PANDAS

PRESENTED BY:
NITISH VIG

Agenda

ICTACADEMY

Pandas Basics

Series

DataFrame

Pandas Operations

Pandas Queries

Pandas Joins and Intersections

PANDAS

Presented By : Nitish Vig

Basics

ICTACADEMY

Pandas is the only library that is used to perform data manipulation and analysis.

Pandas is free software released under the three-clause BSD license.

It was developed by Wes McKinney in 2008.

It is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

It is built on Numpy library, ie., it requires this library to perform actions.

It is installed using : *pip3 install pandas*

*import **pandas** as pd*

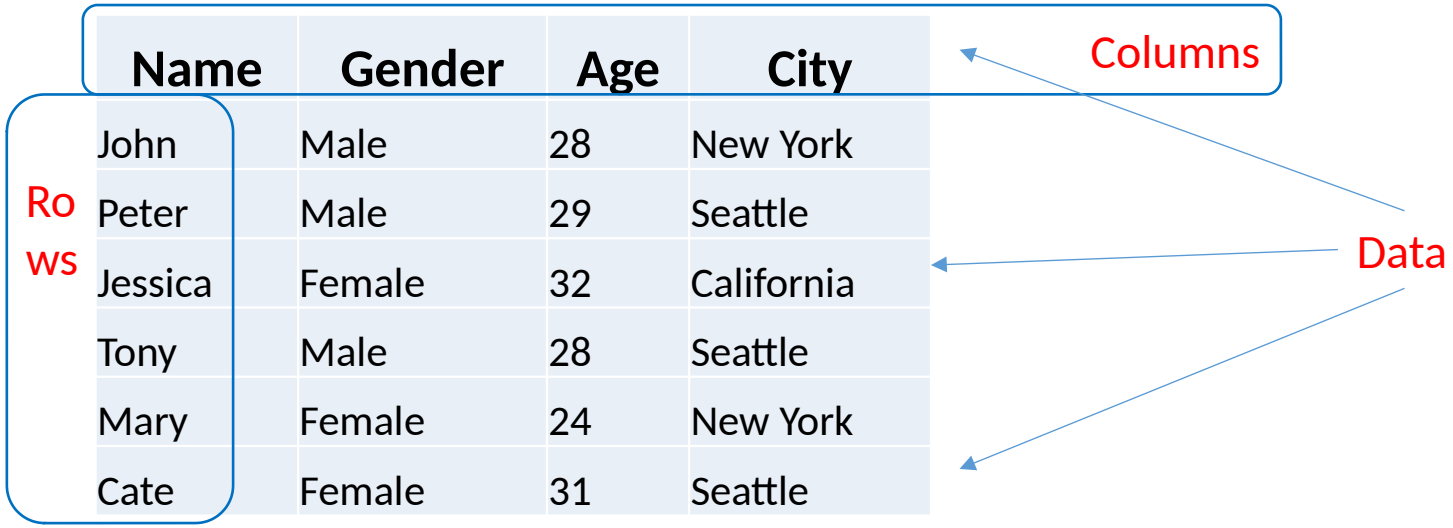
DATAFRAMES

Basics

Syntax : *pandas.DataFrame(data, index, columns, dtype, copy)*

DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns).

Pandas DataFrame consists of three principal components, the **data**, **rows**, and **columns**.



OPERATIONS

Opening files..

ICTACADEMY

Syntax : *pandas.read_excel('fileName.xlsx', sheet_name="sheetName") ;*
pandas.read_csv('fileName.csv', sep="separator")

This command creates a dataframe out of the files.

```
df = pd.read_excel('StudentsPerformance.xlsx', sheet_name="StudentsPerformance")
```

```
df = pd.read_csv('winequality_red.csv', sep = ';')
```

