

Submission P4: Work with the database

Queries executed in mongodb compass

1: Top 10 Movies with total number of votes.

```
Db.imdb_movies.aggregate([
  {$Project :{title:1,
    votes:1 },
  {$sort: votes : -1},
  {$limit :10} ])
```

2: Top/bottom N Movies by year by genre

```
Db.imdb_movies.aggregate([{$lookup
  from: 'Ratings',
  localField: 'imdb_title_id',
  foreignField: 'imdb_title_id',
  as: 'MovieRatings'
},
{$limit :100},
{$group
  _id:{
    Year: "$year",
    Genre: "$genre"
  },
  avg_rating : {$max:"$MovieRatings.us_voters_rating"}}
])
```

3: Top best directors based on overall ratings.

```
Db.imdb_movies.aggregate([{$group
  _id:{
    Director : "$director"
  },
  avg_ratings:{$avg:"$avg_vote"}
},
{$sort:{
  avg_ratings: -1
}
])
```

4: Top best writers based on overall ratings.

```
Db.imdb_movies.aggregate([{$group
  _id:{
    Writer : "$writer"
  },
  avg_ratings:{$avg:"$avg_vote"}
},
$sort:{
  avg_ratings: -1
}
])
```

5: Top movies with Highest Budget

```
Db.imdb_movies.aggregate([{$group
{
  _id:
    {title:"$title"},
  budget:{$max:"$budget"}
},
$sort:{
  budget:-1
}
])
```

6: Top Movies with highest USA Gross Income

```
Db.imdb_movies.aggregate([{$group
{
  _id:
    {title:"$title"},
  USA_Gross_Income:{$max:"$usa_gross_income"}
},
$sort:{
  USA_Gross_Income: -1
}
])
```

7. List the top 50 movies with a mean vote of more than 7

```
Db.imdb_movies.aggregate([{$match avg_vote :{$gte: 7} },
{$sort avg_vote: -1},
{$limit: 50}])
```

8. List the top 10 movies with the highest USA gross

```
Db.imdb_movies.aggregate([{$project title :1, usa_gross_income:1},
{$sort usa_gross_income: -1},
{$limit 10}])
```

9. director by high votes by year

```
Db.imdb_title_principals([{$Project :{
director:1,
year:1,
votes:1
},
{$limit :100},
$group:{
  _id: {year:"$year"},
  High_Votes:
  {
    $max: "$votes"
  }
}])
```

