Movies

Nombre Apellido1 Apellido2

Fecha

Introducción

En este proyecto se desarrolla en Python un análisis básico de datos sobre películas de cine de IMDB. El set de datos que vamos a usar inicialmente se encuentra en la siguiente página:

https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset (https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset)

En ella puede encontrarse información más detallada, así como una descripción precisa de cada columna.
Seguidamente, te toca a ti hacer una breve introducción, completando el fragmento de letra en azul y desarrollándolo a tu antojo. Suprime después este fragmento en verde.

Se plantean los apartados iniciales para su resolución *sin usar la libreria pandas*. Esto se propone así para forzar a practicar con las técnicas, herramientas y conceptos básicos de Python. Más adelante, hay otros apartados propuestos para su resolución con esta librería y otras.

A partir de los datos proporcionados, he conseguido ... pero no he podido ...

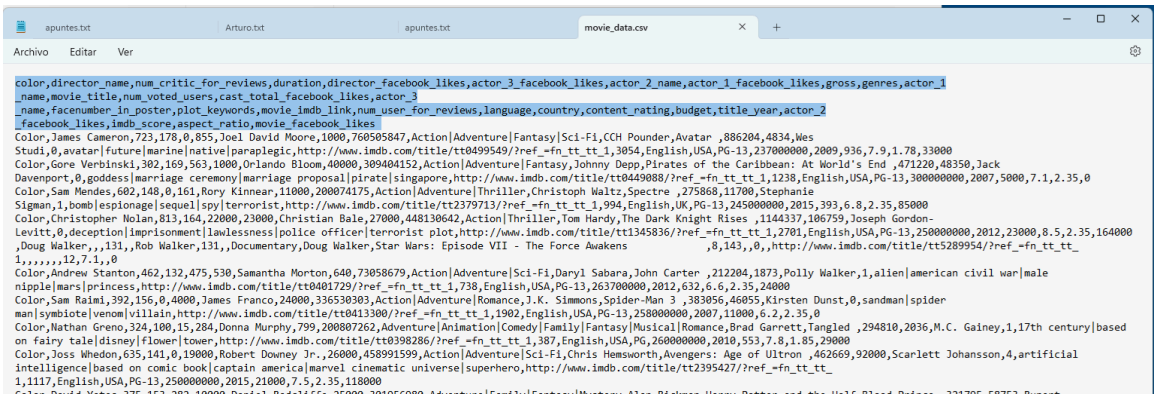
Aunque al final de este notebook detallaré la calificación que calculo honestamente, globalmente, siguiendo las puntuaciones que se asigna a cada apartado, diría que he obtenido una nota de *** sobre 10.

Completa tus datos personales en la cabecera, bajo el rótulo inicial. Completa también el breve apartado anterior, con texto azul, y pon en azul todos los comentarios tuyos, dejando en negro los míos, del enunciado. También, suprime los fragmentos en verde, como éste, que son indicaciones pero que, una vez atendidas, deben desaparecer de la solución que entregas.

Datos de partida

(**Nota previa:** hasta el apartado específico de dataframes, se deben desarrollar las soluciones a los ejercicios propuestos sin usar la librería `pandas`, justamente para obligar a practicar con conceptos básicos de Python.)

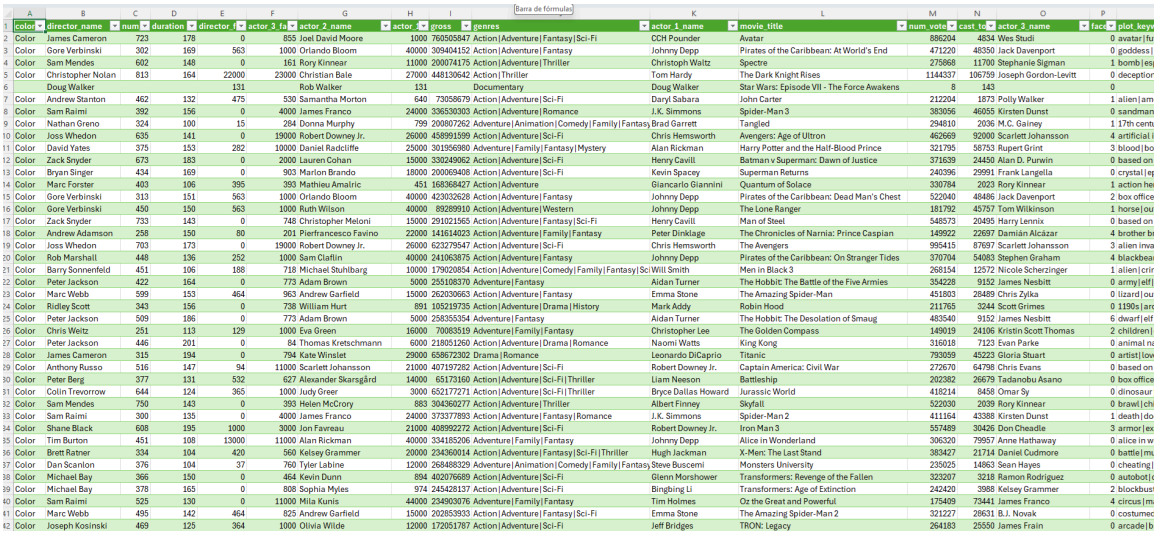
Nuestra tabla de datos es el archivo de texto `movie_data.csv` que se encuentra en la carpeta `data_in`, y puede verse así con cualquier editor:



```
color,director_name,num_critic_for_reviews,duration,director_facebook_likes,actor_3_facebook_likes,actor_2_name,actor_1_facebook_likes,gross,genres,actor_1
_name,movie_title,num_voted_users,cast_total_facebook_likes,actor_3
_name,facenumber_in_poster,plot_keywords,movie_imdb_link,num_user_for_reviews,language,country,content_rating,budget,title_year,actor_2
_facebook_likes,imdb_score,aspect_ratio,movie_facebook_likes
Color,James Cameron,723,178,0,855,Joel David Moore,1000,760505847,Action|Adventure|Fantasy|Sci-Fi,CCH Pounder,Avatar ,886204,4834,Wes
Studi,0,avatar|future|marine|native|paraplegic,http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1,3054,English,USA,PG-13,237000000,2009,936,7.9,1.78,33000
Color,Gore Verbinski,302,169,563,1000,Orlando Bloom,40000,309404152,Action|Adventure|Fantasy,Johnny Depp,Pirates of the Caribbean: At World's End ,471220,48350,Jack
Davenport,0,goddess|marriage ceremony|marriage proposal|pirate|singapore,http://www.imdb.com/title/tt0449088/?ref_=fn_tt_tt_1,1238,English,USA,PG-13,300000000,2007,5000,7.1,2.35,0
Color,Sam Mendes,602,148,0,161,Rory Kinnear,11000,200074175,Action|Adventure|Thriller,Christoph Waltz,Spectre ,275868,11700,Stephanie
Sigman,1,bomb|espionage|sequel|spy|terrorist,http://www.imdb.com/title/tt2379713/?ref_=fn_tt_tt_1,994,English,UK,PG-13,245000000,2015,393,6.8,2.35,85000
Color,Christopher Nolan,813,164,22000,23000,Christian Bale,27000,448130642,Action|Thriller,Tom Hardy,The Dark Knight Rises ,1144337,106759,Joseph Gordon-
Levitt,0,deception|imprisonment|lawlessness|police officer|terrorist plot,http://www.imdb.com/title/tt1345836/?ref_=fn_tt_tt_1,2701,English,USA,PG-13,250000000,2012,23000,8.5,2.35,164000
,Doug Walker,,,131,,Rob Walker,131,,Documentary,Doug Walker,Star Wars: Episode VII - The Force Awakens            ,8,143,,0,,http://www.imdb.com/title/tt5289954/?ref_=fn_tt_tt_
1,,,,,,,12,7.1,,0
Color,Andrew Stanton,462,132,475,530,Samantha Morton,640,73058679,Action|Adventure|Sci-Fi,Daryl Sabara,John Carter ,212204,1873,Polly Walker,1,alien|american civil war|male
nipple|mars|princess,http://www.imdb.com/title/tt0401729/?ref_=fn_tt_tt_1,738,English,USA,PG-13,263700000,2012,632,6.6,2.35,24000
Color,Sam Raimi,392,156,0,4000,James Franco,24000,336530303,Action|Adventure|Romance,J.K. Simmons,Spider-Man 3 ,383056,46055,Kirsten Dunst,0,sandman|spider
man|symbiote|venom|villain,http://www.imdb.com/title/tt0413300/?ref_=fn_tt_tt_1,902,English,USA,PG-13,258000000,2007,11000,6.2,2.35,0
Color,Nathan Greno,324,100,15,284,Donna Murphy,799,200807262,Adventure|Animation|Comedy|Family|Fantasy|Musical|Romance,Brad Garrett,Tangled ,294810,2036,M.C. Gainey,1,17th century|based
on fairy tale|disney|flower|tower,http://www.imdb.com/title/tt0398286/?ref_=fn_tt_tt_1,387,English,USA,PG,260000000,2010,553,7.8,1.85,29000
Color,Joss Whedon,635,141,0,19000,Robert Downey Jr.,26000,458991599,Action|Adventure|Sci-Fi,Chris Hemsworth,Avengers: Age of Ultron ,462669,92000,Scarlett Johansson,4,artificial
intelligence|based on comic book|captain america|marvel cinematic universe|superhero,http://www.imdb.com/title/tt2395427/?ref_=fn_tt_tt_
1,1117,English,USA,PG-13,250000000,2015,21000,7.5,2.35,118000
```

