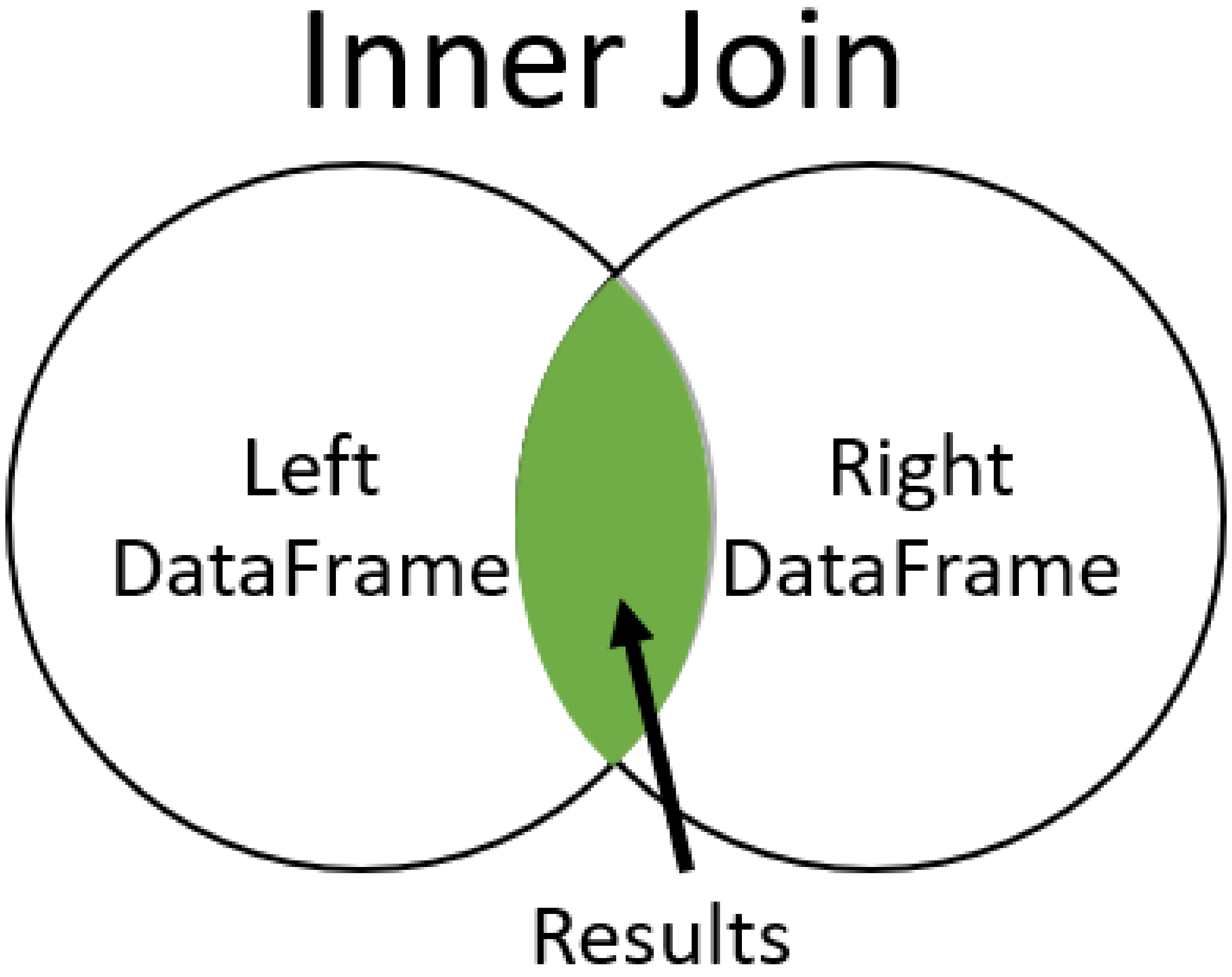
Left join

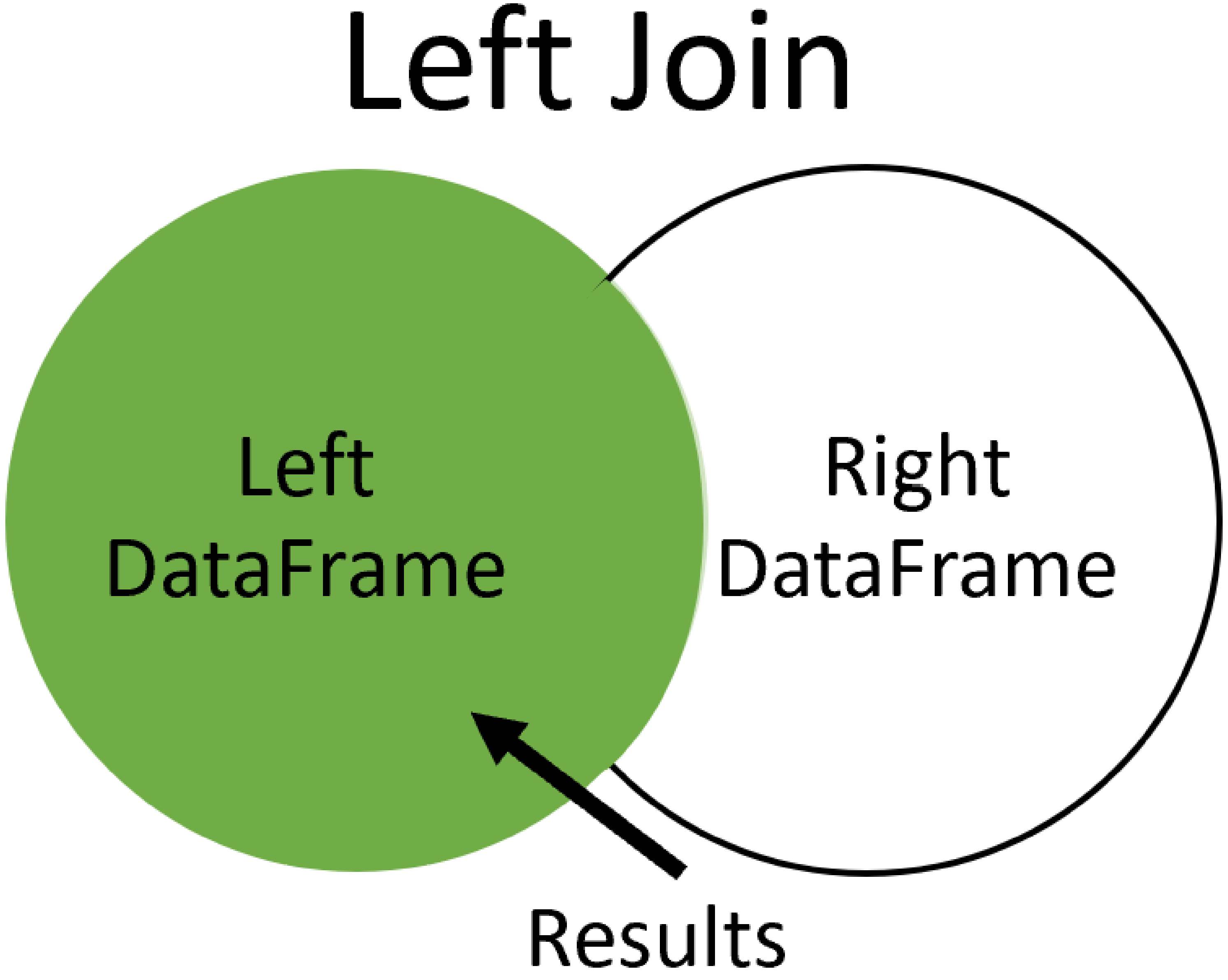
JOINING DATA WITH PANDAS

Aaren Stubberfield
Instructor

Quick review



Left join



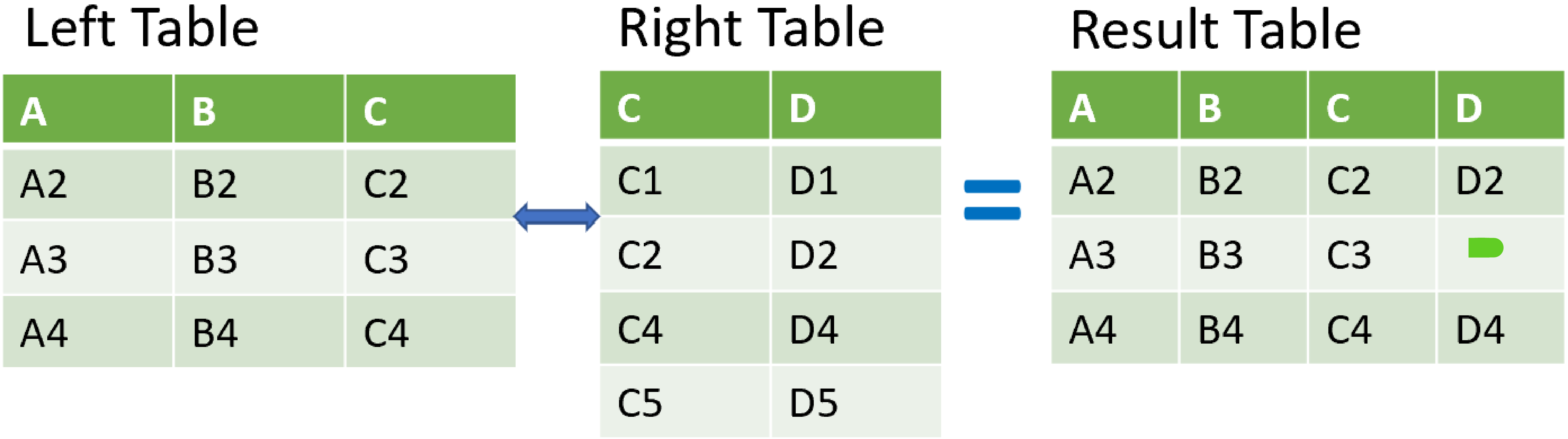
Left Join

