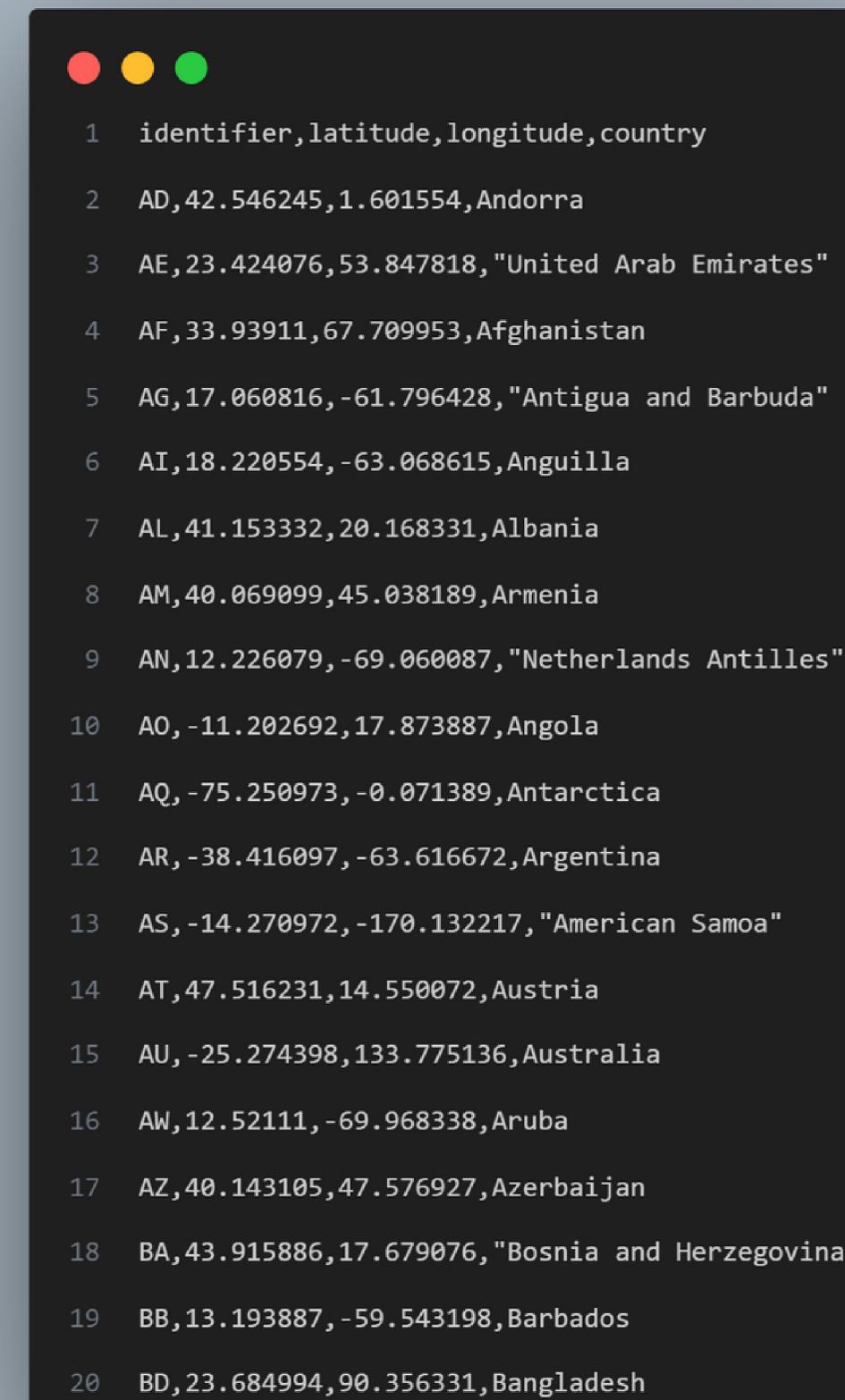
II. METHODOLOGY

1. Project Structure

└ .vscode	└ image	└ .gitattributes
└ launch.json	└ intro.mp4	└ .gitignore
└ settings.json	└ netflix_logo.png	└ ~\$tflix Report.docx
└ resources	└ js	└ Netflix Report.docx
└ css	└ custom.js	└ README.md
# customStyleSheet...	└ map.js	└ scene1.html
# stylemap.css	└ navigation.js	└ scene2.html
# styles.css	└ piechart.js	└ scene3.html
└ data	└ scene1.js	└ text.html
└ geo.csv	└ scene2.js	
└ netflix_clean.csv	└ utils.js	
└ netflix_titles.csv		

II. METHODOLOGY

2. Cleaning Data



1 identifier,latitude,longitude,country
2 AD,42.546245,1.601554,Andorra
3 AE,23.424076,53.847818,"United Arab Emirates"
4 AF,33.93911,67.709953,Afghanistan
5 AG,17.060816,-61.796428,"Antigua and Barbuda"
6 AI,18.220554,-63.068615,Anguilla
7 AL,41.153332,20.168331,Albania
8 AM,40.069099,45.038189,Armenia
9 AN,12.226079,-69.060087,"Netherlands Antilles"
10 AO,-11.202692,17.873887,Angola
11 AQ,-75.250973,-0.071389, Antarctica
12 AR,-38.416097,-63.616672,Argentina
13 AS,-14.270972,-170.132217,"American Samoa"
14 AT,47.516231,14.550072,Austria
15 AU,-25.274398,133.775136,Australia
16 AW,12.52111,-69.968338,Aruba
17 AZ,40.143105,47.576927,Azerbaijan
18 BA,43.915886,17.679076,"Bosnia and Herzegovina"
19 BB,13.193887,-59.543198,Barbados
20 BD,23.684994,90.356331,Bangladesh