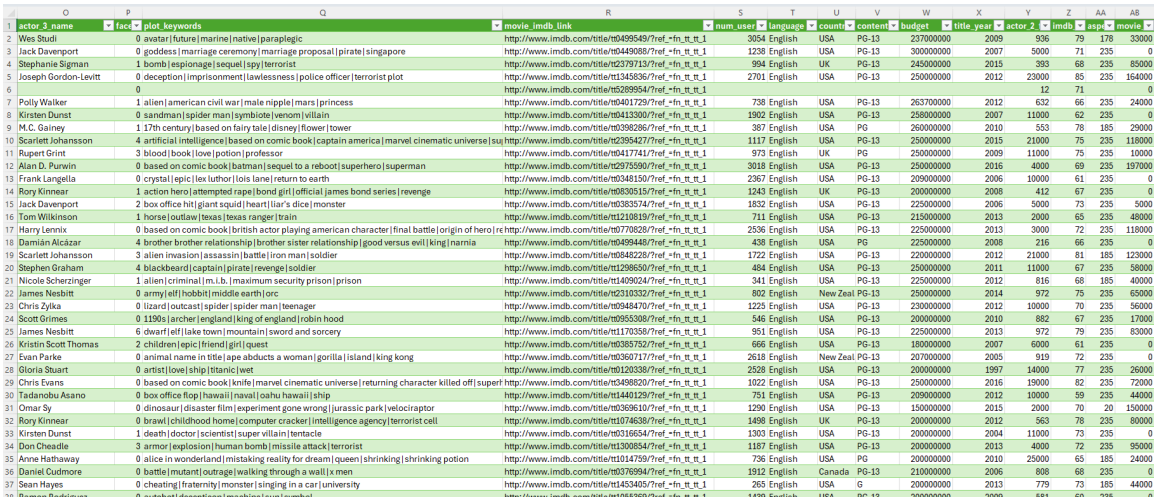
La primera fila es la cabecera. Esta fila cabecera contiene los nombres de los campos, separados por comas. Yo la he marcado en azul para distinguirla fácilmente de las demás filas, que contienen los datos propiamente dichos, esto es, los valores de dichos campos, consignando los datos de cada película en cada línea.

Si abrimos esta tabla con *excell* (importar datos csv con el separador `,`), vemos cada dato en una celda.



color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross	genres	actor_1_name	movie_title	num_voted_users	cast_total_facebook_likes	actor_3_name	facenumber_in_poster	plot_keywords
Color	James Cameron	723	178	0	855	Joel David Moore	1000	760505847	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204	4834	Wes Studi	0	avatar|fu
Color	Gore Verbinski	302	169	563	1000	Orlando Bloom	40000	309404152	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220	48350	Jack Davenport	0	goddess|
Color	Sam Mendes	602	148	0	161	Rory Kinnear	11000	200074175	Action|Adventure|Thriller	Christoph Waltz	Spectre	275868	11700	Stephanie Sigman	1	bomb|es
Color	Christopher Nolan	813	164	22000	23000	Christian Bale	27000	448130642	Action|Thriller	Tom Hardy	The Dark Knight Rises	1144337	106759	Joseph Gordon-Levitt	0	deceptior
	Doug Walker			131		Rob Walker	131		Documentary	Doug Walker	Star Wars: Episode VII - The Force Awakens	8	143		0	
Color	Andrew Stanton	462	132	475	530	Samantha Morton	640	73058679	Action|Adventure|Sci-Fi	Daryl Sabara	John Carter	212204	1873	Polly Walker	1	alien|am
Color	Sam Raimi	392	156	0	4000	James Franco	24000	336530303	Action|Adventure|Romance	J.K. Simmons	Spider-Man 3	383056	46055	Kirsten Dunst	0	sandman
Color	Nathan Greno	324	100	15	284	Donna Murphy	799	200807262	Adventure|Animation|Comedy|Family|Fantasy	Brad Garrett	Tangled	294810	2036	M.C. Gainey	1	17th cent
Color	Joss Whedon	635	141	0	19000	Robert Downey Jr.	26000	458991599	Action|Adventure|Sci-Fi	Chris Hemsworth	Avengers: Age of Ultron	462669	92000	Scarlett Johansson	4	artificial i
Color	David Yates	375	153	282	10000	Daniel Radcliffe	25000	301956980	Adventure|Family|Fantasy|Mystery	Alan Rickman	Harry Potter and the Half-Blood Prince	321795	58753	Rupert Grint	3	blood|bo
Color	Zack Snyder	673	183	0	2000	Lauren Cohan	15000	330249062	Action|Adventure|Sci-Fi	Henry Cavill	Batman v Superman: Dawn of Justice	371639	24450	Alan D. Purwin	0	based on
Color	Bryan Singer	434	169	0	903	Marlon Brando	18000	200069408	Action|Adventure|Sci-Fi	Kevin Spacey	Superman Returns	240396	29991	Frank Langella	0	crystal|ep
Color	Marc Forster	403	106	395	393	Mathieu Amalric	451	168368427	Action|Adventure	Giancarlo Giannini	Quantum of Solace	330784	2023	Rory Kinnear	1	action he
Color	Gore Verbinski	313	151	563	1000	Orlando Bloom	40000	423032628	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: Dead Man's Chest	522040	48486	Jack Davenport	2	box office
Color	Gore Verbinski	450	150	563	1000	Ruth Wilson	40000	89289910	Action|Adventure|Western	Johnny Depp	The Lone Ranger	181792	45757	Tom Wilkinson	1	horse|ou
Color	Zack Snyder	733	143	0	748	Christopher Meloni	15000	291021565	Action|Adventure|Fantasy|Sci-Fi	Henry Cavill	Man of Steel	548573	20495	Harry Lennix	0	based on
Color	Andrew Adamson	258	150	80	201	Pierfrancesco Favino	22000	141614023	Action|Adventure|Family|Fantasy	Peter Dinklage	The Chronicles of Narnia: Prince Caspian	149922	22697	Damián Alcázar	4	brother b
Color	Joss Whedon	703	173	0	19000	Robert Downey Jr.	26000	623279547	Action|Adventure|Sci-Fi	Chris Hemsworth	The Avengers	995415	87697	Scarlett Johansson	3	alien inva
Color	Rob Marshall	448	136	252	1000	Sam Claflin	40000	241063875	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: On Stranger Tides	370704	54083	Stephen Graham	4	blackbea
Color	Barry Sonnenfeld	451	106	188	718	Michael Stuhlbarg	10000	179020854	Action|Adventure|Comedy|Family|Fantasy|Sci	Will Smith	Men in Black 3	268154	12572	Nicole Scherzinger	1	alien|cri
Color	Peter Jackson	422	164	0	773	Adam Brown	5000	255108370	Action|Adventure|Fantasy	Aidan Turner	The Hobbit: The Battle of the Five Armies	354228	9152	James Nesbitt	0	army|elf|
Color	Marc Webb	599	153	464	963	Andrew Garfield	15000	262030663	Action|Adventure|Fantasy	Emma Stone	The Amazing Spider-Man	451803	28489	Chris Zylka	0	lizard|ou
Color	Ridley Scott	343	156	0	738	William Hurt	891	105219735	Action|Adventure|Drama|History	Mark Addy	Robin Hood	211765	3244	Scott Grimes	0	1190s|ar
Color	Peter Jackson	509	186	0	773	Adam Brown	5000	258355354	Action|Adventure|Fantasy	Aidan Turner	The Hobbit: The Desolation of Smaug	483540	9152	James Nesbitt	6	dwarf|elf
Color	Chris Weitz	251	113	129	1000	Eva Green	16000	70083519	Adventure|Family|Fantasy	Christopher Lee	The Golden Compass	149019	24106	Kristin Scott Thomas	2	children|
Color	Peter Jackson	446	201	0	84	Thomas Kretschmann	6000	218051260	Action|Adventure|Drama|Romance	Naomi Watts	King Kong	316018	7123	Evan Parke	0	animal na
Color	James Cameron	315	194	0	794	Kate Winslet	29000	658672302	Drama|Romance	Leonardo DiCaprio	Titanic	793059	45223	Gloria Stuart	0	artist|lov
Color	Anthony Russo	516	147	94	11000	Scarlett Johansson	21000	407197282	Action|Adventure|Sci-Fi	Robert Downey Jr.	Captain America: Civil War	272670	64798	Chris Evans	0	based on
Color	Peter Berg	377	131	532	627	Alexander Skarsgård	14000	65173160	Action|Adventure|Sci-Fi|Thriller	Liam Neeson	Battleship	292382	26679	Tadanobu Asano	0	box office
Color	Colin Trevorrow	644	124	365	1000	Judy Greer	3000	652177271	Action|Adventure|Sci-Fi|Thriller	Bryce Dallas Howard	Jurassic World	418214	8458	Omar Sy	0	dinosaur
Color	Sam Mendes	750	143	0	393	Helen McCrory	883	304360277	Action|Adventure|Thriller	Albert Finney	Skyfall	522030	2039	Rory Kinnear	0	brawl|ch
Color	Sam Raimi	300	135	0	4000	James Franco	24000	373377893	Action|Adventure|Fantasy|Romance	J.K. Simmons	Spider-Man 2	411164	43388	Kirsten Dunst	1	death|do
Color	Shane Black	608	195	1000	3000	Jon Favreau	21000	408992272	Action|Adventure|Sci-Fi	Robert Downey Jr.	Iron Man 3	557489	30426	Don Cheadle	3	armor|ex
Color	Tim Burton	451	108	13000	11000	Alan Rickman	40000	334185206	Action|Family|Fantasy	Johnny Depp	Alice in Wonderland	306320	79957	Anne Hathaway	0	alice in w
Color	Brett Ratner	334	104	420	560	Kelsey Grammer	20000	234360014	Action|Adventure|Fantasy|Sci-Fi|Thriller	Hugh Jackman	X-Men: The Last Stand	383427	21714	Daniel Cudmore	0	battle|mu
Color	Dan Scanlon	376	104	37	760	Tyler Labine	12000	268488329	Adventure|Animation|Comedy|Family|Fantasy	Steve Buscemi	Monsters University	235025	14863	Sean Hayes	0	cheating|
Color	Michael Bay	366	150	0	464	Kevin Dunn	894	402076689	Action|Adventure|Sci-Fi	Glenn Morshower	Transformers: Revenge of the Fallen	323207	3218	Ramon Rodriguez	0	autoboot|d
Color	Michael Bay	378	165	0	808	Sophia Myles	974	245428137	Action|Adventure|Sci-Fi	Bingbing Li	Transformers: Age of Extinction	242420	3988	Kelsey Grammer	2	blockbust
Color	Sam Raimi	525	130	0	11000	Mila Kunis	44000	234903076	Adventure|Family|Fantasy	Tim Holmes	Oz the Great and Powerful	175409	73441	James Franco	4	circus|m
Color	Marc Webb	495	142	464	825	Andrew Garfield	15000	202853933	Action|Adventure|Fantasy|Sci-Fi	Emma Stone	The Amazing Spider-Man 2	321227	28631	B.J. Novak	0	costumec
Color	Joseph Kosinski	469	125	364	1000	Olivia Wilde	12000	172051787	Action|Adventure|Sci-Fi	Jeff Bridges	TRON: Legacy	264183	25550	James Frain	0	arcade|b

Mostramos la hoja excell en dos imágenes por ser muy ancha.



actor_3_name	face	plot_keywords	movie_imdb_link	num_user	language	countr	content	budget	title_year	actor_2	imdb	aspe	movie
Wes Studi	0	avatar|future|marine|native|paraplegic	http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1	3054	English	USA	PG-13	237000000	2009	936	79	178	33000
Jack Davenport	0	goddess|marriage ceremony|marriage proposal|pirate|singapore	http://www.imdb.com/title/tt0449088/?ref_=fn_tt_tt_1	1238	English	USA	PG-13	300000000	2007	5000	71	235	0
Stephanie Sigman	1	bomb|espionage|sequel|spy|terrorist	http://www.imdb.com/title/tt2379713/?ref_=fn_tt_tt_1	994	English	UK	PG-13	245000000	2015	393	68	235	85000
Joseph Gordon-Levitt	0	deception|imprisonment|lawlessness|police officer|terrorist plot	http://www.imdb.com/title/tt1345836/?ref_=fn_tt_tt_1	2701	English	USA	PG-13	250000000	2012	23000	85	235	164000
	0		http://www.imdb.com/title/tt5289954/?ref_=fn_tt_tt_1							12	71		0
Polly Walker	1	alien|american civil war|male nipple|mars|princess	http://www.imdb.com/title/tt0401729/?ref_=fn_tt_tt_1	738	English	USA	PG-13	263700000	2012	632	66	235	24000
Kirsten Dunst	0	sandman|spider man|symbiote|venom|villain	http://www.imdb.com/title/tt0413300/?ref_=fn_tt_tt_1	1902	English	USA	PG-13	258000000	2007	11000	62	235	0
M.C. Gainey	1	17th century|based on fairy tale|disney|flower|tower	http://www.imdb.com/title/tt0398286/?ref_=fn_tt_tt_1	387	English	USA	PG	260000000	2010	553	78	185	29000
Scarlett Johansson	4	artificial intelligence|based on comic book|captain america|marvel cinematic universe|su	http://www.imdb.com/title/tt2395427/?ref_=fn_tt_tt_1	1117	English	USA	PG-13	250000000	2015	21000	75	235	118000
Rupert Grint	3	blood|book|love|potion|professor	http://www.imdb.com/title/tt0417741/?ref_=fn_tt_tt_1	973	English	UK	PG	250000000	2009	11000	75	235	10000
Alan D. Purwin	0	based on comic book|batman|sequel to a reboot|superhero|superman	http://www.imdb.com/title/tt2975590/?ref_=fn_tt_tt_1	3018	English	USA	PG-13	250000000	2016	4000	69	235	197000
Frank Langella	0	crystal|epic|lex luthor|lois lane|return to earth	http://www.imdb.com/title/tt0348150/?ref_=fn_tt_tt_1	2367	English	USA	PG-13	209000000	2006	10000	61	235	0
Rory Kinnear	1	action hero|attempted rape|bond girl|official james bond series|revenge	http://www.imdb.com/title/tt0830515/?ref_=fn_tt_tt_1	1243	English	UK	PG-13	200000000	2008	412	67	235	0
Jack Davenport	2	box office hit|giant squid|heart|liar's dice|monster	http://www.imdb.com/title/tt0383574/?ref_=fn_tt_tt_1	1832	English	USA	PG-13	225000000	2006	5000	73	235	5000
Tom Wilkinson	1	horse|outlaw|texas|texas ranger|train	http://www.imdb.com/title/tt1210819/?ref_=fn_tt_tt_1	711	English	USA	PG-13	215000000	2013	2000	65	235	48000
Harry Lennix	0	based on comic book|british actor playing american character|final battle|origin of hero|re	http://www.imdb.com/title/tt0770828/?ref_=fn_tt_tt_1	2536	English	USA	PG-13	225000000	2013	3000	72	235	118000
Damián Alcázar	4	brother brother relationship|brother sister relationship|good versus evil|king|narnia	http://www.imdb.com/title/tt0499448/?ref_=fn_tt_tt_1	438	English	USA	PG	225000000	2008	216	66	235	0
Scarlett Johansson	3	alien invasion|assassin|battle|iron man|soldier	http://www.imdb.com/title/tt0848228/?ref_=fn_tt_tt_1	1722	English	USA	PG-13	220000000	2012	21000	81	185	123000
Stephen Graham	4	blackbeard|captain|pirate|revenge|soldier	http://www.imdb.com/title/tt1298650/?ref_=fn_tt_tt_1	484	English	USA	PG-13	250000000	2011	11000	67	235	58000
Nicole Scherzinger	1	alien|criminal|m.i.b.|maximum security prison|prison	http://www.imdb.com/title/tt1409024/?ref_=fn_tt_tt_1	341	English	USA	PG-13	225000000	2012	816	68	185	40000
James Nesbitt	0	army|elf|hobbit|middle earth|orc	http://www.imdb.com/title/tt2310332/?ref_=fn_tt_tt_1	802	English	New Zeal	PG-13	250000000	2014	972	75	235	65000
Chris Zylka	0	lizard|outcast|spider|spider man|teenager	http://www.imdb.com/title/tt0948470/?ref_=fn_tt_tt_1	1225	English	USA	PG-13	230000000	2012	10000	70	235	56000
Scott Grimes	0	1190s|archer|england|king of england|robin hood	http://www.imdb.com/title/tt0955308/?ref_=fn_tt_tt_1	546	English	USA	PG-13	200000000	2010	882	67	235	17000
James Nesbitt	6	dwarf|elf|lake town|mountain|sword and sorcery	http://www.imdb.com/title/tt1170358/?ref_=fn_tt_tt_1	951	English	USA	PG-13	225000000	2013	972	79	235	83000
Kristin Scott Thomas	2	children|epic|friend|girl|quest	http://www.imdb.com/title/tt0385752/?ref_=fn_tt_tt_1	666	English	USA	PG-13	180000000	2007	6000	61	235	0
Evan Parke	0	animal name in title|ape abducts a woman|gorilla|island|king kong	http://www.imdb.com/title/tt0360717/?ref_=fn_tt_tt_1	2618	English	New Zeal	PG-13	207000000	2005	919	72	235	0
Gloria Stuart	0	artist|love|ship|titanic|wet	http://www.imdb.com/title/tt0120338/?ref_=fn_tt_tt_1	2528	English	USA	PG-13	200000000	1997	14000	77	235	26000
Chris Evans	0	based on comic book|knife|marvel cinematic universe|returning character killed off|superh	http://www.imdb.com/title/tt3498820/?ref_=fn_tt_tt_1	1022	English	USA	PG-13	250000000	2016	19000	82	235	72000
Tadanobu Asano	0	box office flop|hawaii|naval|oahu hawaii|ship	http://www.imdb.com/title/tt1440129/?ref_=fn_tt_tt_1	751	English	USA	PG-13	209000000	2012	10000	59	235	44000
Omar Sy	0	dinosaur|disaster film|experiment gone wrong|jurassic park|velociraptor	http://www.imdb.com/title/tt0369610/?ref_=fn_tt_tt_1	1290	English	USA	PG-13	150000000	2015	2000	70	20	150000
Rory Kinnear	0	brawl|childhood home|computer cracker|intelligence agency|terrorist cell	http://www.imdb.com/title/tt1074638/?ref_=fn_tt_tt_1	1498	English	UK	PG-13	200000000	2012	563	78	235	80000
Kirsten Dunst	1	death|doctor|scientist|super villain|tentacle	http://www.imdb.com/title/tt0316654/?ref_=fn_tt_tt_1	1303	English	USA	PG-13	200000000	2004	11000	73	235	0
Don Cheadle	3	armor|explosion|human bomb|missile attack|terrorist	http://www.imdb.com/title/tt1300854/?ref_=fn_tt_tt_1	1187	English	USA	PG-13	200000000	2013	4000	72	235	95000
Anne Hathaway	0	alice in wonderland|mistaking reality for dream|queen|shrinking|shrinking potion	http://www.imdb.com/title/tt1014759/?ref_=fn_tt_tt_1	736	English	USA	PG	200000000	2010	25000	65	185	24000
Daniel Cudmore	0	battle|mutant|outrage|walking through a wall|x men	http://www.imdb.com/title/tt0376994/?ref_=fn_tt_tt_1	1912	English	Canada	PG-13	210000000	2006	808	68	235	0
Sean Hayes	0	cheating|fraternity|monster|singing in a car|university	http://www.imdb.com/title/tt1453405/?ref_=fn_tt_tt_1	265	English	USA	G	200000000	2013	779	73	185	44000

