**Python Lab 2: Creating hands-on experience with Pandas library**

1. **Load data from CSV files and understand the data**
2. **Query and index operations on the above data frame**
3. **Insert, delete and update your data**
4. **Apply various filters on the data**
5. **Group, merge, and aggregate data in data frames**
6. **Identify and fix missing values in data**

Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns. CSV stands for “Comma Separated Values.” It is the simplest form of storing data in tabular form as plain text. It is important to know to work with CSV because we mostly rely on CSV data in our day-to-day lives as data scientists.

1. **Load data from CSV files and understand the data**

import pandas as pd

data= pd.read\_csv("Salary\_Data.csv")

print (data)

print (data.columns)

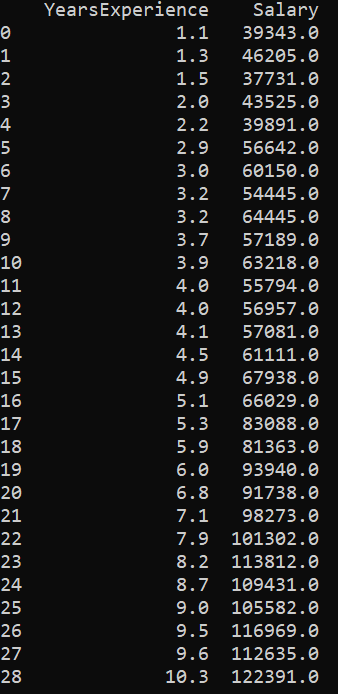
print (data.Salary)

print (data.index)

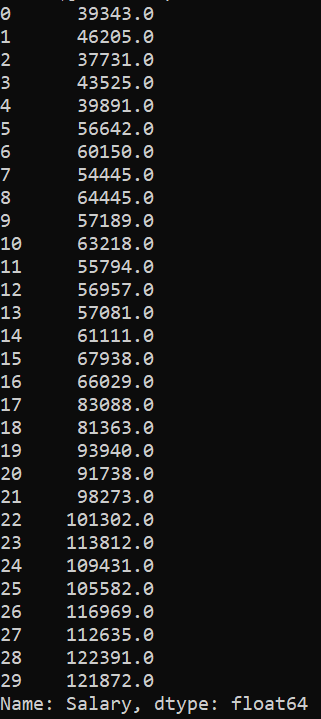
print (data.shape)

print (data.size)

**Output:**













1. **Query and index operations on the above data frame**

There are a lot of ways to pull the elements, rows, and columns from a DataFrame. There is some indexing method in Pandas which help in getting an element from a DataFrame. These indexing methods appear very similar but behave very differently. Pandas support four types of Multi-axes indexing they are:

1. Dataframe.[ ] ; This function also known as indexing operator
2. Dataframe.loc[ ] : This function is used for labels.
3. Dataframe.iloc[ ] : This function is used for positions or integer based

import pandas as pd

data= pd.read\_csv("nba-2.csv", index\_col ="Name")

# retrieving columns by indexing operator

first = data["Age"]

print(first)

#Selecting multiple columns

second = data[["Age", "College", "Salary"]]

print(second)

# Selecting a single row

# retrieving row by .loc method

third = data.loc["Avery Bradley"]

fourth = data.loc["R.J. Hunter"]

print(third, "\n\n\n",fourth)

#Selecting multiple rows

fifth = data.loc[["Avery Bradley", "R.J. Hunter"]]

print(fifth)

# retrieving two rows and three columns by loc method

sixth = data.loc[["Avery Bradley", "R.J. Hunter"],

["Team", "Number", "Position"]]

print(sixth)

# retrieving all rows and some columns by loc method

seventh = data.loc[:, ["Team", "Number", "Position"]]

print(seventh)

#Indexing a DataFrame using .iloc[ ]

# retrieving rows by iloc method

eighth = data.iloc[3]

print(eighth)

#Selecting multiple rows

ninth = data.iloc[[3, 5, 7]]

print(ninth)

# retrieving two rows and two columns by iloc method

tenth = data.iloc [[3, 4], [1, 2]]

print(tenth)

