https://colab.research.google.com/drive/1tTukDXsZ57sYtSZoy-2sZDL3z8HgtGzh?usp=sharing

```
import pandas as pd
import numpy as np
!gdown 1s2TkjSpzNc4SyxqRrQleZyDIHlc7bxnd
!gdown 1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm
movies = pd.read_csv('movies.csv', index_col=0)
directors = pd.read_csv('directors.csv',index_col=0)
data = movies.merge(directors, how='left', left_on='director_id',right_on='id')
data.drop(['director_id','id_y'],axis=1,inplace=True)
```

```
Downloading...
From: https://drive.google.com/uc?id=1s2TkjSpzNc4SyxqRrQleZyDIHlc7bxnd
To: /content/movies.csv
100% 112k/112k [00:00<00:00, 34.8MB/s]
Downloading...
From: https://drive.google.com/uc?id=1Ws-_s1fHZ9nHfGLVUQurbHDvStePlEJm
To: /content/directors.csv
100% 65.4k/65.4k [00:00<00:00, 38.7MB/s]
```

```
data.head()
```

	id_x	budget	popularity	revenue	title	vote_average	vote_count
0	43597	237000000	150	2787965087	Avatar	7.2	11800
1	43598	300000000	139	961000000	Pirates of the Caribbean: At World's End	6.9	4500
2	43599	245000000	107	880674609	Spectre	6.3	4466
3	43600	250000000	112	1084939099	The Dark Knight Rises	7.6	9106
4	43602	258000000	115	890871626	Spider-Man 3	5.9	3576

