NAME: **Meet Raut**

DIV: **S2-1**

ROLL NO: **2201084**

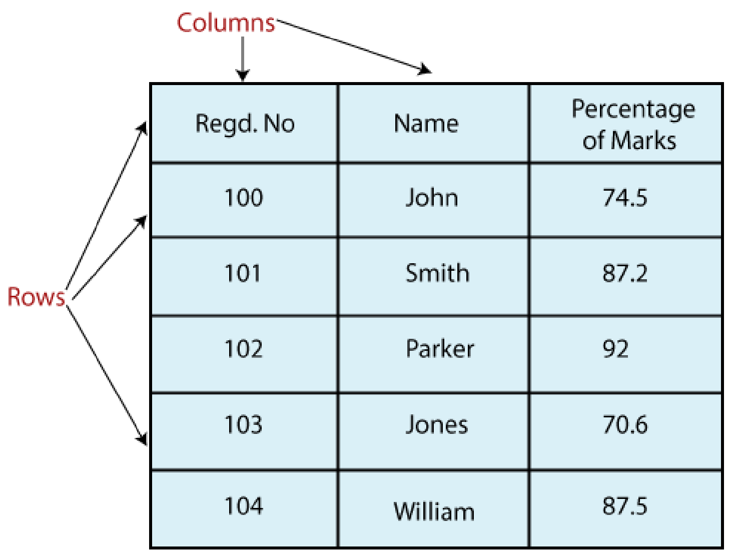
EXPERIMENT – 13:

- ***AIM:*** **To study and implement programs to demonstrate Data Series and Data Frames using Pandas.**

- ***THEORY:***

Pandas DataFrame is a widely used data structure which works with a two-dimensional array with labeled axes (rows and columns). DataFrame is defined as a standard way to store data that has two different indexes, i.e., **row index** and **column index**. It consists of the following properties:

- The columns can be heterogeneous types like int, bool, and so on.
- It can be seen as a dictionary of Series structure where both the rows and columns are indexed. It is denoted as "columns" in case of columns and "index" in case of rows.
- **data:** It consists of different forms like ndarray, series, map, constants, lists, array.
- **index:** The Default np.arrange(n) index is used for the row labels if no index is passed.
- **columns:** The default syntax is np.arrange(n) for the column labels. It shows only true if no index is passed.
- **dtype:** It refers to the data type of each column.
- **copy():** It is used for copying the data



Regd. No	Name	Percentage of Marks
100	John	74.5
101	Smith	87.2
102	Parker	92
103	Jones	70.6
104	William	87.5

Create a DataFrame:

We can create a DataFrame using following ways:

- dict
- Lists
- Numpy ndarrrays
- Series

Column Selection:

We can select any column from the DataFrame.

Column Addition:

We can also add any new column to an existing DataFrame.

Column Deletion:

We can also delete any column from the existing DataFrame.

Row Selection:

We can select any row by passing the row label to a **loc** function. The rows can also be selected by passing the integer location to an **iloc** function. For selecting a row, we have passed the integer location to an **iloc** function. It is another method to select multiple rows using **':'** operator.

Addition of rows:

We can easily add new rows to the DataFrame using **append** function. It adds the new rows at the end.

Deletion of rows:

We can delete or drop any rows from a DataFrame using the **index** label. If in case, the label is duplicate then multiple rows will be deleted.

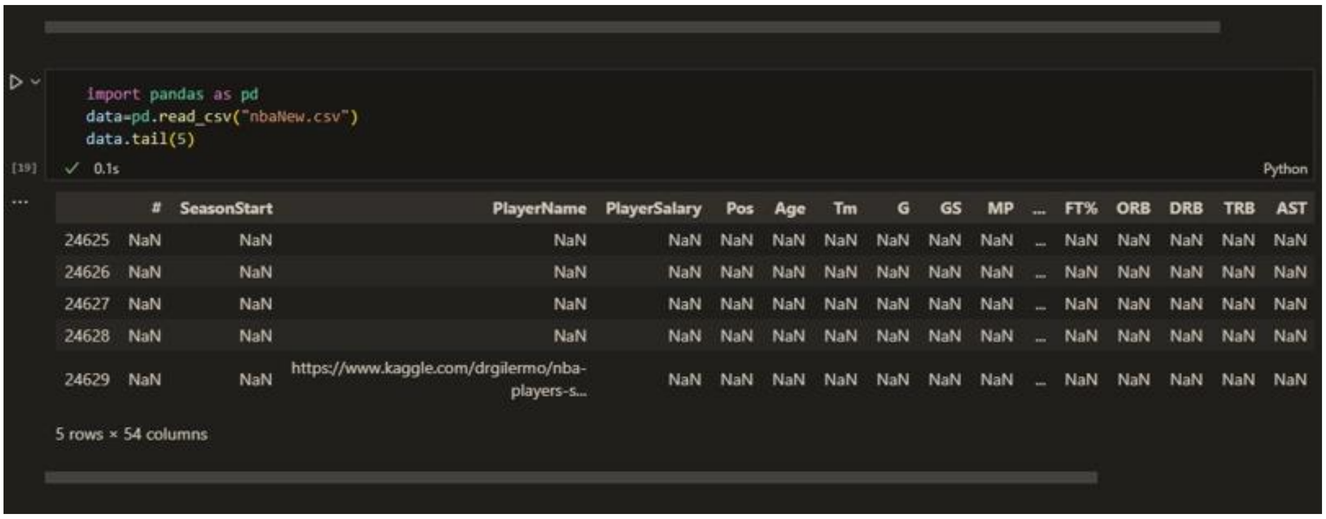
DataFrame Functions

There are lots of functions used in DataFrame which are as follows:

Functions	Description
Pandas DataFrame.append()	Add the rows of other dataframe to the end of the given dataframe.
Pandas DataFrame.apply()	Allows the user to pass a function and apply it to every single value of the Pandas series.
Pandas DataFrame.assign()	Add new column into a dataframe.
Pandas DataFrame.astype()	Cast the Pandas object to a specified dtype.astype() function.
Pandas DataFrame.concat()	Perform concatenation operation along an axis in the DataFrame.
Pandas DataFrame.count()	Count the number of non-NA cells for each column or row.
Pandas DataFrame.describe()	Calculate some statistical data like percentile, mean and std of the numerical values of the Series or DataFrame.
Pandas DataFrame.drop_duplicates()	Remove duplicate values from the DataFrame.
Pandas DataFrame.groupby()	Split the data into various groups.
Pandas DataFrame.head()	Returns the first n rows for the object based on position.
Pandas DataFrame.hist()	Divide the values within a numerical variable into "bins".

