Movie LINQs

Collections are great for storing multiple instances of a data type, but it’s often the case that we need to filter, manipulate, or analyze this data in bulk. C# includes LINQ (language integrated query) which allows us to query our data much like a database.

For this assignment, you will be supplied with a DLL containing movie data to be analyzed. Using this provided DLL, you will write LINQ statements to answer a series of questions.

# Requirements:

* Create a new project (I recommend a console project, but you may use Win Forms or WPF as you wish).
* Download and integrate the provided DLL.
* Using the classes and built-in data from the DLL, write LINQ statements to answer the queries listed below.
  + For each query, copy/paste your full LINQ statement(s) (and any associated code as needed) as well as the answer result for the query into this document. Paste your answer beneath the appropriate query, please.

# Queries

1. What are the years of the oldest and newest movies in the list?

var newestMovie = movies.Aggregate((m1, m2) => m1.Year > m2.Year ? m1 : m2);

var oldestMovie = movies.Aggregate((m1, m2) => m1.Year < m2.Year ? m1 : m2);

Console.WriteLine($"{newestMovie.Title} released in {newestMovie.Year}");

Console.WriteLine($"{oldestMovie.Title} released in {oldestMovie.Year}");

THE LORD OF THE RINGS: THE FELLOWSHIP OF THE RING released in 2001

INTOLERANCE released in 1916

1. Split the total year range in half. Movies on the lower half are considered “classic” while those on the upper half are considered “new”. How many classic movies are there? How many new movies?

var newestMovie = movies.Aggregate((m1, m2) => m1.Year > m2.Year ? m1 : m2);

var oldestMovie = movies.Aggregate((m1, m2) => m1.Year < m2.Year ? m1 : m2);

int middleYear = ((newestMovie.Year - oldestMovie.Year) / 2) + oldestMovie.Year;

var newMovies = movies.Where(movie => movie.Year > middleYear);

var classicMovies = movies.Where(movie => movie.Year <= middleYear);

Console.WriteLine($"{newMovies.Count()} new movies");

Console.WriteLine($"{classicMovies.Count()} classic movies");

60 new movies

40 classic movies

1. Are there more odd years or even years in the list? Show the quantities of both, please.

var oddMovies = movies.Where(movie => movie.Year % 2 == 1);

var evenMovies = movies.Where(movie => movie.Year % 2 == 0);

Console.WriteLine($"Even: {evenMovies.Count()}");

Console.WriteLine($"Odd: {oddMovies.Count()}");

Even: 51

Odd: 49

1. Which year contains the most titles? What is the quantity?

var releaseYears = movies.GroupBy(movie => movie.Year);

var bestReleases = releaseYears.OrderBy(years => years.Count());

var bestRelease = releaseYears.Aggregate((g1, g2) => g1.Count() > g2.Count() ? g1 : g2);

int numOfMovies = bestRelease.Count();

foreach(var release in bestReleases)

{

if(release.Count() >= numOfMovies)

{

numOfMovies = release.Count();

Console.WriteLine($"{release.Key} has {release.Count()} movies.");

}

}

1982 has 4 movies.

1969 has 4 movies.

1976 has 4 movies.

1. Which rating shows up the most? What is the quantity?

var ratingsByMovies = movies.GroupBy(movie => movie.Rating);

var bestRating = ratingsByMovies.Aggregate((m1, m2) => m1.Count() > m2.Count() ? m1 : m2);

Console.WriteLine($"{bestRating.Count()} {bestRating.Key} movies");

32 NonRated movies

1. Which title is the shortest (in character length)? Which is the longest? Your answer should be the titles themselves (but no other data).

var longestestMovie = movies.Aggregate((m1, m2) => m1.Title.Length > m2.Title.Length ? m1 : m2);

var shortestMovie = movies.Aggregate((m1, m2) => m1.Title.Length < m2.Title.Length ? m1 : m2);

Console.WriteLine($"{longestestMovie.Title}");

Console.WriteLine($"{shortestMovie.Title}");

THE LORD OF THE RINGS: THE FELLOWSHIP OF THE RING

JAWS

1. Which rating has the widest range in years?

var ratings = movies.GroupBy(movie => movie.Rating);

int highestCount = 0;

MPAARating popularRating = MPAARating.Unknown;

var newestYear = 0;

var oldestYear = 0;

foreach( var rating in ratings)

{

var newestMovie = rating.Aggregate((m1, m2) => m1.Year > m2.Year ? m1 : m2);

var oldestMovie = rating.Aggregate((m1, m2) => m1.Year < m2.Year ? m1 : m2);

int range = newestMovie.Year - oldestMovie.Year;

if(range > highestCount)

{

highestCount = range;

popularRating = rating.Key;

newestYear = newestMovie.Year;

oldestYear = oldestMovie.Year;

}

}

Console.WriteLine($"{popularRating}: {oldestYear}-{newestYear}");

G: 1931-1995

1. How many films are in each decade of the total year range?

var decadeMovies = movies.GroupBy(movies => (movies.Year / 10) \* 10);

decadeMovies = decadeMovies.OrderBy(decade => decade.Key);

foreach(var decade in decadeMovies)

{

Console.WriteLine($"{decade.Key}-{decade.Key + 9} has {decade.Count()} movies.");

}

1910-1919 has 1 movies.