Left DataFrame

Right DataFrame

Results

Left join



Left Table

A	B	C
A2	B2	C2
A3	B3	C3
A4	B4	C4

Right Table

C	D
C1	D1
C2	D2
C4	D4
C5	D5

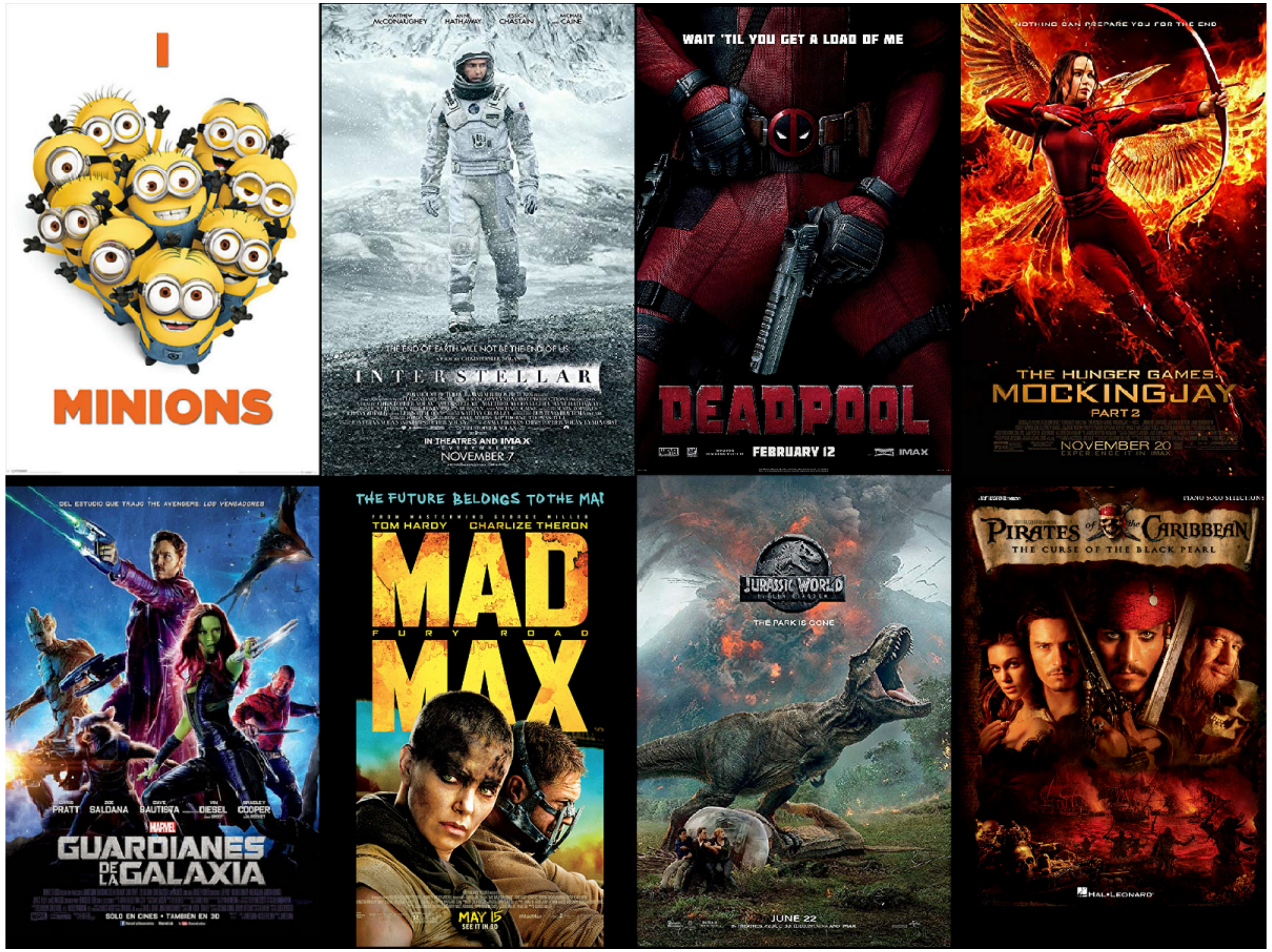
Result Table

A	B	C	D
A2	B2	C2	D2
A3	B3	C3	🟩
A4	B4	C4	D4

Here we want to use left join to merge them n the key column C, htla2y enu fl result, howa rg3 el left table kolo
lakn mn el right table, howa mgbsh gher bs el D2 wl D4, 34an homa dol elly mwgud el C bta3thum fl C bta3t el left table.