2. Cleaning Data

```
● ● ●

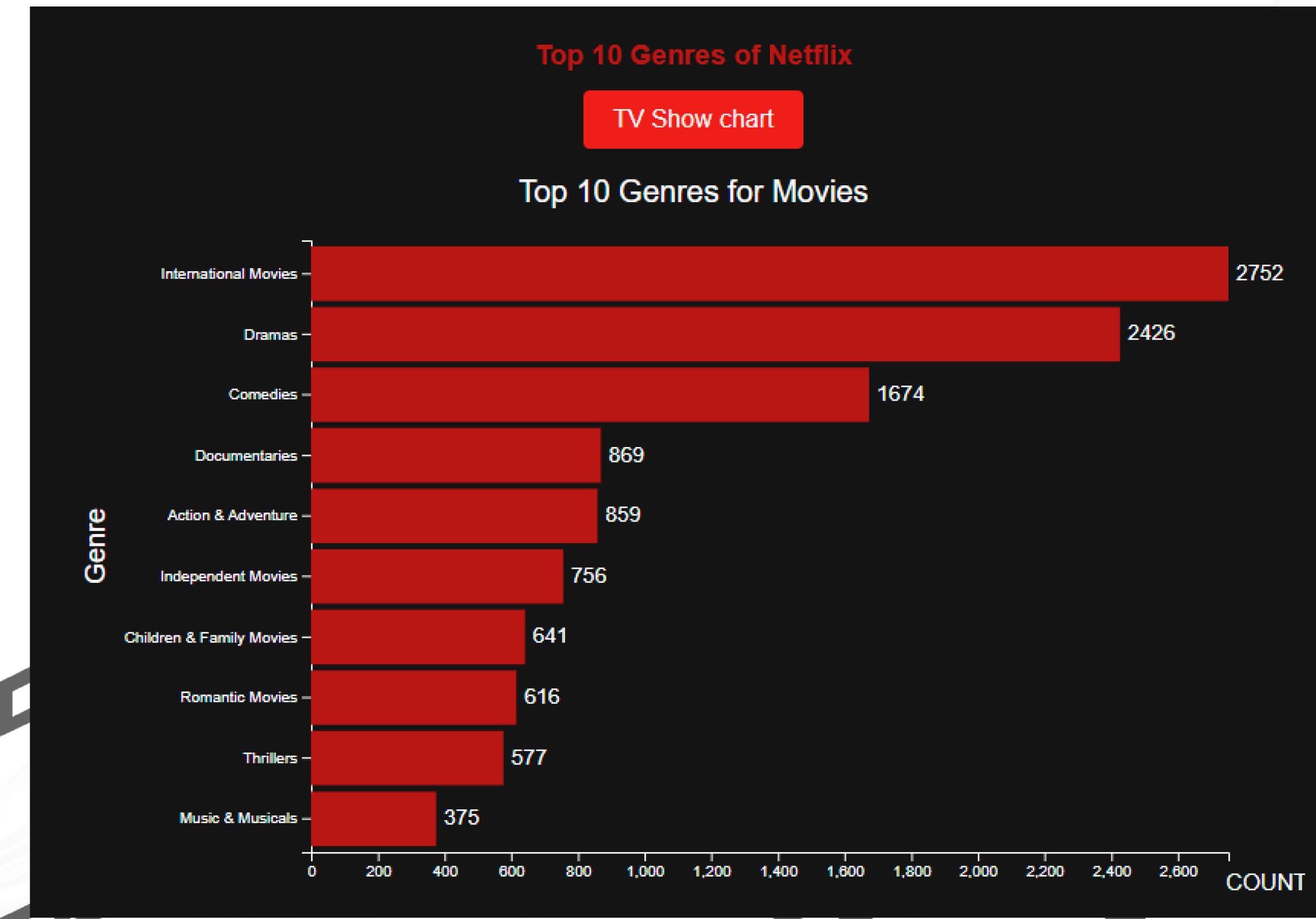
1  function processData(data, type) {
2      const filteredData = data.filter(d => d.type === type);
3
4      const categoryCount = {};
5      filteredData.forEach(d => {
6          d.listed_in.split(', ').forEach(category => {
7              if (categoryCount[category]) {
8                  categoryCount[category]++;
9              } else {
10                  categoryCount[category] = 1;
11              }
12          });
13      });
14
15      const sortedCategories = Object.keys(categoryCount).sort((a, b) => categoryCount[b] - categoryCount[a]);
16      const topCategories = sortedCategories.slice(0, 10);
17
18      return topCategories.map(category => ({ category, value: categoryCount[category] }));
19  }
20
21  d3.csv('resources/data/netflix_titles.csv').then(data => {
22      drawScene2(data);
23  });


```

2. Cleaning Data

The data we got will look like this:

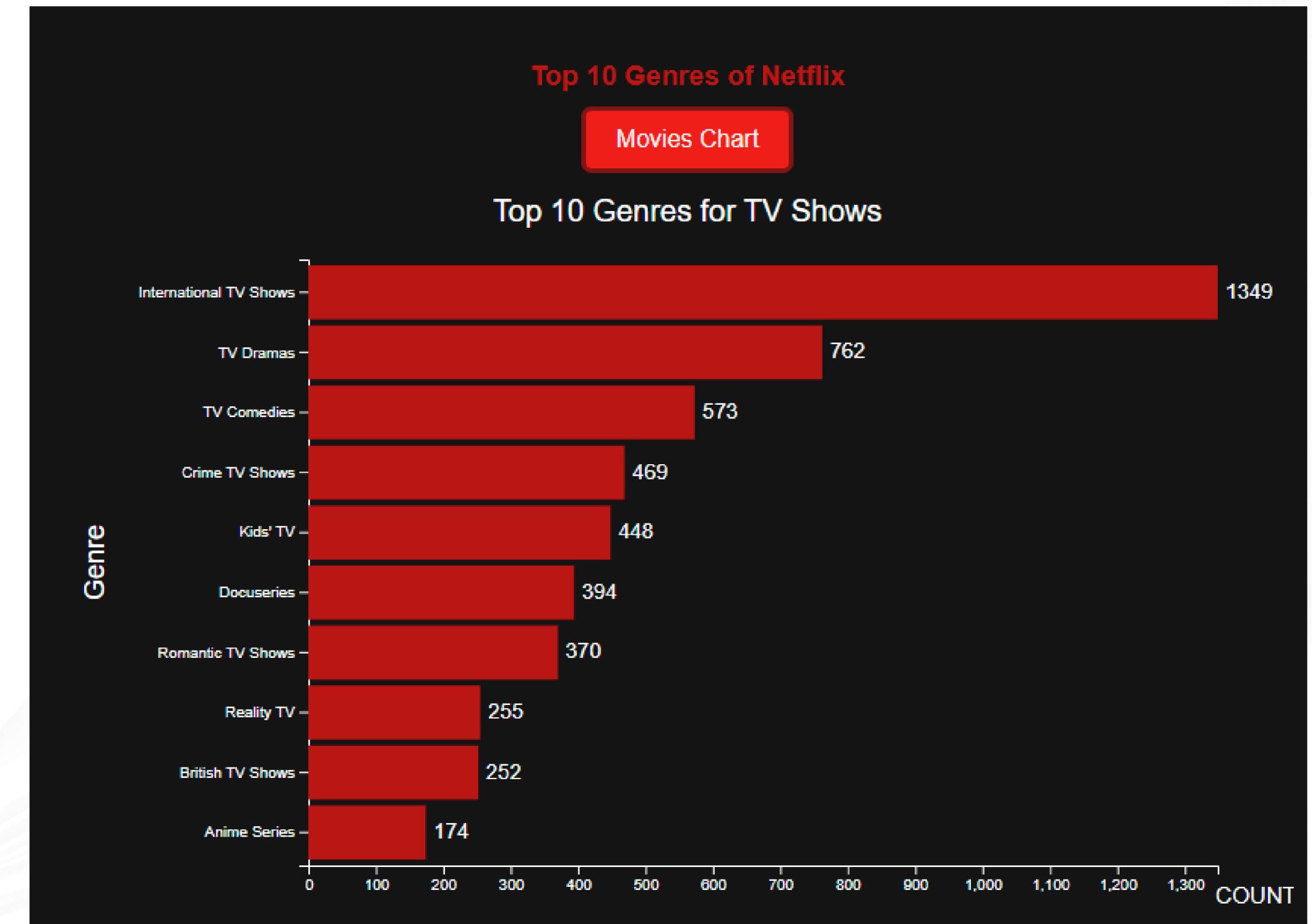
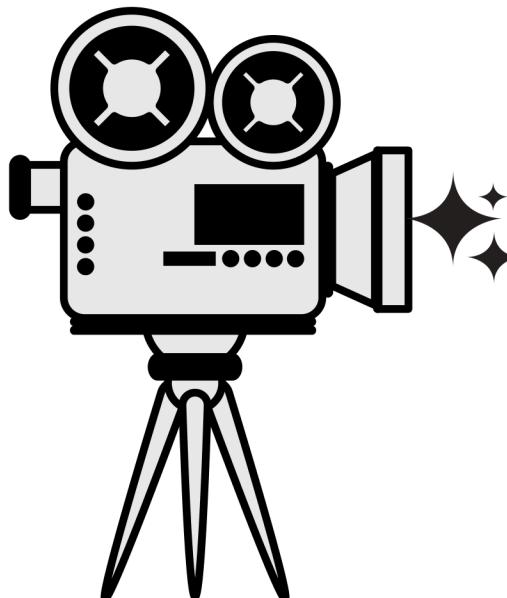
Movies Chart



2. Cleaning Data

The data we got will look like this:

TV Show Chart



3. Chart usage

Chart name	Purpose	Description
Pie chart	To compare the number of TV Shows and Movies viewers	Compare using requirements between TV Shows and Movies of users from Netflix
Area Chart	To show trends over time, compare magnitudes, and visualize cumulative totals	The number of TV Shows and Movies added to a platform over a timeline from 2008 to 2021.