Reading the top rows

ICTACADEMY

Syntax : *dataframe.head(integerNumber)*

This command helps to read top rows in dataframe. Default value of integerNumber = 1.

```
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

Information about DataFrame

ICTACADEMY

Syntax : *dataframe.info ()*

This command displays information about dataset such as, datatype, size, rows, columns and non-null values.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
gender                                  1000 non-null object
race/ethnicity                          1000 non-null object
parental level of education             1000 non-null object
lunch                                   1000 non-null object
test preparation course                 1000 non-null object
math score                              1000 non-null int64
reading score                           1000 non-null int64
writing score                           1000 non-null int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

Description about DataFrame

ICTACADEMY

Syntax : *dataframe.describe (percentiles=None, include=None, exclude=None)*

This command displays statistical values of the dataset.
We can include different percentiles, or include / exclude columns.

```
df.describe(percentiles=[.5, 0.33], include=[np.number])
```

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
33%	60.00000	63.000000	62.000000
50%	66.00000	70.000000	69.000000
max	100.00000	100.000000	100.000000

Description about DataFrame

ICTACADEMY

Syntax : *dataframe.describe (percentiles=None, include=None, exclude=None)*

```
df.describe(include="all")
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
count	1000	1000	1000	1000	1000	1000.00000	1000.000000	1000.000000
unique	2	5	6	2	2	NaN	NaN	NaN
top	female	group C	some college	standard	none	NaN	NaN	NaN
freq	518	319	226	645	642	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	NaN	66.08900	69.169000	68.054000
std	NaN	NaN	NaN	NaN	NaN	15.16308	14.600192	15.195657
min	NaN	NaN	NaN	NaN	NaN	0.00000	17.000000	10.000000
25%	NaN	NaN	NaN	NaN	NaN	57.00000	59.000000	57.750000
50%	NaN	NaN	NaN	NaN	NaN	66.00000	70.000000	69.000000
75%	NaN	NaN	NaN	NaN	NaN	77.00000	79.000000	79.000000
max	NaN	NaN	NaN	NaN	NaN	100.00000	100.000000	100.000000

Count Rows, Columns in DataFrame

ICTACADEMY

Syntax : *dataframe.shape*

This will show the total number of rows and columns in dataframe.

```
df.shape
```

```
(1000, 8)
```

We can also see the rows and columns individually.

```
df.shape[0]
```

```
1000
```

```
df.shape[1]
```

```
8
```

Column Names in DataFrame

ICTACADEMY

Syntax : *dataframe.columns*

This will show the column names in dataframe.

```
df.columns
```

```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

Rows, columns of DataFrame

Selecting Column in DataFrame

ICTACADEMY

Syntax : *dataframe [colName] ;* *dataframe [[colName1, colName2, ...]]*

This will show the data for the specified column names.

```
df[["race/ethnicity", "reading score", "writing score"]].head()
```

	race/ethnicity	reading score	writing score
0	group B	72	74
1	group C	90	88
2	group B	95	93
3	group A	57	44
4	group C	78	75

Creating a new Column in DataFrame

ICTACADEMY

Syntax : *dataFrame[colName] = df[OPERATION]*

It creates a new column depending on the operation performed on a single or multiple columns.

```
df['total score'] = df["math score"] + df["reading score"] + df["writing score"]
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	total score
0	female	group B	bachelor's degree	standard	none	72	72	74	218
1	female	group C	some college	standard	completed	69	90	88	247
2	female	group B	master's degree	standard	none	90	95	93	278
3	male	group A	associate's degree	free/reduced	none	47	57	44	148
4	male	group C	some college	standard	none	76	78	75	229

Deleting Column in DataFrame

ICTACADEMY

Syntax : *dataframe.drop(columnName, axis, inplace)*

This will delete the specified columns from the dataframe.

