4.0 Selecting Values from Dataframe/Series

May 19, 2019 5:18 PM

Since dataframe has column labels and row labels, data from dataframe could be extracted using column labels & row labels or using row & column index

Selecting using column and row labels: dataframe.loc [row label, column label]

Selecting using column and row index: dataframe.iloc [row index, column index]

Series only have one column, so we only need to specify the row label or row index

Row/column label or row index/column index can be one of
- A single label/index
- A list or array of labels/index
- A slice object with labels/index
- A Boolean array
    - Boolean array is discussed later

- Different label selection methods:

Select by Label	Explicit Syntax	Shorthand Convention
Single column from dataframe	df.loc[:,"col1"]	df["col1"]
List of columns from dataframe	df.loc[:,["col1","col7"]]	df[["col1","col7"]]
Slice of columns from dataframe	df.loc[:,"col1":"col4"]	
Single row from dataframe	df.loc["row4"]	
List of rows from dataframe	df.loc[["row1", "row8"]]	
Slice of rows from dataframe	df.loc["row3":"row5"]	df["row3":"row5"]
Single item from series	s.loc["item8"]	s["item8"]
List of items from series	s.loc[["item1","item7"]]	s[["item1","item7"]]
Slice of items from series	s.loc["item2":"item4"]	s["item2":"item4"]

Not sure why shorthand convention is available for some of the desired selection method.

Also note that, shorthand convention is only viable when we need certain rows but all columns or vice versa

There is also an alternate shorthand but it is not commonly used

Select by integer position	Explicit Syntax	Shorthand Convention
Single column from dataframe	`df.iloc[:,3]`	
List of columns from dataframe	`df.iloc[:,[3,5,6]]`	
Slice of columns from dataframe	`df.iloc[:,3:7]`	
Single row from dataframe	`df.iloc[20]`	
List of rows from dataframe	`df.iloc[[0,3,8]]`	
Slice of rows from dataframe	`df.iloc[3:5]`	`df[3:5]`
Single items from series	`s.iloc[8]`	`s[8]`
List of item from series	`s.iloc[[2,8,1]]`	`s[[2,8,1]]`
Slice of items from series	`s.iloc[5:10]`	`s[5:10]`

For sanity's sake, I might use explicit syntax all the time that means including column labels and row labels explicitly.

For example, for selecting a single row,

Column Labels & Row Labels: df.loc ["row4", :])
Column Index & Row Index: df.iloc [3,:]

Key distinction: When slicing: When using index number, the data for the last index is not included. However, when using labels, the data for the last label is included. This is because when specifying slicing using labels, it is more intuitive to specify up to the column/row we want the data for