Pandas DataFrame.iterrows()	Iterate over the rows as (index, series) pairs.
Pandas DataFrame.mean()	Return the mean of the values for the requested axis.
Pandas DataFrame.melt()	Unpivots the DataFrame from a wide format to a long format.
Pandas DataFrame.merge()	Merge the two datasets together into one.
Pandas DataFrame.pivot_table()	Aggregate data with calculations such as Sum, Count, Average, Max, and Min.
Pandas DataFrame.query()	Filter the dataframe.
Pandas DataFrame.sample()	Select the rows and columns from the dataframe randomly.
Pandas DataFrame.shift()	Shift column or subtract the column value with the previous row value from the dataframe.
Pandas DataFrame.sort()	Sort the dataframe.
Pandas DataFrame.sum()	Return the sum of the values for the requested axis by the user.
Pandas DataFrame.to_excel()	Export the dataframe to the excel file.
Pandas DataFrame.transpose()	Transpose the index and columns of the dataframe.
Pandas DataFrame.where()	Check the dataframe for one or more conditions.

- **OUTPUT:**

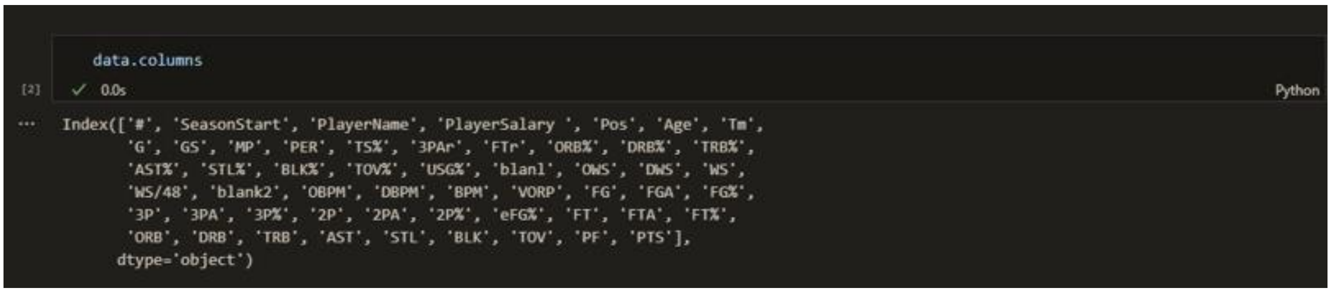


```python
import pandas as pd
data=pd.read_csv("nbaNew.csv")
data.tail(5)
```

[19] ✓ 0.1s Python

	#	SeasonStart	PlayerName	PlayerSalary	Pos	Age	Tm	G	GS	MP	...	FT%	ORB	DRB	TRB	AST
24625	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
24626	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
24627	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
24628	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
24629	NaN	NaN	https://www.kaggle.com/drgilermo/nba-players-s...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN

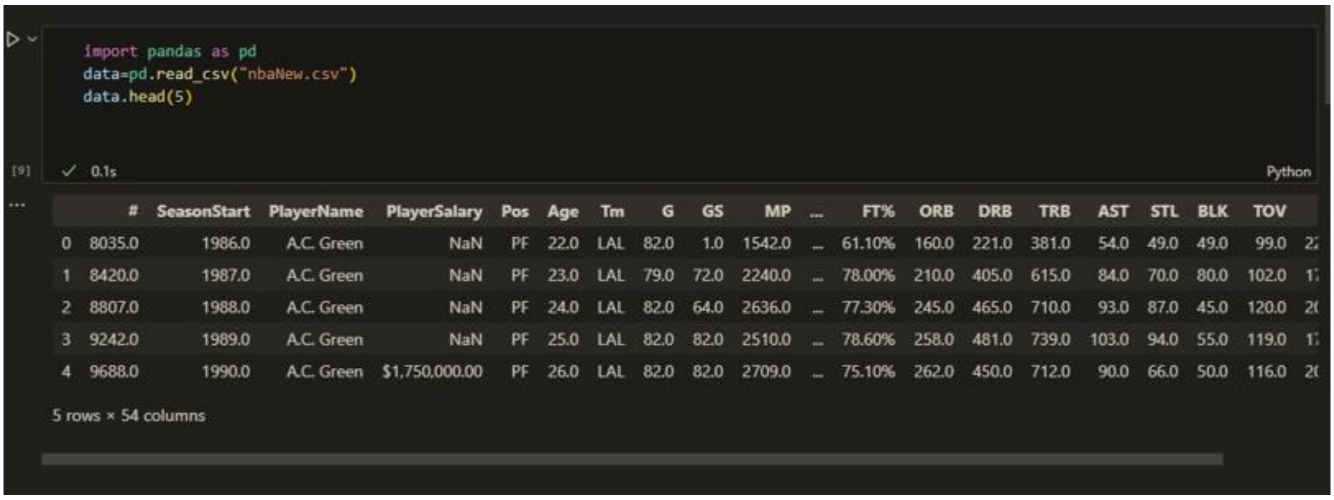
5 rows × 54 columns



```python
data.columns
```

[2] ✓ 0.0s Python

```
Index(['#', 'SeasonStart', 'PlayerName', 'PlayerSalary ', 'Pos', 'Age', 'Tm',
       'G', 'GS', 'MP', 'PER', 'TS%', '3PAr', 'FTr', 'ORB%', 'DRB%', 'TRB%',
       'AST%', 'STL%', 'BLK%', 'TOV%', 'USG%', 'blanl', 'OWS', 'DWS', 'WS',
       'WS/48', 'blank2', 'OBPM', 'DBPM', 'BPM', 'VORP', 'FG', 'FGA', 'FG%',
       '3P', '3PA', '3P%', '2P', '2PA', '2P%', 'eFG%', 'FT', 'FTA', 'FT%',
       'ORB', 'DRB', 'TRB', 'AST', 'STL', 'BLK', 'TOV', 'PF', 'PTS'],
      dtype='object')
```