1920-1929 has 3 movies.

1930-1939 has 12 movies.

1940-1949 has 11 movies.

1950-1959 has 16 movies.

1960-1969 has 17 movies.

1970-1979 has 20 movies.

1980-1989 has 8 movies.

1990-1999 has 11 movies.

2000-2009 has 1 movies.

1. Which ratings in the MPAARating enum are never used?

var ratingsByMovies = movies.GroupBy(movie => movie.Rating);

List<MPAARating> ratings = Enum.GetValues(typeof(MPAARating)).Cast<MPAARating>().ToList<MPAARating>();

foreach (var rating in ratingsByMovies)

{

if (ratings.Contains(rating.Key))

{

ratings.Remove(rating.Key);

}

}

ratings.Print();

Unknown, NC17, Unrated

1. Display all films, first by rating (lowest to highest enum value), then by title (alphabetically ascending).

var ratings = movies.GroupBy(movie => movie.Rating);

ratings = ratings.OrderBy(movie => movie.Key);

foreach(var rating in ratings)

{

var titles = rating.OrderBy(movie => movie.Title);

foreach(var movie in titles)

{

Console.WriteLine(movie);

}

}

2001: A SPACE ODYSSEY (1968) - G

BEN-HUR (1959) - G

CITY LIGHTS (1931) - G

GONE WITH THE WIND (1939) - G

MODERN TIMES (1936) - G

SINGIN' IN THE RAIN (1952) - G

SNOW WHITE AND THE SEVEN DWARFS (1937) - G

THE SOUND OF MUSIC (1965) - G

TOY STORY (1995) - G

A STREETCAR NAMED DESIRE (1951) - PG

ALL THE PRESIDENT'S MEN (1976) - PG

AMERICAN GRAFFITI (1973) - PG

ANNIE HALL (1977) - PG

BUTCH CASSIDY AND THE SUNDANCE KID (1969) - PG

CABARET (1972) - PG

CASABLANCA (1942) - PG

CITIZEN KANE (1941) - PG

DR. STRANGELOVE (1964) - PG

E.T. THE EXTRA-TERRESTRIAL (1982) - PG

HIGH NOON (1952) - PG

IT'S A WONDERFUL LIFE (1946) - PG

JAWS (1975) - PG

LAWRENCE OF ARABIA (1962) - PG

RAIDERS OF THE LOST ARK (1981) - PG

REAR WINDOW (1954) - PG

ROCKY (1976) - PG

STAR WARS (1977) - PG

THE AFRICAN QUEEN (1951) - PG

THE BRIDGE ON THE RIVER KWAI (1957) - PG

THE GRADUATE (1967) - PG

THE WIZARD OF OZ (1939) - PG

TOOTSIE (1982) - PG

VERTIGO (1958) - PG

FORREST GUMP (1994) - PG13

SPARTACUS (1960) - PG13

THE LORD OF THE RINGS: THE FELLOWSHIP OF THE RING (2001) - PG13

THE SIXTH SENSE (1999) - PG13

TITANIC (1997) - PG13

A CLOCKWORK ORANGE (1971) - R

APOCALYPSE NOW (1979) - R

BLADE RUNNER (1982) - R

BONNIE AND CLYDE (1967) - R

CHINATOWN (1974) - R

DO THE RIGHT THING (1989) - R

EASY RIDER (1969) - R

GOODFELLAS (1990) - R

M\*A\*S\*H (1970) - R

MIDNIGHT COWBOY (1969) - R

NASHVILLE (1975) - R

NETWORK (1976) - R

ONE FLEW OVER THE CUCKOO'S NEST (1975) - R

PLATOON (1986) - R

PSYCHO (1960) - R

PULP FICTION (1994) - R

RAGING BULL (1980) - R

SAVING PRIVATE RYAN (1998) - R

SCHINDLER'S LIST (1993) - R

SOPHIE'S CHOICE (1982) - R

TAXI DRIVER (1976) - R

THE DEER HUNTER (1978) - R

THE FRENCH CONNECTION (1971) - R

THE GODFATHER (1972) - R

THE GODFATHER PART II (1974) - R

THE LAST PICTURE SHOW (1971) - R

THE SHAWSHANK REDEMPTION (1994) - R

THE SILENCE OF THE LAMBS (1991) - R

THE WILD BUNCH (1969) - R

UNFORGIVEN (1992) - R

12 ANGRY MEN (1957) - Not Rated

A NIGHT AT THE OPERA (1935) - Not Rated

ALL ABOUT EVE (1950) - Not Rated

BRINGING UP BABY (1938) - Not Rated

DOUBLE INDEMNITY (1944) - Not Rated

DUCK SOUP (1933) - Not Rated

IN THE HEAT OF THE NIGHT (1967) - Not Rated

INTOLERANCE (1916) - Not Rated

IT HAPPENED ONE NIGHT (1934) - Not Rated