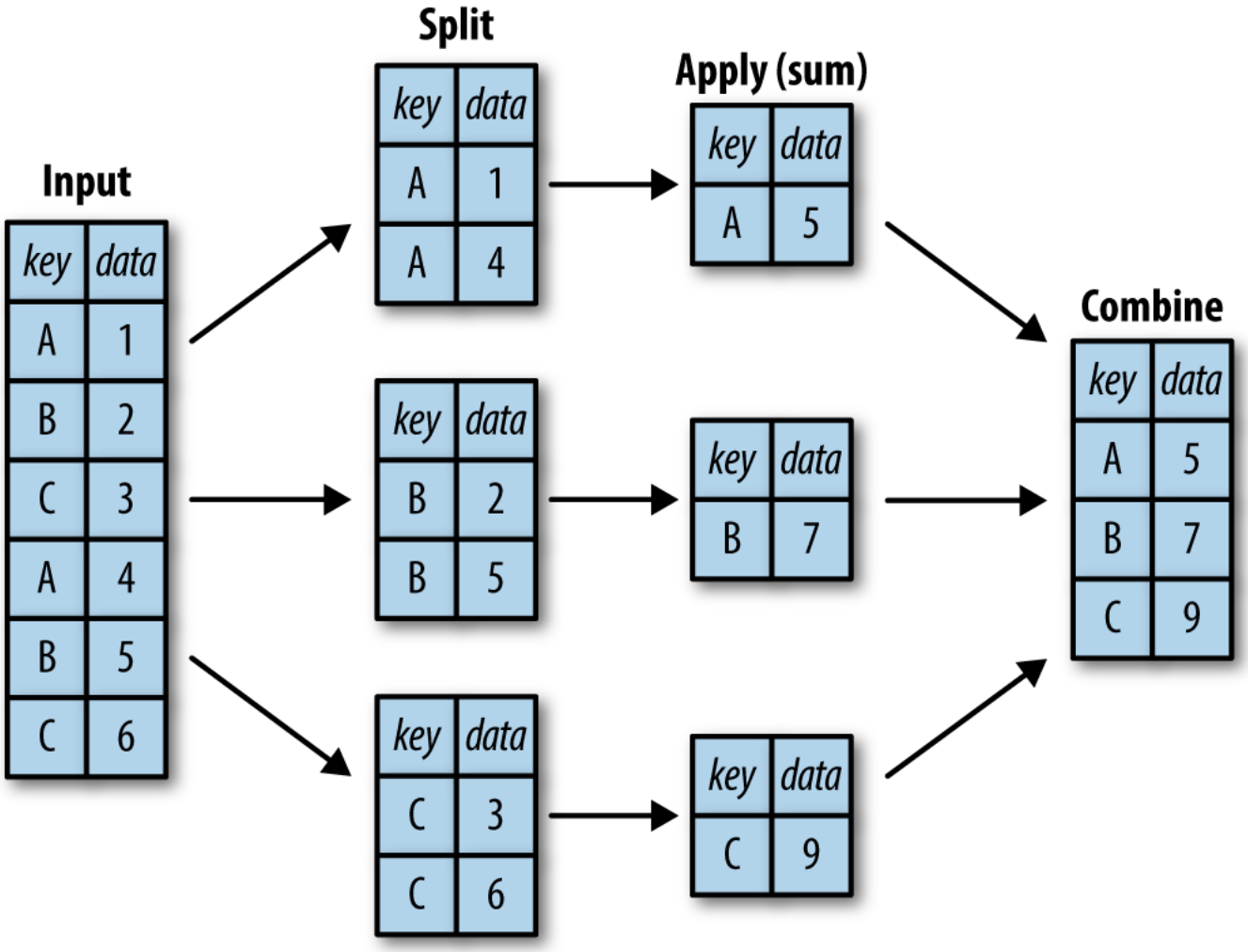
```
# Grouping
```

```
# How do we know the count of movies directed by Christopher Nolan?
data.loc[data['director_name'] == 'Christopher Nolan','title'].count()
```

```
8
```

```
# How would you do the same thing for every director?
data["director_name"].value_counts()
```

```
Steven Spielberg      26
Martin Scorsese       19
Clint Eastwood        19
Woody Allen           18
Ridley Scott          16
                      ..
Tim Hill               5
Jonathan Liebesman     5
Roman Polanski         5
Larry Charles          5
Nicole Holofcener      5
Name: director_name, Length: 199, dtype: int64
```




```
data.groupby("director_name")
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7f1f4c3c65e0>
```

```
data.groupby("director_name").ngroups
```

```
199
```

```
data.groupby("director_name").groups
```

```
{'Adam McKay': [176, 323, 366, 505, 839, 916], 'Adam Shankman': [265, 300, 350, 404, 458, 843, 999, 1231], 'Alejandro
González Iñárritu': [106, 749, 1015, 1034, 1077, 1405], 'Alex Proyas': [95, 159, 514, 671, 873], 'Alexander Payne':
[793, 1006, 1101, 1211, 1281], 'Andrew Adamson': [11, 43, 328, 501, 947], 'Andrew Niccol': [533, 603, 701, 722, 1439],
'Andrzej Bartkowiak': [349, 549, 754, 911, 924], 'Andy Fickman': [517, 681, 909, 926, 973, 1023], 'Andy Tennant': [314,
320, 464, 593, 676, 885], 'Ang Lee': [99, 134, 748, 840, 1089, 1110, 1132, 1184], 'Anne Fletcher': [610, 650, 736, 789,
1206], 'Antoine Fuqua': [310, 338, 424, 467, 576, 808, 818, 1105], 'Atom Egoyan': [946, 1128, 1164, 1194, 1347, 1416],
'Barry Levinson': [313, 319, 471, 594, 878, 898, 1013, 1037, 1082, 1143, 1185, 1345, 1378], 'Barry Sonnenfeld': [13, 48,
90, 205, 591, 778, 783], 'Ben Stiller': [209, 212, 547, 562, 850], 'Bill Condon': [102, 307, 902, 1233, 1381], 'Bobby
Farrelly': [352, 356, 481, 498, 624, 630, 654, 806, 928, 972, 1111], 'Brad Anderson': [1163, 1197, 1350, 1419, 1430],
'Brett Ratner': [24, 39, 188, 207, 238, 292, 405, 456, 920], 'Brian De Palma': [228, 255, 318, 439, 747, 905, 919, 1088,
1232, 1261, 1317, 1354], 'Brian Helgeland': [512, 607, 623, 742, 933], 'Brian Levant': [418, 449, 568, 761, 860, 1003],
'Brian Robbins': [416, 441, 669, 962, 988, 1115], 'Bryan Singer': [6, 32, 33, 44, 122, 216, 297, 1326], 'Cameron Crowe':
[335, 434, 488, 503, 513, 698], 'Catherine Hardwicke': [602, 695, 724, 937, 1406, 1412], 'Chris Columbus': [117, 167,
204, 218, 229, 509, 656, 897, 996, 1086, 1129], 'Chris Weitz': [17, 500, 794, 869, 1202, 1267], 'Christopher Nolan': [3,
45, 58, 59, 74, 565, 641, 1341], 'Chuck Russell': [177, 410, 657, 1069, 1097, 1339], 'Clint Eastwood': [369, 426, 447,
482, 490, 520, 530, 535, 645, 727, 731, 786, 787, 899, 974, 986, 1167, 1190, 1313], 'Curtis Hanson': [494, 579, 606,
711, 733, 1057, 1310], 'Danny Boyle': [527, 668, 1083, 1085, 1126, 1168, 1287, 1385], 'Darren Aronofsky': [113, 751,
1187, 1328, 1363, 1458], 'Darren Lynn Bousman': [1241, 1243, 1283, 1338, 1440], 'David Ayer': [50, 273, 741, 1024, 1146,
1407], 'David Cronenberg': [541, 767, 994, 1055, 1254, 1268, 1334], 'David Fincher': [62, 213, 253, 383, 398, 478, 522,
555, 618, 785], 'David Gordon Green': [543, 862, 884, 927, 1376, 1418, 1432, 1459], 'David Koepp': [443, 644, 735, 1041,
