Data Management Technologies

Hooshyar Hosna (Student Com20)

Contents

Domain Analysis and Description 2
Motivation 2
Business Process Modeling 2
Business questions 3
Conceptual Design 3
Dimensions 3
Measures 4
Logical Design 5
Implementation 6
Data collection and pre-processing 6
SQL script 7
Approximate size of tables 9
Querying 10
Answering business questions 10
Question 1: Which age ratings have won the most awards? 10
Question 2: Which directors have won the most Oscar awards between 2000 to 2016? 12
SQL OLAP Extensions 14
CUBE 14
ROLLUP 15
GROUPING SETS 16
Window functions 18
Ranking query 18
Window query 19
Period-to-period comparison 20
Conclusion 22
References 23

Domain Analysis and Description

The **Academy Awards**, popularly known as the **Oscars**, are awards for artistic and technical merit in the film industry. Although recently many cinema enthusiasts have questioned their prestige and the rating of the ceremony has dropped, the Oscars are still regarded by many as the most prestigious and significant awards in the entertainment industry worldwide, and the truth is even if Oscars are not the best indicators of quality, they are indeed the most important awards in the eyes of ordinary viewers and film companies.

One of the main reasons that give Oscars so much influence, is the impact of the awards on how much they make in Boxoffice. For example, it was predicted that the movie *The King's speech* (2010) to grow 30 million dollars but after it won 4 Oscars, the number went up to 400 million dollars. So, it is not a surprise that everyone involved in producing a movie and especially film companies try to be at least on the final nomination list.

Motivation

The **Academy of Motion Picture Arts and Sciences** is the organization that votes for the Oscar winners, and they have been accused of caring more about public's opinion rather than the real quality when it comes to choosing Oscar's winner -a fair assumption considering *Titanic* won the Oscar for best picture in 70th Academy Awards in 1998- but in recent years the Academy Awards faced a much bigger critique which says the awards are rigged. In other words, there is a formula to win an Oscar, and knowing that can increase the chances of winning significantly. This formula is a combination of elements such as the genre, the main actor's gender, history and fame, the theme of the story, its relation to current political events, the runtime, the age of the targeted audience, and the first language of the movie. A data warehouse will be built to analyze different aspects of this theory.

Business Process Modeling

The analyzed and modeled business process is any movie that won at least one Oscar award from 2000 to 2016 in one of the *Big Five* categories:

- Best Picture
- Best Director
- Best Actor
- Best Actress
- Best Screenplay

Winning an award in these categories establishes a movie as a movie of high quality in the public's opinion and they have more weight than other categories such as costume design. That is why these five categories are the focus of the DW.

The primary event of this DW is a movie winning an Oscar for one of these categories in years between 2000 and 2016.

Business questions

The business questions are about the factors which are believed to prove there is a formula for increasing the chance of winning an Oscar.

- What is the average score that is given to movies by viewers?
- What is the average score that is given to movies by critics?
- Does the viewers' idea of a great movie, match one of critics?
- What is the average duration of the movies?
- How much the movies have been sold?
- To which countries do most of the movies belong?
- In which language do most of the characters talk?
- To which gender most of the main characters belong?
- Is there a connection between age restriction and the gender of main character?
- Which age groups have won the most awards?
- In which period of the year most movies were released?
- What genre do most movies belong to?
- What is the average number of award nominations the movies got?
- Which directors/actors have won the most Oscar awards between 2000 to 2016?

Conceptual Design

Dimensions

There are six dimensions that describe an analysis coordinate of the fact:

1. **Title**: This dimension indicates the title of the movie and includes five attributes
   - Synopsis
   - IMDb page
   - Genre
   - Number of award nominations
   - Number of won awards
   - Number of won Oscars.

   The first two attributes are descriptive. There is a many-to-many association between *Title* and *Genre* since each movie can be categorized in more than one genre and each genre includes more than one movie

   Number of award nominations demonstrates the number of all types of awards that the movie has been nominated for and not just the Oscars.

2. **Release date:** A hierarchy which goes from release date to the day of release, to the month, to the year.
3. **Country of origin**: The country that produced the movie (this is not the country of movie's location). The attributes are:

- Continent
- Language

4. **Rating**: This dimension is about the scores the movies received from two different websites which are the attributes:
   - IMDb attribute shows the viewers' opinions since it is a platform anyone can easily go to and rate a movie and everyone's vote has the same weight. In addition to the score, the IMDb hierarchy includes the number of votes that resulted in the score and the number of reviews written by viewers.
   - Metacritic Score which is the score from Metacritic website and the indicator for the opinions of critics.
5. **Cast**: For this dimension, two main roles of the movies' casts are considered
   - Director
   - Main actor

   The IMDb pages of both are included as descriptive attributes. The gender of the main actors is also mentioned in this dimension.
6. **Age restriction**: Films' age rates are decided by the **Motion Picture Association film rating system** to protect children from unsuitable and even harmful content in movies. The age rates are:
   - G: General Audiences
   - PG: Parental Guidance Suggested
   - PG-13: Parents Strongly Cautioned
   - R: Restricted (Contains some adult material)
   - NC-17: Adults Only

   If a movie is not suitable for all ages, its IMDb page expresses why. As a result, the attribute named reason is optional.

Measures

Two numerical properties describe quantitative aspects of the fact:

1. **Gross**: Refers to gross earnings of a movie and is an additive measure.
2. **Duration**: How long each movie is in minutes. Since summing this measure along any dimension, makes no sense, this measure is non-additive.

The dimensional fact model is shown in figure 1.