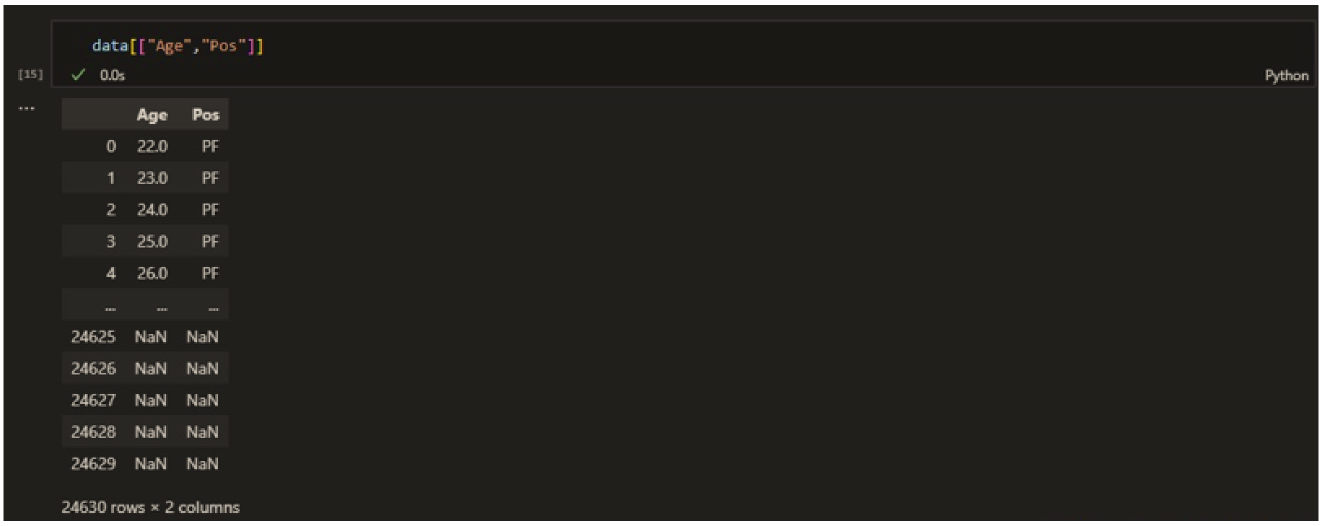


```python
import pandas as pd
data=pd.read_csv("nbaNew.csv")
data.head(5)
```

[9] ✓ 0.1s Python

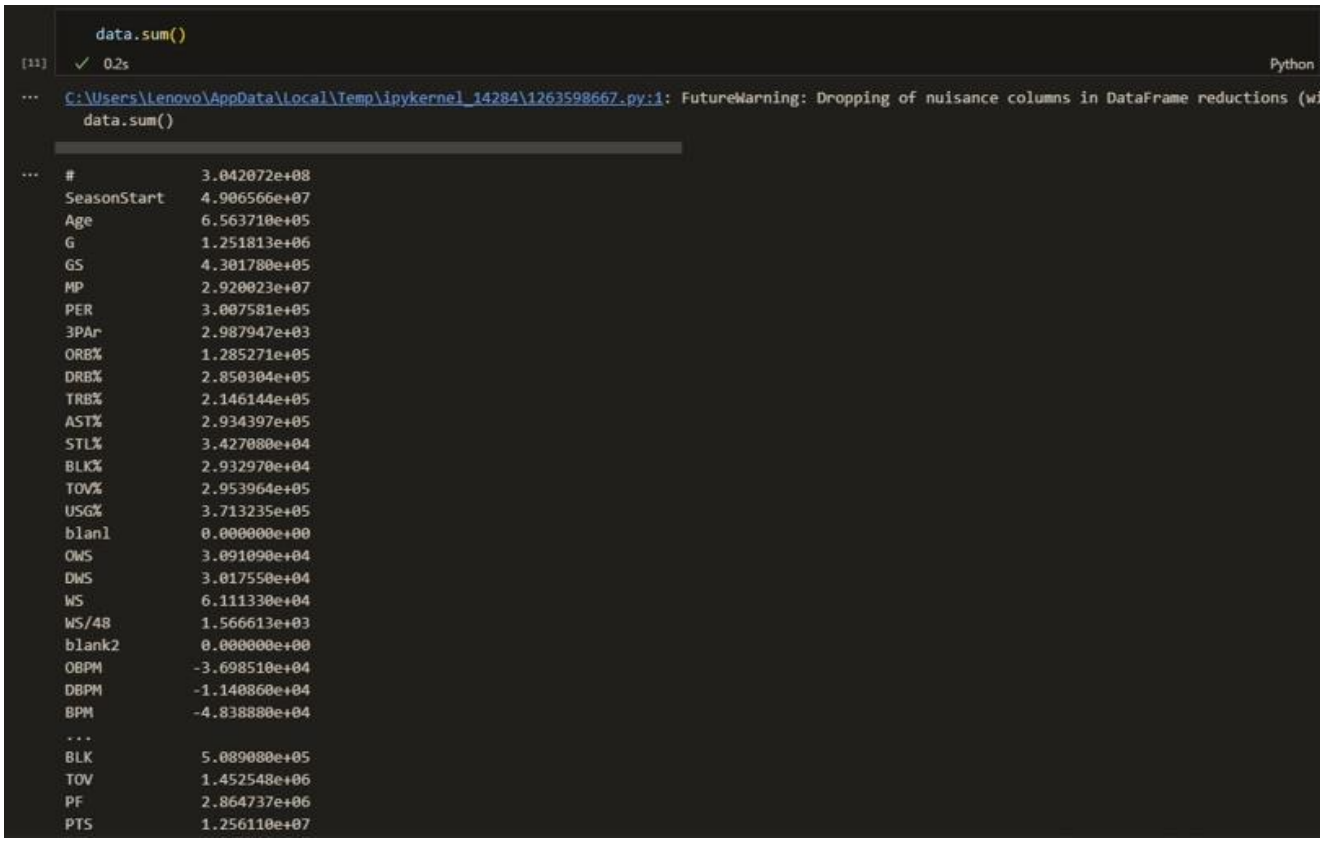
	#	SeasonStart	PlayerName	PlayerSalary	Pos	Age	Tm	G	GS	MP	...	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV
0	8035.0	1986.0	A.C. Green	NaN	PF	22.0	LAL	82.0	1.0	1542.0	...	61.10%	160.0	221.0	381.0	54.0	49.0	49.0	99.0
1	8420.0	1987.0	A.C. Green	NaN	PF	23.0	LAL	79.0	72.0	2240.0	...	78.00%	210.0	405.0	615.0	84.0	70.0	80.0	102.0
2	8807.0	1988.0	A.C. Green	NaN	PF	24.0	LAL	82.0	64.0	2636.0	...	77.30%	245.0	465.0	710.0	93.0	87.0	45.0	120.0
3	9242.0	1989.0	A.C. Green	NaN	PF	25.0	LAL	82.0	82.0	2510.0	...	78.60%	258.0	481.0	739.0	103.0	94.0	55.0	119.0
4	9688.0	1990.0	A.C. Green	$1,750,000.00	PF	26.0	LAL	82.0	82.0	2709.0	...	75.10%	262.0	450.0	712.0	90.0	66.0	50.0	116.0