Axis = 1, means columns
Axis = 0, means rows (default)

inplace = False (default), means that the column is not permanently deleted
inplace = True, means that the column is permanently deleted

Deleting Column in DataFrame

ICTACADEMY

Syntax : *dataframe.drop(columnName, axis, inplace)*

```
df.drop('total score', axis = 1)
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	total score
0	female	group B	bachelor's degree	standard	none	72	72	74	218
1	female	group C	some college	standard	completed	69	90	88	247
2	female	group B	master's degree	standard	none	90	95	93	278
3	male	group A	associate's degree	free/reduced	none	47	57	44	148
4	male	group C	some college	standard	none	76	78	75	229

Deleting Column in DataFrame

ICTACADEMY

Syntax : *dataframe.drop(columnName, axis, inplace)*

```
df.drop('total score', axis = 1, inplace = True)
df.head()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-109-3af03d7918a9> in <module>
----> 1 df.drop('total score', axis = 1, inplace = True)
      2 df.head()

~\Anaconda3\lib\site-packages\pandas\core\frame.py in drop(self, labels, axis, index, columns, level, inplace, error
s)
   3938                                            index=index, columns=columns,
   3939                                            level=level, inplace=inplace,
-> 3940                                            errors=errors)
   3941
   3942     @rewrite_axis_style_signature('mapper', [('copy', True),

~\Anaconda3\lib\site-packages\pandas\core\generic.py in drop(self, labels, axis, index, columns, level, inplace, erro
rs)
   3778         for axis, labels in axes.items():
   3779             if labels is not None:
-> 3780                 obj = obj._drop_axis(labels, axis, level=level, errors=errors)
   3781
   3782         if inplace:

~\Anaconda3\lib\site-packages\pandas\core\generic.py in _drop_axis(self, labels, axis, level, errors)
   3810                 new_axis = axis.drop(labels, level=level, errors=errors)
   3811             else:
-> 3812                 new_axis = axis.drop(labels, errors=errors)
   3813             result = self.reindex(**{axis_name: new_axis})
   3814
```

Deleting Rows in DataFrame

ICTACADEMY

Syntax : *dataframe.drop(rowName, axis, inplace)*

This will delete the specified rows from the dataframe.

Axis = 1, means columns
Axis = 0, means rows (default)

inplace = False (default), means that the column is not permanently deleted
inplace = True, means that the column is permanently deleted

Deleting Column in DataFrame

ICTACADEMY

Syntax : *dataframe.drop(rowName, axis, inplace)*

```
df.drop(5, inplace=True)
```

```
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
1	female	group B	bachelor's degree	standard	none	72	72	74
2	female	group C	some college	standard	completed	69	90	88
3	female	group B	master's degree	standard	none	90	95	93
4	male	group A	associate's degree	free/reduced	none	47	57	44
6	female	group B	associate's degree	standard	none	71	83	78

Slicing DataFrame

ICTACADEMY

Syntax : *dataframe.loc(<rowName selection>, <columnName selection>)*
dataframe.iloc(<rowName selection>, <columnName selection>)

Both “loc” and “iloc” are used for slicing operation in dataframe.

“loc” works when you know the row or column name.
“iloc” works when you know the row or column position number.

If any selection is not mentioned, then by default, staring point is to be considered.

Slicing DataFrame

ICTACADEMY

Syntax : *dataframe.loc(<rowName selection>, <columnName selection>) ;*
dataframe.iloc(<rowName selection>, <columnName selection>)

```
df.loc[5 : 12, ['lunch', "writing score"]]
```

	lunch	writing score
5	standard	75
6	standard	78
7	standard	92
8	free/reduced	39
9	free/reduced	67
10	free/reduced	50
11	standard	52
12	standard	43

```
df.iloc[5 : 12, 2 : 4]
```

	parental level of education	lunch
6	associate's degree	standard
7	some college	standard
8	some college	free/reduced
9	high school	free/reduced
10	high school	free/reduced
11	associate's degree	standard
12	associate's degree	standard

Slicing using iloc

ICTACADEMY

Syntax : *dataframe.iloc(<rowName selection>, <columnName selection>)*

```
df = pd.read_excel('StudentsPerformance.xlsx', sheet_name="StudentsPerformance", index_col=0)
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
1	female	group B	bachelor's degree	standard	none	72	72	74
2	female	group C	some college	standard	completed	69	90	88
3	female	group B	master's degree	standard	none	90	95	93
4	male	group A	associate's degree	free/reduced	none	47	57	44
5	male	group C	some college	standard	none	76	78	75

1

2

Slicing using iloc

ICTACADEMY

Syntax : *dataframe.iloc(<rowName selection>, <columnName selection>)*

1