3. Chart usage

Chart name	Purpose	Description
Horizontal Bar Charts	View ranking of the minimum degree requirement	The horizontal bar chart illustrates how popular each film type in Netflix
Bubble Charts	To visualize and compare relationships between the number of users in each countries.	Displays a world map with various red circles of different size of TV Shows and Movies in each countries

4. IMPLEMENTATION WITH D3.JS



PIE CHART

```
 1 // Chọn phần tử SVG và thiết lập thuộc tính chiều rộng và chiều cao
 2 const svgPie = d3.select("#piechart").html("")
 3   .append("svg")
 4   .attr("width", widthPie+500)
 5   .attr("height", heightPie + 300)
 6   .append("g")
 7   .attr("transform", "translate(" + 250 + "," + (350) + ")");
 8
 9   svgPie.append("text")
10   .attr("x", 0)
11   .attr("y", -300) // Điều chỉnh vị trí của tiêu đề
12   .attr("text-anchor", "middle")
13   .style("font-size", "25px")
14   .style("font-weight", "bold")
15   .attr("fill", "white")
16   .text("Total Movies & TV Shows on Netflix");
17
18
19 // Dữ liệu Movies và TV shows
20 d3.csv("resources/data/netflix_titles.csv")
21 .then(function (data) {
22   const totalMovies = data.filter(d => d.type === "Movie").length;
23   const totalTVShows = data.filter(d => d.type === "TV Show").length;
24   const total = totalMovies + totalTVShows;
25
26   const moviesPercentage = (totalMovies / total) * 100;
27   const tvShowsPercentage = (totalTVShows / total) * 100;
28
29   const pieData = [
30     { category: "Movies", percent: moviesPercentage },
31     { category: "TV Shows", percent: tvShowsPercentage }
32   ];
33
34   draw_pie(pieData);
35 })
36 .catch(function (error) {
37   console.log(error);
38 });
39
```

AREA CHART

```
● ○ ■
1 function drawScene1(data) {
2     const svg = d3.select("#chart").html("");
3     const width = 1500, height = 700, margin = {top: 50, right: 0, bottom: 50, left: 125};
4
5     const parseTime = d3.timeParse("%Y-%m-%d");
6     const x = d3.scaleTime().range([0, width - margin.left - margin.right]);
7     const y = d3.scaleLinear().range([height - margin.top - margin.bottom, 0]);
8
9     const area = d3.area()
10        .x(d => x(d.date))
11        .y0(y(0))
12        .y1(y(0));
13
14    const g = svg.append("g")
15        .attr("transform", `translate(${margin.left}, ${margin.top})`);
16
17    data.forEach(d => {
18        d.date = parseTime(d.date_added.split(" ")[0]);
19        d.value = +d.show_id;
20    });
21
22    const sumData = d3.rollups(data, v => v.length, d => d.type, d => d.date.getFullYear());
23
24    const newData = [];
25    sumData.forEach(d => {
26        d[1].forEach(e => {
27            newData.push({date: new Date(e[0], 0, 1), type: d[0], value: e[1]});
28        });
29    });
30
31    newData.sort((a, b) => a.date - b.date);
32
33    x.domain(d3.extent(newData, d => d.date));
34    y.domain([0, d3.max(newData, d => d.value)]);
35
36    g.selectAll(".area")
37        .data(d3.groups(newData, d => d.type))
38        .join("path")
39        .attr("class", "area")
40        .attr("fill", d => d[0] === "Movie" ? "#b20710" : "#f75151") // màu hình
41        .attr("stroke", "black")
42        .attr("stroke-width", 3)
43        .attr("opacity", 0.8)
44        .attr("d", d => area(d[1]))
45        .transition()
46        .duration(2000)
47        .attr("d", d => {
48            area.y1(d => y(d.value));
49            return area(d[1]);
50        });
51
```

HORIZONTAL BAR CHART

```
1  function drawChart(container, data, title) {
2      const svg = d3.select(container).append("svg")
3          .attr("width", 800)
4          .attr("height", 500);
5
6      const width = 800, height = 500;
7      const margin = { top: 50, right: 50, bottom: 50, left: 150 }; // Increase left margin to 150
8
9      const categories = data.map(d => d.category);
10     const maxValue = d3.max(data, d => d.value);
11
12     const yScale = d3.scaleBand()
13         .domain(categories)
14         .range([margin.top, height - margin.bottom])
15         .padding(0.1);
16
17     const xScale = d3.scaleLinear()
18         .domain([0, maxValue])
19         .range([margin.left, width - margin.right]);
20
21     const xAxis = d3.axisBottom(xScale);
22     const yAxis = d3.axisLeft(yScale);
23
24     svg.append("g")
25         .attr("class", "x-axis")
26         .attr("transform", `translate(0, ${height - margin.bottom})`)
27         .call(xAxis);
28
29     svg.append("g")
30         .attr("class", "y-axis")
31         .attr("transform", `translate(${margin.left}, 0)`)
32         .call(yAxis);
33
34     const tooltip = d3.select("body").append("div")
35         .attr("class", "tooltip")
36         .style("position", "absolute")
37         .style("background-color", "white")
38         .style("padding", "5px")
39         .style("border", "1px solid black")
40         .style("border-radius", "5px")
41         .style("visibility", "hidden")
42         .text("");
```

NETFLIX

A SNAPSHOT OF MOVIES AND TV SHOWS IN NETFLIX

PEOPLE have seldom noticed how many movies and TV shows are collected by Netflix. It seems like that most of the time, people can always find what they want to watch on Netflix. Based on a recent article, Netflix is the most-watched TV streaming service in Australia. Visualising all the Movies and TV data in Netflix might be an approach to understand its success.