Librerías y constantes globales

Pongamos todas las librerías necesarias al principio, tal como propone el estilo `pep-8`. Ej.: PEP 8 -- Style Guide for Python Code (https://www.python.org/dev/peps/pep-0008/).

De paso, en éste y otros lugares de Internet podrás encontrar detalles sobre estilo y presentación de código con un estilo estándar y cuidado. Examina esta página y trata de seguir esas indicaciones.

In [1]:
```
# Librerías:

# Esta celda debe ser completada por el estudiante
```

In [2]:
```
# Constantes globales:

# Esta celda debe ser completada por el estudiante
```

Parte A. Ejercicios básicos sin usar `pandas` [2 puntos]

Esta parte inicial debe realizarse sin usar la librería `pandas`. Para practicar con esta librería, se plantean otros apartados más abajo.

A.1. Exploración inicial básica del archivo de datos

Deseamos cargar el archivo de datos, que tiene un formato `csv`. En este apartado, te recomiendo fuertemente usar la librería `cvs`, que deberás importar en la primera celda del script, más arriba, no aquí. (En los siguientes apartados, ya no mencionaré qué librerías usar ni recordaré dónde se han de importar.)

Observa también que el test de funcionamiento te da el nombre de la función que deber definir y algún otro identificador como es, por ejemplo, la constante `MOVIES_DATA`, que debes definir también más arriba, en la segunda celda de este script.

Finalmente, observando el test, verás que se carga por separado la cabecera y las filas de datos.

In [3]:
```
# Esta celda debe ser completada por el estudiante
```

In [4]:
```
# Test de funcionamiento

full_header, full_list_data = load_full_data(MOVIES_DATA)

print(full_header)
print()
print(full_list_data[0:5])
```

```
['color', 'director_name', 'num_critic_for_reviews', 'duration', 'direc
tor_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name', 'actor_1
_facebook_likes', 'gross', 'genres', 'actor_1_name', 'movie_title', 'nu
m_voted_users', 'cast_total_facebook_likes', 'actor_3_name', 'facenumbe
r_in_poster', 'plot_keywords', 'movie_imdb_link', 'num_user_for_review
s', 'language', 'country', 'content_rating', 'budget', 'title_year', 'a
ctor_2_facebook_likes', 'imdb_score', 'aspect_ratio', 'movie_facebook_l
ikes']

[['Color', 'James Cameron', '723', '178', '0', '855', 'Joel David Moor
e', '1000', '760505847', 'Action|Adventure|Fantasy|Sci-Fi', 'CCH Pounde
r', 'Avatar\xa0', '886204', '4834', 'Wes Studi', '0', 'avatar|future|ma
rine|native|paraplegic', 'http://www.imdb.com/title/tt0499549/?ref_=fn_
tt_tt_1', '3054', 'English', 'USA', 'PG-13', '237000000', '2009', '93
6', '7.9', '1.78', '33000'], ['Color', 'Gore Verbinski', '302', '169',
'563', '1000', 'Orlando Bloom', '40000', '309404152', 'Action|Adventure
|Fantasy', 'Johnny Depp', "Pirates of the Caribbean: At World's End\xa
0", '471220', '48350', 'Jack Davenport', '0', 'goddess|marriage ceremon
y|marriage proposal|pirate|singapore', 'http://www.imdb.com/title/tt044
9088/?ref_=fn_tt_tt_1', '1238', 'English', 'USA', 'PG-13', '300000000',
'2007', '5000', '7.1', '2.35', '0'], ['Color', 'Sam Mendes', '602', '14
8', '0', '161', 'Rory Kinnear', '11000', '200074175', 'Action|Adventure
|Thriller', 'Christoph Waltz', 'Spectre\xa0', '275868', '11700', 'Steph
anie Sigman', '1', 'bomb|espionage|sequel|spy|terrorist', 'http://www.i
mdb.com/title/tt2379713/?ref_=fn_tt_tt_1', '994', 'English', 'UK', 'PG-
13', '245000000', '2015', '393', '6.8', '2.35', '85000'], ['Color', 'Ch
ristopher Nolan', '813', '164', '22000', '23000', 'Christian Bale', '27
000', '448130642', 'Action|Thriller', 'Tom Hardy', 'The Dark Knight Ris
es\xa0', '1144337', '106759', 'Joseph Gordon-Levitt', '0', 'deception|i
mprisonment|lawlessness|police officer|terrorist plot', 'http://www.imd
b.com/title/tt1345836/?ref_=fn_tt_tt_1', '2701', 'English', 'USA', 'PG-
13', '250000000', '2012', '23000', '8.5', '2.35', '164000'], ['', 'Doug
Walker', '', '', '131', '', 'Rob Walker', '131', '', 'Documentary', 'Do
ug Walker', 'Star Wars: Episode VII - The Force Awakens\xa0
', '8', '143', '', '0', '', 'http://www.imdb.com/title/tt5289954/?ref_=
fn_tt_tt_1', '', '', '', '', '', '', '12', '7.1', '', '0']]
```

Ahora, querríamos ver las posiciones de los identificadores de los campos, dados en la línea cabecera.

In [5]:
```
# Esta celda debe ser completada por el estudiante
```

In [6]:
```
# Test de funcionamiento

print(list_of_enumerated_headers)
```

```
[(0, 'color'), (1, 'director_name'), (2, 'num_critic_for_reviews'), (3,
'duration'), (4, 'director_facebook_likes'), (5, 'actor_3_facebook_like
s'), (6, 'actor_2_name'), (7, 'actor_1_facebook_likes'), (8, 'gross'),
(9, 'genres'), (10, 'actor_1_name'), (11, 'movie_title'), (12, 'num_vot
ed_users'), (13, 'cast_total_facebook_likes'), (14, 'actor_3_name'), (1
5, 'facenumber_in_poster'), (16, 'plot_keywords'), (17, 'movie_imdb_lin
k'), (18, 'num_user_for_reviews'), (19, 'language'), (20, 'country'),
(21, 'content_rating'), (22, 'budget'), (23, 'title_year'), (24, 'actor
_2_facebook_likes'), (25, 'imdb_score'), (26, 'aspect_ratio'), (27, 'mo
vie_facebook_likes')]
```

A.2. Campos principales de una película

Los campos (columnas) del archivo son demasiados. No nos interesan todos ellos. Dada una lista con todos los campos, se pide extraer otra lista sólo con los campos con los que vamos a trabajar en los siguientes apartados: `movie_title`, `title_year`, `director_name`, `actor_1_name`, `language`, `country`, `color`, `budget`, `imdb_score` y `movie_imdb_link`.