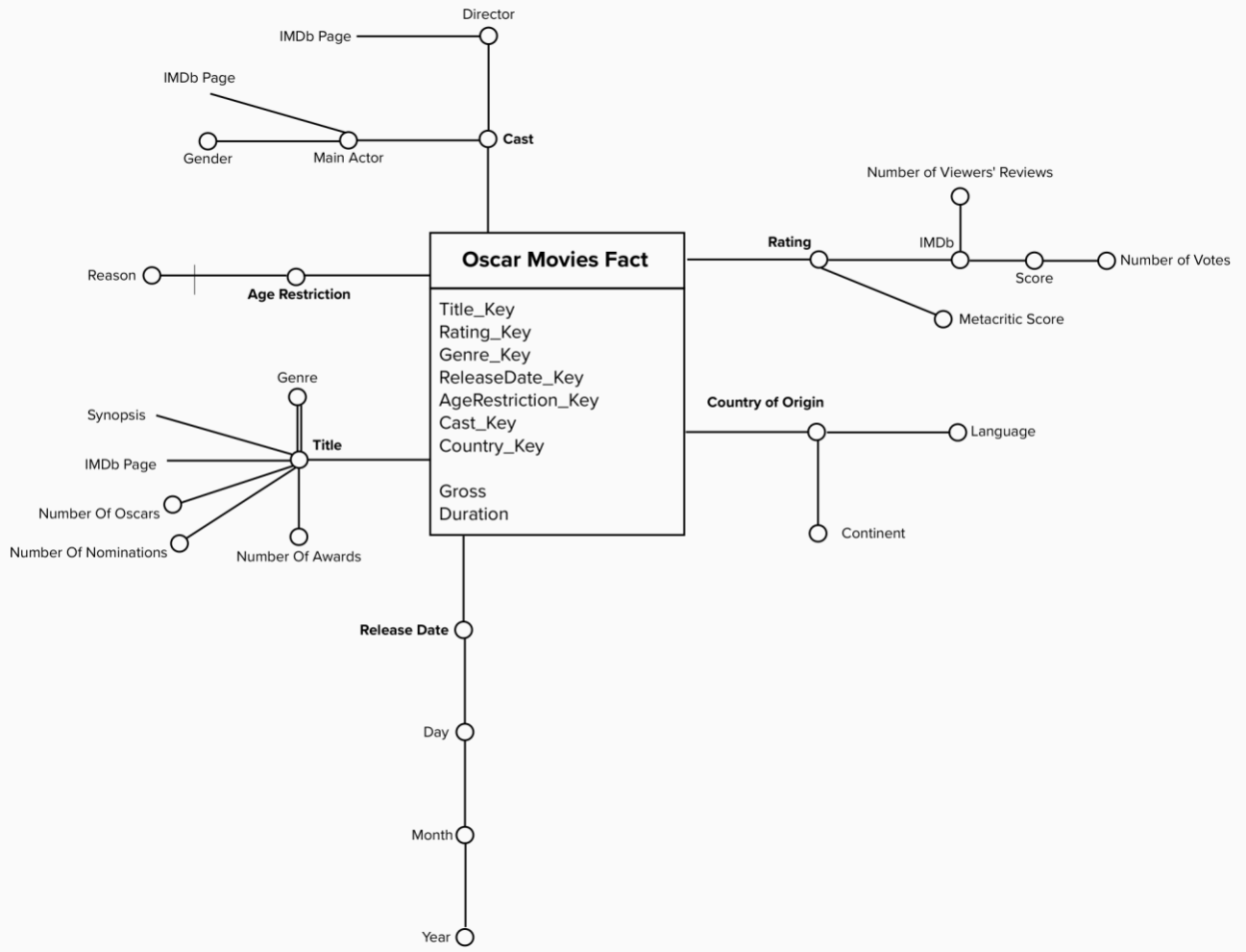

Figure 1- Dimensional fact model of the *Oscar Movies fact*[1]

Logical Design

The star schema of the fact model is drawn by *QuickDBD*, and it can be seen in figure 2. The type of domain is illustrated in each row and the key symbol indicates the primary key. There are two key symbols in the *Genre_Bridge* table meaning the primary key of this table is a combination of two values.

As it was mentioned before, there is a multiple arc that models the many-to-many relationship between genres and titles. To handle this arc, a bridge table, named *Genre_Bridge*, has been used. And to make it easier to understand the connection between *Title* and *Genre* tables, it is drawn as a snowflake schema. The process of finding the genre(s) of a movie will be explained later.

The tables are connected through foreign keys e.g., *title_key* of *Oscar_Movies_Fact* table is a foreign key that references to the primary key of *Title* table, *Title_ID*.

[1] Drawn by mural.co/

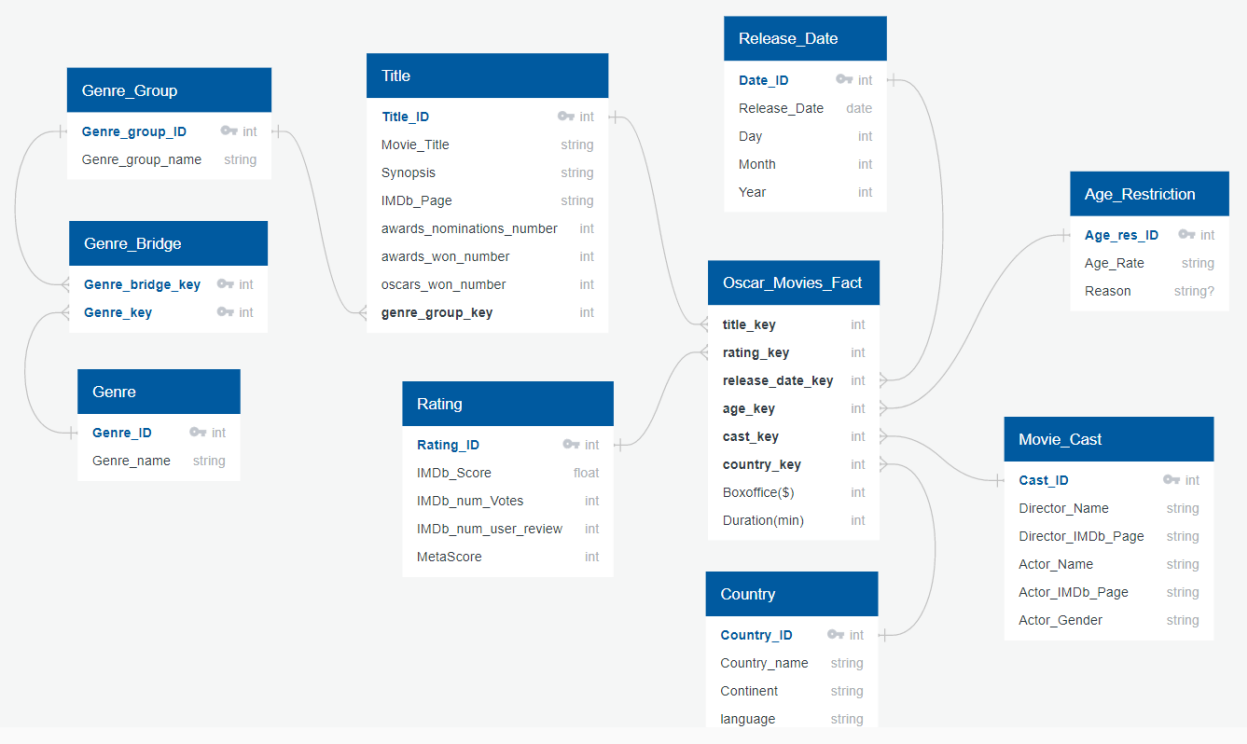

Figure 2 - The star schema of the *Oscar Movies* fact model

Implementation

Data collection and pre-processing

Predictably, the data that is completely aligned with this business process was not found so the closest available match was used. This data that was found on Kaggle originally includes 1183 rows and 119 columns and is about all the movies that have been nominated for an Oscar in any category. All the pre-processing steps including cleaning are done with Python programming language.