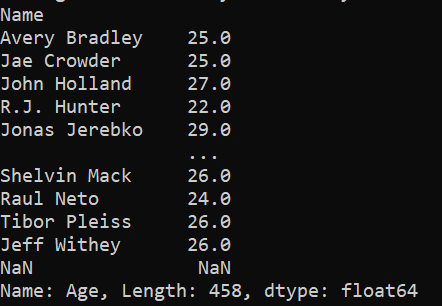
# retrieving all rows and some columns by iloc method

eleventh = data.iloc [:, [1, 2]]

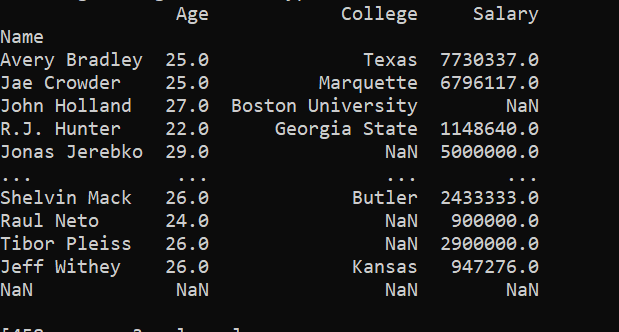
print(eleventh)

**Output:**

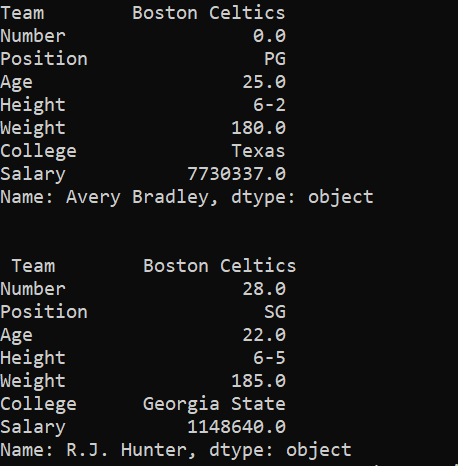
One)



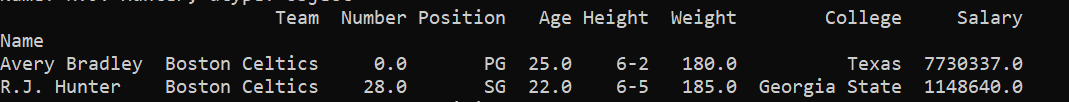
Two



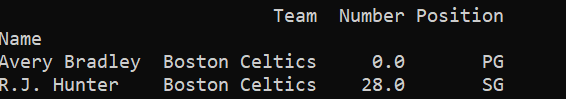
Three & Four



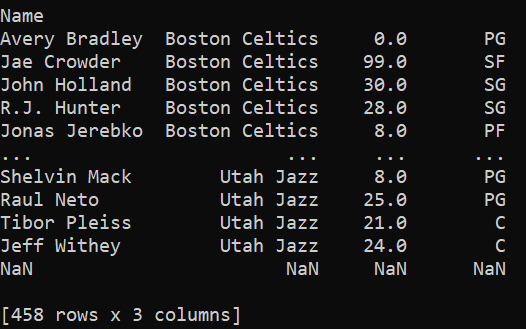
Fifth



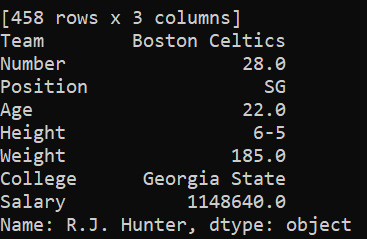
Sixth



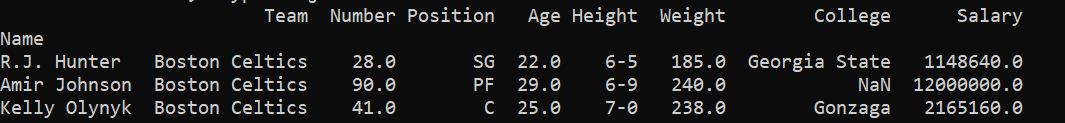
Seventh



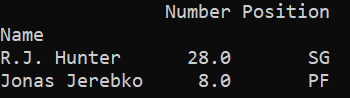
Eighth



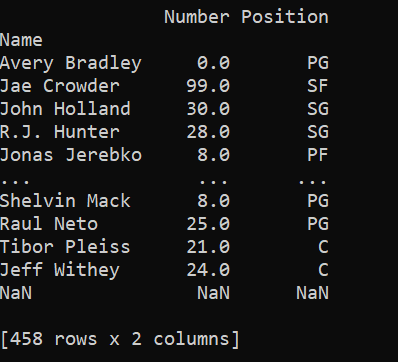
Ninth



Tenth



Eleventh



1. **Insert, delete and update your data**
2. Insert Syntax:

DataFrameName.insert(loc, column, value, allow\_duplicates = False)

import pandas as pd

# reading csv file

data = pd.read\_csv("pokemon-3.csv",index\_col ="#")

# displaying dataframe - Output 1

print (data)

# inserting column with static value in data frame

data.insert(2, "Team", "Any")

# displaying data frame again - Output 2

# list output

print (data)

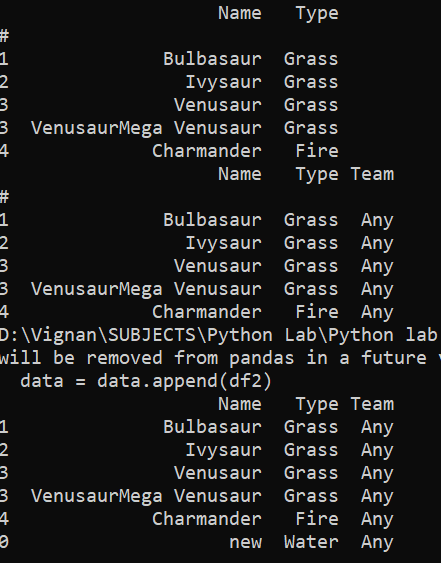
# inserting row using append

df2 = pd.DataFrame ({ 'Name': ['new'], 'Type':['Water'],'Team': ['Any'] })

data = data.append(df2, ignore\_index = True)

print (data)

**Output:**



1. Delete a row in dataframe

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

Delete rows and columns

import pandas as pd

# making data frame from csv file

data = pd.read\_csv("nba-2.csv", index\_col ="Name" )

print (data)

# dropping passed values

data.drop(["Avery Bradley", "John Holland", "R.J. Hunter",

"R.J. Hunter"], inplace = True)

# display

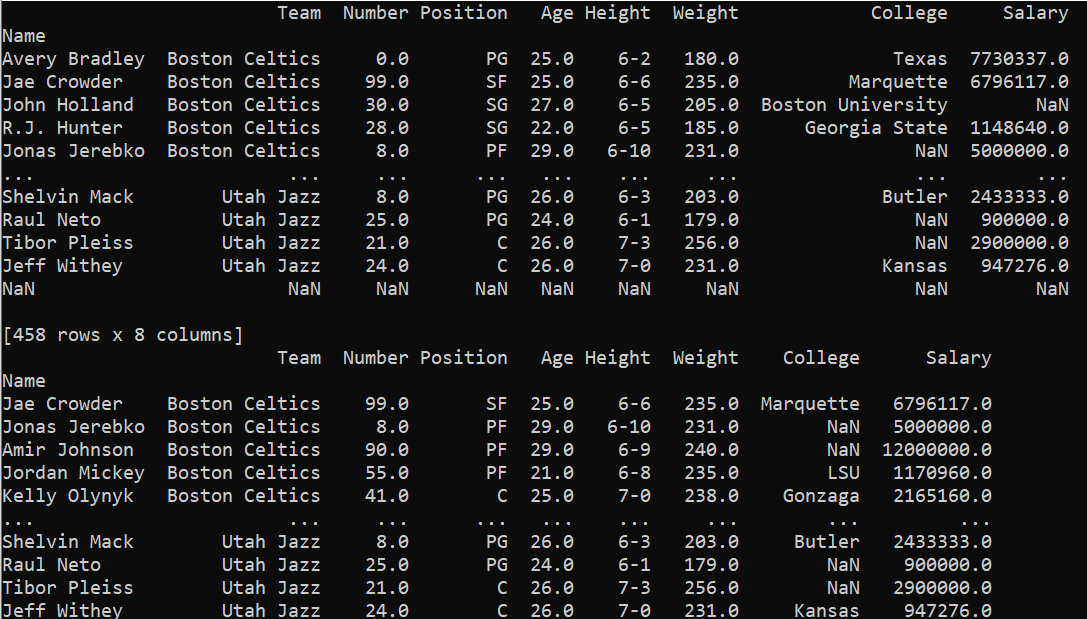
print (data)