[CLICK HERE TO KNOW MORE](#)

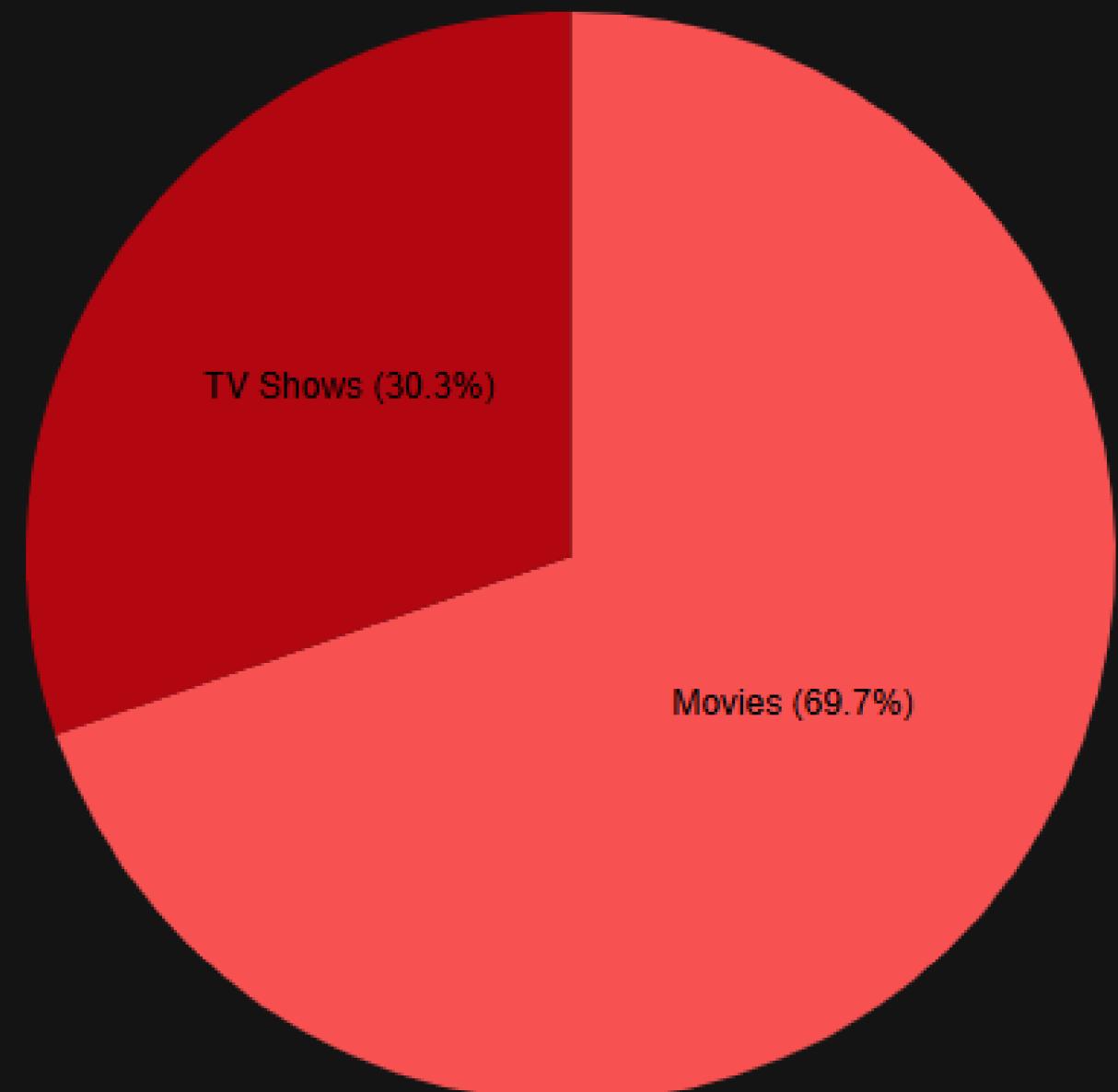
IV. RESULT

Evolution of Netflix's Content

This scene visualizes the growth in the quantity of content that Netflix has produced over time, and the type of the content. The chart below presents a timeline on the x-axis and the count of Movies and TV Shows on the y-axis.

Explore the progression of Netflix's content production over the years and observe how their focus has shifted between **Movies and TV shows**.