In the first step, any movie that has not won an Oscar is removed from the data. Some columns such as *Los Angeles Film Critics Association* -which gives information about the awards of this association- are also deleted because they did not play a role in answering the defined business questions.

The different genres of each movie were in one column as one string and they needed to be separated. The *Genre* table includes all the genres of the data. The *Genre_Group* table contains groups of genres, and each group belongs to at least one Oscar-winning movie. To make the understanding of the process easier, each group is named by combining its genres e.g., *Biography_Drama*.

After removing the awards' categories other than the five that have been already mentioned, a column is created to show the number of Oscar awards that each movie has won.

At the same time, some of the data that are needed to build the data warehouse, could not be found in this database. This information that has been found by visiting the IMDb page of each movie is of directors, actors, link to IMDb pages, country of origin, the continent of origin, the language of the movie, and the reasons for movies' age rate. Also, the null values of the data have been filled with the help of IMDb and Wikipedia.

SQL script

The SQL script below creates the dimension tables and the fact table:

```
--- The dimension tables
CREATE TABLE age_restriction(
age_res_ID int PRIMARY KEY,
age_rate VARCHAR(6) NOT NULL,
reason character varying);

CREATE TABLE Movie_cast(
cast_ID int PRIMARY KEY,
Director_name character varying NOT NULL,
Director_IMDb_page character varying,
Actor_name character varying,
Actor_IMDb_page character varying,
Actor_Gender VARCHAR(6));

CREATE TABLE country(
country_id int PRIMARY KEY,
country_name character varying NOT NULL,
language character varying NOT NULL,
continent character varying NOT NULL);

CREATE TABLE Genre(
Genre_ID int PRIMARY KEY,
Genre_name character varying NOT NULL);

CREATE TABLE Genre_Group(
Genre_group_ID int PRIMARY KEY,
Genre_group_name character varying);

CREATE TABLE Genre_Bridge(
Genre_Bridge_key int,
Genre_key int,
FOREIGN KEY (Genre_Bridge_key) REFERENCES Genre_Group(Genre_group_ID),
FOREIGN KEY (Genre_key) REFERENCES Genre(Genre_ID),
PRIMARY KEY (Genre_Bridge_key, Genre_key));

CREATE TABLE Rating(
```

```
Rating_ID int PRIMARY KEY,
IMDb_score float(2) NOT NULL,
IMDb_num_Votes int NOT NULL,
IMDb_num_user_review int,
metascore int);

CREATE TABLE Release_Date(
Date_ID int PRIMARY KEY,
release_date date,
day int,
month int,
year int);

CREATE TABLE Title(
Title_ID int PRIMARY KEY,
Movie_title character varying NOT NULL,
synopsis character varying,
IMDb_Page character varying,
awards_nominations_number int,
awards_won_number int,
oscars_won_number int NOT NULL,
Genre_Group_key int,
FOREIGN KEY (genre_group_key) REFERENCES genre_group(genre_group_ID));

--- The fact table:
CREATE TABLE Movie_Fact(
title_key int,
Rating_key int,
release_date_key int,
age_key int,
cast_key int,
Country_key int,
boxoffice int,
duration int,
FOREIGN KEY (title_key) REFERENCES title(title_ID),
FOREIGN KEY (Rating_key) REFERENCES rating(rating_ID),
FOREIGN KEY (release_date_key) REFERENCES release_date(date_ID),
FOREIGN KEY (age_key) REFERENCES age_restriction(age_res_ID),
FOREIGN KEY (cast_key) REFERENCES Movie_cast(cast_ID),
FOREIGN KEY (Country_key) REFERENCES Country(Country_ID));
```

All the datasets have been created in Python and implemented in these tables with *copy* command, such as the example below:

```
\copy Title from 'adressOfFile\nameOfFile.csv' CSV HEADER;
```

The script below and figure 3 show how we can find the genre(s) of a movie. For this example, the movie *12 Years a Slave* has been chosen.

```
SELECT genre_name
FROM title JOIN genre_group ON genre_group_key = genre_group_id
    JOIN genre_bridge ON genre_group_id = genre_bridge_key
    JOIN genre ON genre_key = genre_id
WHERE movie_title = '12 Years a Slave'
```