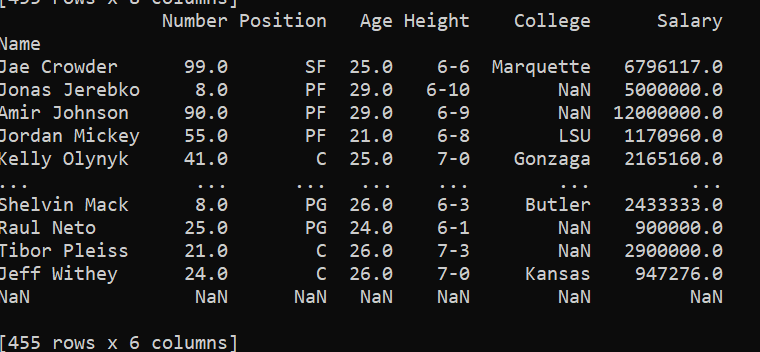
# dropping passed columns

data.drop(["Team", "Weight"], axis = 1, inplace = True)

# display

print (data)





1. Update a dataframe

import pandas as pd

# making data frame from csv file

data = pd.read\_csv("pokemon-3.csv", index\_col ="Name" )

print (data)

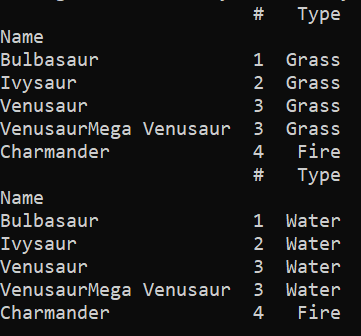
#update

data['Type'] = data['Type'].replace({'Grass': 'Water'})

# display

print (data)

Output:



1. **Apply various filters on the data**

Pandas dataframe.filter() function is used to Subset rows or columns of dataframe according to labels in the specified index

(a) Use filter() function to filter out any three columns of the dataframe.

# importing pandas as pd

import pandas as pd

# Creating the dataframe

df = pd.read\_csv("nba-2.csv")

# Print the dataframe

print (df)

# applying filter function

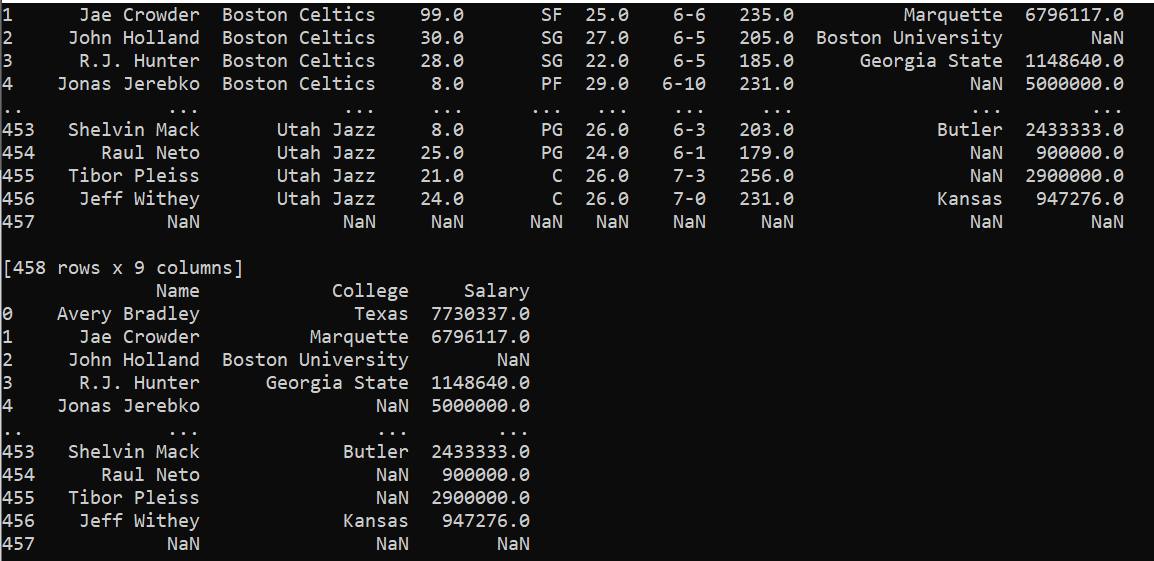
print (df.filter(["Name", "College", "Salary"]))