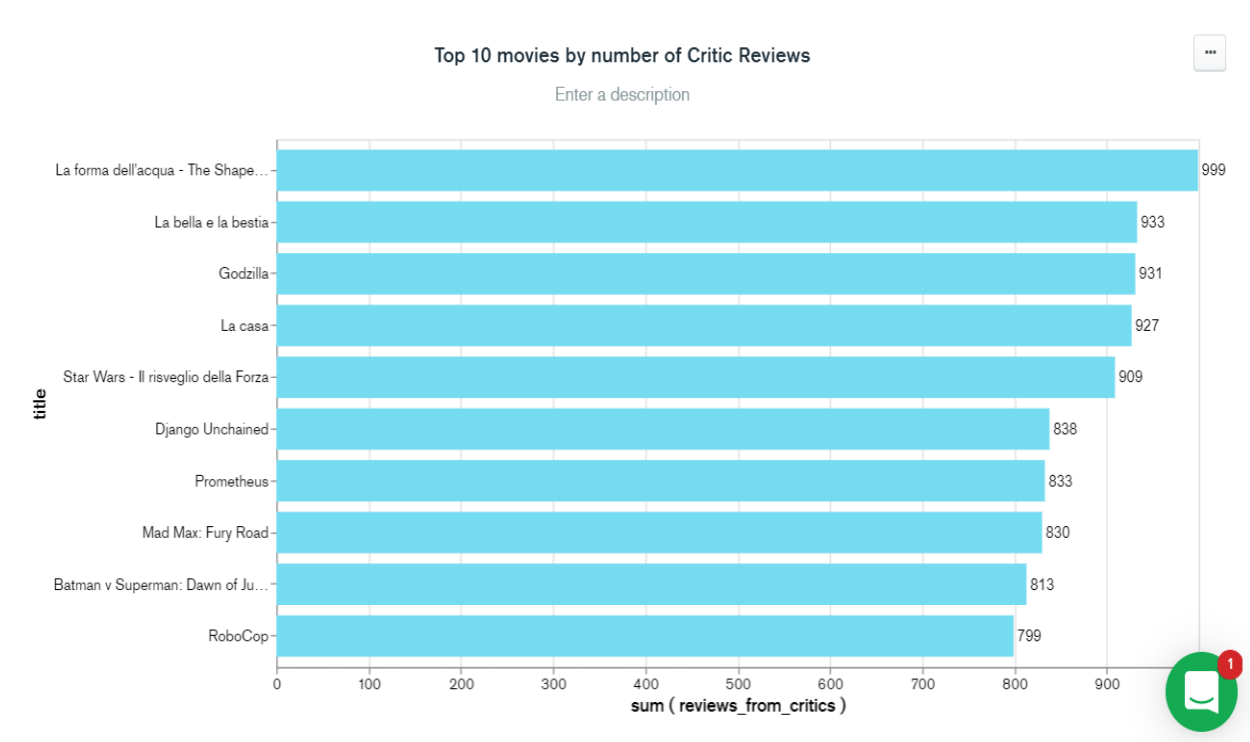
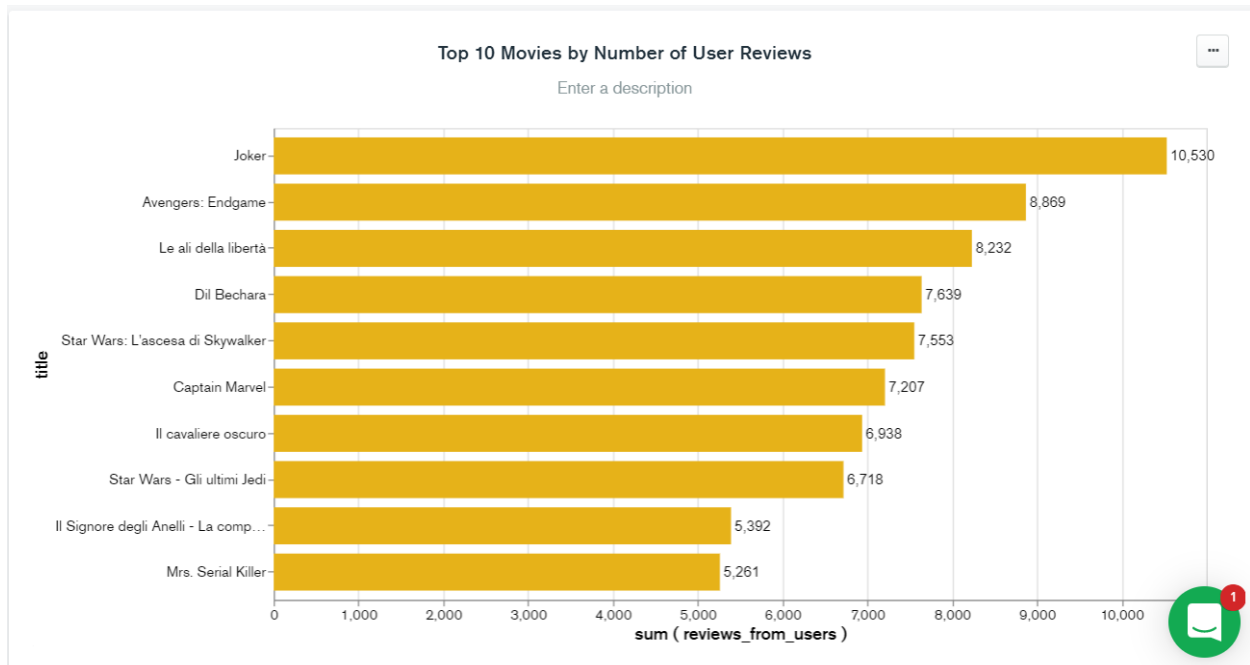
10. Movie with longest duration, return by title, year and duration.

```
Db.imdb_movies.aggregate([$group:{
  _id: {year:"$year",title:"$title"},
  duration: {
    $avg: "$duration"
  }
},
$Sort:{
  duration: 1
}])
```

11. Movie is in 1950 year which has duration greater than 100

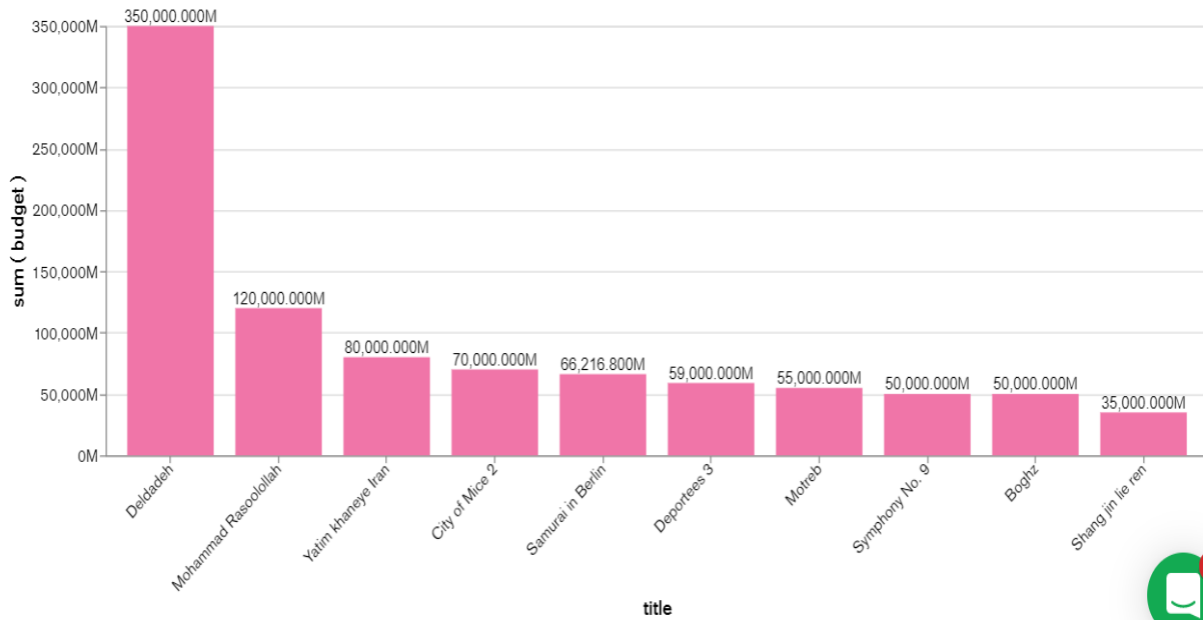
```
Db.imdb_movies.aggregate([$match: {
  year:{$in:[1950]}
},
$match:{
  duration:{$gte:100}
},
$sort: {
  duration: -1
}])
```