In [7]:
```
# Esta celda debe ser completada por el estudiante
```

In [8]:
```
# Test de funcionamiento

print(main_data_from_item(full_header))

print()

datos_avatar_2009 = main_data_from_item(full_list_data[0])
print(datos_avatar_2009)

print()

datos_star_wars_7 = main_data_from_item(full_list_data[4])
print(datos_star_wars_7)
```

```
['movie_title', 'title_year', 'director_name', 'actor_1_name', 'languag
e', 'country', 'color', 'budget', 'imdb_score', 'movie_imdb_link']

['Avatar\xa0', '2009', 'James Cameron', 'CCH Pounder', 'English', 'US
A', 'Color', '237000000', '7.9', 'http://www.imdb.com/title/tt0499549/?
ref_=fn_tt_tt_1']

['Star Wars: Episode VII - The Force Awakens\xa0            ', '', 'Dou
g Walker', 'Doug Walker', '', '', '', '', '7.1', 'http://www.imdb.com/t
itle/tt5289954/?ref_=fn_tt_tt_1']
```

A.3. Algunos ajustes en los campos

Observa la anomalía en el string del título de la película. También, queremos tratar algunos campos como numéricos, ya sea enteros (el año y el presupuesto) o reales (la valoración). (Algunos campos numéricos enteros están vacíos en el archivo de datos; para ellos, el valor imputado será `-1`.) Además, en las urls de las películas no necesitaremos el fragmento final, iniciado con `?ref_`.

In [9]:
```
# Esta celda debe ser completada por el estudiante
```

In [10]:
```
# Test de funcionamiento

print(datatypes_arranged(datos_avatar_2009))

print(datatypes_arranged(datos_star_wars_7))
```

```
['Avatar', 2009, 'James Cameron', 'CCH Pounder', 'English', 'USA', 'Col
or', 237000000, 7.9, 'http://www.imdb.com/title/tt0499549/']
['Star Wars: Episode VII - The Force Awakens', -1, 'Doug Walker', 'Doug
Walker', '', '', '', -1, 7.1, 'http://www.imdb.com/title/tt5289954/']
```

A.4. Recuperación de alguna información básica

Diseña funciones para recuperar la siguiente información:

- El conjunto de valores posibles del campo `Color`.
- Los títulos de película de nuestro archivo (limpios de caracteres extraños), junto con la cantidad de calificadores (un entero), que contengan la subcadena "Victor".

In [11]:
```
# Esta celda debe ser completada por el estudiante
```

In [12]:
```
# Test de funcionamiento

print(colores)
print(pelis_victor)
```

```
{'', 'Color', ' Black and White'}
[('Victor Frankenstein', 159), ('Victor Frankenstein', 159), ('The Youn
g Victoria', 188), ('Victor Frankenstein', 159), ('Raising Victor Varga
s', 59)]
```

B Datos en un diccionario [2 puntos]

B.1. Carga únicamente de datos que vamos a usar

En lugar de cargar *todos* los datos del archivo, luego generar otro archivo con los datos que nos interesan y luego ajustarlos, es mejor diseñar una función que recorre el archivo de datos y, de cada línea del archivo, *únicamente* retiene la parte que nos interesa, cargando la cabecera por un lado y, por otro, la lista de datos de interés, ajustados como ya hemos dicho.

Cargaremos esta información en un diccionario, cuyas claves serán pares `(título, año)` y cuyos valores se recogen en una lista con el resto de los campos.

Además, los datos de cada película tras la clave `(título, año)` están repetidas en el archivo de datos, con todos los datos idénticos; y lo que es peor: alguna que está repetida

In [13]:
```
# Esta celda debe ser completada por el estudiante
```

In [14]:
```
# Test de funcionamiento

main_header, main_dict_data = load_main_data(MOVIES_DATA)

print(main_header)
print()
for title_year, pieces in list(main_dict_data.items())[:5]:
    print(title_year, " -> ", pieces)
```

```
['movie_title', 'title_year', 'director_name', 'actor_1_name', 'languag
e', 'country', 'color', 'budget', 'imdb_score', 'movie_imdb_link']

('Avatar', 2009)  ->  ['James Cameron', 'CCH Pounder', 'English', 'US
A', 'Color', 237000000, 7.9, 'http://www.imdb.com/title/tt0499549/']
("Pirates of the Caribbean: At World's End", 2007)  ->  ['Gore Verbinsk
i', 'Johnny Depp', 'English', 'USA', 'Color', 300000000, 7.1, 'http://w
ww.imdb.com/title/tt0449088/']
('Spectre', 2015)  ->  ['Sam Mendes', 'Christoph Waltz', 'English', 'U
K', 'Color', 245000000, 6.8, 'http://www.imdb.com/title/tt2379713/']
('The Dark Knight Rises', 2012)  ->  ['Christopher Nolan', 'Tom Hardy',
'English', 'USA', 'Color', 250000000, 8.5, 'http://www.imdb.com/title/t
t1345836/']
('Star Wars: Episode VII - The Force Awakens', -1)  ->  ['Doug Walker',
'Doug Walker', '', '', '', -1, 7.1, 'http://www.imdb.com/title/tt528995
4/']
```

Y ahora podemos suprimir la variable `full_list_data`

In [15]:
```
# Esta celda debe ser completada por el estudiante
```

In [16]:
```
# Test de funcionamiento

try:
    print(full_list_data)
except:
    print('La variable full_list_data está suprimida correctamente')
```

```
La variable full_list_data está suprimida correctamente
```

B.2. Recuperación de alguna información

Diseña funciones para averiguar la siguiente información:

- ¿Qué títulos de películas han sido dirigidas por "James Cameron" (o por el director que se desee)?

In [17]:
```
# Esta celda debe ser completada por el estudiante
```

In [18]:
```
# Test de funcionamiento

movies_anno_for_director(main_dict_data, "James Cameron")
```

Out[18]:
```
[('Avatar', 2009),
 ('Titanic', 1997),
 ('Terminator 2: Judgment Day', 1991),
 ('True Lies', 1994),
 ('The Abyss', 1989),
 ('Aliens', 1986),
 ('The Terminator', 1984)]
```

- ¿Qué directores han dirigido el número máximo de películas?

In [19]:
```
# Esta celda debe ser completada por el estudiante
```

In [20]:
```
# Test de funcionamiento

print(directors_max_movies(main_dict_data))
```

```
(['Steven Spielberg'], 26)
```

- Para cada año de un intervalo dado de años, ¿cuántas películas se han realizado? Esta información debe recuperrase en un diccionario convencional (no por defecto), y luego debe mostrarse en una lista con los años en orden ascendente..

In [21]:
```
# Esta celda debe ser completada por el estudiante
```

In [22]:
```
# Test de funcionamiento

num_movies = years_num_movies(main_dict_data, 2000, 2015)

print(num_movies)
```

```
{2009: 253, 2007: 198, 2015: 211, 2012: 214, 2010: 225, 2006: 235, 200
8: 223, 2013: 231, 2011: 224, 2014: 243, 2005: 216, 2004: 207, 2003: 16
9, 2001: 183, 2002: 204, 2000: 169}
```

In [23]:
```
# Esta celda debe ser completada por el estudiante
```

In [24]:
```
# Test de funcionamiento

print(num_movies_sorted)
```

```
[(2000, 169), (2001, 183), (2002, 204), (2003, 169), (2004, 207), (200
5, 216), (2006, 235), (2007, 198), (2008, 223), (2009, 253), (2010, 22
5), (2011, 224), (2012, 214), (2013, 231), (2014, 243), (2015, 211)]
```

- Diseña también una operación que, partiendo de nuestro diccionario de los datos principales, almacene en un archivo los datos siguientes de cada película, sin la cabecera: el título, el idioma, el año, el país y el presupuesto. El separador será en este caso el carácter `|`. El archivo usado para el almacenamiento es el siguiente:

    ``` python
    FEW_FIELDS = "algunos_campos.txt"
    ```

In [25]:
```
# Esta celda debe ser completada por el estudiante
```

In [26]:
```
# Test de funciomaniemto

store_file(main_dict_data, FEW_FIELDS)

! dir algunos*.*

print()

with open(FEW_FIELDS) as f:
    for i in range(5):
        print(f.readline())
```

```
 El volumen de la unidad C es Windows
 El n£mero de serie del volumen es: BEF4-40B1

 Directorio de C:\Users\CPareja\Jupyter\Q - enunciados\-- IMDB 5000 Mov
ie Dataset

29/08/2024  09:56           210.832 algunos_campos.txt
               1 archivos        210.832 bytes
               0 dirs  253.565.710.336 bytes libres

Avatar|2009|English|USA|237000000

Pirates of the Caribbean: At World's End|2007|English|USA|300000000

Spectre|2015|English|UK|245000000

The Dark Knight Rises|2012|English|USA|250000000

Star Wars: Episode VII - The Force Awakens|-1|||-1
```

B.3. Un conteo sencillo con `defaultdict`

Deseamos saber con qué directores y número de veces ha actuado cada actor como actor principal. Se pide realizar este conteo en un `defaultdict` cuyas claves serán nombres de los actores y cuyos valores tendrán la estructura de un `defaultdict`, cuyas claves serán los nombres de los directores y cuyos valores serán los núneros contabilizados.