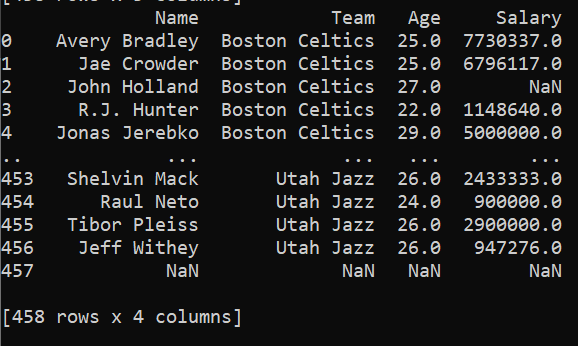
## Using regular expression to extract all

# columns which has letter 'a' or 'A' in its name.

print (df.filter(regex ='[aA]'))

**Output:**





**5. Group, merge, and aggregate data in data frames**

# import module

import pandas as pd

# Creating our dataset

df = pd.DataFrame([[9, 4, 8, 9],

[8, 10, 7, 6],

[7, 6, 8, 5]],

columns=['Maths', 'English',

'Science', 'History'])

# display dataset

print(df)

#The sum() function is used to calculate the sum of every value

print (df.sum())

#The describe() function is used to get a summary of our dataset

print (df.describe())

#We used agg() function to calculate the sum, min, and max of each column in our dataset.

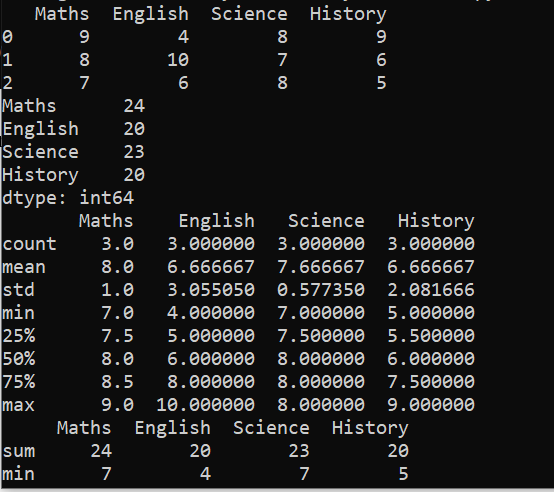
print (df.agg(['sum', 'min', 'max']))

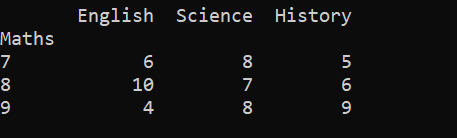
#Grouping is used to group data using some criteria from our dataset

a = df.groupby('Maths')

print(a.first())

Output:





1. **Identify and fix missing values in data**

import pandas as pd

import numpy as np

# create two columns of randomly generated values, replace a few examples with NaNs

data = {"X1": [np.nan, 0.7636183 , 0.61735332, 0.73848657, np.nan,

0.71623709, 0.73075927, np.nan, 0.71073827, 0.54693503],

"X2": [0.87505771, 0.77210971, 0.64369448, 0.54238232, 0.0710951 ,

0.6854597 , np.nan, 0.20935994, 0.54764129, np.nan ]}

df = pd.DataFrame(data)

print(df)

#Imputation Method 1: Mean or Median

df\_mean\_imputed = df.fillna(df.mean())

print (df\_mean\_imputed)

df\_median\_imputed = df.fillna(df.median())

print (df\_median\_imputed)

#Imputation Method 2: Zero

df\_zero\_imputed = df.fillna(0)

print (df\_zero\_imputed)

Output:

