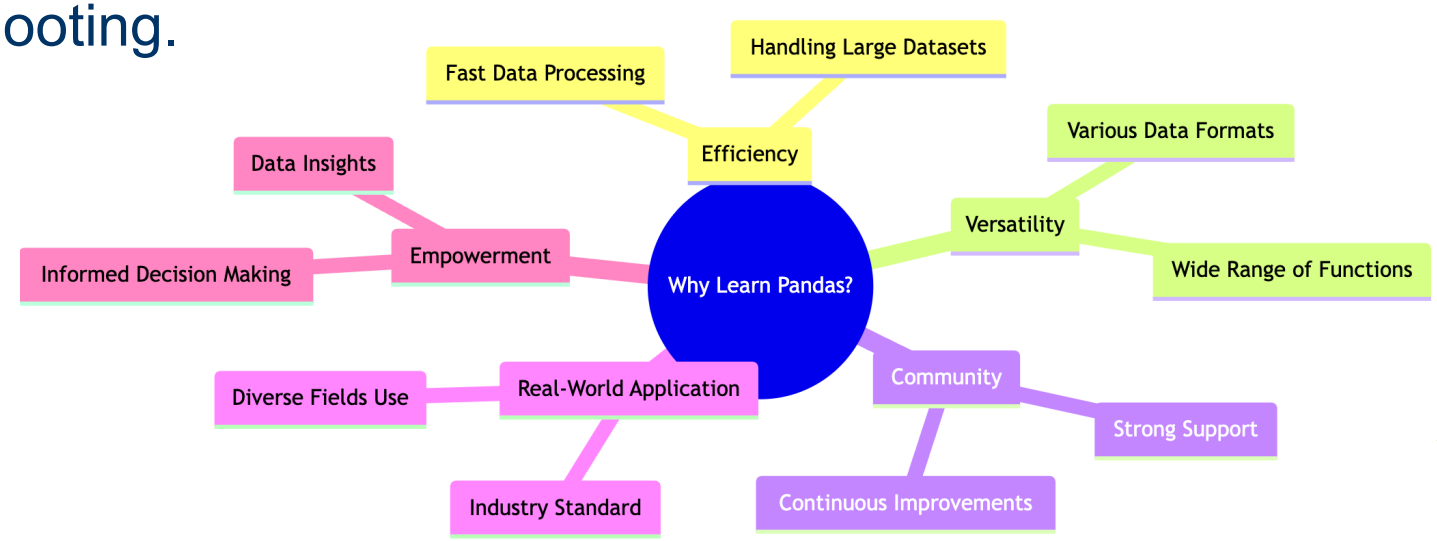
Pandas: Basics

Why Pandas?

- Versatility in Data Handling: Effortlessly manage and manipulate large datasets.
- Time Efficiency: Accelerate data analysis with intuitive functions.
- Data Cleaning & Preparation: Streamline the process of cleaning and preparing data for analysis.
- Real-World Applications: Widely used in industries for data analysis in diverse fields including financial modeling, etc.
- Community Support: Benefit from an active community for learning and troubleshooting.