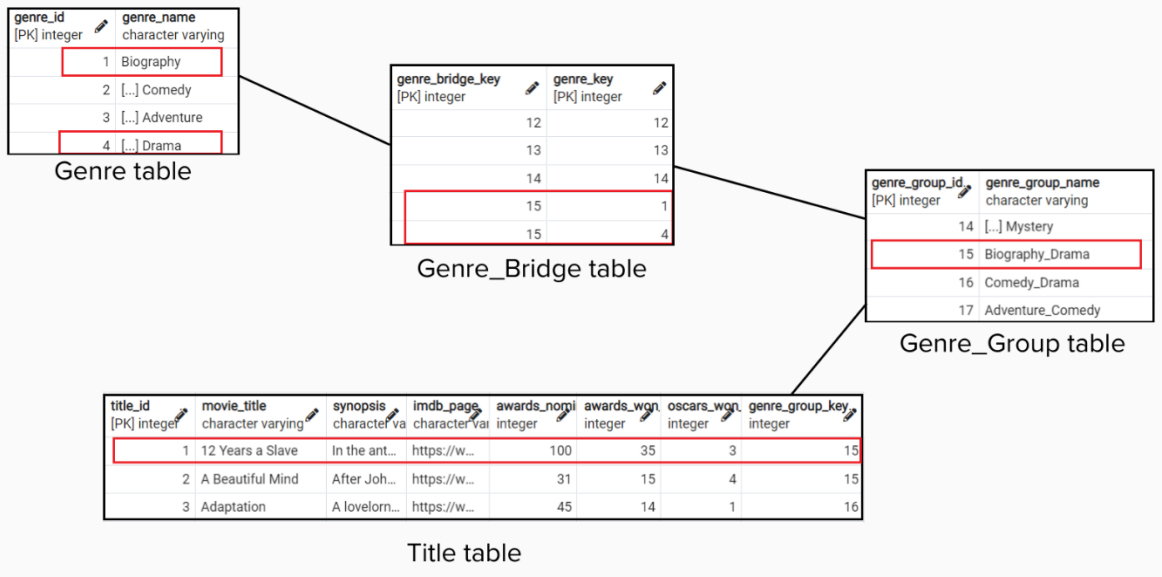

Figure 3 – The steps of finding the genres of *12 Years a Slave*

The four tables that are shown in figure 3 have been joined. *Title*'s foreign key, *genre_group_key* refers to *Genre_Group*'s primary key, *genre_group_id*. This primary key points to one or two *genre_key* in *Genre_Bridge*, from there, the genres can be found in the *Genre* table.

	genre_name character varying
1	Biography
2	[...] Drama

Figure 4 - Results of the query

Approximate size of tables

To estimate the size of each table, the size of each row should be calculated. We start with approximating the size of one row of *Title* table:

$$23(starting\ point) + 4(Title\_ID) + 40(Movie\_title) + 280(\text{synopsis}) + 60(IMDb\_page) + 4 \times 5(other\ integers) = 427\ bytes$$

There are 87 rows in this table, therefore the size of table is approximately 37 KB. The results of estimation for other tables are illustrated in table 1.

Table 1 - approximate size of each table

Name of the table	Size of each row(bytes)	Number of rows	Size of the table (KB)
Age_restriction	59	10	0.6
Movie_cast	241	87	21
Country	107	9	1
Genre	47	14	0.7
Genre_Group	47	33	1.5
Genre_Bridge	31	52	1.6
Rating	41	87	3.6
Release_Date	47	78	3.7
Title	427	87	37
Movie_fact (fact table)	55	87	4.8
Total	-	-	75.5

Querying

Answering business questions

Question 1: Which age ratings have won the most awards?

To answer this query, two tables need to be joined: *age_restriction* and *title* (because the number of awards can be found in this table).

	age_res_id [PK] integer	age_rate character varying (6)	reason character varying
1	1	R	violence
2	2	R	language
3	3	R	sexual content
4	4	R	drug use
5	5	PG-13	violence
6	6	PG-13	language
7	7	PG-13	sexual content
8	8	PG-13	drug use
9	9	PG	[null]

Figure 5 - *age_restriction* table

	title_id [PK] integer	movie_title character varying	synopsis	imdb_page	awards_nominations_nu	awards_won_number integer	oscars_won_number integer	genre_group_key integer
1	1	12 Years a Slave	In the ant...	https://w...	100	35	3	15
2	2	A Beautiful Mind	After Joh...	https://w...	31	15	4	15
3	3	Adaptation	A lovelor...	https://w...	45	14	1	16
4	4	Almost Famous	A high-sc...	https://w...	55	22	1	17

Figure 6 - Several rows of *title* table

There is no condition on which, we can join these two tables. That is why the fact table is also used in the inner join.

	title_key integer	rating_key integer	release_date_key integer	age_key integer	cast_key integer	country_key integer	boxoffice integer	duration integer
1	1	1	1	1	1	9	56670000	134
2	2	2	2	7	2	9	170710000	135
3	3	3	3	2	3	9	22250000	114
4	4	4	4	2	4	9	32520000	122
5	5	5	5	2	5	8	136020000	120

Figure 7 - Several rows of the fact table

```
SELECT age_rate, reason, SUM(oscars_won_number) AS Oscar,
    SUM(awards_won_number) AS Awards
FROM age_restriction JOIN movie_fact ON age_res_ID = age_key
    JOIN title ON title_key = title_ID
GROUP BY (age_rate,reason)
ORDER BY SUM(oscars_won_number) DESC
```

	age_rate character varying (6)	reason character varying	oscar bigint	awards bigint
1	R	language	37	420
2	R	violence	33	302
3	R	sexual content	20	190
4	PG-13	sexual content	18	202
5	PG-13	violence	11	112
6	PG-13	language	10	180
7	R	drug use	3	18
8	PG-13	drug use	2	25
9	PG	[null]	1	10

Figure 8 - The result of the query

The column *oscar* and *awards* show the number of Oscars and the number of all awards that have been won, respectively.

For visualization, the R programming language has been used and the bar chart can be seen in figure 7.

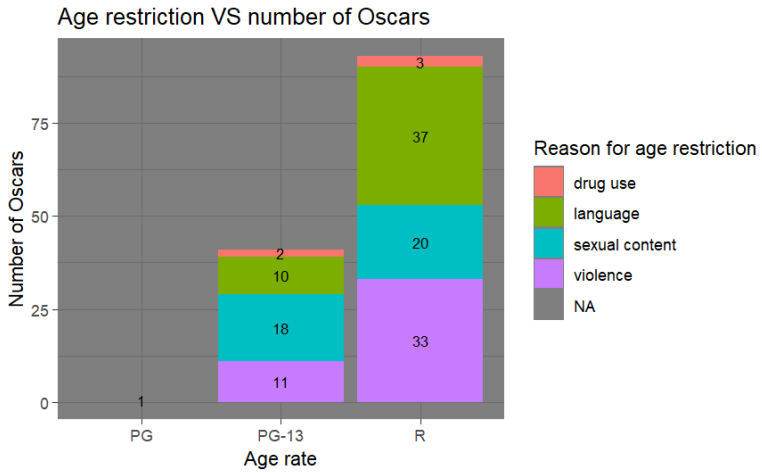

Figure 9 - Result of the query as bar chart

As it was mentioned before, there are five categories in the film rating system. There are no G (general audiences) or NC-17 (adults only) movies between the winners of 2000 to 2016 and only one PG movie. Between the two remaining age categories, around 70% of Oscars belong to movies of R rating which contain some adult material.

This can suggest that the movies with an R rating, have a better chance of being chosen as Oscar winners, especially if we look at the reasons, we can see that the most frequent reason (or in other words "adult material") is language, something that can be easily removed from a movie without damaging its quality.