Visualization charts from dashboard designed in MongoCharts



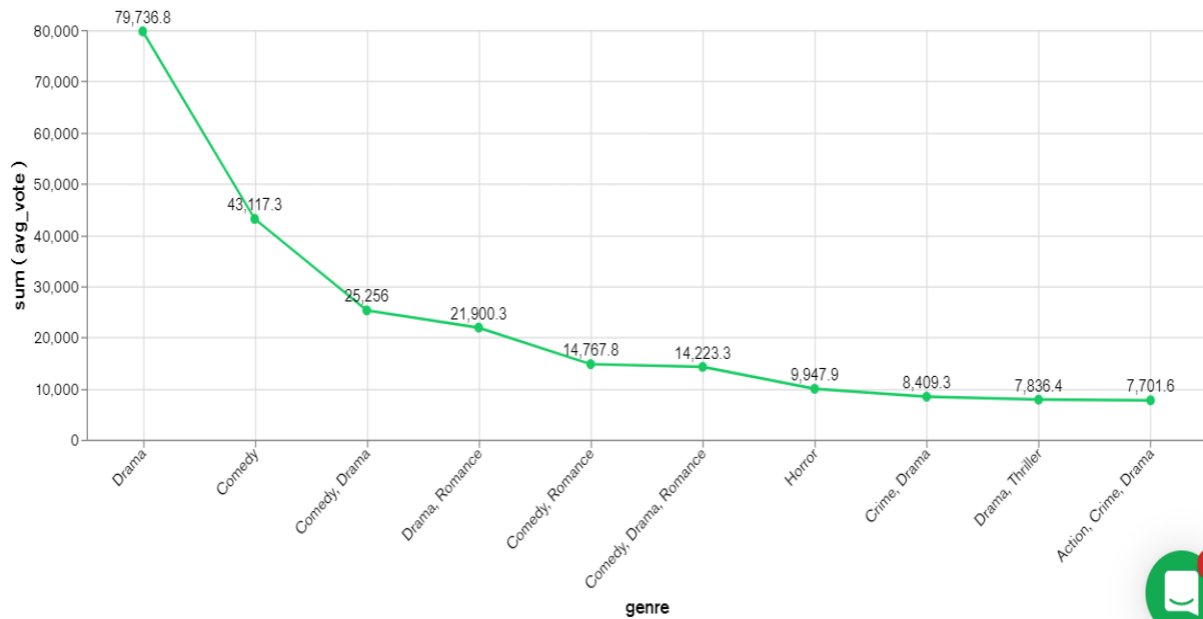
Top 10 movies by Budget

Enter a description



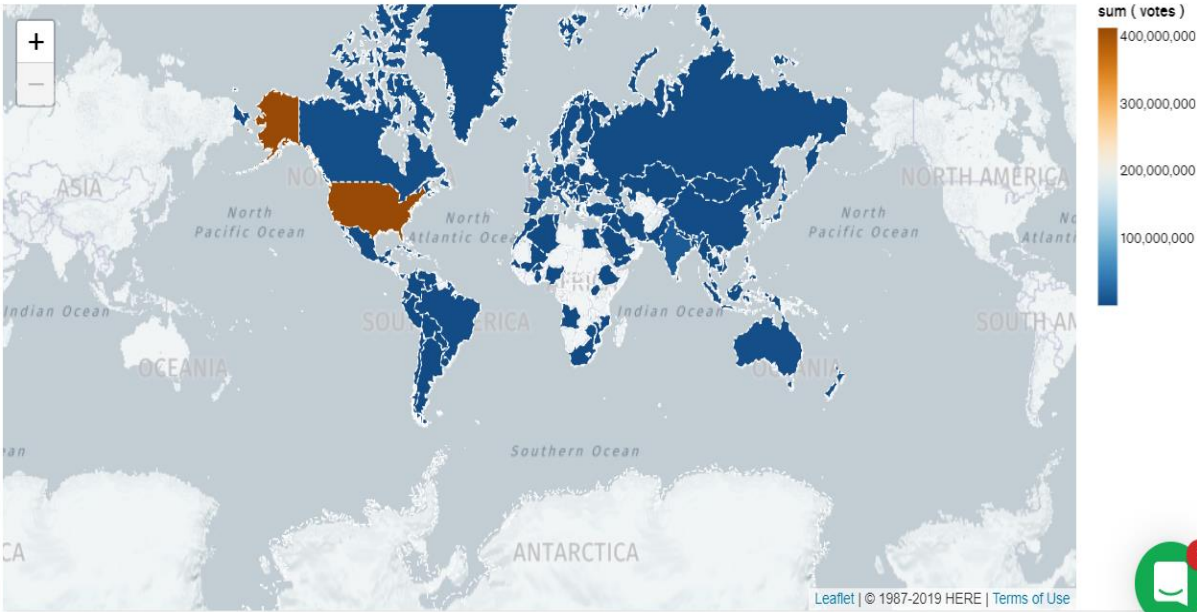
Top 10 Genres by Average Votes

Enter a description



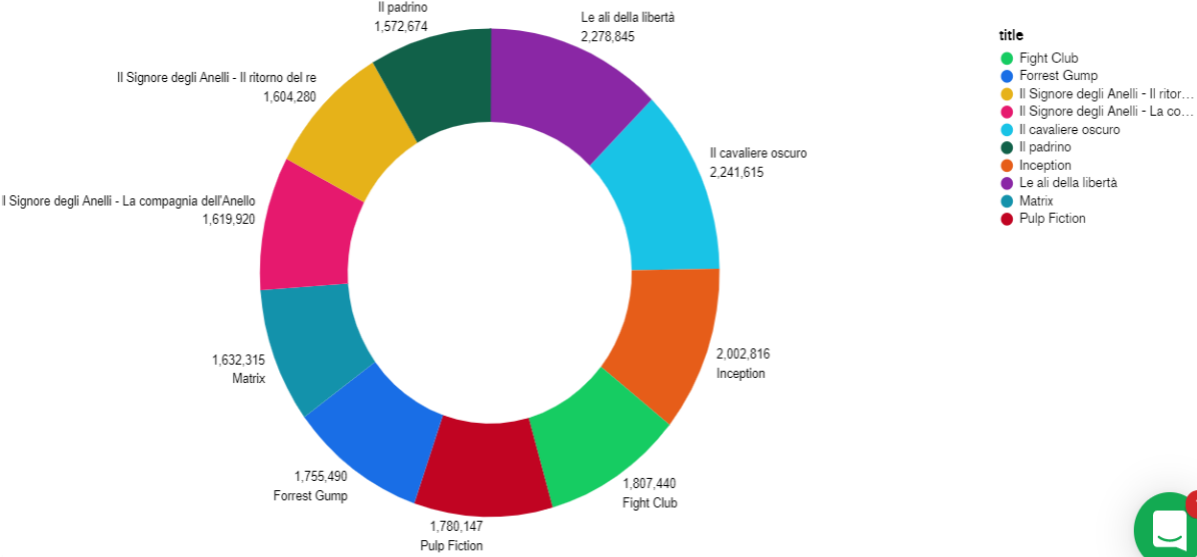