5 rows × 54 columns



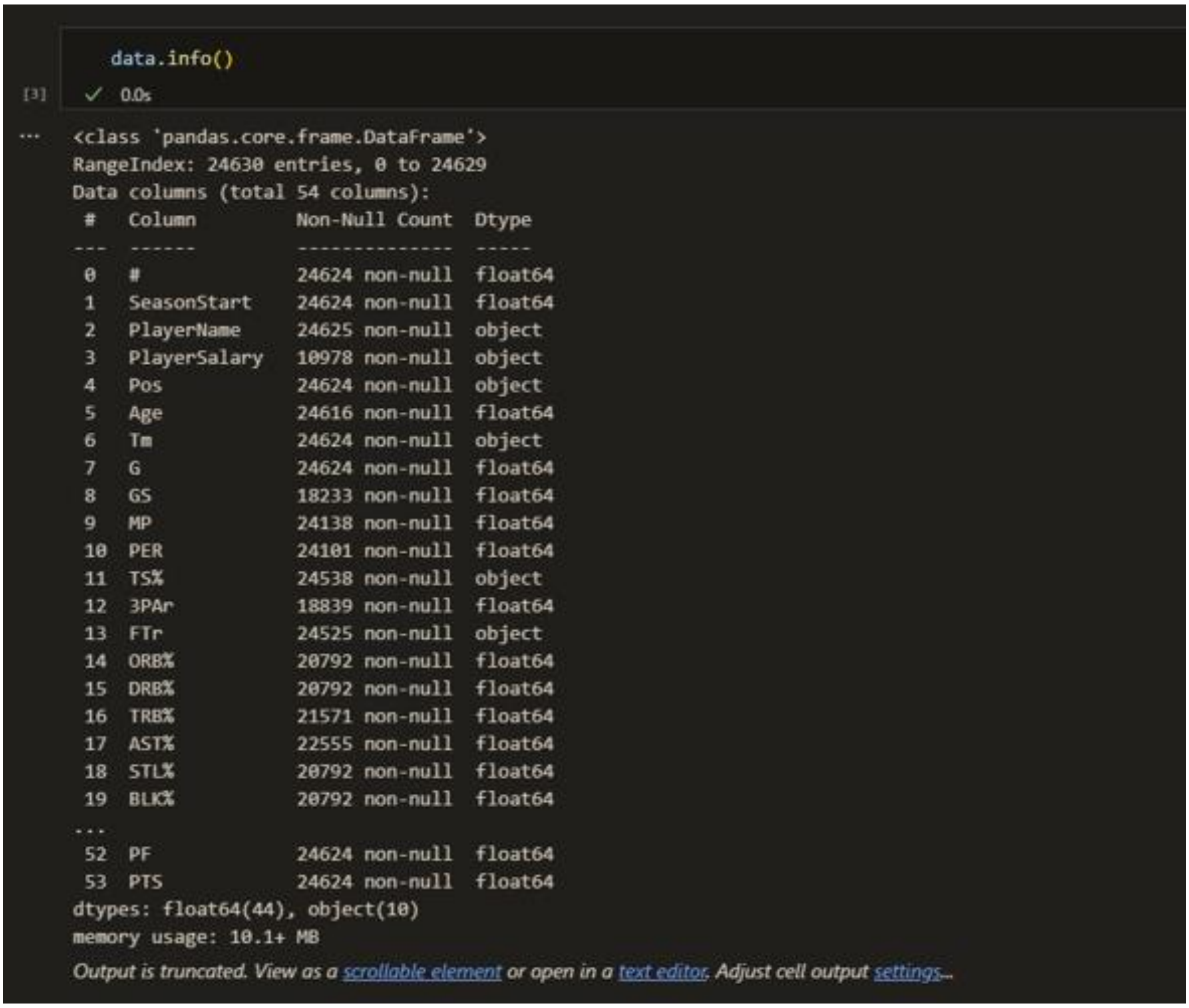
```
data[["Age","Pos"]]
```
[15] ✓ 0.0s Python

```
         Age   Pos
    0    22.0  PF
    1    23.0  PF
    2    24.0  PF
    3    25.0  PF
    4    26.0  PF
  ...    ...   ...
24625    NaN   NaN
24626    NaN   NaN
24627    NaN   NaN
24628    NaN   NaN
24629    NaN   NaN

24630 rows × 2 columns
```



```
data.sum()
```
[11] ✓ 0.2s Python

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_14284\1263598667.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (wi
  data.sum()