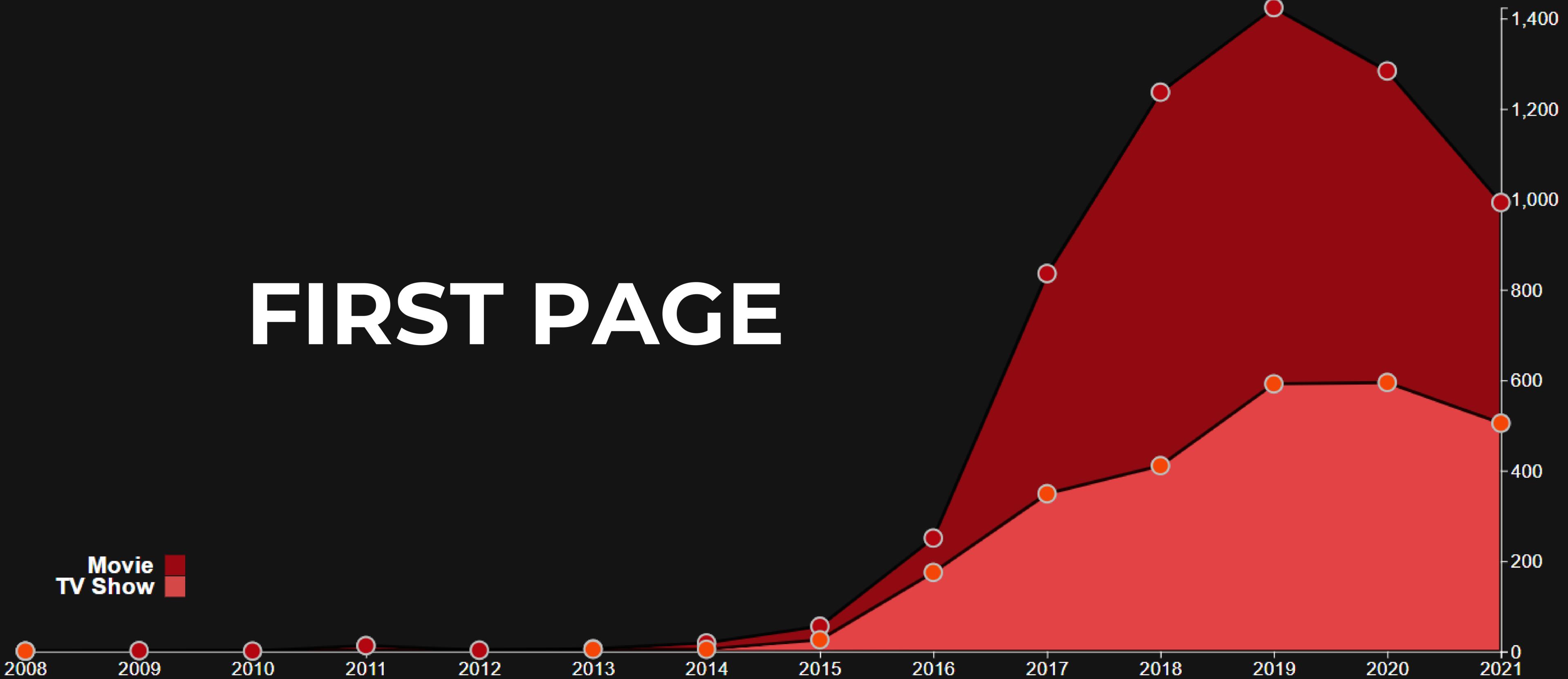
Total Movies & TV Shows on Netflix



FIRST PAGE

Content Growth: Movies and TV Shows added over the years

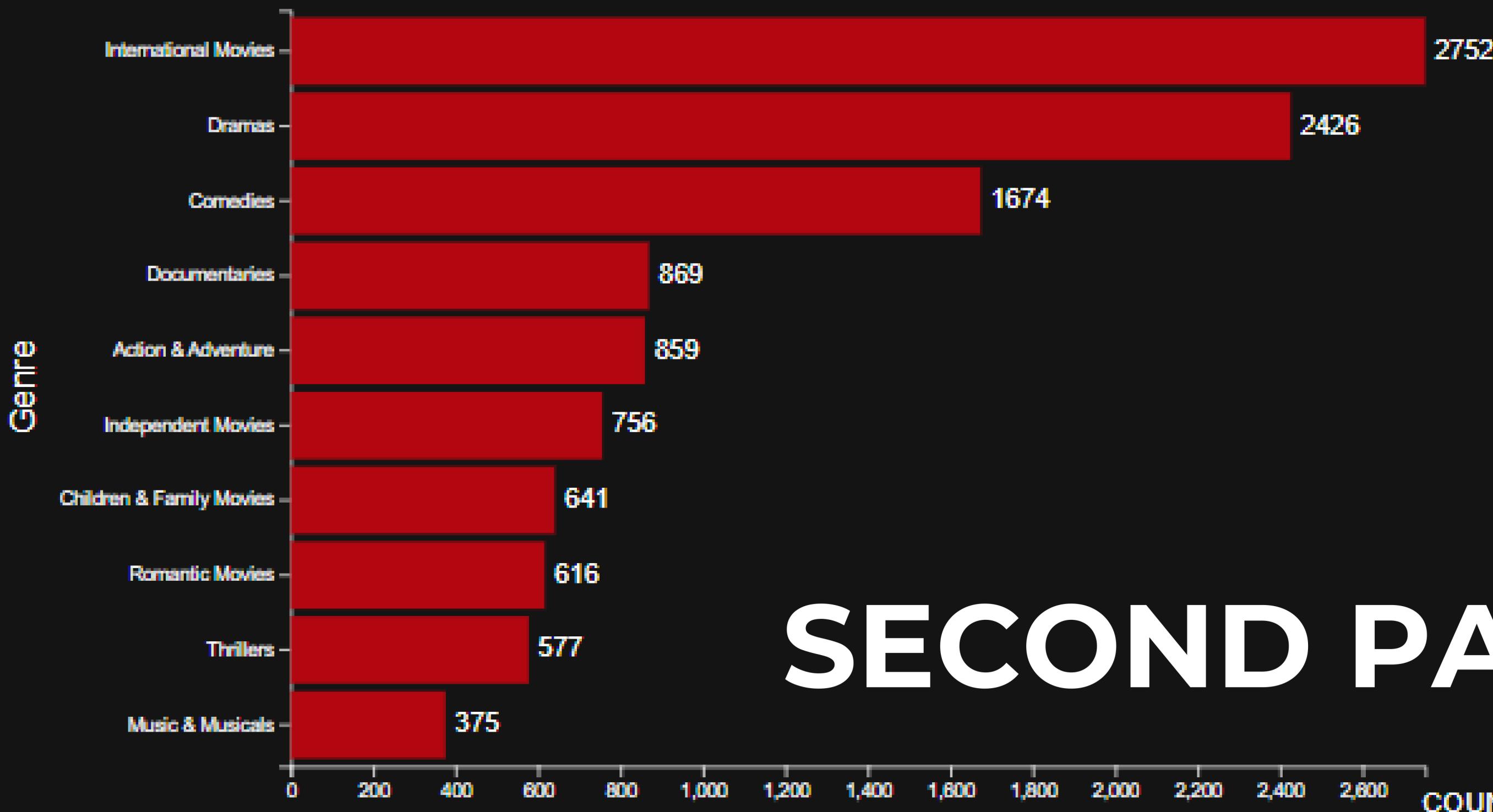
FIRST PAGE



Top 10 Genres of Netflix

TV Show chart

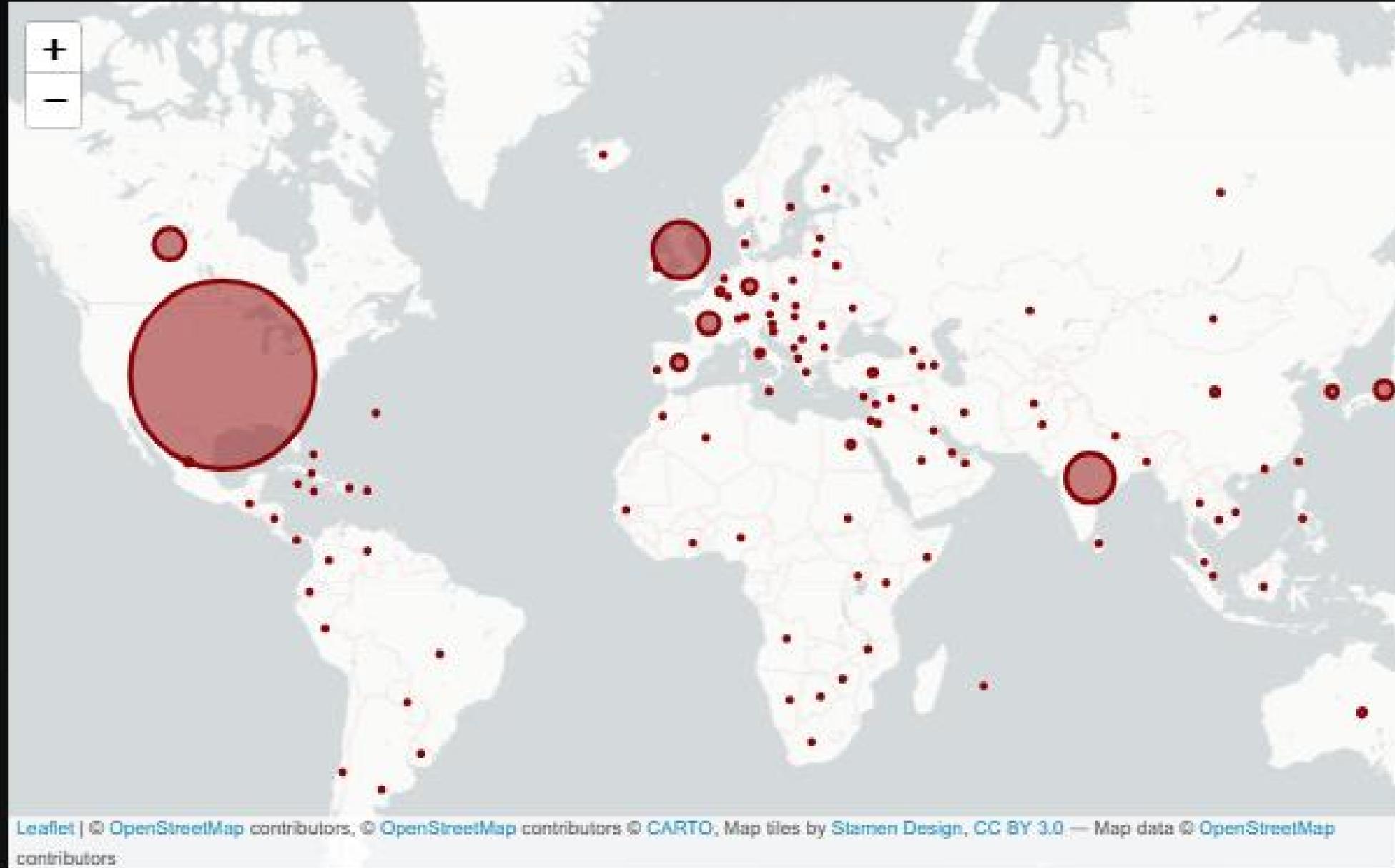
Top 10 Genres for Movies



SECOND PAGE

Interactive Map

Explore the number of movies and shows in the netflix library per production country. When using the year filter, be aware that there are significantly fewer entries in the database pre-2015 compared to later years. Alternatively, the genre filter allows you to look at one or more genres in particular. To revert your selection and display all genres at once, simply select all (Ctrl + A / ⌘ + A).

