Merging data in Python

Paul Bradshaw

This week:

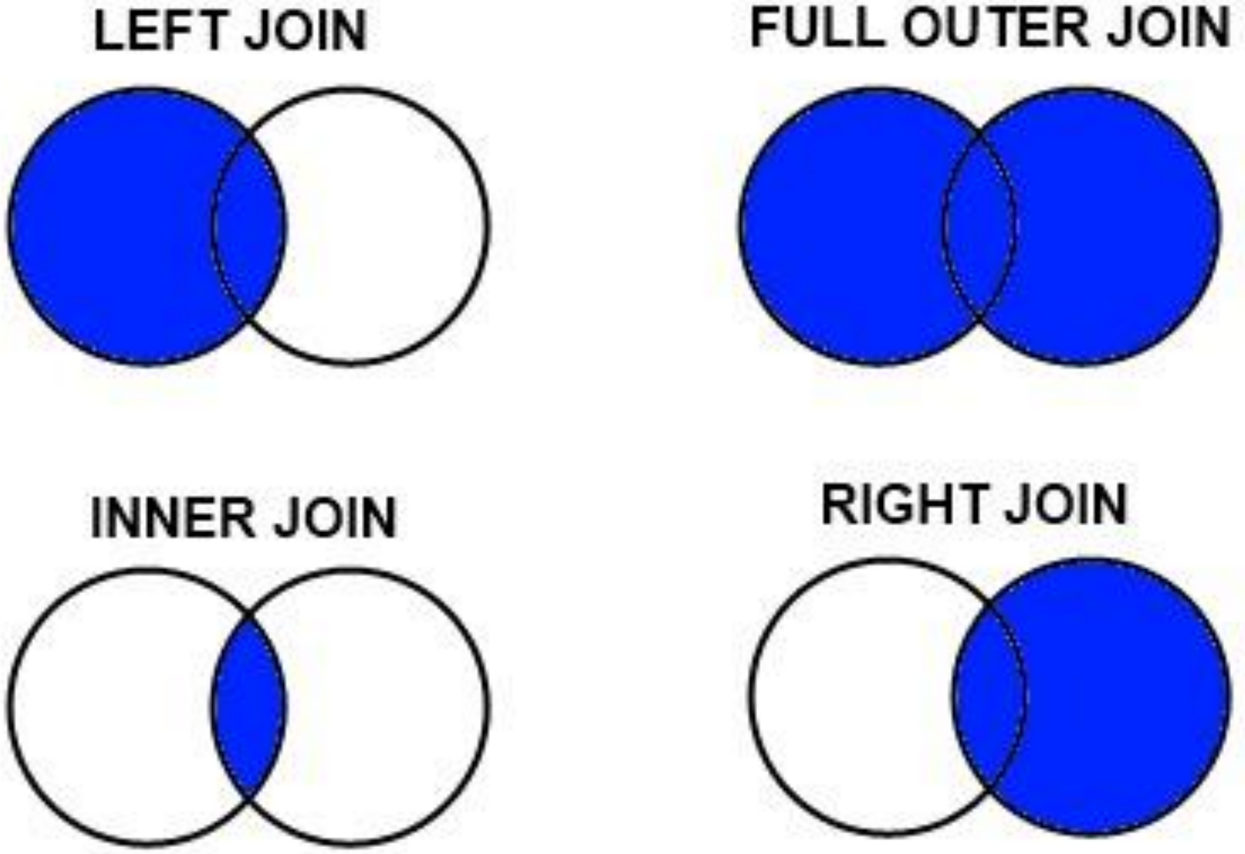
- Merging data in pandas
- Different types of ‘join’

The **merge( )** function

- Use **pd.merge( )** to merge two dataframes
- Must have a column in common
- Specify dataframes with: **left=, right=,**
- Specify column with: **left_on=, right_on=,**
- Specify type of join with: **how=**

joins

- **how=‘inner’** - keeps the first dataframe
- **how=‘outer’** - keeps the second dataframe
- **how=‘left’** - keeps the first dataframe
- **how=‘right’** - keeps the second dataframe



https://stackoverflow.com/questions/38549/what-is-the-difference-between-inner-join-and-outer-join

Example

```
newdf = pd.merge(left=requestsdata,
  right=foidata,
  left_on="Government body",
  right_on="Government body",
  how="inner")
```

The .append( ) method

- Use **DATAFRAMENAME.append( )** to append a second dataframe to a first (e.g. different periods)
- Assign to a variable to keep results
- Make sure dataframes include data on period so you can distinguish, e.g do this first for each: **DATAFRAMENAME[‘period’] = ‘2022’**

Example

```
df2021[‘period’] = 2021

df2022[‘period’] = 2022

bothdfs = df2021.append(df2022)
```

Key points

- Use pd.**merge** to combine dataframes in order to add context such as population, etc.
- Different **joins** will result in different data being discarded (or not) based on whether it matches
- Add **.append** to a dataframe to add other dataframes from different periods underneath