1209], 'David Lynch': [583, 1161, 1264, 1340, 1456], 'David O. Russell': [422, 556, 609, 896, 982, 989, 1229, 1304],
'David R. Ellis': [582, 634, 756, 888, 934], 'David Zucker': [569, 619, 965, 1052, 1175], 'Dennis Dugan': [217, 260,
267, 293, 303, 718, 780, 977, 1247], 'Donald Petrie': [427, 507, 570, 649, 858, 894, 1106, 1331], 'Doug Liman': [52,
148, 251, 399, 544, 1318, 1451], 'Edward Zwick': [92, 182, 346, 566, 791, 819, 825], 'F. Gary Gray': [308, 402, 491,
523, 697, 833, 1272, 1380], 'Francis Ford Coppola': [487, 559, 622, 646, 772, 1076, 1155, 1253, 1312], 'Francis
Lawrence': [63, 72, 109, 120, 679], 'Frank Coraci': [157, 249, 275, 451, 577, 599, 963], 'Frank Oz': [193, 355, 473,
580, 712, 813, 987], 'Garry Marshall': [329, 496, 528, 571, 784, 893, 1029, 1169], 'Gary Fleder': [518, 667, 689, 867,
981, 1165], 'Gary Winick': [258, 797, 798, 804, 1454], 'Gavin O'Connor': [820, 841, 939, 953, 1444], 'George A. Romero':
[250, 1066, 1096, 1278, 1367, 1396], 'George Clooney': [343, 450, 831, 966, 1302], 'George Miller': [78, 103, 233, 287,
1250, 1403, 1450], 'Gore Verbinski': [1, 8, 9, 107, 119, 633, 1040], 'Guillermo del Toro': [35, 252, 419, 486, 1118],
'Gus Van Sant': [595, 1018, 1027, 1159, 1240, 1311, 1398], 'Guy Ritchie': [124, 215, 312, 1093, 1225, 1269, 1420],
```

```
'Harold Ramis': [425, 431, 558, 586, 788, 1137, 1166, 1325], 'Ivan Reitman': [274, 643, 816, 883, 910, 935, 1134, 1242],
'James Cameron': [0, 19, 170, 173, 344, 1100, 1320], 'James Ivory': [1125, 1152, 1180, 1291, 1293, 1390, 1397], 'James
Mangold': [140, 141, 557, 560, 829, 845, 958, 1145], 'James Wan': [30, 617, 1002, 1047, 1337, 1417, 1424], 'Jan de
Bont': [155, 224, 231, 270, 781], 'Jason Friedberg': [812, 1010, 1012, 1014, 1036], 'Jason Reitman': [792, 1092, 1213,
1295, 1299], 'Jaume Collet-Serra': [516, 540, 640, 725, 1011, 1189], 'Jay Roach': [195, 359, 389, 397, 461, 703, 859,
1072], 'Jean-Pierre Jeunet': [423, 485, 605, 664, 765], 'Joe Dante': [284, 525, 638, 1226, 1298, 1428], 'Joe Wright':
[85, 432, 553, 803, 814, 855], 'Joel Coen': [428, 670, 691, 707, 721, 889, 906, 980, 1157, 1238, 1305], 'Joel
Schumacher': [128, 184, 348, 484, 572, 614, 652, 764, 876, 886, 1108, 1230, 1280], 'John Carpenter': [537, 663, 686,
861, 938, 1028, 1080, 1102, 1329, 1371], 'John Glen': [601, 642, 801, 847, 864], 'John Landis': [524, 868, 1276, 1384,
1435], 'John Madden': [457, 882, 1020, 1249, 1257], 'John McTiernan': [127, 214, 244, 351, 534, 563, 648, 782, 838,
1074], 'John Singleton': [294, 489, 732, 796, 1120, 1173, 1316], 'John Whitesell': [499, 632, 763, 1119, 1148], 'John
Woo': [131, 142, 264, 371, 420, 675, 1182], 'Jon Favreau': [46, 54, 55, 382, 759, 1346], 'Jon M. Chu': [100, 225, 810,
1099, 1186], 'Jon Turteltaub': [64, 180, 372, 480, 760, 846, 1171], 'Jonathan Demme': [277, 493, 1000, 1123, 1215],
'Jonathan Liebesman': [81, 143, 339, 1117, 1301], 'Judd Apatow': [321, 710, 717, 865, 881], 'Justin Lin': [38, 123, 246,
1437, 1447], 'Kenneth Branagh': [80, 197, 421, 879, 1094, 1277, 1288], 'Kenny Ortega': [412, 852, 1228, 1315, 1365],
'Kevin Reynolds': [53, 502, 639, 1019, 1059], ...}
```

```
data.groupby("director_name").get_group('Kenny Ortega')
```

