Creating Data Frames from lists

Let us go through the details of creating Data Frames using collections.

- Pandas Data Frame is a two-dimensional labeled array capable of holding attributes of any data type.
- It is similar to multi column excel spreadsheet or a database table.
- We can create Data Frame using list of tuples or list of dicts.
- We can also create Data Frames using data from files. We will have a look at it later.

```
import pandas as pd
```

Note

Creating Pandas Data Frame using list of tuples.

```
sals_ld = [(1, 1500.0), (2, 2000.0, 10.0), (3, 2200.00)]
```

```
sals_df = pd.DataFrame(sals_ld)
```

```
sals_df
```

	0	1	2
0	1	1500.0	NaN
1	2	2000.0	10.0
2	3	2200.0	NaN

```
sals_df = pd.DataFrame(sals_ld, columns=['id', 'sal', 'comm'])
```

```
sals_df
```

	id	sal	comm
0	1	1500.0	NaN
1	2	2000.0	10.0
2	3	2200.0	NaN

```
sals_df['id']
```

```
0    1
1    2
2    3
Name: id, dtype: int64
```

```
sals_df[['id', 'sal']]
```

	id	sal
0	1	1500.0
1	2	2000.0
2	3	2200.0

Note

Creating Pandas Data Frame using list of dicts.

```
sals_ld = [
    {'id': 1, 'sal': 1500.0},
    {'id': 2, 'sal': 2000.0},
    {'id': 3, 'sal': 2200.0}
]
```

Note

Column names will be inherited automatically using keys from the dict.

```
sals_df = pd.DataFrame(sals_ld)
```

```
sals_df
```

	id	sal
0	1	1500.0
1	2	2000.0
2	3	2200.0

```
sals_df['id']
```

```
0    1
1    2
2    3
Name: id, dtype: int64
```

```
sals_ld = [
    {'id': 1, 'sal': 1500.0},
    {'id': 2, 'sal': 2000.0, 'comm': 10},
    {'id': 3, 'sal': 2200.0}
]
```

```
pd.DataFrame?
```

```
Init signature:
pd.DataFrame(
    data=None,
    index:Union[Collection, NoneType]=None,
    columns:Union[Collection, NoneType]=None,
    dtype:Union[_ForwardRef('ExtensionDtype'), str, numpy.dtype, Type[Union[str, float, int,
complex]]], NoneType]=None,
    copy:bool=False,
)
Docstring:
Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns).
Arithmetic operations align on both row and column labels. Can be
thought of as a dict-like container for Series objects. The primary
pandas data structure.

Parameters
----------
data : ndarray (structured or homogeneous), Iterable, dict, or DataFrame
    Dict can contain Series, arrays, constants, or list-like objects.

    .. versionchanged:: 0.23.0
       If data is a dict, column order follows insertion-order for
       Python 3.6 and later.

    .. versionchanged:: 0.25.0
       If data is a list of dicts, column order follows insertion-order
       for Python 3.6 and later.

index : Index or array-like
    Index to use for resulting frame. Will default to RangeIndex if
    no indexing information part of input data and no index provided.
columns : Index or array-like
    Column labels to use for resulting frame. Will default to
    RangeIndex (0, 1, 2, ..., n) if no column labels are provided.
dtype : dtype, default None
    Data type to force. Only a single dtype is allowed. If None, infer.
copy : bool, default False
    Copy data from inputs. Only affects DataFrame / 2d ndarray input.

See Also
--------
DataFrame.from_records : Constructor from tuples, also record arrays.
DataFrame.from_dict : From dicts of Series, arrays, or dicts.
read_csv : Read a comma-separated values (csv) file into DataFrame.
read_table : Read general delimited file into DataFrame.
read_clipboard : Read text from clipboard into DataFrame.

Examples
--------
Constructing DataFrame from a dictionary.

>>> d = {'col1': [1, 2], 'col2': [3, 4]}
>>> df = pd.DataFrame(data=d)
>>> df
   col1  col2
0     1     3
1     2     4

Notice that the inferred dtype is int64.

>>> df.dtypes
col1    int64
col2    int64
dtype: object

To enforce a single dtype:

>>> df = pd.DataFrame(data=d, dtype=np.int8)
>>> df.dtypes
col1    int8
col2    int8
dtype: object

Constructing DataFrame from numpy ndarray:

>>> df2 = pd.DataFrame(np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]),
...                    columns=['a', 'b', 'c'])
>>> df2
   a  b  c
0  1  2  3
1  4  5  6
2  7  8  9
File:           /opt/anaconda3/envs/beakerx/lib/python3.6/site-packages/pandas/core/frame.py
Type:           type
Subclasses:     SubclassedDataFrame
```

```
sals_ld
```

```
[{'id': 1, 'sal': 1500.0},
 {'id': 2, 'sal': 2000.0, 'comm': 10},
 {'id': 3, 'sal': 2200.0}]
```

```
sals_df = pd.DataFrame(sals_ld)
```

```
sals_df
```

	id	sal	comm
0	1	1500.0	NaN
1	2	2000.0	10.0
2	3	2200.0	NaN