Question 2: Which directors have won the most Oscar awards between 2000 to 2016?

To answer this question the fact table and these three dimension-tables will be needed:

1. *movie_cast*
2. *title*
3. *release_date*

The tables *title* and *movie_fact* were shown in the previous query and the other two can be seen here.

	cast_id [PK] integer	director_name character varying	director_imdb_page character varying	actor_name character varying	actor_imdb_page character varying	actor_gender character varying (6)
1	1	Steve McQueen	https://www.imdb....	Chiwetel Ejiofor	https://www.imd...	male
2	2	Ron Howard	https://www.imdb....	Russell Crowe	https://www.imd...	male
3	3	Spike Jonze	https://www.imdb....	Nicolas Cage	https://www.imd...	male
4	4	Cameron Crowe	https://www.imdb....	Billy Crudup	https://www.imd...	male

Figure 10 - Several rows of *movie_cast* table

	date_id [PK] integer	release_date date	day integer	month integer	year integer
1	1	2013-11-08	8	11	2013
2	2	2002-01-04	4	1	2002
3	3	2003-02-14	14	2	2003
4	4	2000-09-22	22	9	2000

Figure 11 - Several rows of *release_date* table

```
SELECT IDK.Director_name, maxOscars, boxoffice/100000 AS gross_in_million,duration,
movie_title, year
FROM (
    SELECT movie_cast.Director_name, maxOscars, cast_id
    FROM(
        SELECT Director_name, maxOscars
        FROM
            (SELECT MAX(count) as maxOscars
            FROM (
                SELECT Director_name, count(*) as count
                FROM movie_cast
                GROUP BY Director_name) as x
                    ) max_director,
            (SELECT Director_name, count(*) as count
            FROM movie_cast
            GROUP BY Director_name) num_director
        WHERE maxOscars = count) AS max_director_table,
    movie_cast
    WHERE max_director_table.Director_name = movie_cast.Director_name) AS IDK JOIN
movie_fact
ON cast_id = cast_key JOIN title
ON title_id = title_key JOIN release_date
ON date_id = release_date_key
ORDER BY boxoffice DESC
```

	director_name character varying	maxoscars bigint	gross_in_million integer	duration integer	movie_title character varying	year integer
1	Tom Hooper	3	1487	158	Les MisÃ©rables	2012
2	Tom Hooper	3	1388	118	The King's Speech	2010
3	Woody Allen	3	568	94	Midnight in Paris	2011
4	Woody Allen	3	334	98	Blue Jasmine	2013
5	Woody Allen	3	232	96	Vicky Cristina Barcelona	2008
6	Tom Hooper	3	127	119	The Danish Girl	2016

Figure 12 - Result of the query

There are two directors the won three Oscar awards - more than any other director- in years between 2000 and 2016. The name of these movies is in the results. Studying them, their story, and even their technical aspects can help to understand the preferences of the members og the Academy better.

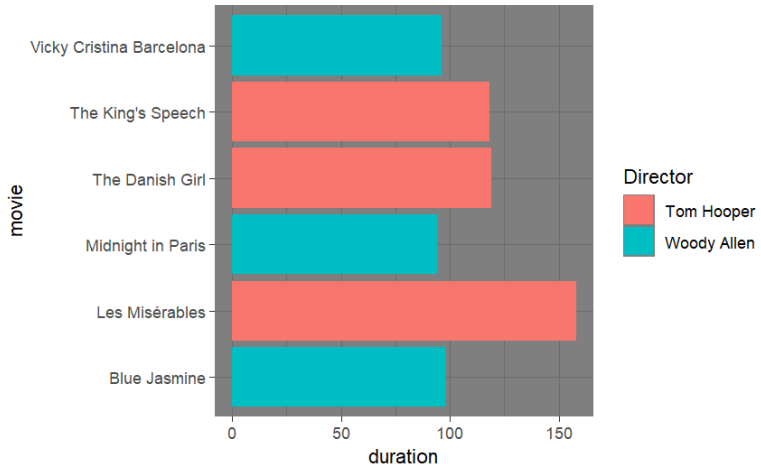

Figure 13 - Movies directed by Hooper and Allen, two directors whose movies won more Oscars than any other director.

SQL OLAP Extensions

First, a virtual table that contains data from all the tables is created to make accessing information within the fact and dimensions, easier.

```
CREATE OR REPLACE VIEW movie_view AS
SELECT *
FROM Movie_Fact, age_restriction, Movie_cast, country, Genre_Group,
    Rating, Release_Date, Title
WHERE age_key = age_res_ID
    AND cast_key = cast_ID
    AND Country_key = country_id
    AND genre_group_key = Genre_group_ID
    AND Rating_key = Rating_ID
    AND release_date_key = Date_ID
    AND title_key = Title_ID
```

This virtual table is called *movie_view.*

CUBE

How many awards the movies that are not produced by United States have won? In which language they are spoken?

```
SELECT country_name AS country, language, SUM(awards_won_number) AS awards,
    SUM(oscars_won_number) AS Oscars
FROM movie_view
```

```
WHERE country_name NOT IN ('United States')
GROUP BY CUBE (country_name, language)
```

	country character varying	language character varying	awards bigint	oscars bigint
1	[null]	[null]	456	54
2	Ireland	english	13	1
3	New Zealand	english	48	3
4	spain	Spanish	4	1
5	France	english	146	18
6	United Kingdom	english	225	28
7	France	french	10	1
8	spain	english	3	1

Figure 14 – Several rows of the results of the CUBE query

Between countries outside the United States, the most Oscars and awards belong to the United Kingdom, after that it is France. The interesting point is that between 26 movies that have been produced in countries with non-English languages, only 2 movies are spoken in a language other than English.