	id_x	budget	popularity	revenue	title	vote_average	vote_count	yea
412	44316	60000000	15	0	This Is It	6.7	247	200
852	45315	28000000	18	39514713	Hocus Pocus	6.4	471	199
1228	46513	0	21	0	High School Musical 3: Senior Year	6.2	821	200
1315	47004	0	21	7000000	High School Musical 2	6.1	843	200
1365	47328	4200000	16	0	High School Musical	6.1	1000	200

```
# to Groupby aggregates
# count of the movies titles for every director
data.groupby("director_name")["title"].count()
```

```
director_name
Adam McKay                      6
Adam Shankman                   8
Alejandro González Iñárritu     6
Alex Proyas                     5
Alexander Payne                 5
                               ..
Wes Craven                     10
Wolfgang Petersen               7
Woody Allen                    18
Zack Snyder                     7
Zhang Yimou                     6
Name: title, Length: 199, dtype: int64
```

```
data.groupby("director_name")["title"].count()
```

```
director_name
Adam McKay                      6
Adam Shankman                   8
Alejandro González Iñárritu     6
Alex Proyas                     5
Alexander Payne                 5
                               ..
Wes Craven                     10
Wolfgang Petersen               7
Woody Allen                    18
Zack Snyder                     7
Zhang Yimou                     6
Name: title, Length: 199, dtype: int64
```

```
data.groupby("director_name")["year"].aggregate(["min", "max"])
```