New dataset

THE MOVIE DB



Movies table

```
movies = pd.read_csv('tmdb_movies.csv')
print(movies.head())
print(movies.shape)
```

```
  id     original_title    popularity       release_date
0 257    Oliver Twist      20.415572        2005-09-23
1 14290  Better Luck ...   3.877036         2002-01-12
2 38365  Grown Ups         38.864027        2010-06-24
3 9672   Infamous          3.6808959999...  2006-11-16
4 12819  Alpha and Omega   12.300789        2010-09-17
(4803, 4)
```

Tagline table

```
taglines = pd.read_csv('tmdb_taglines.csv')
print(taglines.head())
print(taglines.shape)
```

```
  id      tagline
0 19995   Enter the World of Pandora.
1 285     At the end of the world, the adventure begins.
2 206647  A Plan No One Escapes
3 49026   The Legend Ends
4 49529   Lost in our world, found in another.
(3955, 2)
```

Merge with left join

```
movies_taglines = movies.merge(taglines, on='id', how='left')
print(movies_taglines.head())
```

```
  id     original_title   popularity       release_date  tagline
0 257    Oliver Twist     20.415572        2005-09-23    NaN
1 14290  Better Luck ...  3.877036         2002-01-12    Never undere...
2 38365  Grown Ups        38.864027        2010-06-24    Boys will be...
3 9672   Infamous         3.6808959999...  2006-11-16    There's more...
4 12819  Alpha and Omega  12.300789        2010-09-17    A Pawsome 3D...
```

Number of rows returned

```
print(movies_taglines.shape)
```

```
(4805, 5)
```

In one-to-one relation ships, the left or the right join, will return the same number of rows of the corresponding table however, in one-to-many, it will return more rows.

inner join will not return all the data, so sometimes it is a waste if we used the inner join, however, if we want to keep all our data, then we could use the left or right join instead.

Let's practice!