```
df.iloc[0]
```

```
gender                         female
race/ethnicity                 group B
parental level of education    bachelor's degree
lunch                          standard
test preparation course            none
math score                           72
reading score                        72
writing score                        74
Name: 1, dtype: object
```

2

```
df.iloc[3, 3]
```

```
'free/reduced'
```

Slicing using iloc

ICTACADEMY

Syntax : *dataframe.loc(<rowName selection>, <columnName selection>)*

```
df = pd.read_excel('StudentsPerformance.xlsx', sheet_name="StudentsPerformance", index_col=0)
df.head()
```

1

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
1	female	group B	bachelor's degree	standard	none	72	72	74
2	female	group C	some college	standard	completed	69	90	88
3	female	group B	master's degree	standard	none	90	95	93
4	male	group A	associate's degree	free/reduced	none	47	57	44
5	male	group C	some college	standard	none	76	78	75

2

Slicing using iloc

ICTACADEMY

Syntax : *dataframe.loc(<rowName selection>, <columnName selection>)*

1

```
df.loc[ : 8 ,['reading score', "parental level of education"]]
```

	reading score	parental level of education
1	72	bachelor's degree
2	90	some college
3	95	master's degree
4	57	associate's degree
5	78	some college
6	83	associate's degree
7	95	some college
8	43	some college

2

```
df.loc[2 ,'reading score']
```

```
90
```

QUERIES

Retrieve Rows

Syntax : *dataframe[dataframe[columnName].condition's]*

It retrieve rows that return True for the following condition.

It will show null, if condition is False.

The data is shown for all rows against it's column names

Retrieve Rows

ICTACADEMY

DataSet :

```
df = pd.read_excel('StudentsPerformance.xlsx', sheet_name="StudentsPerformance", index_col=0)
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
1	female	group B	bachelor's degree	standard	none	72	72	74
2	female	group C	some college	standard	completed	69	90	88
3	female	group B	master's degree	standard	none	90	95	93
4	male	group A	associate's degree	free/reduced	none	47	57	44
5	male	group C	some college	standard	none	76	78	75

Retrieve Rows

ICTACADEMY

Retrieve data for which column “lunch” has value ‘standard’.

```
df[df['lunch'] == 'standard'].head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
1	female	group B	bachelor's degree	standard	none	72	72	74
2	female	group C	some college	standard	completed	69	90	88
3	female	group B	master's degree	standard	none	90	95	93
5	male	group C	some college	standard	none	76	78	75
6	female	group B	associate's degree	standard	none	71	83	78

Retrieve Rows

ICTACADEMY

Retrieve data for which column “math score” has value ‘ > 70’.

```
df[df['math score'] > 70].count()
```

```
gender                         391
race/ethnicity                 391
parental level of education    391
lunch                          391
test preparation course        391
math score                     391
reading score                  391
writing score                  391
dtype: int64
```

Retrieve Rows

ICTACADEMY

Retrieve data for columns [‘math score’, ‘reading score’, “writing score”], for which column “test preparation course” has value ‘completed’.

```
df[["math score", "reading score", "writing score"]][df['test preparation course'] == 'completed'].head()
```

	math score	reading score	writing score
2	69	90	88
7	88	95	92
9	64	64	67
14	78	72	70
19	46	42	46

Retrieve Rows

ICTACADEMY

Retrieve data for column ‘gender’, for which column “race/ethnicity” has value ‘group A’.

```
df["gender"][df['race/ethnicity'] == 'group A'].value_counts()
```

```
male      53
female    36
Name: gender, dtype: int64
```

ICTACADEMY

THANK YOU

Reach out to me at: **nitish@ictacademy.in**