director_name	min	max
Adam McKay	2004	2015
Adam Shankman	2001	2012
Alejandro González Iñárritu	2000	2015
Alex Proyas	1994	2016
Alexander Payne	1999	2013
...	...	...
Wes Craven	1984	2011
Wolfgang Petersen	1981	2006
Woody Allen	1977	2013
Zack Snyder	2004	2016

```
# Filter out director names with max budget > 100M?
data.groupby("director_name").filter(lambda x: x["budget"].max() >= 100)
```

	id_x	budget	popularity	revenue	title	vote_average	vote_coun
0	43597	237000000	150	2787965087	Avatar	7.2	1180
1	43598	300000000	139	961000000	Pirates of the Caribbean: At World's End	6.9	450
2	43599	245000000	107	880674609	Spectre	6.3	446
3	43600	250000000	112	1084939099	The Dark Knight Rises	7.6	910
4	43602	258000000	115	890871626	Spider-Man 3	5.9	357
...	...	...	...	...	...	...	.
1460	48363	0	3	321952	The Last Waltz	7.9	6
1461	48370	27000	19	3151130	Clerks	7.4	75
1462	48375	0	7	0	Rampage	6.0	13
1463	48376	0	3	0	Slacker	6.4	7
1464	48395	220000	14	2040920	El Mariachi	6.6	23