JOINING DATA WITH PANDAS

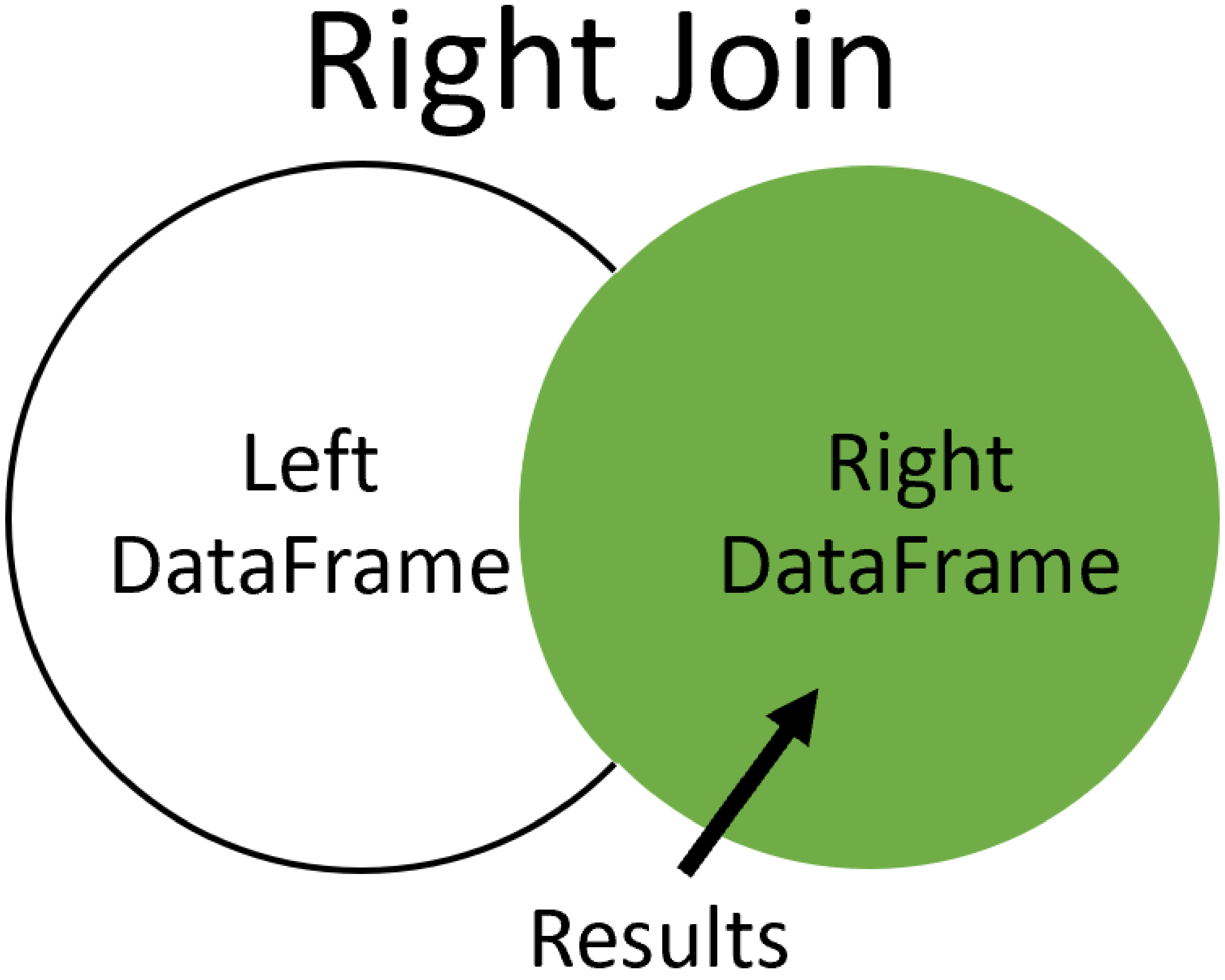
Other joins

JOINING DATA WITH PANDAS

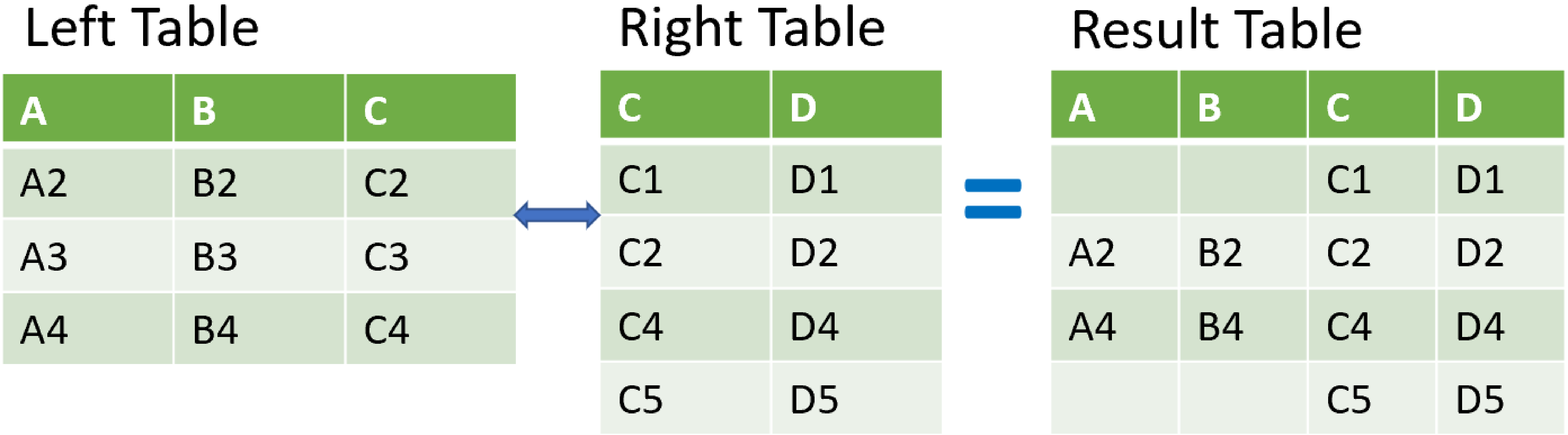
Aaren Stubberfield

Instructor

Right join



Right join



Left Table

A	B	C
A2	B2	C2
A3	B3	C3
A4	B4	C4

Right Table

C	D
C1	D1
C2	D2
C4	D4
C5	D5

Result Table

A	B	C	D
		C1	D1
A2	B2	C2	D2
A4	B4	C4	D4
		C5	D5

Looking at data

```
movie_to_genres = pd.read_csv('tmdb_movie_to_genres.csv')
tv_genre = movie_to_genres[movie_to_genres['genre'] == 'TV Movie']
print(tv_genre)
```

```
       movie_id  genre
4998   10947     TV Movie
5994   13187     TV Movie
7443   22488     TV Movie
10061  78814     TV Movie
10790  153397    TV Movie
10835  158150    TV Movie
11096  205321    TV Movie
11282  231617    TV Movie
```