#              3.042072e+08
SeasonStart    4.906566e+07
Age            6.563710e+05
G              1.251813e+06
GS             4.301780e+05
MP             2.920023e+07
PER            3.007581e+05
3PAr           2.987947e+03
ORB%           1.285271e+05
DRB%           2.850304e+05
TRB%           2.146144e+05
AST%           2.934397e+05
STL%           3.427080e+04
BLK%           2.932970e+04
TOV%           2.953964e+05
USG%           3.713235e+05
blanl          0.000000e+00
OWS            3.091090e+04
DWS            3.017550e+04
WS             6.111330e+04
WS/48          1.566613e+03
blank2         0.000000e+00
OBPM          -3.698510e+04
DBPM          -1.140860e+04
BPM           -4.838880e+04
...
BLK            5.089080e+05
TOV            1.452548e+06
PF             2.864737e+06
PTS            1.256110e+07
```

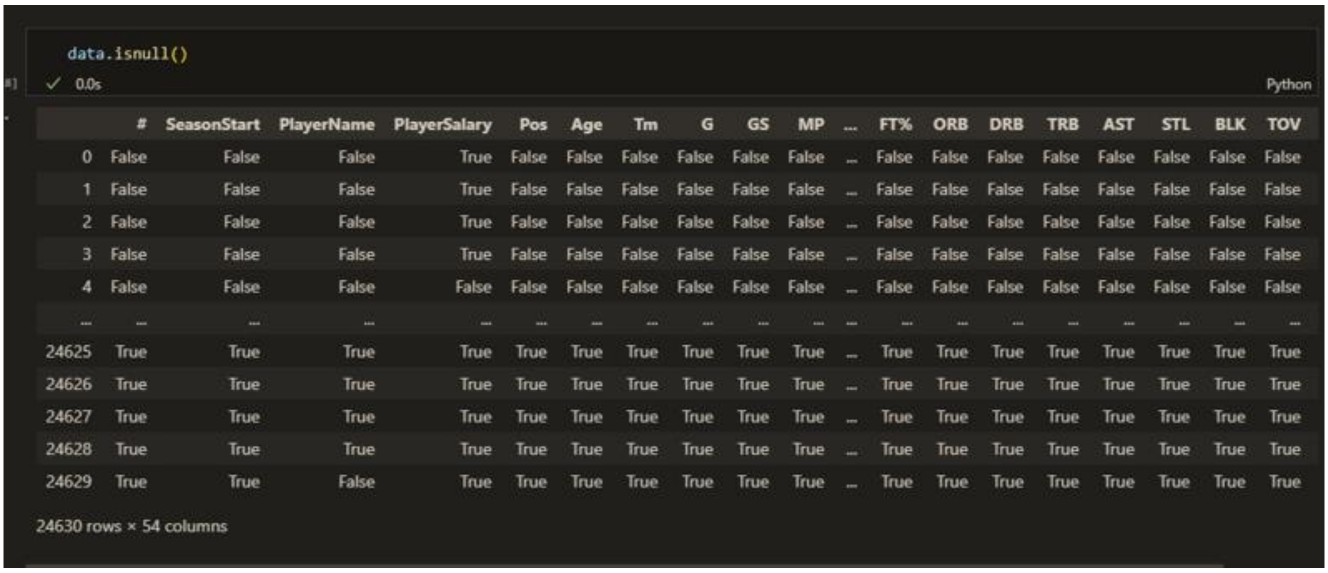


```
data.info()
```

[3] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24630 entries, 0 to 24629
Data columns (total 54 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   #              24624 non-null  float64
 1   SeasonStart    24624 non-null  float64
 2   PlayerName     24625 non-null  object
 3   PlayerSalary   10978 non-null  object
 4   Pos            24624 non-null  object
 5   Age            24616 non-null  float64
 6   Tm             24624 non-null  object
 7   G              24624 non-null  float64
 8   GS             18233 non-null  float64
 9   MP             24138 non-null  float64
 10  PER            24101 non-null  float64
 11  TS%            24538 non-null  object
 12  3PAr           18839 non-null  float64
 13  FTr            24525 non-null  object
 14  ORB%           20792 non-null  float64
 15  DRB%           20792 non-null  float64
 16  TRB%           21571 non-null  float64
 17  AST%           22555 non-null  float64
 18  STL%           20792 non-null  float64
 19  BLK%           20792 non-null  float64
...
 52  PF             24624 non-null  float64
 53  PTS            24624 non-null  float64
dtypes: float64(44), object(10)
memory usage: 10.1+ MB
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

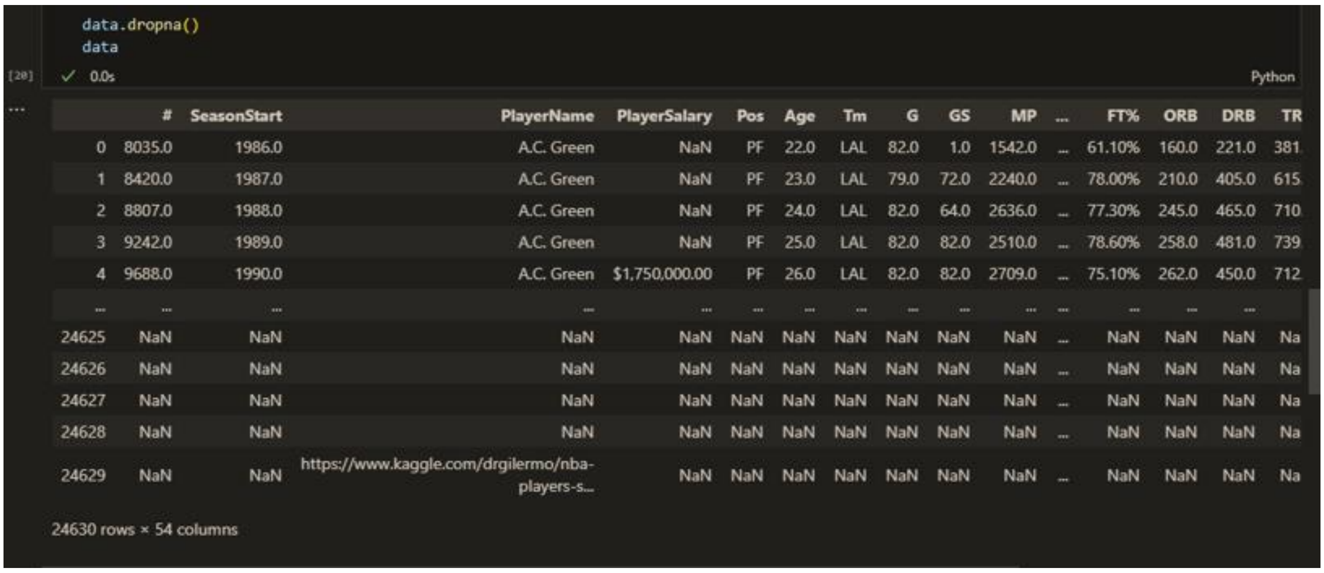