In [27]:
```
# Esta celda debe ser completada por el estudiante
```

In [28]:
```
# Test de funcionamiento

num_collaborations = actor_directors(main_dict_data)

print(type(num_collaborations))
key_a, value_a = list(num_collaborations.items())[0]
print(type(key_a), type(value_a))
key_b, value_b = list(value_a.items())[0]
print(type(key_b), type(value_b))

print()

print(num_collaborations)
```

```
<class 'collections.defaultdict'>
<class 'str'> <class 'collections.defaultdict'>
<class 'str'> <class 'int'>

defaultdict(<function actor_directors.<locals>.<lambda> at 0x000001C4
E090B380>, {'CCH Pounder': defaultdict(<class 'int'>, {'James Camero
n': 1, 'Peter Hyams': 1, 'Fred Dekker': 1, 'Ernest R. Dickerson':
1}), 'Johnny Depp': defaultdict(<class 'int'>, {'Gore Verbinski': 5,
'Rob Marshall': 2, 'Tim Burton': 6, 'James Bobin': 1, 'Michael Mann':
1, 'Florian Henckel von Donnersmarck': 1, 'Wally Pfister': 1, 'David
Koepp': 2, 'Scott Cooper': 1, 'Albert Hughes': 1, 'Roman Polanski':
1, 'Wes Craven': 1, 'Mike Newell': 1, 'Rand Ravich': 1, 'Ted Demme':
1, 'Robert Rodriguez': 1, 'Marc Forster': 1, 'Jeremy Leven': 1, 'Laur
ence Dunmore': 1, 'Terry Gilliam': 1, 'Lasse Hallström': 1, 'Oliver S
tone': 1, 'Rachel Talalay': 1, 'Kevin Smith': 2}), 'Christoph Waltz':
defaultdict(<class 'int'>, {'Sam Mendes': 1, 'David Yates': 1, 'Miche
l Gondry': 1, 'Tim Burton': 1}), 'Tom Hardy': defaultdict(<class 'in
t'>, {'Christopher Nolan': 1, 'George Miller': 1, 'McG': 1, 'Stuart B
aird': 1, 'Daniel Espinosa': 1, 'Brian Helgeland': 1, "Gavin O'Conno
```

B.4. Print seleccionado

Demasiada información. Deseamos imprimir únicamente, para cada actor, las colaboraciones que superen un mínimo, dato entrada. Si un actor no tiene ninguna colaboración que supere dicho mínimo, lógicamente no debe mostrarse.

In [29]:
```
# Esta celda debe ser completada por el estudiante
```

```
Johnny Depp  ->  [('Gore Verbinski', 5), ('Tim Burton', 6)]
Leonardo DiCaprio  ->  [('Martin Scorsese', 5)]
Robert De Niro  ->  [('Martin Scorsese', 7)]
Bill Murray  ->  [('Wes Anderson', 5)]
Clint Eastwood  ->  [('Clint Eastwood', 10)]
Woody Allen  ->  [('Woody Allen', 10)]
```

C. Algunos gráficos sencillos [1 punto]

C.1 Un modelo de gráfica

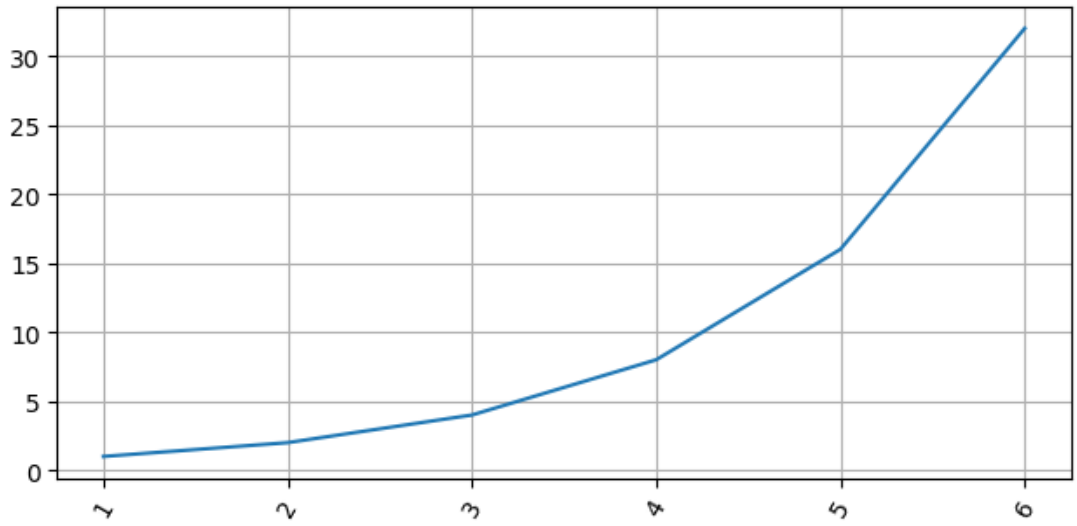
Vamos a diseñar un modelo de gráfica sencillo que nos sirva para las siguientes representaciones. Tomará como parámetro una lista de pares (x, y), y opcionalmente los tres rótulos explicativos que necesitamos incluir. Además, queremos que las etiquetas de las abcisas aparezcan inclinadas, para poder luego mostrar intervalos de edad.

In [30]:
```
# Esta celda debe ser completada por el estudiante
```

In [31]:
```
# Pruebas de funcionamiento:

representar_xxx_yyy([(1, 8), (2, 4), (3, 2), (4, 1), (5, 0.5), (6, 0.25)

representar_xxx_yyy([(1, 1), (2, 2), (3, 4), (4, 8), (5, 16), (6, 32)])
```



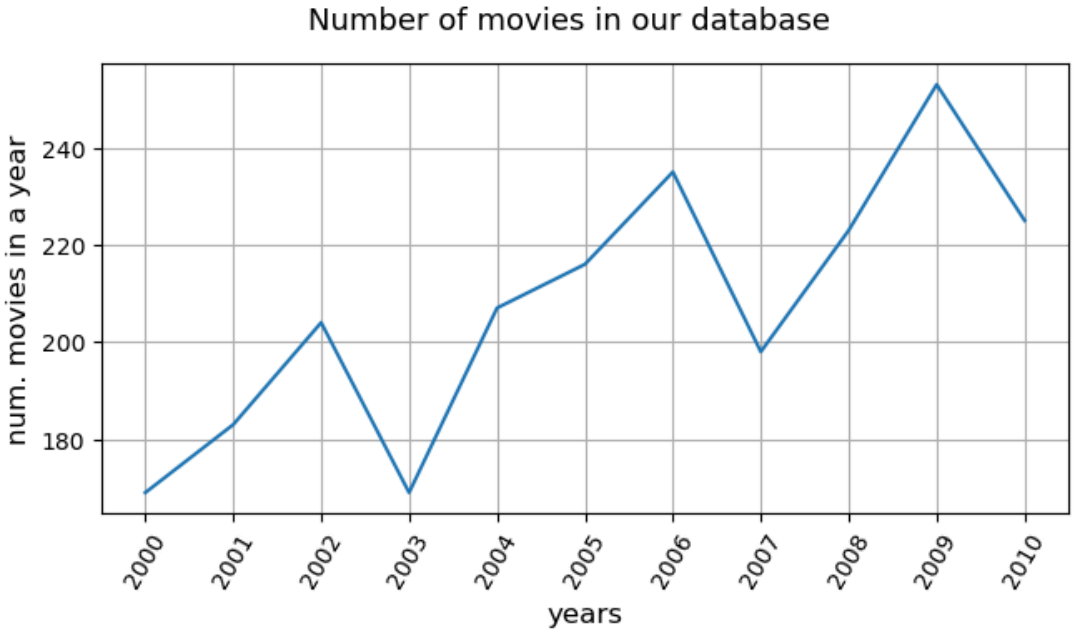
Una gráfica concreta

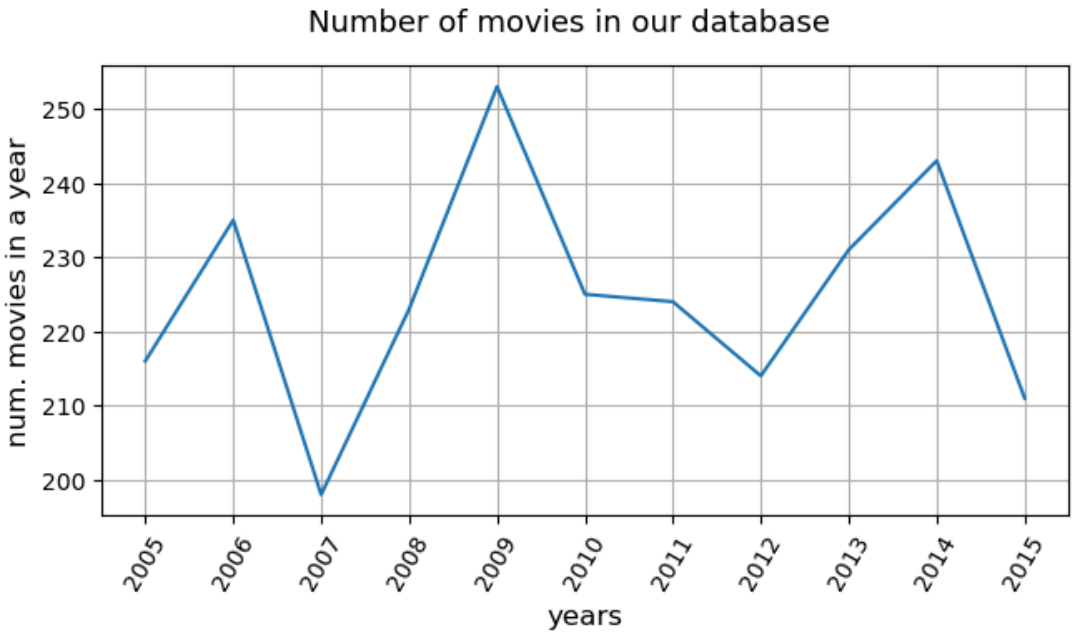
Deseamos representar el número de películas de nuestra base de datos que se han producido en un intervalo de años dado.

