

A) Row Selection

▼ Example #1: Extracting single Row

In this example, Name column is made as the index column and then two single rows are extracted one by one in the form of series using index label of rows.

```
# importing pandas package
import pandas as pd

# making data frame from csv file
data = pd.read_csv("/content/drive/MyDrive/nba.csv", index_col = "Name")

# retrieving row by loc method
first = data.loc["Avery Bradley"]
second = data.loc["R.J. Hunter"]

print(first, "\n\n", second)
```

Team	Boston Celtics
Number	0.0
Position	PG
Age	25.0
Height	6-2
Weight	180.0
College	Texas
Salary	7730337.0

Name: Avery Bradley, dtype: object

Team	Boston Celtics
Number	28.0
Position	SG
Age	22.0
Height	6-5
Weight	185.0
College	Georgia State
Salary	1148640.0

Name: R.J. Hunter, dtype: object

▼ Example #2: Multiple parameters

In this example, Name column is made as the index column and then two single rows are extracted at the same time by passing a list as parameter.

```
# importing pandas package
import pandas as pd

# making data frame from csv file
data = pd.read_csv("/content/drive/MyDrive/nba.csv", index_col = "Name")
```

```
# retrieving rows by loc method
rows = data.loc[["Avery Bradley", "R.J. Hunter"]]

# checking data type of rows
print(type(rows))

# display
rows
```

```
<class 'pandas.core.frame.DataFrame'>
```

	Team	Number	Position	Age	Height	Weight	College	Salary
Name								
Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640.0

▼ Example #3: Extracting multiple rows with same index

In this example, Team name is made as the index column and one team name is passed to .loc method to check if all values with same team name have been returned or not.

```
# importing pandas package
import pandas as pd

# making data frame from csv file
data = pd.read_csv("/content/drive/MyDrive/nba.csv", index_col = "Team")

# retrieving rows by loc method
rows = data.loc["Utah Jazz"]

# checking data type of rows
print(type(rows))

# display
rows
```

```
<class 'pandas.core.frame.DataFrame'>
```

	Name	Number	Position	Age	Height	Weight	College	Salary
Team								
Utah Jazz	Trevor Booker	33.0	PF	28.0	6-8	228.0	Clemson	4775000.0
Utah Jazz	Trey Burke	3.0	PG	23.0	6-1	191.0	Michigan	2658240.0
Utah Jazz	Alec Burks	10.0	SG	24.0	6-6	214.0	Colorado	9463484.0
Utah Jazz	Dante Exum	11.0	PG	20.0	6-6	190.0	NaN	3777720.0
Utah Jazz	Derrick Favors	15.0	PF	24.0	6-10	265.0	Georgia Tech	12000000.0

▼ Example #4: Extracting rows between two index labels

In this example, two index label of rows are passed and all the rows that fall between those two index label have been returned (Both index labels Inclusive).

```
# importing pandas package
import pandas as pd

# making data frame from csv file
data = pd.read_csv("/content/drive/MyDrive/nba.csv", index_col = "Name")

# retrieving rows by loc method
rows = data.loc["Avery Bradley":"Isaiah Thomas"]

# checking data type of rows
print(type(rows))

# display
rows
```

```
<class 'pandas.core.frame.DataFrame'>
```

Team Number Position Age Height Weight

College

Salary

B) Row Addition

In Order to add a Row in Pandas DataFrame, we can concat the old dataframe with new one.

```
# importing pandas module
import pandas as pd

# making data frame
df = pd.read_csv("/content/drive/MyDrive/nba.csv")

df.head(10)
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337
1	Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117
2	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	Na
3	R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640
4	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000
5	Amir Johnson	Boston Celtics	90.0	PF	29.0	6-9	240.0	NaN	12000000
6	Jordan Mickey	Boston Celtics	55.0	PF	21.0	6-8	235.0	LSU	1170960
7	Kelly Olynyk	Boston Celtics	41.0	C	25.0	7-0	238.0	Gonzaga	2165160
8	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360
9	Marcus Smart	Boston Celtics	36.0	PG	22.0	6-4	220.0	Oklahoma State	3431040

Example #1: Adding row at the top of given dataframe by concatenating the old dataframe with new one.