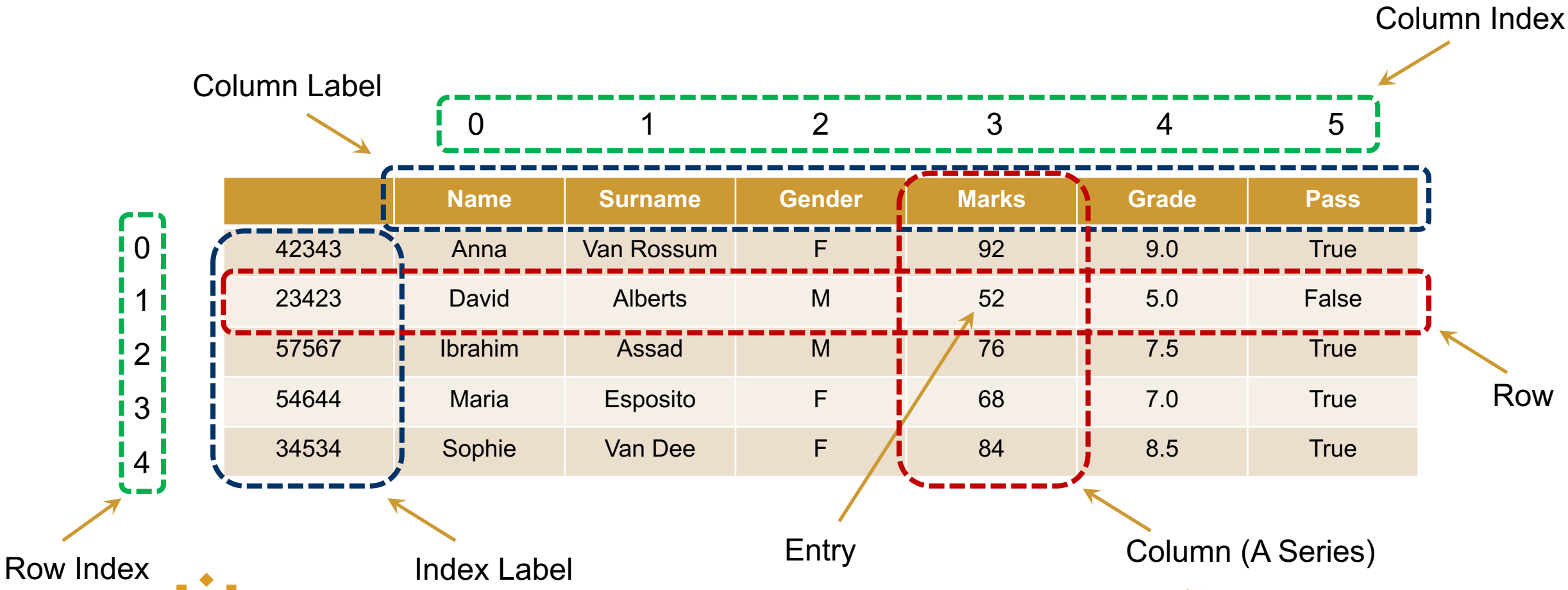
Learning Pandas makes you a better data scientist.

What is Pandas?

- Pandas is a Python package for fast, easy, and intuitive processing of big tabular data.
- The basic data structures in Pandas are DataFrames.
- Unlike NumPy arrays, Pandas DataFrames allow for storing mixture of variables with different data types. Thus, they are more suitable for handling real-world messy data.

DataFrames in Pandas

- DataFrames are the primary data structure in Pandas.



Creating Pandas Dataframes

1) From NumPy arrays
2) From Python lists
3) From Python dictionaries
4) From files

Creating DataFrames

- From a NumPy arrays:
  - The data is stored in a NumPy array
    - Up to 2D arrays can be used
    - Elements should have the same datatype
  - Index label is defined as a list
  - Column label is defined as a list

```
# creating the Numpy array
grades = numpy.array([[92, 9.0],[52, 5.0],[76, 7.5],[68, 7.0],[84, 8.5]])

# creating a list of row names
student_numbers = [42343, 23423, 57567,54644, 34534]

# creating a list of column names
column_labels = ['Marks', 'Grade']

# creating the dataframe
df = pandas.DataFrame(data = grades,
                      index = student_numbers,
                      columns = column_labels)
```

Creating DataFrames

- From a Python list:
  - The data in the table is defined as a list of lists
    - Each inner list is a row in the table
  - It can handle a mixture of datatypes

```
# creating list
grades = [['Anna', 'van Rossum', 'F', 92, 9.0, True],
          ['David', 'Alberts', 'M', 52, 5.0, False],
         ['Ibrahim', 'Assad', 'M', 76, 7.5, True],
         ['Maria', 'Esposito', 'F', 68, 7.0, True],
         ['Sophie', 'van Dee', 'F', 84, 8.5, True]]

# creating a list of row names
student_numbers = [42343, 23423, 57567,54644, 34534]

# creating a list of column names
column_labels = ['Name', 'Suname', 'Gender', 'Marks', 'Grade', 'Pass']

# creating the dataframe
df = pandas.DataFrame(data = grades,
                      index = student_numbers,
                      columns = column_labels)
```

Creating DataFrames

- From a dictionary of lists:
  - Column labels are used as keys of the dictionary
  - The values are lists of entries in each column

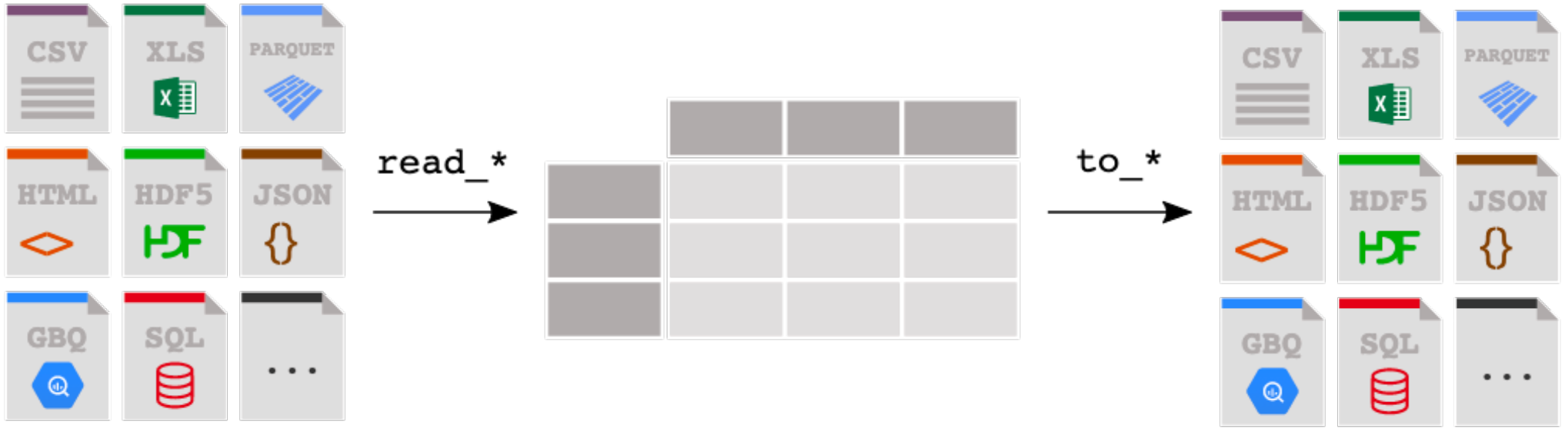
```
ADP_grades = {'Name' : ['Anna', 'David', 'Ibrahim', 'Maria', 'Sophie'],
              'Surname' : ['van Rossum', 'Alberts', 'Assad', 'Esposito', 'van Dee'],
              'Gender' : ['F','M','M','F','F'],
             'Marks': [92, 52, 76, 68, 84],
             'Grade': [9.0, 5.0, 7.5, 7, 8.5],
             'Pass': [True, False, True, True, True]}

# creating a list of row names
student_numbers = [42343, 23423, 57567,54644, 34534]

ADP_grades = pandas.DataFrame(ADP_grades, index=student_numbers)
```