In [32]:
```
# Esta celda debe ser completada por el estudiante
```

In [33]:
```
# Test de funcionamiento

repr_movies_years(main_dict_data, 2000, 2010)
repr_movies_years(main_dict_data, 2005, 2015)
```





D. Acceso a las urls de imdb y webscraping [2 puntos]

D.1. Recuperación de las URLs

Con sencillas instrucciones, deseamos recuperar todas las *urls* de las películas de nuestro archivo, mostrando cuántas son, la primera de ellas o las diez primeras por ejemplo:

In [34]:
```
# Esta celda debe ser completada por el estudiante
```

In [35]:
```
# Test de funcionamiento

print(len(urls))

print()

print(first_url_movie)

print()

print(first_ten_urls)
```

```
4919

http://www.imdb.com/title/tt0499549/ (http://www.imdb.com/title/tt0499549/)

['http://www.imdb.com/title/tt0499549/', 'http://www.imdb.com/title/tt0449088/', 'http://www.imdb.com/title/tt2379713/', 'http://www.imdb.com/title/tt1345836/', 'http://www.imdb.com/title/tt5289954/', 'http://www.imdb.com/title/tt0401729/', 'http://www.imdb.com/title/tt0413300/', 'http://www.imdb.com/title/tt0398286/', 'http://www.imdb.com/title/tt2395427/', 'http://www.imdb.com/title/tt0417741/']
```

In [36]:
```
# Una forma de evitar el error "Requests 403 forbidden" al hacer web scr

HEADERS = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0
        "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9
        "Accept-Language": "en-US,en;q=0.5",
        "Accept-Encoding": "gzip, deflate",
        "Connection": "keep-alive",
        "Upgrade-Insecure-Requests": "1",
        "Sec-Fetch-Dest": "document",
        "Sec-Fetch-Mode": "navigate",
        "Sec-Fetch-Site": "none",
        "Sec-Fetch-User": "?1",
        "Cache-Control": "max-age=0",
    }
```

D.2. Carga de la estructura sintáctica de una URL

Ahora, deseamos extraer el código `html` de una película.

In [37]:
```
# Esta celda debe ser completada por el estudiante
```

In [38]:

```
# Test de funcionamiento

soup = soup_movie(first_url_movie)

print(str(soup)[:1000])
print()
print("... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... .
print()
print(str(soup)[-1000:])
```

```
<!DOCTYPE html>
<html lang="en-US" xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:o
g="http://opengraphprotocol.org/schema/"><head><meta charset="utf-8"/><
meta content="width=device-width" name="viewport"/><script>if(typeof ue
t === 'function'){ uet('bb', 'LoadTitle', {wb: 1}); }</script><script>w
indow.addEventListener('load', (event) => {
        if (typeof window.csa !== 'undefined' && typeof window.csa ===
'function') {
            var csaLatencyPlugin = window.csa('Content', {
                element: {
                    slotId: 'LoadTitle',
                    type: 'service-call'
                }
            });
            csaLatencyPlugin('mark', 'clickToBodyBegin', 172491821336
1);
        }
    })</script><title>Avatar (2009) - IMDb</title><meta content="Avata
r: Directed by James Cameron. With Sam Worthington, Zoe Saldana, Sigour
ney Weaver, Stephen Lang. A paraplegic Marine dispatched to the moon Pa
ndora on a unique mission becomes torn between following his orders and
pr

... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
... ... ... ... ... ... ...

g(e+c);return!!e}function n(){for(var e=RegExp("^https://(.*\.(images|s
sl-images|media)-amazon\.com|"+c.location.hostname+")/images/","i"),d=
{},h=0,k=c.performance.getEntriesByType("resource"),l=!1,b,a,m,f=0;f<k.
length;f++)if(a=k[f],0<a.transferSize&&a.transferSize>=a.encodedBodySiz
e&&(b=e.exec(String(a.name)))&&3===b.length){a:{b=a.serverTiming||[];fo
r(a=0;a<b.length;a++)if("provider"===b[a].name){b=b[a].description;brea
k a}b=void 0}b&&(l||(l=g(b,"_cdn_fr")),
a=d[b]=(d[b]||0)+1,a>h&&(m=b,h=a))}g(m,"_cdn_mp")}d.ue&&"function"===ty
peof d.ue.tag&&c.performance&&c.location&&n()},"cdnTagging")(ue_csm,win
dow);


}

/* ⚠ */
</script>
</div>
<noscript>
<img alt="" height="1" src="//fls-na.amazon.com/1/batch/1/OP/A1EVAM02EL
8SFB:143-8406919-7416635:WKYP1V6N2RW327XH73XJ$uedata=s:%2Fuedata%2Fueda
ta%3Fnoscript%26id%3DWKYP1V6N2RW327XH73XJ:0" style="display:none;visibi
lity:hidden;" width="1"/>
</noscript>
<script>window.ue && ue.count && ue.count('CSMLibrarySize', 60231)</scr
ipt></div></body></html>
```

D.3. Extracción de algunas piezas de información de una URL

Y ahora, con dicho código, deseamos extraer la siguiente información, referida a la película `first_url_movie` :

- La etiqueta completa del título de la película
- La descripción (sólo el contenido).

- La lista de los actores del *reparto principal*
- La información sobre el presupuesto

In [39]:
```
# Esta celda debe ser completada por el estudiante
```

```
<title>Avatar (2009) - IMDb</title>

Avatar: Directed by James Cameron. With Sam Worthington, Zoe Saldana, S
igourney Weaver, Stephen Lang. A paraplegic Marine dispatched to the mo
on Pandora on a unique mission becomes torn between following his order
s and protecting the world he feels is his home.

['Sam Worthington', 'Zoe Saldana', 'Sigourney Weaver', 'Michelle Rodrig
uez', 'Stephen Lang', 'Giovanni Ribisi', 'Joel David Moore', 'CCH Pound
er', 'Wes Studi', 'Laz Alonso', 'Dileep Rao', 'Matt Gerald', 'Sean Anth
ony Moran', 'Jason Whyte', 'Scott Lawrence', 'Kelly Kilgour', 'James Pa
trick Pitt', 'Sean Patrick Murphy']

$237,000,000 (estimated)
```

D.4. Actores que intervienen en una lista de URLs

Necesitamos crear un archivo con los actores del reparto principal de las películas de IMDB, dada la lista de sus URLs.

In [40]:
```
# Esta celda debe ser completada por el estudiante
```

In [41]:

```
# Test de funcionamiento

# OJO: esta operación puede llevar bastante tiempo.
# Para esta prueba, usamos un número limitado de películas.

gather_actors("actors_3_first_movies.txt", urls[:3])

! type actors_3_first_movies.txt
```

Sam Worthington
Zoe Saldana
Sigourney Weaver
Michelle Rodriguez
Stephen Lang
Giovanni Ribisi
Joel David Moore
CCH Pounder
Wes Studi
Laz Alonso
Dileep Rao
Matt Gerald
Sean Anthony Moran
Jason Whyte
Scott Lawrence
Kelly Kilgour
James Patrick Pitt
Sean Patrick Murphy
Johnny Depp
Orlando Bloom
Keira Knightley
Geoffrey Rush
Jack Davenport
Bill Nighy
Jonathan Pryce
Lee Arenberg
Mackenzie Crook
Kevin McNally
David Bailie
Stellan Skarsgård
Tom Hollander
Naomie Harris
Martin Klebba
David Schofield
Lauren Maher
Dermot Keaney
Daniel Craig
Christoph Waltz
Léa Seydoux
Ralph Fiennes
Monica Bellucci
Ben Whishaw
Naomie Harris
Dave Bautista
Andrew Scott
Rory Kinnear
Jesper Christensen
Alessandro Cremona
Stephanie Sigman
Tenoch Huerta
Adriana Paz
Domenico Fortunato
Marco Zingaro
Stefano Elfi DiClaudia

In [42]:
```
# La siguente llamada llevaría un tiempo realmente largo:

# import time # para cronometrar esta función, que tarda mucho

# reloj_inicio = time.time()
# gather_actors("actors_all_movies.txt", urls)
# reloj_fin = time.time()

# print("Tiempo invertido: %s segundos." % (reloj_fin - reloj_inicio))
```

