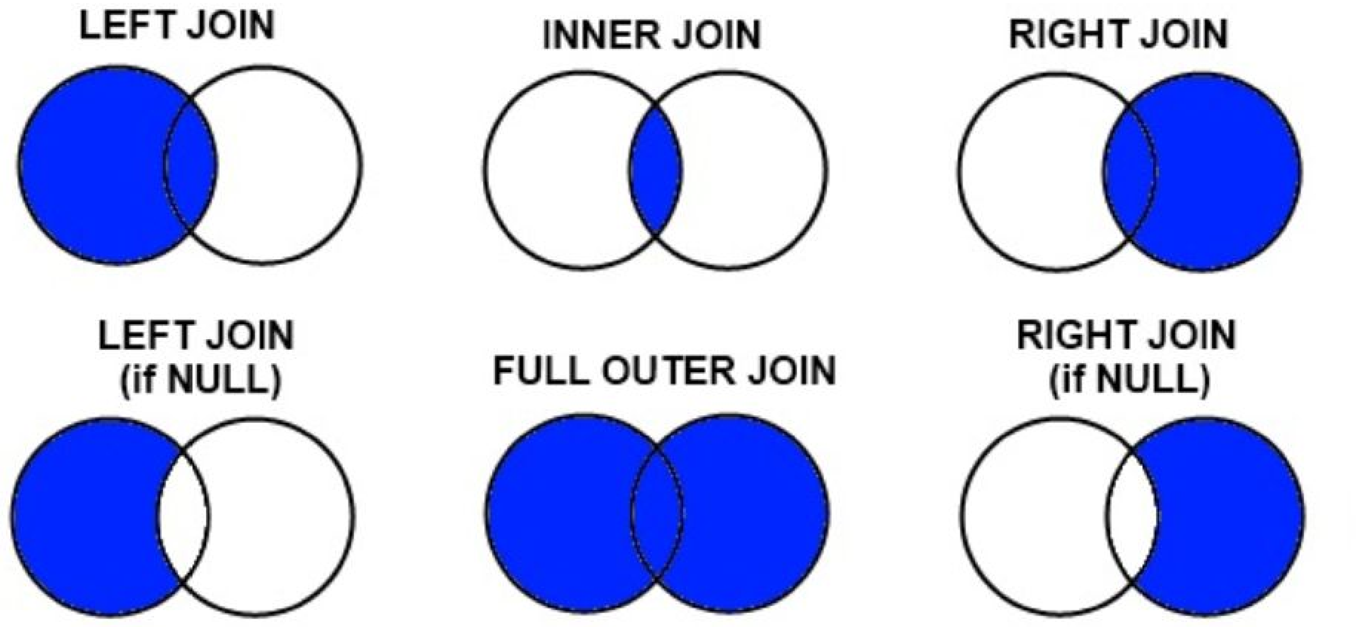
T5 - Data analysis techniques and methodologies



- Merging on Dataframes Columns

We can merge Dataframes N:1 and N:N
pandas.merge(<Dataframe_1>, ..., <Dataframe_n>) -> Looks for strictly coincidences index and labels
pandas.merge(<Dataframes>, on= <ColumnLabels>) -> Label exact coincidence values, not indexes.
pandas.merge(<dfs>, on = <CoLabels>, how= {‘inner’, ‘right’, ‘left’, ‘outer’ }) -> Order to merge:
    inner: for labels, after indexes df1, …, after indexes dfn. Default value for merge
    right: for Colabels, after labels dfn, …, after labels df1. NaN not permitted on right dfs labels
    left: for CoLabels, after labels df1, …, after labels dfn. NaN not permitted on left dfs labels
    outer: same as inned, but permits NaN for any non combination.
pandas.merge(…., suffixes=<suffix list for labels not in CoLabels>)

- Merging on DataFrames Indexes

Merge index to index -> **left_index = True, right_index = True**
Merge label with index -> left_on = < list of labels>, right_index = True
                                        |-> right_on=< list of labels>, left_index = True