KING KONG (1933) - Not Rated

MR. SMITH GOES TO WASHINGTON (1939) - Not Rated

NORTH BY NORTHWEST (1959) - Not Rated

ON THE WATERFRONT (1954) - Not Rated

SHANE (1953) - Not Rated

SOME LIKE IT HOT (1959) - Not Rated

SULLIVAN'S TRAVELS (1941) - Not Rated

SUNRISE (1927) - Not Rated

SUNSET BLVD. (1950) - Not Rated

SWING TIME (1936) - Not Rated

THE APARTMENT (1960) - Not Rated

THE BEST YEARS OF OUR LIVES (1946) - Not Rated

THE GENERAL (1927) - Not Rated

THE GOLD RUSH (1925) - Not Rated

THE GRAPES OF WRATH (1940) - Not Rated

THE MALTESE FALCON (1941) - Not Rated

THE PHILADELPHIA STORY (1940) - Not Rated

THE SEARCHERS (1956) - Not Rated

THE TREASURE OF THE SIERRA MADRE (1948) - Not Rated

TO KILL A MOCKINGBIRD (1962) - Not Rated

WEST SIDE STORY (1961) - Not Rated

WHO'S AFRAID OF VIRGINIA WOOLF? (1966) - Not Rated

YANKEE DOODLE DANDY (1942) - Not Rated

1. Group the movies by number of words in the title. List the results by group (meaning the number of words in the title) and the number of films in that group. Clearly label your results, please.

var groupTitleLength = movies.GroupBy(movie => movie.Title.Split(' ').Count());

groupTitleLength = groupTitleLength.OrderBy(movie => movie.Key);

foreach(var length in groupTitleLength)

{

Console.WriteLine($"{length.Key} word long titles has {length.Count()} movies.");

}

1 word long titles has 20 movies.

2 word long titles has 31 movies.

3 word long titles has 21 movies.

4 word long titles has 15 movies.

5 word long titles has 5 movies.

6 word long titles has 7 movies.

10 word long titles has 1 movies.

1. My teenager is 15 and may watch any film that is rated G, PG, or PG-13. How many films from the master list could they watch? Which films are they?

var kidsMovies = movies.Where(movie => movie.Rating == MPAARating.PG13 || movie.Rating == MPAARating.PG || movie.Rating == MPAARating.G);

Console.WriteLine($"There are {kidsMovies.Count()} movies that are non-adult friendly.");

StringBuilder build = new StringBuilder();

foreach(var movie in kidsMovies)

{

build.Append($"{movie.Title}, ");

}

build.Length -= 2;

Console.WriteLine($"These films are: {build}");

There are 38 movies that are non-adult friendly.

These films are: CITIZEN KANE, CASABLANCA, SINGIN' IN THE RAIN, GONE WITH THE WIND, LAWRENCE OF ARABIA, VERTIGO, THE WIZARD OF OZ, CITY LIGHTS, STAR WARS, 2001: A SPACE ODYSSEY, THE GRADUATE, IT'S A WONDERFUL LIFE, E.T. THE EXTRA-TERRESTRIAL, HIGH NOON, SNOW WHITE AND THE SEVEN DWARFS, ANNIE HALL, THE BRIDGE ON THE RIVER KWAI, DR. STRANGELOVE, THE SOUND OF MUSIC, A STREETCAR NAMED DESIRE, REAR WINDOW, THE LORD OF THE RINGS: THE FELLOWSHIP OF THE RING, JAWS, ROCKY, AMERICAN GRAFFITI, CABARET, THE AFRICAN QUEEN, RAIDERS OF THE LOST ARK, TOOTSIE, BUTCH CASSIDY AND THE SUNDANCE KID, FORREST GUMP, ALL THE PRESIDENT'S MEN, MODERN TIMES, SPARTACUS, TITANIC, THE SIXTH SENSE, TOY STORY, BEN-HUR

# What you’ll need

* LINQ
* Referencing external DLLs

# You should check out

* MSDN documentation on the Enumerable class

# Rubric

**Automatic 0** – You fail to use LINQ in each query, your answers are not readable, or your doc is not able to be graded for any reason.

You will be awarded 8.33 points per query you answer correctly.