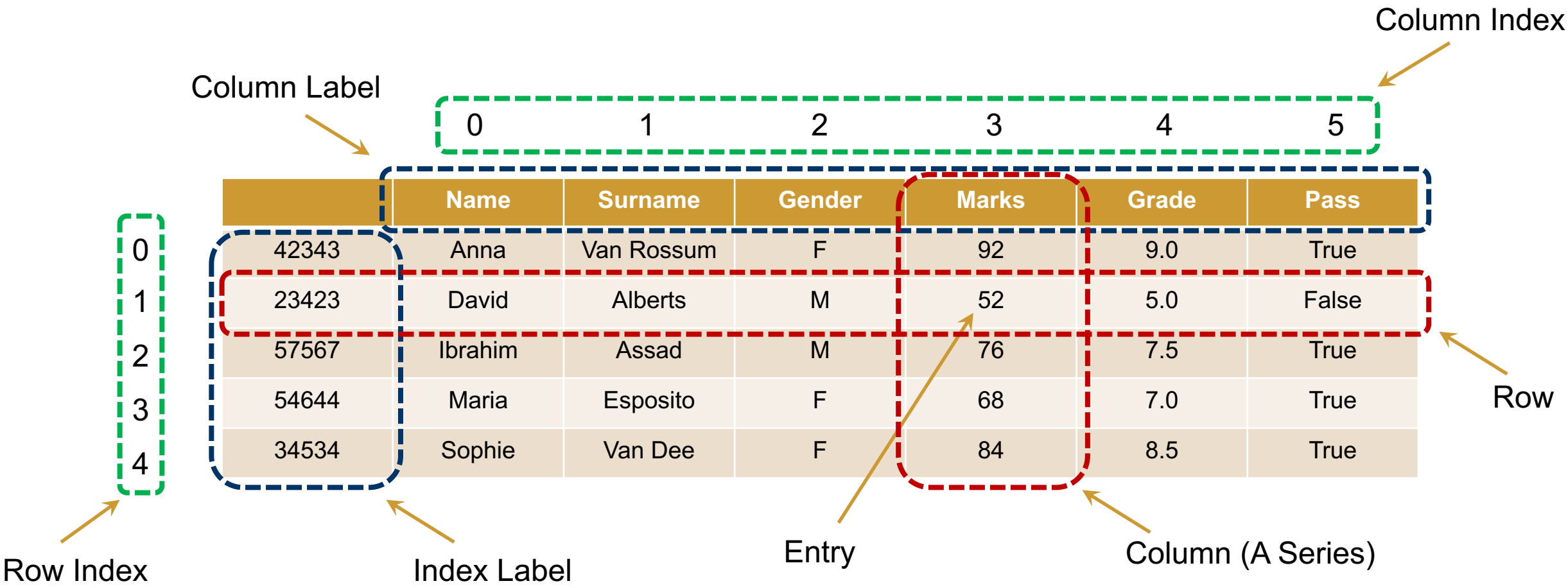
Creating DataFrames

- From a file:



Questions?

DataFrames Indexing in Pandas



Indexing DataFrames

1. Attribute indexing: Selecting a column using column label as an attribute

```
# Attribute Indexing
ADP_grades.Grade
```

```
42343    9.0
23423    5.0
57567    7.5
54644    7.0
34534    8.5
Name: Grade, dtype: float64
```

Indexing DataFrames

2. NumPy Style Selection:

i) List of column labels

```
# Numpy style Indexing for columns
ADP_grades[['Marks', 'Grade']]
```

	Marks	Grade
42343	92	9.0
23423	52	5.0
57567	76	7.5
54644	68	7.0
34534	84	8.5

ii) Slice of row indices

```
# Numpy style Indexing for rows
ADP_grades[2:]
```

	Name	Suname	Gender	Marks	Grade	Pass
57567	Ibrahim	Assad	M	76	7.5	True
54644	Maria	Esposito	F	68	7.0	True
34534	Sophie	van Dee	F	84	8.5	True

Indexing DataFrames

3. Selection using labels (using loc attribute):

```
# Label Indexing for rows
ADP_grades.loc[57567:54644]
```

	Name	Suname	Gender	Marks	Grade	Pass
57567	Ibrahim	Assad	M	76	7.5	True
54644	Maria	Esposito	F	68	7.0	True

```
# Label Indexing for rows and columns
ADP_grades.loc[57567:54644,['Name', 'Grade', 'Pass']]
```

	Name	Grade	Pass
57567	Ibrahim	7.5	True
54644	Maria	7.0	True

```
# Label Indexing for columns
ADP_grades.loc[:,['Name', 'Grade', 'Pass']]
```

	Name	Grade	Pass
42343	Anna	9.0	True
23423	David	5.0	False
57567	Ibrahim	7.5	True
54644	Maria	7.0	True
34534	Sophie	8.5	True

Indexing DataFrames

4. Selection using location (using iloc attribute):

```
# Selection by index for rows
ADP_grades.iloc[3:5]
```

	Name	Suname	Gender	Marks	Grade	Pass
54644	Maria	Esposito	F	68	7.0	True
34534	Sophie	van Dee	F	84	8.5	True

```
# Selection by index for columns
ADP_grades.iloc[:, 2:4]
```

	Gender	Marks
42343	F	92
23423	M	52
57567	M	76
54644	F	68
34534	F	84

```
# Selection by index for rows and columns
ADP_grades.iloc[3:5, [1,3,5]]
```

	Suname	Marks	Pass
54644	Esposito	68	True
34534	van Dee	84	True

Indexing DataFrames

5. Selection by random sampling

```
#Selection by sampling
ADP_grades.sample(3)
```

	Name	Suname	Gender	Marks	Grade	Pass
42343	Anna	van Rossum	F	92	9.0	True
34534	Sophie	van Dee	F	84	8.5	True
57567	Ibrahim	Assad	M	76	7.5	True

Questions?

Thanks!