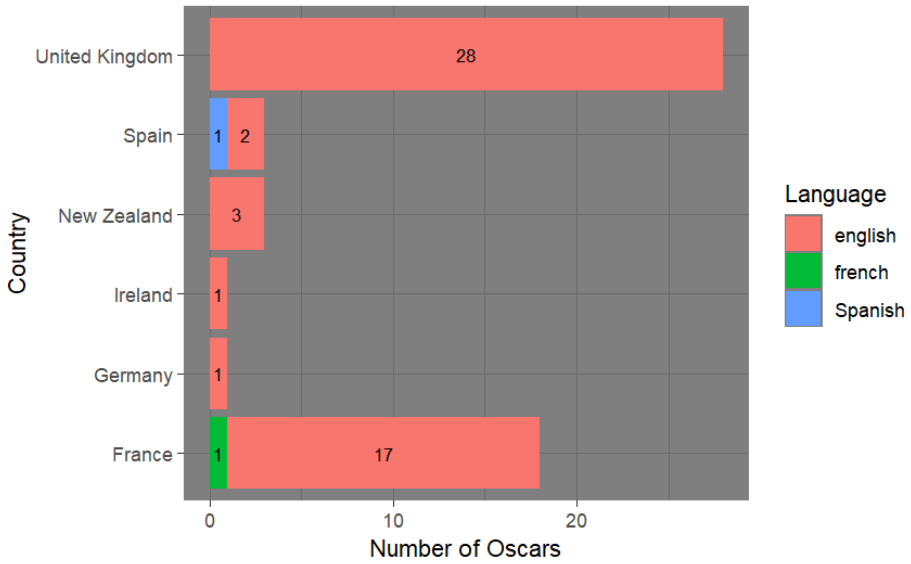

Figure 15 – Number of Oscar awards of movies that were produced by countries other than the United States.

ROLLUP

Which genres did appear the most in Oscar-winning movies in each year and in total?

```
SELECT genre_name, movie_view.year, COUNT(*) AS count,
    SUM(awards_won_number) AS awards
FROM movie_view
    JOIN genre_bridge ON genre_group_ID = genre_bridge_key
```

```
    JOIN genre ON genre_key = genre_id
GROUP BY ROLLUP(genre_name,movie_view.year)
ORDER BY year DESC, count DESC
```

Both the results of the query and the bar chart show that the favorite genres of Academy are drama and biography. Almost half of the movies that won an Oscar were drama movies (drama can be their first or second genre). Looking at the awards column, we can see that the same thing can be said about other types of awards too.

	genre_name character varying	year integer	count bigint	awards bigint
1	[null]	[null]	163	2721
2	[...] Drama	[null]	79	1337
3	Biography	[null]	26	386
4	[...] Comedy	[null]	15	290
33	[...] Drama	2012	7	132
34	[...] Comedy	2012	2	50
35	Biography	2012	2	44

Figure 16 - Results of the ROLLUP query (several rows from different part of results' table are put together). The count column shows the number of Oscars each genre has won.

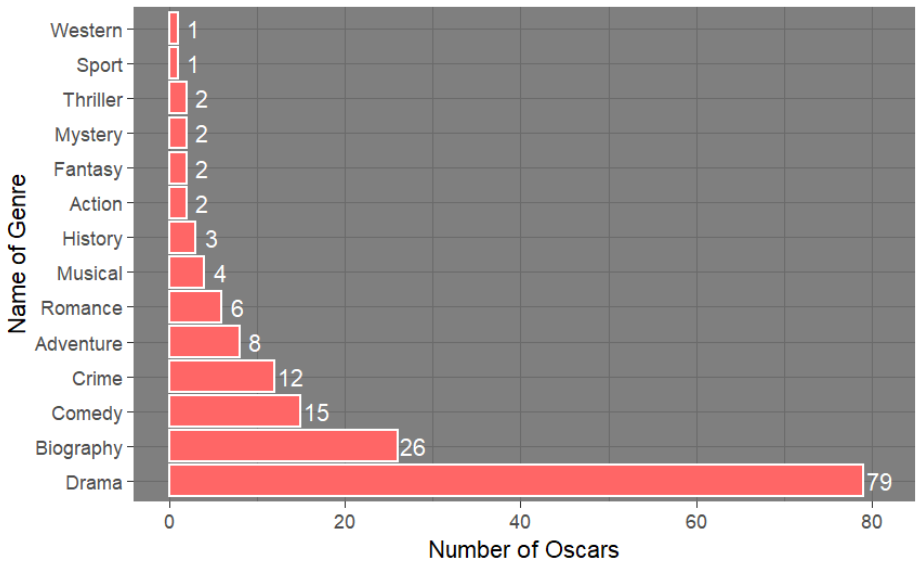

Figure 17 - Number of Oscar awards each genre has won between 2000 to 2016

GROUPING SETS

Movies with an IMDb score greater than 9 and/or a Metacritic score greater than 90 are considered to be excellent. How much profit did these movies make based on these two groups of attributes?

1. genre and age rating
2. country and language

```
SELECT age_rate,genre_group_name AS genres, country_name, language,
    trunc((SUM(boxoffice::numeric)/1000000),2) AS profit_in_million
FROM movie_view
WHERE (imdb_score >= 9 OR metascore >= 90)
GROUP BY GROUPING SETS ((age_rate,genre_group_name),
                        (country_name,language))
ORDER BY SUM(boxoffice) DESC
```

	age_rate character varying (6)	genres character varying	country_name character varying	language character varying	profit_in_million numeric
1	[null]	[null]	United States	english	1335.44
2	PG-13	Adventure_Drama	[null]	[null]	651.10
3	PG-13	Action_Crime	[null]	[null]	533.32
4	[null]	[null]	New Zealand	english	377.02