```
new_row = pd.DataFrame({'Name':'Geeks', 'Team':'Boston', 'Number':3,
                        'Position':'PG', 'Age':33, 'Height':'6-2',
                        'Weight':189, 'College':'MIT', 'Salary':99999},
                        index =[0])

# simply concatenate both dataframes
df = pd.concat([new_row, df]).reset_index(drop = True)
df.head(5)
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Geeks	Boston	3.0	PG	33.0	6-2	189.0	MIT	99999.0

Example #2: Adding row at the top of given dataframe by concatenating the old dataframe with new one.

4	R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640.0
---	-------------	----------------	------	----	------	-----	-------	---------------	-----------

```
new_row = pd.DataFrame({'Name': 'Geeks', 'Team': 'Boston', 'Number': 3,
                        'Position': 'PG', 'Age': 33, 'Height': '6-2',
                        'Weight': 189, 'College': 'MIT', 'Salary': 99999}, index = [0])
```

```
# Concatenate new_row with df
df = pd.concat([new_row, df[:]]).reset_index(drop = True)
df.head(5)
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Geeks	Boston	3.0	PG	33.0	6-2	189.0	MIT	99999.0
1	Geeks	Boston	3.0	PG	33.0	6-2	189.0	MIT	99999.0
2	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
3	Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117.0
4	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN

C) Row Deletion

In Order to delete a row in Pandas DataFrame, we can use the drop() method. Rows is deleted by dropping Rows by index label.

Pandas provide data analysts a way to delete and filter data frame using .drop() method. Rows or columns can be removed using index label or column name using this method.

Syntax:

```
DataFrame.drop(labels=None, axis=0, index=None, columns=None, level=None,
               inplace=False, errors='raise')
```

Parameters:

labels: String or list of strings referring row or column name.

axis: int or string value, 0 'index' for Rows and 1 'columns' for Columns.

index or columns: Single label or list. index or columns are an alternative to axis and cannot be used together.

level: Used to specify level in case data frame is having multiple level index. inplace: Makes changes in original Data Frame if True.

errors: Ignores error if any value from the list doesn't exists and drops rest of the values when errors = 'ignore'

Return type: Dataframe with dropped values

Example: Dropping Rows by index label

In this code, A list of index labels is passed and the rows corresponding to those labels are dropped using .drop() method.

```
# importing pandas module
import pandas as pd

# making data frame from csv file
data = pd.read_csv("/content/drive/MyDrive/nba.csv", index_col = "Name" )

# dropping passed values
data.drop(["Avery Bradley", "John Holland", "R.J. Hunter",
          "R.J. Hunter"], inplace = True)

# display
data
```

	Team	Number	Position	Age	Height	Weight	College	Salary
Name								
Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117.0
Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
Amir Johnson	Boston Celtics	90.0	PF	29.0	6-9	240.0	NaN	12000000.0
Jordan Mickey	Boston Celtics	55.0	PF	21.0	6-8	235.0	LSU	1170960.0
Kelly Olynyk	Boston Celtics	41.0	C	25.0	7-0	238.0	Gonzaga	2165160.0
...
Shelvin Mack	Utah Jazz	8.0	PG	26.0	6-3	203.0	Butler	2433333.0
Raul Neto	Utah Jazz	25.0	PG	24.0	6-1	179.0	NaN	900000.0
Tibor Pleiss	Utah Jazz	21.0	C	26.0	7-3	256.0	NaN	2900000.0
Jeff Withey	Utah Jazz	24.0	C	26.0	7-0	231.0	Kansas	947276.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

455 rows × 8 columns