```
data.isnull()
```

✓ 0.0s Python

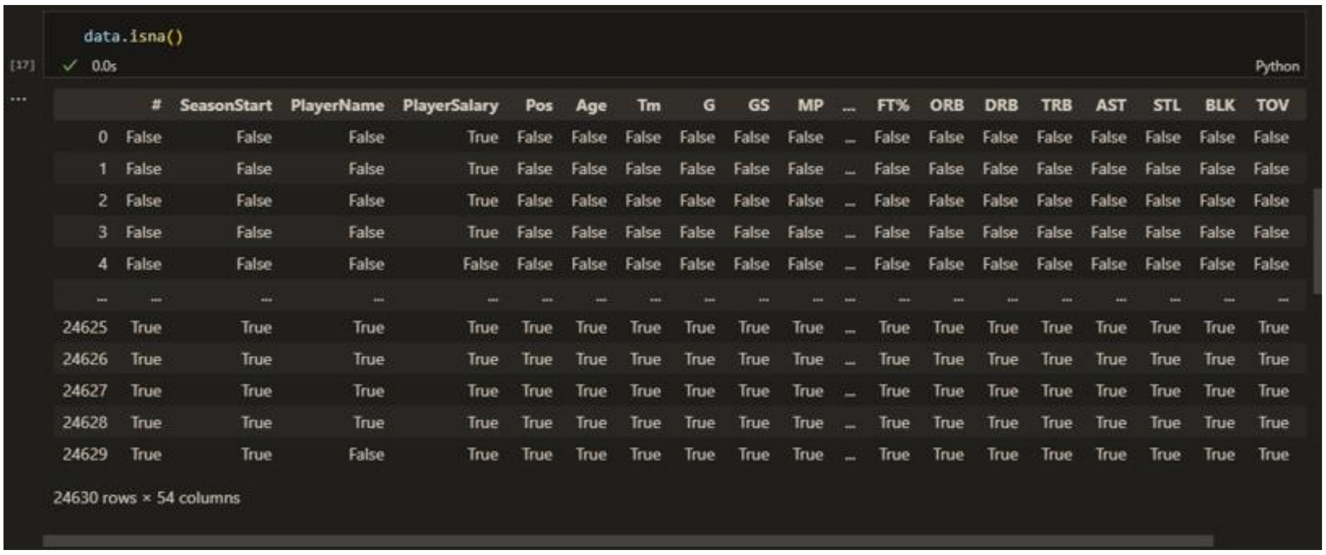
	#	SeasonStart	PlayerName	PlayerSalary	Pos	Age	Tm	G	GS	MP	...	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV
0	False	False	False	True	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
1	False	False	False	True	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
2	False	False	False	True	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
3	False	False	False	True	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
24625	True	True	True	True	True	True	True	True	True	True	...	True	True	True	True	True	True	True	True
24626	True	True	True	True	True	True	True	True	True	True	...	True	True	True	True	True	True	True	True
24627	True	True	True	True	True	True	True	True	True	True	...	True	True	True	True	True	True	True	True
24628	True	True	True	True	True	True	True	True	True	True	...	True	True	True	True	True	True	True	True
24629	True	True	False	True	True	True	True	True	True	True	...	True	True	True	True	True	True	True	True

24630 rows × 54 columns



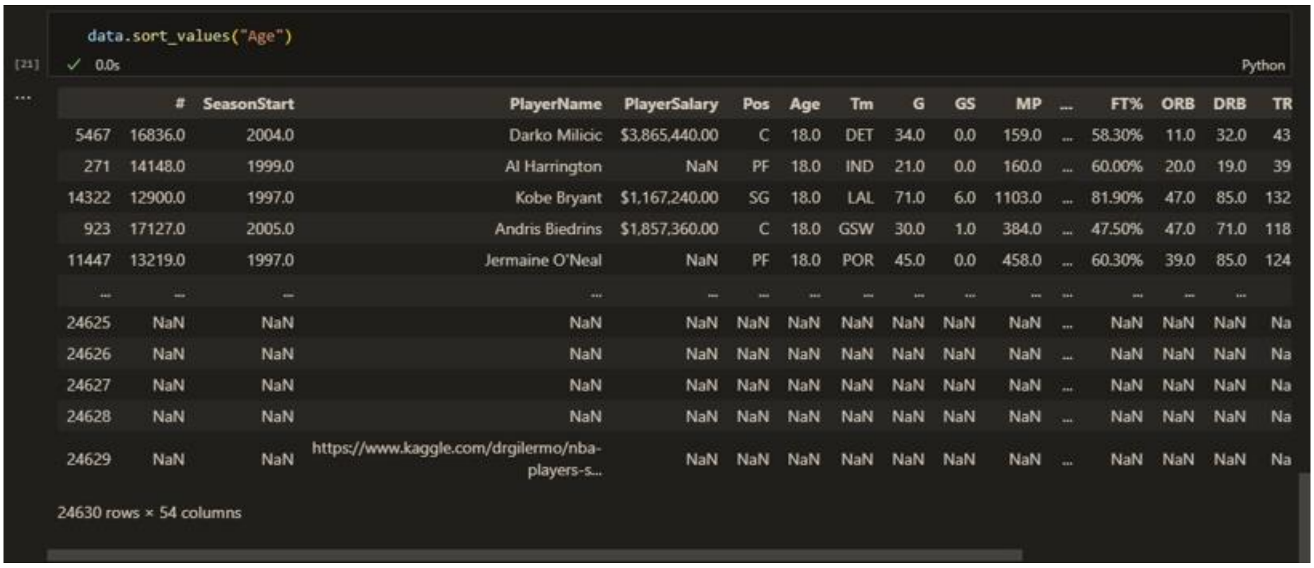
```
data.dropna()
data
```

```
[20]  ✓ 0.0s                                                                                   Python