E. Pandas [2 puntos]

E.1. El primer paso es la carga del archivo en un dataframe

In [43]:
```
# Esta celda debe ser completada por el estudiante
```

In [44]:
```
# Test de funcionamiento

tabla_completa = load_dataframe(MOVIES_DATA)

tabla_completa
```

Out[44]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	act
0	Color	James Cameron	723.0	178.0	0.0	
1	Color	Gore Verbinski	302.0	169.0	563.0	
2	Color	Sam Mendes	602.0	148.0	0.0	
3	Color	Christopher Nolan	813.0	164.0	22000.0	
4	NaN	Doug Walker	NaN	NaN	131.0	
...	...	...	...	...	...	
5038	Color	Scott Smith	1.0	87.0	2.0	
5039	Color	NaN	43.0	43.0	NaN	
5040	Color	Benjamin Roberds	13.0	76.0	0.0	
5041	Color	Daniel Hsia	14.0	100.0	0.0	
5042	Color	Jon Gunn	43.0	90.0	16.0	

5043 rows × 28 columns

E.2. Tabla de los campos principales

A pertir de la tabla anterior, construimos otra con sólo algumos de los campos:

In [45]:
```
# Esta celda debe ser completada por el estudiante
```

In [46]:
```
# test de comprobación

tabla_breve = fields_selected_dataframe(tabla_completa)

tabla_breve
```

Out[46]:

	movie_title	color	director_name	language	country	actor_1_name	
0	Avatar	Color	James Cameron	English	USA	CCH Pounder	http://www.im
1	Pirates of the Caribbean: At World's End	Color	Gore Verbinski	English	USA	Johnny Depp	http://www.im
2	Spectre	Color	Sam Mendes	English	UK	Christoph Waltz	http://www.im
3	The Dark Knight Rises	Color	Christopher Nolan	English	USA	Tom Hardy	http://www.im
4	Star Wars: Episode VII - The Force Awakens	NaN	Doug Walker	NaN	NaN	Doug Walker	http://www.im

E.3. Columnas de una tabla

¿Cuáles son las columnas de nuestrsa `tabla_breve` ?

In [47]:
```
# Esta celda debe ser completada por el estudiante
```

Out[47]:
```
Index(['movie_title', 'color', 'director_name', 'language', 'country',
       'actor_1_name', 'movie_imdb_link'],
      dtype='object')
```

E.4. Campos missing

Algunos campos muestran un valod `NaN` . Deseamos cambiarlo por una cadena de caracteres: `"Desc"` .

In [48]:
```
# Esta celda debe ser completada por el estudiante
```

In [49]:
```
# Test de comprobación

tabla_breve
```

Out[49]:

	movie_title	color	director_name	language	country	actor_1_name	
0	Avatar	Color	James Cameron	English	USA	CCH Pounder	http://www.im
1	Pirates of the Caribbean: At World's End	Color	Gore Verbinski	English	USA	Johnny Depp	http://www.im
2	Spectre	Color	Sam Mendes	English	UK	Christoph Waltz	http://www.im
3	The Dark Knight Rises	Color	Christopher Nolan	English	USA	Tom Hardy	http://www.im
4	Star Wars: Episode VII - The Force Awakens	Desc	Doug Walker	Desc	Desc	Doug Walker	http://www.im

E.5. Director → películas y número de películas

Función que averigua la lista de títulos de películas de un director dado:

In [50]:
```
# Esta celda debe ser completada por el estudiante
```

In [51]:
```
# Test de comprobación:

tabla_tits = titulos_de_director_df(tabla_breve, "James Cameron")
tabla_tits
```

Out[51]:

	movie_title
0	Avatar
26	Titanic
288	Terminator 2: Judgment Day
291	True Lies
606	The Abyss
2486	Aliens
3575	The Terminator

In [52]:
```
list_tits = tabla_tits["movie_title"].to_list()
print(list_tits)
```

```
['Avatar\xa0', 'Titanic\xa0', 'Terminator 2: Judgment Day\xa0', 'True L
ies\xa0', 'The Abyss\xa0', 'Aliens\xa0', 'The Terminator\xa0']
```

Deseamos saber qué directores han dirigido el máximo número de películas, junto con ese número de películas.

In [53]:
```
# Esta celda debe ser completada por el estudiante
```

In [54]:
```
# test de comprobación:

directors_max_movies_df(tabla_breve)
```

Out[54]:
```
(['Steven Spielberg'], 26)
```

Parte F. Un cálculo masivo con map-reduce [0,5 puntos]

En este apartado se ha de realizar un programa aparte, *basado en la técnica de map-reduce*, que calcule, para cada idioma, en qué países en que se han producido películas y la suma de los presupuestos de dichas películas. Cuando el idioma o el país o el presupuesto no se conozcan, no se considerará esta película.

```
C:\...> python language_budget_countries.py -q algunos_campos.txt
```

El programa funcionará necesariamente con la técnica map-reduce, que podemos poner en juego con la librería `mrjob` .

El funcionamiento del mismo se puede activar también desde aquí:

In [55]:
```
# Hagamos una llamada al programa de consola desde aquí:

! python language_budget_countries.py -q algunos_campos.txt
```

```
"Aboriginal"    [["UK","Australia"],86000000]
"Arabic"        [["Turkey","Egypt","France","United Arab Emirates"],1
1225000]
"Aramaic"       [["USA"],30000000]
"Bosnian"       [["USA"],13000000]
"Cantonese"     [["Hong Kong","China"],154500000]
"Chinese"       [["China"],12000000]
"Czech" [["Czech Republic"],84450000]
"Danish"        [["Denmark"],50100000]
"Dari"  [["USA","Afghanistan"],20046000]
"Dutch" [["Netherlands"],32150000]
"Dzongkha"      [["Australia"],1800000]
"English"       [["USA","UK","New Zealand","Canada","Australia","Germ
any","China","New Line","France","Japan","Spain","Hong Kong","Czech R
epublic","South Korea","Peru","Italy","Aruba","Denmark","Libya","Belg
ium","Ireland","South Africa","Switzerland","Romania","West German
y","Chile","Hungary","Russia","Mexico","Panama","Greece","Netherland
s","Norway","Official site","Bulgaria","Iran","Georgia","India","Thai
land","Nigeria","Bahamas","Iceland","Brazil","Poland","Kyrgyzstan","P
```

In [56]:
```
# Para que el resultado se almacene en un archivo:

! python language_budget_countries.py -q algunos_campos.txt > language_c
```

La siguiente celda me permite ver tu programa cómodamente desde aquí.

In [57]:
```
def print_file(filename):
    with open(filename, "r") as f:
        for line in f:
            print(line, end="")

print_file("language_budget_countries.py")
```

Parte G. Un apartado libre [0.5 puntos]

Dejo este apartado a tu voluntad. Inventa tú mismo el enunciado y resuélvelo. El enunciado deberá estar **relacionado con el análisis de datos y con el tema de este proyecto**. También, la idea es mostrar algún aspecto de programación en Python no contemplado o alguna técnica o librería que no has puesto en juego en los apartados anteriores. Concretamente, se valorará el uso de **la librería pandas**, que hemos estudiado de un modo bastante sucinto en este módulo y tratado de forma insuficiente en este proyecto de programación, o quizá puedes también usar alguna otra librería gráfica, distinta de matplotlib.

En la evaluación, si este apartado está bien o muy bien, anota un 0,3 o 0,4. El 0,5 lo reservaremos para las situaciones en que se presente algo brillante, con alguna idea original o alguna técnica novedosa o complejidad especial o algún gráfico vistoso. Especialmente quien opta a un 9,5 o más, debe esmerarse en plantear este apartado a la altura de esa calificación.

Tras eliminar el párrafo anterior, en verde, sustituye éste por tu enunciado, cuya fuente aparecerá en azul oscuro.

In [58]:
```
# Este apartado debe ser completado por el estudiante
```

In [59]:
```
# Pruebas de funcionamiento, también tarea del estudiante:
```

Datos personales

- **Apellidos:** .....
- **Nombre:** .....
- **Email:** .....
- **Fecha:** .....

Ficha de autoevaluación

Aquí vienen comentarios del estudiante. Lo siguiente es un ejemplo posible obviamente ... elimina este párrafo y redacta el tuyo propio, en azul.

Apartado	Calificación	Comentario
a)	2.0 / 2.5	Completamente resuelto
b)	0.0 / 2.0	No lo he conseguido
c)	0.0 / 1.5	No he entendido el enunciado
d)	0.25 / 1.0	Sólo he conseguido una parte mínima
e)	0.0 / 2.0	No lo he conseguido
f)	0.5 / 0.5	No lo he conseguido más que mínimamente
g)	0.0 / 0.5	No he logrado el correcto funcionamiento
Total	**2.75** / 10.0	**Suspenso**

Ayuda recibida y fuentes utilizadas

... comentarios del estudiante ... Pon tú este párrafo con tus propias observaciones. Elimina este párrafo en verde.

Comentario adicional

... Este apartado es optativo. Si lo completas, ponlo en azul; si no, suprímelo con su título.

In [ ]:
```
# Esta celda se ha de respetar: está aquí para comprobar
# el funcionamiento de algunas funciones por parte de tu profesor
```