1465 rows × 12 columns

```
# can we directly filter rows with budget >100M and find unique names in that filter?
names = data.loc[data["budget"] >= 100, "director_name"]
names.unique()
```

```
array(['James Cameron', 'Gore Verbinski', 'Sam Mendes',
       'Christopher Nolan', 'Sam Raimi', 'Zack Snyder', 'Bryan Singer',
       'Marc Forster', 'Andrew Adamson', 'Rob Marshall',
       'Barry Sonnenfeld', 'Peter Jackson', 'Ridley Scott', 'Chris Weitz',
       'Peter Berg', 'Tim Burton', 'Brett Ratner', 'Michael Bay',
       'Martin Campbell', 'McG', 'James Wan', 'Mike Newell',
       'Guillermo del Toro', 'Steven Spielberg', 'Justin Lin',
       'Roland Emmerich', 'Robert Zemeckis', 'Lilly Wachowski',
       'Jon Favreau', 'Martin Scorsese', 'Rob Cohen', 'David Ayer',
       'Tom Shadyac', 'Doug Liman', 'Kevin Reynolds', 'David Fincher',
       'Francis Lawrence', 'Jon Turteltaub', 'Wolfgang Petersen',
       'Michael Apted', 'Oliver Stone', 'Shawn Levy', 'George Miller',
       'Ron Howard', 'Kenneth Branagh', 'Jonathan Liebesman',
       'M. Night Shyamalan', 'Joe Wright', 'Rob Minkoff', 'Lee Tamahori',
       'Edward Zwick', 'Alex Proyas', 'Richard Donner', 'Ang Lee',
       'Jon M. Chu', 'Bill Condon', 'Louis Leterrier',
       'Alejandro González Iñárritu', 'Paul Greengrass', 'Phillip Noyce',
       'Darren Aronofsky', 'Chris Columbus', 'Robert Schwentke',
       'Guy Ritchie', 'Paul Verhoeven', 'John McTiernan',
       'Joel Schumacher', 'John Woo', 'Tim Story', 'James Mangold',
       'Roger Donaldson', 'Steven Soderbergh', 'Raja Gosnell',
```

```
            'Jan de Bont', 'Frank Coraci', 'Michael Mann', 'Peter Chelsom',
            'Tony Scott', 'Paul Weitz', 'Adam McKay', 'Chuck Russell',
            'Quentin Tarantino', 'Simon West', 'Peter Hyams', 'Tom Tykwer',
            'Zhang Yimou', 'Frank Oz', 'Luc Besson', 'Mark Waters',
            'Renny Harlin', 'Ben Stiller', 'Dennis Dugan', 'Sydney Pollack',
            'Brian De Palma', 'Paul W.S. Anderson', 'Nancy Meyers',
            'Peter Segal', 'George A. Romero', 'Todd Phillips', 'Gary Winick',
            'Adam Shankman', 'Les Mayfield', 'Ivan Reitman', 'Stephen Hopkins',
            'Jonathan Demme', 'Terry Gilliam', 'Joe Dante', 'John Singleton',
            'Mike Nichols', 'F. Gary Gray', 'Antoine Fuqua', 'Robert Luketic',
            'Barry Levinson', 'Andy Tennant', 'Judd Apatow', 'Garry Marshall',
            'Cameron Crowe', 'George Clooney', 'Andrzej Bartkowiak',
            'Bobby Farrelly', 'Jay Roach', 'Lawrence Kasdan', 'Clint Eastwood',
            'Larry Charles', 'Taylor Hackford', 'Roman Polanski',
            'Robert Rodriguez', 'Rob Reiner', 'Tim Hill', 'Robert Redford',
            'Kenny Ortega', 'Brian Robbins', 'Brian Levant',
            'David O. Russell', 'Jean-Pierre Jeunet', 'Harold Ramis',
            'Donald Petrie', 'Joel Coen', 'Rod Lurie', 'David Koepp',
            'Uwe Boll', 'Stephen Herek', 'John Madden', 'Wayne Wang',
            'Francis Ford Coppola', 'Neil Jordan', 'Spike Lee',
            'Brian Helgeland', 'Jaume Collet-Serra', 'Andy Fickman',
            'Gary Fleder', 'John Landis', 'Danny Boyle', 'Andrew Niccol',
            'John Carpenter', 'Wes Anderson', 'David Cronenberg',
            'David Gordon Green', 'Richard LaGravenese', 'Stephen Frears',
            'David Zucker', 'Curtis Hanson', 'David R. Ellis', 'David Lynch',
            'Gus Van Sant', 'John Glen', 'Catherine Hardwicke',
            'Anne Fletcher', 'Wes Craven', 'John Whitesell',
            'Nicholas Stoller', 'Stephen Daldry', 'Paul Thomas Anderson',
            'Kirk Jones', 'Kevin Smith', 'Scott Hicks', 'Jason Reitman',
            'Alexander Payne', 'Woody Allen', 'Jason Friedberg',
            "Gavin O'Connor", 'Lasse Hallström', 'Miguel Arteta',
            'Malcolm D. Lee', 'Steve Miner', 'Richard Linklater',
            'Atom Egoyan', 'Sidney Lumet', 'Mira Nair', 'Tyler Perry',
            'Michael Moore', 'James Ivory', 'Michael Winterbottom',
            'Brad Anderson', 'Michael Polish', 'Mike Leigh',
            'Darren Lynn Bousman', 'Paul Schrader', 'Nicole Holofcener'],
```

```
# Find all the movies directed by high budget directors?
# High Budget Directors - directed at least 1 100M$ movie

# def something(x):
#   print(x) ----> try checking if this is df or a row
#   return x["b"]
data.groupby("director_name").filter(lambda x: x["budget"].max() >= 100)


# Group based apply


# Find all the risky movies
# Risky Movie - whose budget is higher than average revenue of its directors

def func(x):
  x["risky"] = x["budget"] - x["revenue"].mean() >= 0
  return x

data_risky = data.groupby("director_name").apply(func)


data_risky.loc[data_risky["risky"]]
```

	id_x	budget	popularity	revenue	title	vote_average	vote_coun
7	43608	200000000	107	586090727	Quantum of Solace	6.1	296
12	43614	380000000	135	1045713802	Pirates of the Caribbean: On Stranger Tides	6.4	494
15	43618	200000000	37	310669540	Robin Hood	6.2	139
20	43624	200000000	64	303025485	Battleship	5.5	211

```
# transform --> post-read
# transform is just like apply, but applied one column
```

Stand

```
# Who is the most productive director?
# Assume: Number of movies
data_agg = data.groupby("director_name")[["year", "title"]].aggregate(
    {
        "year": ["min", "max"],
        "title": "count"
    }
)
data_agg
```

	year		title
	min	max	count
director_name			
Adam McKay	2004	2015	6
Adam Shankman	2001	2012	8
Alejandro González Iñárritu	2000	2015	6
Alex Proyas	1994	2016	5
Alexander Payne	1999	2013	5
...	...	...	...
Wes Craven	1984	2011	10
Wolfgang Petersen	1981	2006	7
Woody Allen	1977	2013	18
Zack Snyder	2004	2016	7
Zhang Yimou	2002	2014	6