         #  SeasonStart                             PlayerName  PlayerSalary  Pos   Age   Tm     G    GS      MP  ...    FT%    ORB    DRB   TR
    0  8035.0   1986.0                               A.C. Green           NaN   PF  22.0  LAL  82.0   1.0  1542.0  ...  61.10%  160.0  221.0  381
    1  8420.0   1987.0                               A.C. Green           NaN   PF  23.0  LAL  79.0  72.0  2240.0  ...  78.00%  210.0  405.0  615
    2  8807.0   1988.0                               A.C. Green           NaN   PF  24.0  LAL  82.0  64.0  2636.0  ...  77.30%  245.0  465.0  710
    3  9242.0   1989.0                               A.C. Green           NaN   PF  25.0  LAL  82.0  82.0  2510.0  ...  78.60%  258.0  481.0  739
    4  9688.0   1990.0                               A.C. Green $1,750,000.00   PF  26.0  LAL  82.0  82.0  2709.0  ...  75.10%  262.0  450.0  712
  ...     ...      ...                                      ...           ...  ...   ...  ...   ...   ...     ...  ...     ...    ...    ...
24625     NaN      NaN                                      NaN           NaN  NaN   NaN  NaN   NaN   NaN     NaN  ...     NaN    NaN    NaN   Na
24626     NaN      NaN                                      NaN           NaN  NaN   NaN  NaN   NaN   NaN     NaN  ...     NaN    NaN    NaN   Na
24627     NaN      NaN                                      NaN           NaN  NaN   NaN  NaN   NaN   NaN     NaN  ...     NaN    NaN    NaN   Na
24628     NaN      NaN                                      NaN           NaN  NaN   NaN  NaN   NaN   NaN     NaN  ...     NaN    NaN    NaN   Na
24629     NaN      NaN  https://www.kaggle.com/drgilermo/nba-           NaN  NaN   NaN  NaN   NaN   NaN     NaN  ...     NaN    NaN    NaN   Na
                                                  players-s...

24630 rows × 54 columns
```



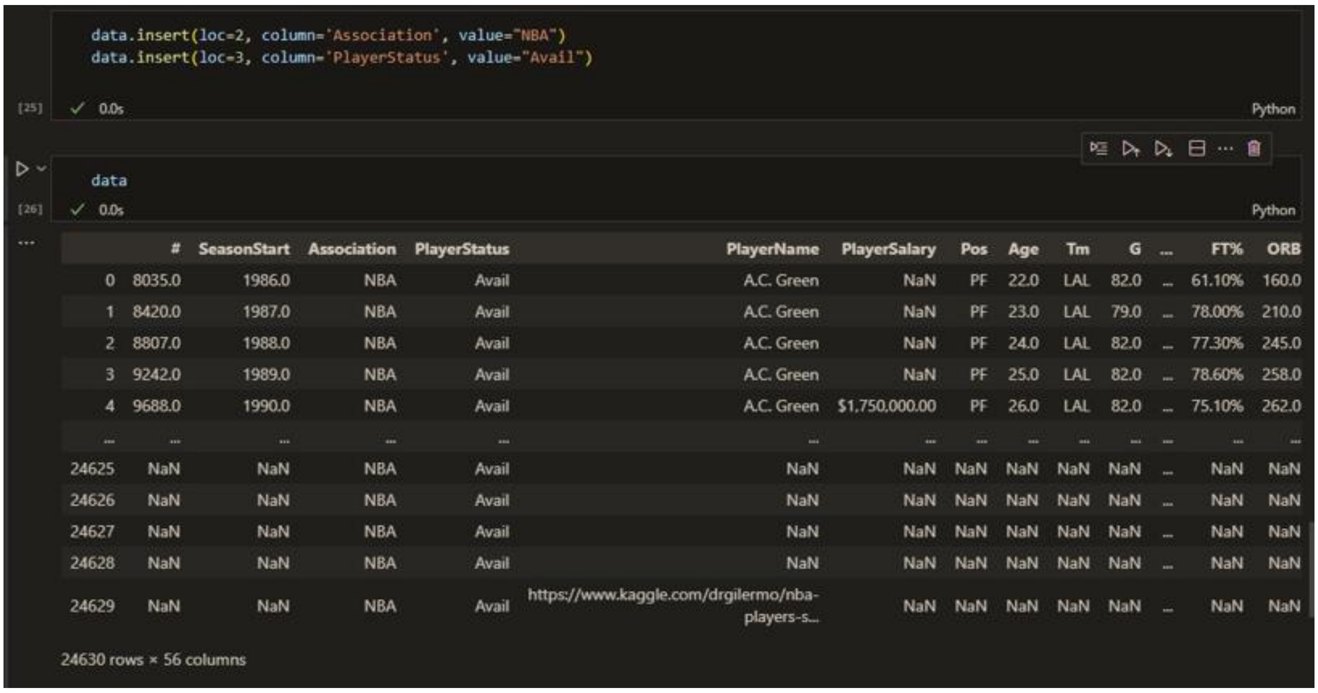
```
data.isna()
```

```
[17]  ✓ 0.0s                                                                                   Python

           #  SeasonStart  PlayerName  PlayerSalary    Pos    Age     Tm      G     GS     MP  ...    FT%    ORB    DRB    TRB    AST    STL    BLK    TOV
    0  False        False       False          True  False  False  False  False  False  False  ...  False  False  False  False  False  False  False  False
    1  False        False       False          True  False  False  False  False  False  False  ...  False  False  False  False  False  False  False  False
    2  False        False       False          True  False  False  False  False  False  False  ...  False  False  False  False  False  False  False  False
    3  False        False       False          True  False  False  False  False  False  False  ...  False  False  False  False  False  False  False  False
    4  False        False       False         False  False  False  False  False  False  False  ...  False  False  False  False  False  False  False  False
  ...    ...          ...         ...           ...    ...    ...    ...    ...    ...    ...  ...    ...    ...    ...    ...    ...    ...    ...    ...
24625   True         True        True          True   True   True   True   True   True   True  ...   True   True   True   True   True   True   True   True
24626   True         True        True          True   True   True   True   True   True   True  ...   True   True   True   True   True   True   True   True
24627   True         True        True          True   True   True   True   True   True   True  ...   True   True   True   True   True   True   True   True
24628   True         True        True          True   True   True   True   True   True   True  ...   True   True   True   True   True   True   True   True
24629   True         True       False          True   True   True   True   True   True   True  ...   True   True   True   True   True   True   True   True