Figure 18 - Results of GROUPING SETS query

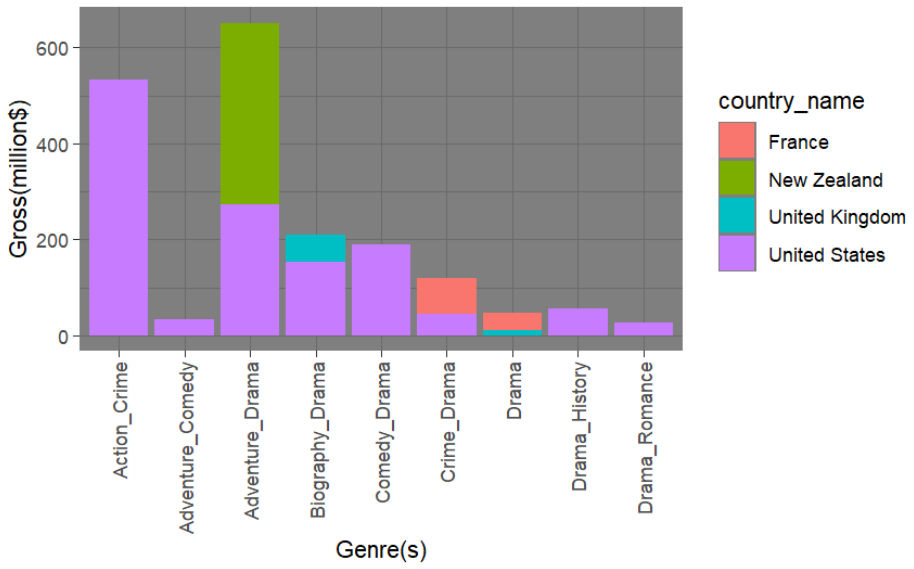

Figure 19 – "Excellent" movies' gross based on genre(s) and country

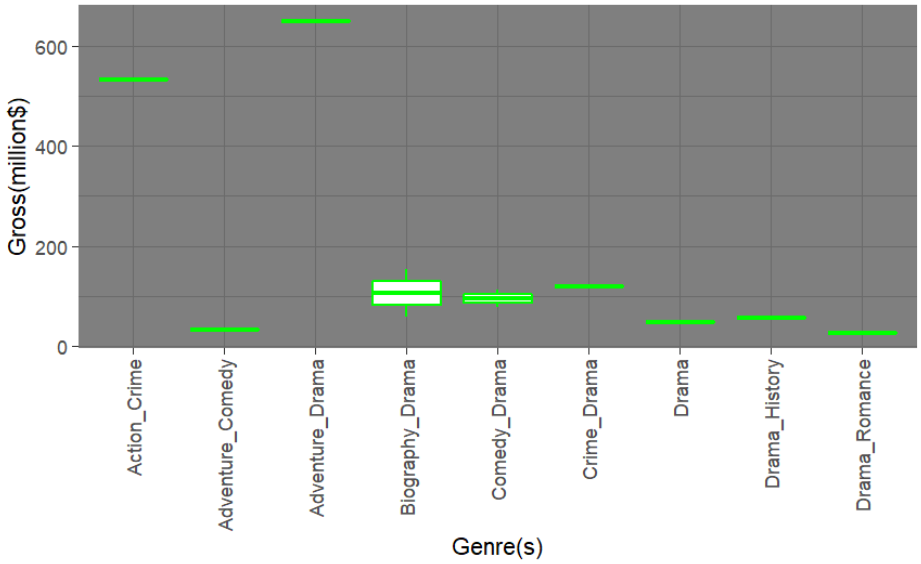

Figure 20 – Box plot of "excellent" movies' gross based on genre(s)

Window functions

Ranking query

The purpose of this query is assigning a rank and a dense rank to each combination of age rate and IMDb score based on the profit they have made.

```
SELECT
    concat(FLOOR(imdb_score), '-', FLOOR(imdb_score) + 1) AS IMDb_score,
    age_rate,
    SUM(boxoffice)/1000000 AS Gross_in_million,
    RANK() OVER ( PARTITION BY concat(FLOOR(imdb_score), '-', FLOOR(imdb_score) + 1)
            ORDER BY SUM(boxoffice)) AS rank_by_gross,
    DENSE_RANK() OVER ( PARTITION BY concat(FLOOR(imdb_score), '-', FLOOR(imdb_score)
+ 1)
            ORDER BY SUM(boxoffice)) AS DENSErank_by_gross
FROM movie_view
GROUP BY concat(FLOOR(imdb_score), '-', FLOOR(imdb_score) + 1),age_rate
```

	imdb_score text	age_rate character varying (6)	gross_in_million bigint	rank_by_gross bigint	denserank_by_gross bigint
1	6-7	PG-13	103	1	1
2	7-8	PG	124	1	1
3	7-8	R	1856	2	2
4	7-8	PG-13	1961	3	3
5	8-9	PG-13	1030	1	1
6	8-9	R	1683	2	2
7	9-10	PG-13	533	1	1

Figure 21 - Results of the ranking query

Since there are no two groups that have the same rank, the values of *rank_by_gross* and *denserank_by_gross* are the same.

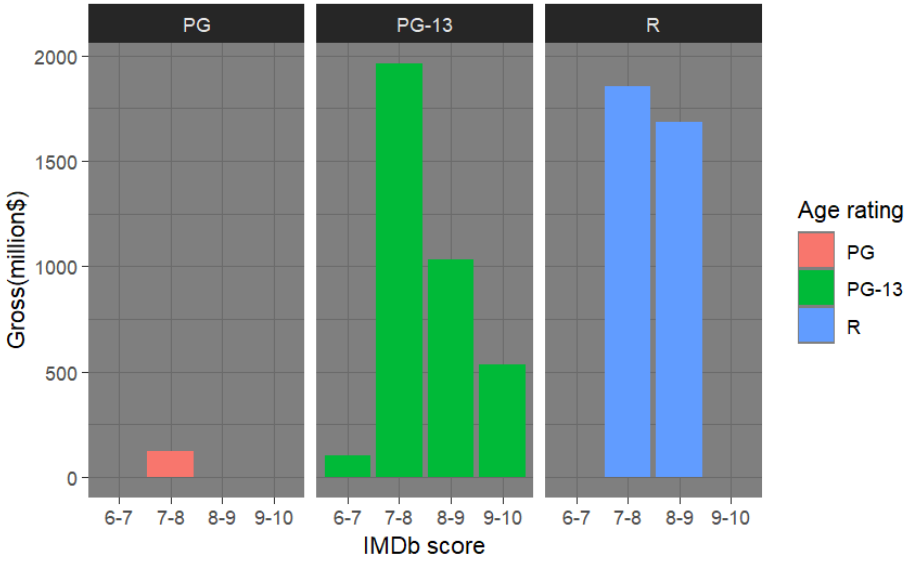

Figure 22 – Movies' profit based on IMDb score and age rating

Window query

We would like to know on average how many reviews have been written by users about Oscar-winning movies that were released from 2010 to 2016, based on the month of release. We also want to know the difference in the average number of votes between each row and the previous/next one.

```
SELECT
    to_char(to_timestamp (month::text, 'MM'), 'TMmonth') AS Release_Month,
    LAG(ROUND(AVG(imdb_num_user_review))) OVER(ORDER BY month) ,
    ROUND(AVG(imdb_num_user_review)) AS num_of_reviews,
    LEAD(ROUND(AVG(imdb_num_user_review))) OVER(ORDER BY month)
```

```
FROM movie_view
WHERE release_date > '12/31/2009'
AND release_date < '01/01/2017'
GROUP BY month
```

	release_month text	lag numeric	num_of_reviews numeric	lead numeric
1	january	[null]	642	130
2	february	642	130	335
3	june	130	335	572
4	august	335	572	894
5	october	572	894	581
6	november	894	581	603
7	december	581	603	[null]