- ## Joining Dataframes with same indexes

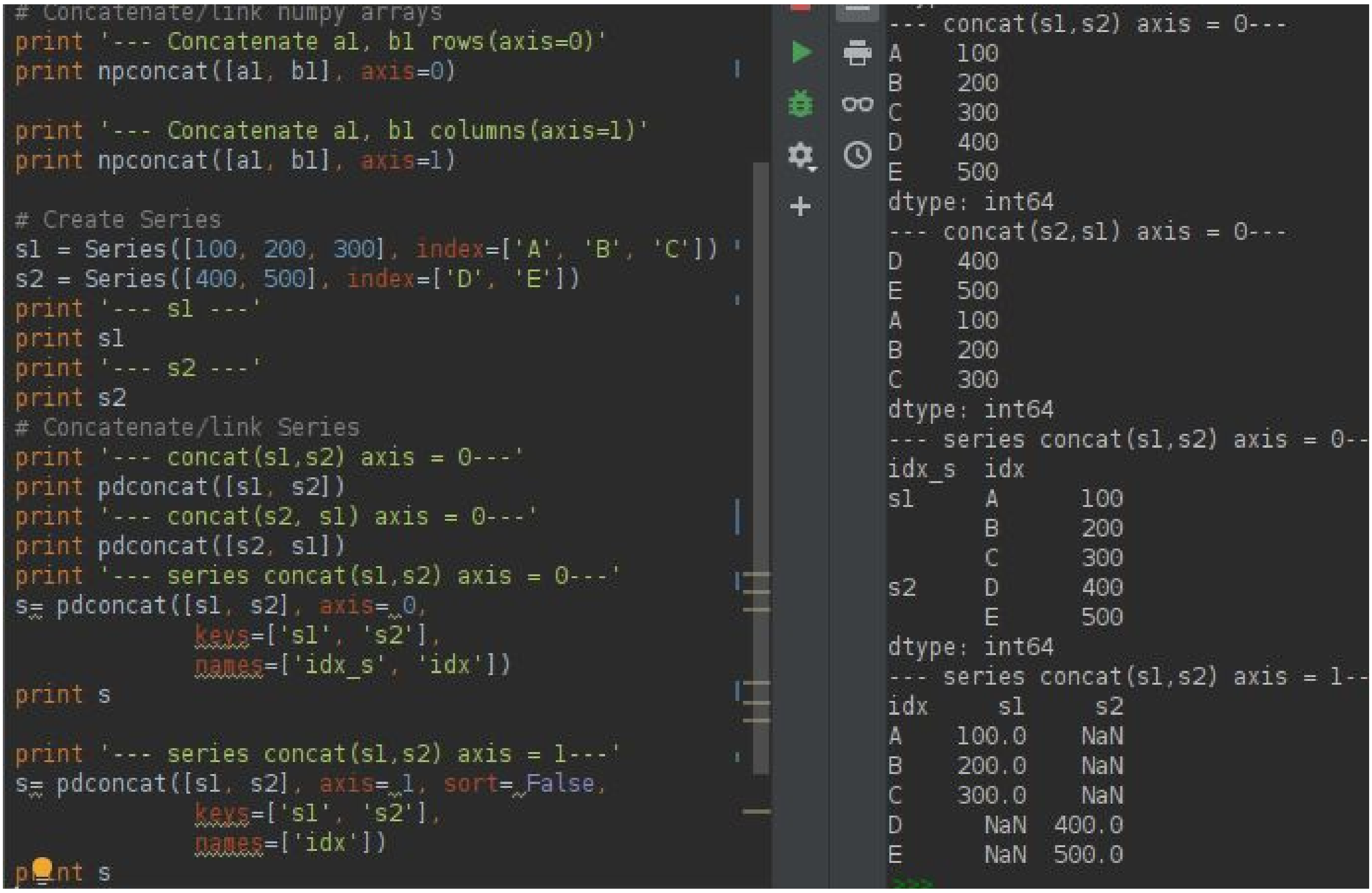
Joining dataframe to other dataframe: adding combinations and columns for items:

Example1: **df1.join(df2)**

df1	**data**	**df2**	**profit**	**df1.j(df2)**	**data**	**profit**
O	0	**O**	10	**L**	2	NaN
U	1	**O**	20	**O**	0	10.0
L	2	**U**	20	**O**	0	20.0
O	3			**O**	3	10.0
U	4			**O**	3	20.0
				U	1	20.0
				U	4	20.0

- ## Concatenation of Series and Dataframes

Concatenate/link them along specific access: **axis = 0** -> rows, **axis = 1** -> columns
For new behaviour (not sort by default) on **axis = 1** -> **sort = False**
Label indexes by **names=**
Set subindexes by **keys=**



```
# Concatenate/link numpy arrays
print '--- Concatenate a1, b1 rows(axis=0)'
print npconcat([a1, b1], axis=0)

print '--- Concatenate a1, b1 columns(axis=1)'
print npconcat([a1, b1], axis=1)

# Create Series
s1 = Series([100, 200, 300], index=['A', 'B', 'C'])
s2 = Series([400, 500], index=['D', 'E'])
print '--- s1 ---'
print s1
print '--- s2 ---'
print s2
# Concatenate/link Series
print '--- concat(s1,s2) axis = 0---'
print pdconcat([s1, s2])
print '--- concat(s2, s1) axis = 0---'
print pdconcat([s2, s1])
print '--- series concat(s1,s2) axis = 0---'
s= pdconcat([s1, s2], axis= 0,
             keys=['s1', 's2'],
             names=['idx_s', 'idx'])
print s

print '--- series concat(s1,s2) axis = 1---'
s= pdconcat([s1, s2], axis= 1, sort= False,
             keys=['s1', 's2'],
             names=['idx'])
print s
```

```
--- concat(s1,s2) axis = 0---
A    100
B    200
C    300
D    400
E    500
dtype: int64
--- concat(s2,s1) axis = 0---
D    400
E    500
A    100
B    200
C    300
dtype: int64
--- series concat(s1,s2) axis = 0--
idx_s  idx
s1     A      100
       B      200
       C      300
s2     D      400
       E      500
dtype: int64
--- series concat(s1,s2) axis = 1--
idx     s1     s2
A    100.0    NaN
B    200.0    NaN
C    300.0    NaN
D      NaN  400.0
E      NaN  500.0
```