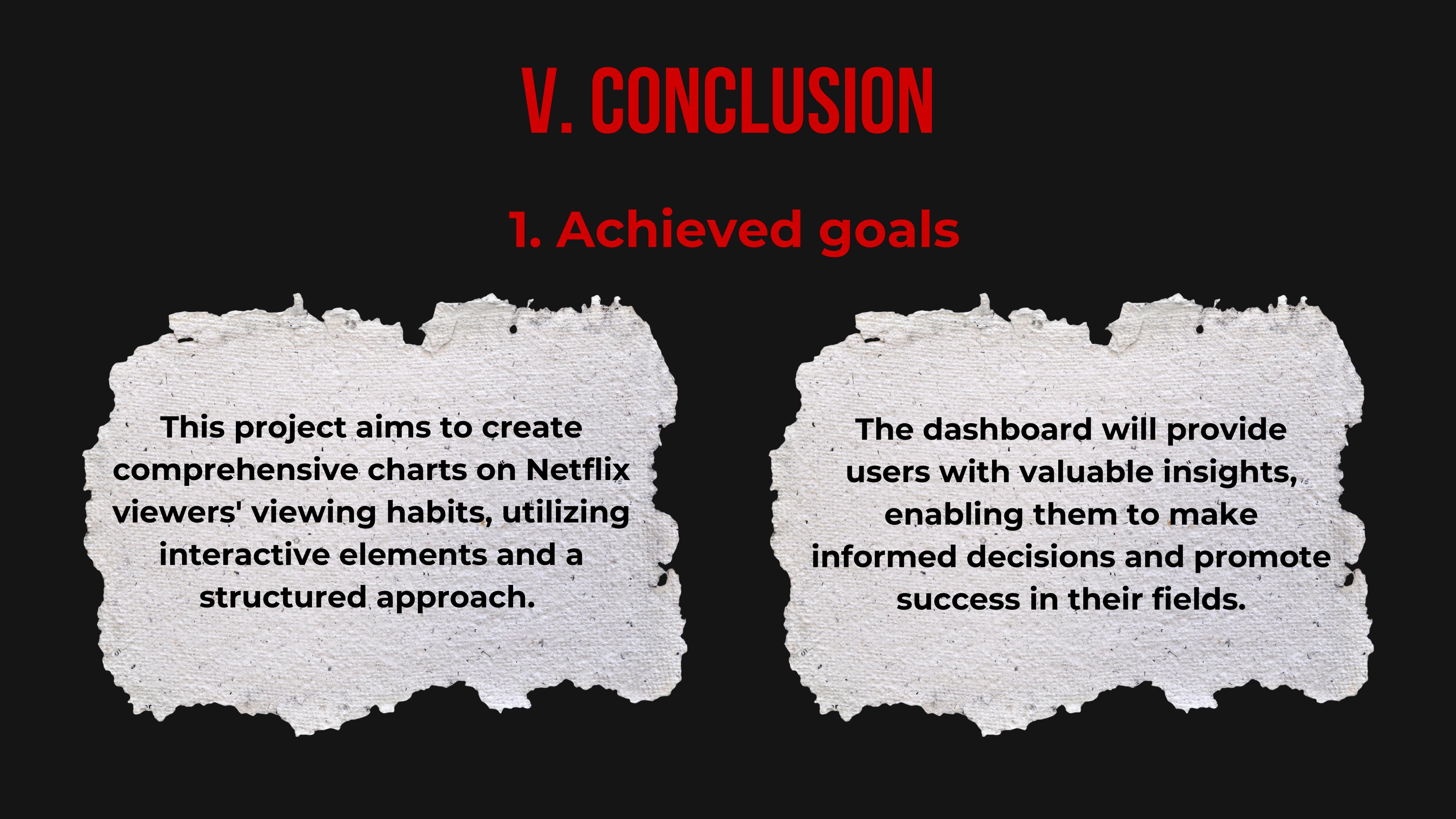


SCENCE 3



V. CONCLUSION

1. Achieved goals



This project aims to create comprehensive charts on Netflix viewers' viewing habits, utilizing interactive elements and a structured approach.

The dashboard will provide users with valuable insights, enabling them to make informed decisions and promote success in their fields.

V. CONCLUSION

2. Future work



Future work aims to enhance the dashboard's usability and functionality by adding detailed text descriptions, drill-down tools, interactive filters, and real-time data updates. This will enhance user experience and provide more insightful data analysis.



V. CONCLUSION

3. Concluding

**We want to thank the lecturer
for helping and supporting us.**

**We're grateful for the clear and
well-documented resources that
have helped us learn and grow.**

We're excited to keep moving forward and learn more !

**We also want to
congratulate the team
members for working well
together and achieving great
results through effective
teamwork.**

**Our collaborative effort has
resulted in creating a dashboard
that's easy to use and helps
people explore data in a better
way.**

VI. REFERENCES

1. Kaggle: Your Machine Learning and Data Science Community
2. Resources/css/stylemap.css
3. <https://cdnjs.cloudflare.com/ajax/libs/twitterbootstrap/4.3.1/css/bootstrap.min.css>
4. <https://stackpath.bootstrapcdn.com/fontawesome/4.7.0/css/fontawesome.min.css>
5. <https://unpkg.com/leaflet@1.4.0/dist/leaflet.css>

THANKS FOR
PLAYING

▶ End