Figure 23 – Results of LAG and LEAD query

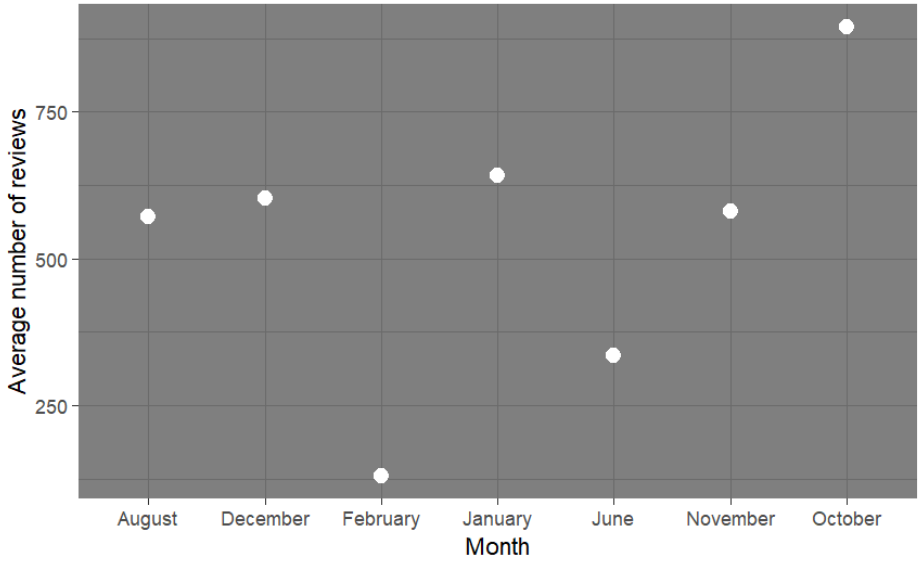

Figure 24 – Average number of reviews that have been written about 2006 movies

Period-to-period comparison

How have the average duration of movies that were released in first 3 months changed each year from 2000 to 2016?

```
SELECT
    year AS release_year,
    avg_duration, Pre_year_avg_duration,
    avg_duration - Pre_year_avg_duration AS Difference,
    trunc((avg_duration -
Pre_year_avg_duration)::numeric/Pre_year_avg_duration::numeric,2)
        AS Difference_per
```

```
FROM (
    SELECT year,ROUND(AVG(duration)) AS avg_duration,
        LAG(ROUND(AVG(duration))) OVER (ORDER BY year) AS Pre_year_avg_duration
    FROM release_date JOIN movie_fact
        ON date_ID = release_date_key JOIN rating
        ON rating_id = rating_key
    WHERE month IN (1,2,3)
    GROUP BY year
    ORDER BY year) AS vote_table
```

	release_year integer	avg_duration numeric	pre_year_avg_duration numeric	difference numeric	difference_per numeric
1	2000	131	[null]	[null]	[null]
2	2001	135	131	4	0.03
3	2002	118	135	-17	-0.12
4	2003	121	118	3	0.02
5	2004	109	121	-12	-0.09
6	2005	129	109	20	0.18
7	2006	124	129	-5	-0.03
8	2007	123	124	-1	0.00
9	2008	158	123	35	0.28

Figure 25 – Several rows of the result

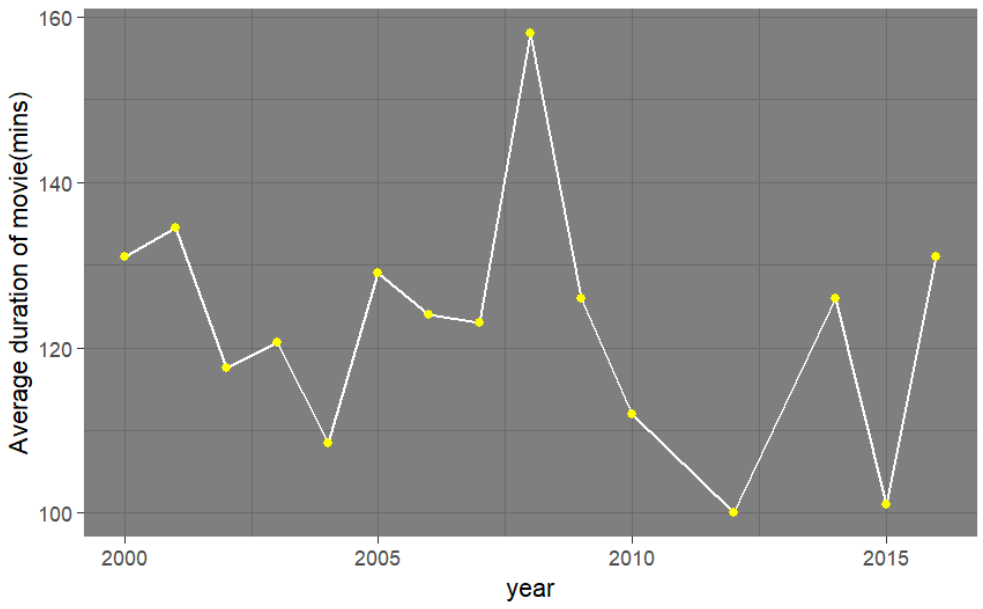

Figure 26 – Average duration of movies that have been released in each year's first 3 months

The average duration of movies fluctuates between 120 and 130 minutes, supporting the idea that longer movies have a better chance at winning Oscar awards.

Conclusion

The motivation of this project was discovering the similarities between movies that have won at least one Oscar award in the top five categories and to see if there is a formula for creating movies that have a high chance of winning.

Related data was gathered from various sources and pre-processing steps were done by Python programming language. Then a data warehouse was built, and queries were asked by SQL scripts in pgAdmin4. The visualizations related to queries were done by R language.

Studying the results tell us significant similarities between those movies, most of them are dramas, R rated, spoken in English, produced by the United States, and have a runtime of over two hours. We cannot be sure that the most famous award for movies is rigged but it is safe to say that the Academy has certain preferences.

References

1. Zelazko, Alicja. "Who Votes for the Academy Awards?". *Encyclopedia Britannica*, Invalid Date, link
2. Nathaniel “There's a formula to winning the Oscars, and it's all in the statistics”, 2020, INSIDER, link
3. Kaggle