Filtering the data

```
m = movie_to_genres['genre'] == 'TV Movie'
tv_genre = movie_to_genres[m]
print(tv_genre)
```

```
       movie_id  genre
4998   10947     TV Movie
5994   13187     TV Movie
7443   22488     TV Movie
10061  78814     TV Movie
10790  153397    TV Movie
10835  158150    TV Movie
11096  205321    TV Movie
11282  231617    TV Movie
```

Data to merge

```
  id     title             popularity       release_date
0 257    Oliver Twist      20.415572        2005-09-23
1 14290  Better Luck ...   3.877036         2002-01-12
2 38365  Grown Ups         38.864027        2010-06-24
3 9672   Infamous          3.6808959999...  2006-11-16
4 12819  Alpha and Omega   12.300789        2010-09-17
```

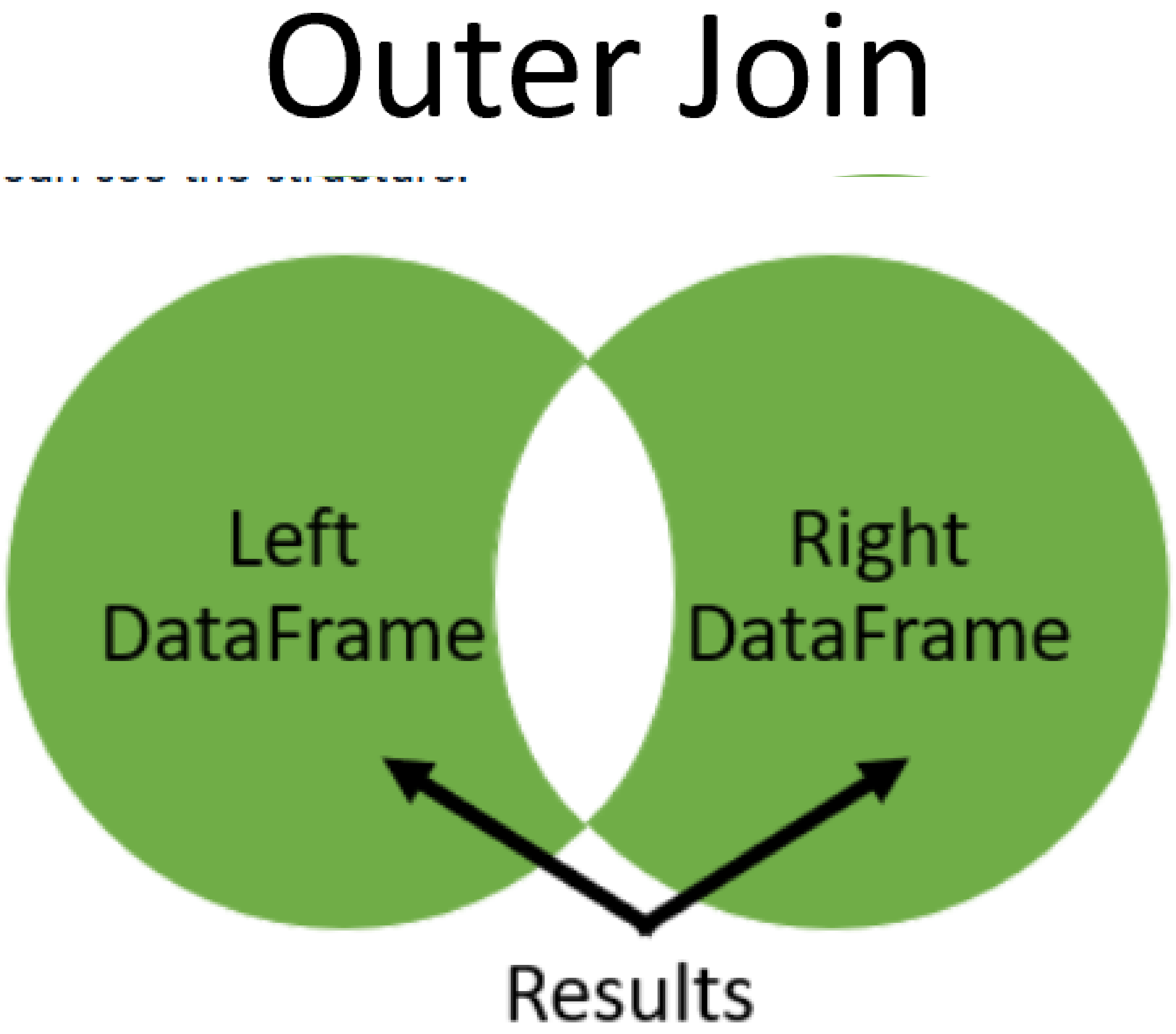
```
       movie_id  genre
4998   10947     TV Movie
5994   13187     TV Movie
7443   22488     TV Movie
10061  78814     TV Movie
10790  153397    TV Movie
```

Merge with right join

```
tv_movies = movies.merge(tv_genre, how='right',
                         left_on='id', right_on='movie_id')
print(tv_movies.head())
```

```
  id      title             popularity  release_date  movie_id  genre
0 153397  Restless          0.812776    2012-12-07    153397    TV Movie
1 10947   High School ...   16.536374   2006-01-20    10947     TV Movie
2 231617  Signed, Seal...   1.444476    2013-10-13    231617    TV Movie
3 78814   We Have Your...   0.102003    2011-11-12    78814     TV Movie
4 158150  How to Fall ...   1.923514    2012-07-21    158150    TV Movie
```

Outer join



Outer Join

Outer join

Left Table

A	B	C
A2	B2	C2
A3	B3	C3
A4	B4	C4

Right Table

C	D
C1	D1
C2	D2
C4	D4
C5	D5

Result Table

A	B	C	D
		C1	D1
A2	B2	C2	D2
A3	B3	C3	
A4	B4	C4	D4
		C5	D5

Datasets for outer join

```
m = movie_to_genres['genre'] == 'Family'
family = movie_to_genres[m].head(3)
```

```
  movie_id  genre
0 12        Family
1 35        Family
2 105       Family
```

```
m = movie_to_genres['genre'] == 'Comedy'
comedy = movie_to_genres[m].head(3)
```

```
  movie_id  genre
0 5         Comedy
1 13        Comedy
2 35        Comedy
```

Merge with outer join

```
family_comedy = family.merge(comedy, on='movie_id', how='outer',
                             suffixes=('_fam', '_com'))
print(family_comedy)
```

```
  movie_id  genre_fam  genre_com
0 12        Family     NaN
1 35        Family     Comedy
2 105       Family     NaN
3 5         NaN        Comedy
4 13        NaN        Comedy
```

Let's practice!

JOINING DATA WITH PANDAS

Merging a table to itself

JOINING DATA WITH PANDAS

Aaren Stubberfield
Instructor

Sequel movie data

```
print(sequel.head())
```

```
     id     title          sequel
0 19995   Avatar         NaN
1 862     Toy Story      863
2 863     Toy Story 2    10193
3 597     Titanic        NaN
4 24428   The Avengers NaN
```

Merging a table to itself

Left Table

id	title	sequel
19995	Avatar	
862	Toy Story	863
863	Toy Story 2	10193
597	Titanic	
24428	The Ave...	

Right Table

id	title	sequel
19995	Avatar	
862	Toy Story	863
863	Toy Story 2	10193
597	Titanic	
24428	The Ave...	

Result Table

id_x	title_x	sequel_x	id_y	title_y	sequel_y
862	Toy Story	863	863	Toy Story 2	10193
863	Toy Story 2	10193	10193	Toy Story 3	

Merge Columns

Merging a table to itself

```
original_sequels = sequels.merge(sequels, left_on='sequel', right_on='id',
                                 suffixes=('_org','_seq'))
print(original_sequels.head())
```

```
  id_org  title_org        sequel_org  id_seq  title_seq        sequel_seq
0 862     Toy Story        863         863     Toy Story 2      10193
1 863     Toy Story 2      10193       10193   Toy Story 3      NaN
2 675     Harry Potter...  767         767     Harry Potter...  NaN
3 121     The Lord of ...  122         122     The Lord of ...  NaN
4 120     The Lord of ...  121         121     The Lord of ...  122
```

Continue format results

```
print(original_sequels[,['title_org','title_seq']].head())
```

```
  title_org         title_seq
0 Toy Story         Toy Story 2
1 Toy Story 2       Toy Story 3
2 Harry Potter...   Harry Potter...
3 The Lord of ...   The Lord of ...
4 The Lord of ...   The Lord of ...
```

Merging a table to itself with left join

```
original_sequels = sequels.merge(sequels, left_on='sequel', right_on='id',
                                 how='left', suffixes=('_org','_seq'))
print(original_sequels.head())
```

```
  id_org  title_org      sequel_org  id_seq  title_seq     sequel_seq
0 19995   Avatar         NaN         NaN     NaN           NaN
1 862     Toy Story      863         863     Toy Story 2   10193
2 863     Toy Story 2    10193       10193   Toy Story 3   NaN
3 597     Titanic        NaN         NaN     NaN           NaN
4 24428   The Avengers   NaN         NaN     NaN           NaN
```

When to merge at table to itself

Common situations:

- Hierarchical relationships
- Sequential relationships
- Graph data

Let's practice!