199 rows × 3 columns

```
data_agg.columns
```

```
MultiIndex([( 'year',   'min'),
            ( 'year',   'max'),
            ('title', 'count')],
           )
```

```
data_agg["year"]
```

director_name	min	max
Adam McKay	2004	2015

```
data_agg["title"]
```

director_name	count
Adam McKay	6
Adam Shankman	8
Alejandro González Iñárritu	6
Alex Proyas	5
Alexander Payne	5
...	...
Wes Craven	10
Wolfgang Petersen	7
Woody Allen	18
Zack Snyder	7
Zhang Yimou	6

199 rows × 1 columns

```
data_agg.columns = ["_".join(tup) for tup in data_agg.columns]
```

```
data_agg
```

director_name	year_min	year_max	title_count
Adam McKay	2004	2015	6
Adam Shankman	2001	2012	8
Alejandro González Iñárritu	2000	2015	6
Alex Proyas	1994	2016	5
Alexander Payne	1999	2013	5
...	...	...	...
Wes Craven	1984	2011	10
Wolfgang Petersen	1981	2006	7
Woody Allen	1977	2013	18
Zack Snyder	2004	2016	7
Zhang Yimou	2002	2014	6

199 rows × 3 columns

```
data_agg = data.groupby("director_name")[["year", "title"]].aggregate(
    year_max = ("year", "max"),
    year_min = ("year", "min"),
    title_count = ("title", "count")
).reset_index()
data_agg
```

	director_name	year_max	year_min	title_count
0	Adam McKay	2015	2004	6
1	Adam Shankman	2012	2001	8
2	Alejandro González Iñárritu	2015	2000	6
3	Alex Proyas	2016	1994	5

```
data_agg["yrs_active"] = data_agg["year_max"] - data_agg["year_min"]
data_agg
```

	director_name	year_max	year_min	title_count	yrs_active
0	Adam McKay	2015	2004	6	11
1	Adam Shankman	2012	2001	8	11
2	Alejandro González Iñárritu	2015	2000	6	15
3	Alex Proyas	2016	1994	5	22
4	Alexander Payne	2013	1999	5	14
...	...	...	...	...	...
194	Wes Craven	2011	1984	10	27
195	Wolfgang Petersen	2006	1981	7	25
196	Woody Allen	2013	1977	18	36
197	Zack Snyder	2016	2004	7	12
198	Zhang Yimou	2014	2002	6	12

199 rows × 5 columns

```
data_agg["movie_per_yr"] = data_agg["title_count"]/data_agg["yrs_active"]
data_agg
```

	director_name	year_max	year_min	title_count	yrs_active	movie_per_yr
0	Adam McKay	2015	2004	6	11	0.545455
1	Adam Shankman	2012	2001	8	11	0.727273
2	Alejandro González Iñárritu	2015	2000	6	15	0.400000
3	Alex Proyas	2016	1994	5	22	0.227273
4	Alexander Payne	2013	1999	5	14	0.357143
...	...	...	...	...	...	...
194	Wes Craven	2011	1984	10	27	0.370370
195	Wolfgang Petersen	2006	1981	7	25	0.280000
196	Woody Allen	2013	1977	18	36	0.500000
197	Zack Snyder	2016	2004	7	12	0.583333
198	Zhang Yimou	2014	2002	6	12	0.500000

199 rows × 6 columns

```
data_agg.sort_values("movie_per_yr", ascending=False)[["director_name", "movie_per_yr"]]
```

	director_name	movie_per_yr
190	Tyler Perry	1.285714
169	Shawn Levy	0.916667
158	Robert Rodriguez	0.727273
1	Adam Shankman	0.727273
...	...	...
104	Lawrence Kasdan	0.185185
109	Luc Besson	0.172414

✓ 0s completed at 22:44