Countries by Number of Votes

Enter a description



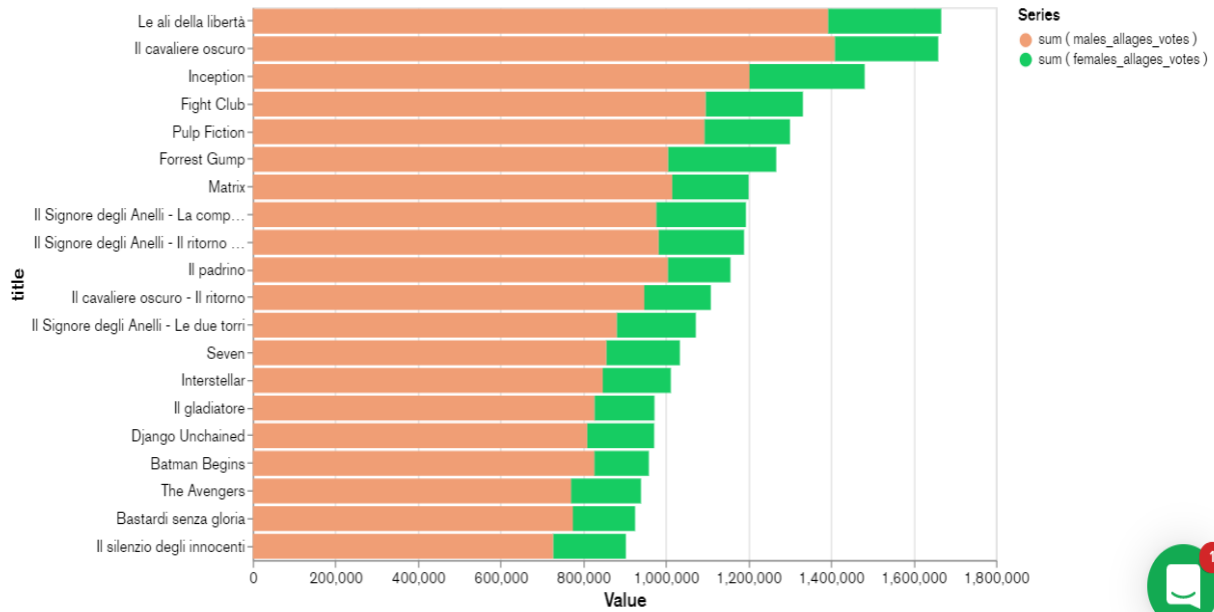
Movies by Total Votes

Enter a description



Movies by Number of Votes by Males and Females

Enter a description



Place of Birth by Number of Actors

Enter a description