JOINING DATA WITH PANDAS

Merging on indexes

JOINING DATA WITH PANDAS

Aaren Stubberfield
Instructor

Table with an index

```
  id     title             popularity  release_date
0 257    Oliver Twist      20.415572   2005-09-23
1 14290  Better Luck ...   3.877036    2002-01-12
2 38365  Grown Ups         38.864027   2010-06-24
3 9672   Infamous          3.680896    2006-11-16
4 12819  Alpha and Omega   12.300789   2010-09-17
```

```
       title             popularity  release_date
id
257    Oliver Twist      20.415572   2005-09-23
14290  Better Luck ...   3.877036    2002-01-12
38365  Grown Ups         38.864027   2010-06-24
9672   Infamous          3.680896    2006-11-16
12819  Alpha and Omega   12.300789   2010-09-17
```

in the below table, we set the id as the index.

Setting an index

```
movies = pd.read_csv('tmdb_movies.csv', index_col=['id'])
print(movies.head())
```

```
       title             popularity  release_date
id
257    Oliver Twist      20.415572   2005-09-23
14290  Better Luck ...   3.877036    2002-01-12
38365  Grown Ups         38.864027   2010-06-24
9672   Infamous          3.680896    2006-11-16
12819  Alpha and Omega   12.300789   2010-09-17
```

Merge index datasets

```
       title             popularity  release_date
id
257    Oliver Twist      20.415572   2005-09-23
14290  Better Luck ...   3.877036    2002-01-12
38365  Grown Ups         38.864027   2010-06-24
9672   Infamous          3.680896    2006-11-16
```

```
        tagline
id
19995   Enter the Wo...
285     At the end o...
206647  A Plan No On...
49026   The Legend Ends
```

Merging on index

```
movies_taglines = movies.merge(taglines, on='id', how='left')
print(movies_taglines.head())
```

```
       title             popularity  release_date  tagline
id
257    Oliver Twist      20.415572   2005-09-23    NaN
14290  Better Luck ...   3.877036    2002-01-12    Never undere...
38365  Grown Ups         38.864027   2010-06-24    Boys will be...
9672   Infamous          3.680896    2006-11-16    There's more...
12819  Alpha and Omega   12.300789   2010-09-17    A Pawsome 3D...
```

MultiIndex datasets

```
samuel = pd.read_csv('samuel.csv',
                     index_col=['movie_id',
                                'cast_id'])
print(samuel.head())
```

```
                   name
movie_id  cast_id
184       3        Samuel L. Jackson
319       13       Samuel L. Jackson
326       2        Samuel L. Jackson
329       138      Samuel L. Jackson
393       21       Samuel L. Jackson
```

```
casts = pd.read_csv('casts.csv',
                    index_col=['movie_id',
                               'cast_id'])
print(casts.head())
```

```
                   character
movie_id  cast_id
5         22       Jezebel
          23       Diana
          24       Athena
          25       Elspeth
          26       Eva
```

MultiIndex merge

```
samuel_casts = samuel.merge(casts, on=['movie_id','cast_id'])
print(samuel_casts.head())
print(samuel_casts.shape)
```

```
                   name              character
movie_id  cast_id
184       3        Samuel L. Jackson  Ordell Robbie
319       13       Samuel L. Jackson  Big Don
326       2        Samuel L. Jackson  Neville Flynn
329       138      Samuel L. Jackson  Arnold
393       21       Samuel L. Jackson  Rufus
(67, 2)
```

Index merge with left_on and right_on

```
       title             popularity  release_date
id
257    Oliver Twist      20.415572   2005-09-23
14290  Better Luck ...   3.877036    2002-01-12
38365  Grown Ups         38.864027   2010-06-24
9672   Infamous          3.680896    2006-11-16
```

```
          genre
movie_id
5         Crime
5         Comedy
11        Science Fiction
11        Action
```

Index merge with left_on and right_on

```
movies_genres = movies.merge(movie_to_genres, left_on='id', left_index=True,
                             right_on='movie_id', right_index=True)
print(movies_genres.head())
```

```
    id  title       popularity  release_date  genre
5    5  Four Rooms  22.876230   1995-12-09    Crime
5    5  Four Rooms  22.876230   1995-12-09    Comedy
11  11  Star Wars   126.393695  1977-05-25    Science Fiction
11  11  Star Wars   126.393695  1977-05-25    Action
11  11  Star Wars   126.393695  1977-05-25    Adventure
```

Let's practice!

JOINING DATA WITH PANDAS