24630 rows × 54 columns
```



```
data.sort_values("Age")
```

```
[21]  ✓ 0.0s                                                                                   Python

           #  SeasonStart                             PlayerName  PlayerSalary  Pos   Age   Tm     G   GS      MP  ...    FT%   ORB   DRB   TR
 5467  16836.0   2004.0                            Darko Milicic $3,865,440.00    C  18.0  DET  34.0  0.0   159.0  ...  58.30%  11.0  32.0   43
  271  14148.0   1999.0                            Al Harrington           NaN   PF  18.0  IND  21.0  0.0   160.0  ...  60.00%  20.0  19.0   39
14322  12900.0   1997.0                              Kobe Bryant $1,167,240.00   SG  18.0  LAL  71.0  6.0  1103.0  ...  81.90%  47.0  85.0  132
  923  17127.0   2005.0                          Andris Biedrins $1,857,360.00    C  18.0  GSW  30.0  1.0   384.0  ...  47.50%  47.0  71.0  118
11447  13219.0   1997.0                          Jermaine O'Neal           NaN   PF  18.0  POR  45.0  0.0   458.0  ...  60.30%  39.0  85.0  124
  ...      ...      ...                                      ...           ...  ...   ...  ...   ...  ...     ...  ...     ...   ...   ...
24625      NaN      NaN                                      NaN           NaN  NaN   NaN  NaN   NaN  NaN     NaN  ...     NaN   NaN   NaN   Na
24626      NaN      NaN                                      NaN           NaN  NaN   NaN  NaN   NaN  NaN     NaN  ...     NaN   NaN   NaN   Na
24627      NaN      NaN                                      NaN           NaN  NaN   NaN  NaN   NaN  NaN     NaN  ...     NaN   NaN   NaN   Na
24628      NaN      NaN                                      NaN           NaN  NaN   NaN  NaN   NaN  NaN     NaN  ...     NaN   NaN   NaN   Na
24629      NaN      NaN  https://www.kaggle.com/drgilermo/nba-           NaN  NaN   NaN  NaN   NaN  NaN     NaN  ...     NaN   NaN   NaN   Na
                                                  players-s...

24630 rows × 54 columns
```



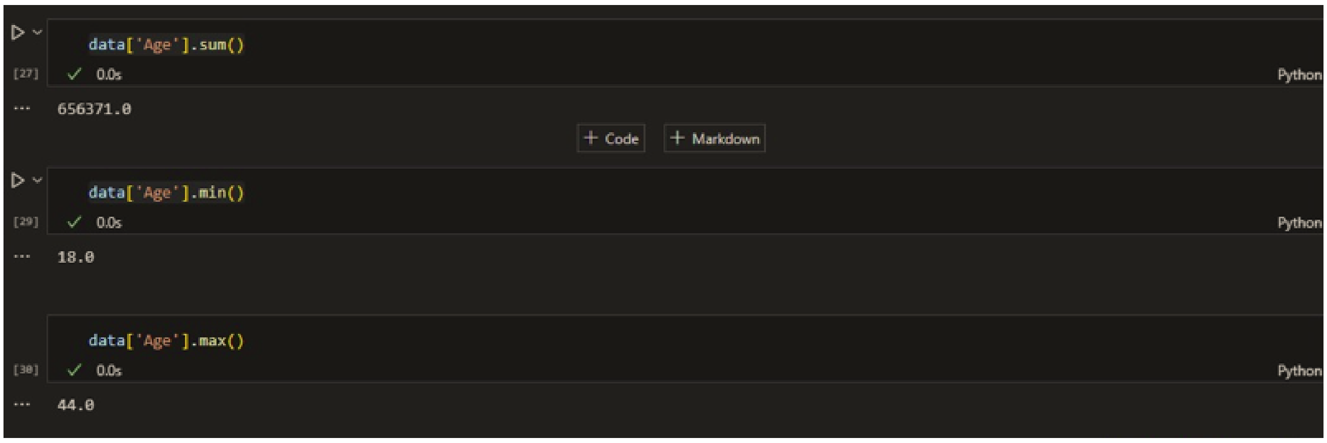
```
data.insert(loc=2, column='Association', value="NBA")
data.insert(loc=3, column='PlayerStatus', value="Avail")
```

[25] ✓ 0.0s Python

```
data
```

[26] ✓ 0.0s Python

```
             #  SeasonStart  Association  PlayerStatus                          PlayerName    PlayerSalary  Pos   Age   Tm     G  ...     FT%    ORB
    0   8035.0       1986.0          NBA         Avail                          A.C. Green             NaN   PF  22.0  LAL  82.0  ...  61.10%  160.0
    1   8420.0       1987.0          NBA         Avail                          A.C. Green             NaN   PF  23.0  LAL  79.0  ...  78.00%  210.0
    2   8807.0       1988.0          NBA         Avail                          A.C. Green             NaN   PF  24.0  LAL  82.0  ...  77.30%  245.0
    3   9242.0       1989.0          NBA         Avail                          A.C. Green             NaN   PF  25.0  LAL  82.0  ...  78.60%  258.0
    4   9688.0       1990.0          NBA         Avail                          A.C. Green   $1,750,000.00   PF  26.0  LAL  82.0  ...  75.10%  262.0
  ...      ...          ...          ...           ...                                 ...             ...  ...   ...  ...   ...  ...     ...    ...
24625      NaN          NaN          NBA         Avail                                 NaN             NaN  NaN   NaN  NaN   NaN  ...     NaN    NaN
24626      NaN          NaN          NBA         Avail                                 NaN             NaN  NaN   NaN  NaN   NaN  ...     NaN    NaN
24627      NaN          NaN          NBA         Avail                                 NaN             NaN  NaN   NaN  NaN   NaN  ...     NaN    NaN
24628      NaN          NaN          NBA         Avail                                 NaN             NaN  NaN   NaN  NaN   NaN  ...     NaN    NaN
24629      NaN          NaN          NBA         Avail  https://www.kaggle.com/drgilermo/nba-             NaN  NaN   NaN  NaN   NaN  ...     NaN    NaN
                                                                                 players-s...

24630 rows × 56 columns
```



```
data['Age'].sum()
```

[27] ✓ 0.0s Python

```
656371.0
```

```
data['Age'].min()
```

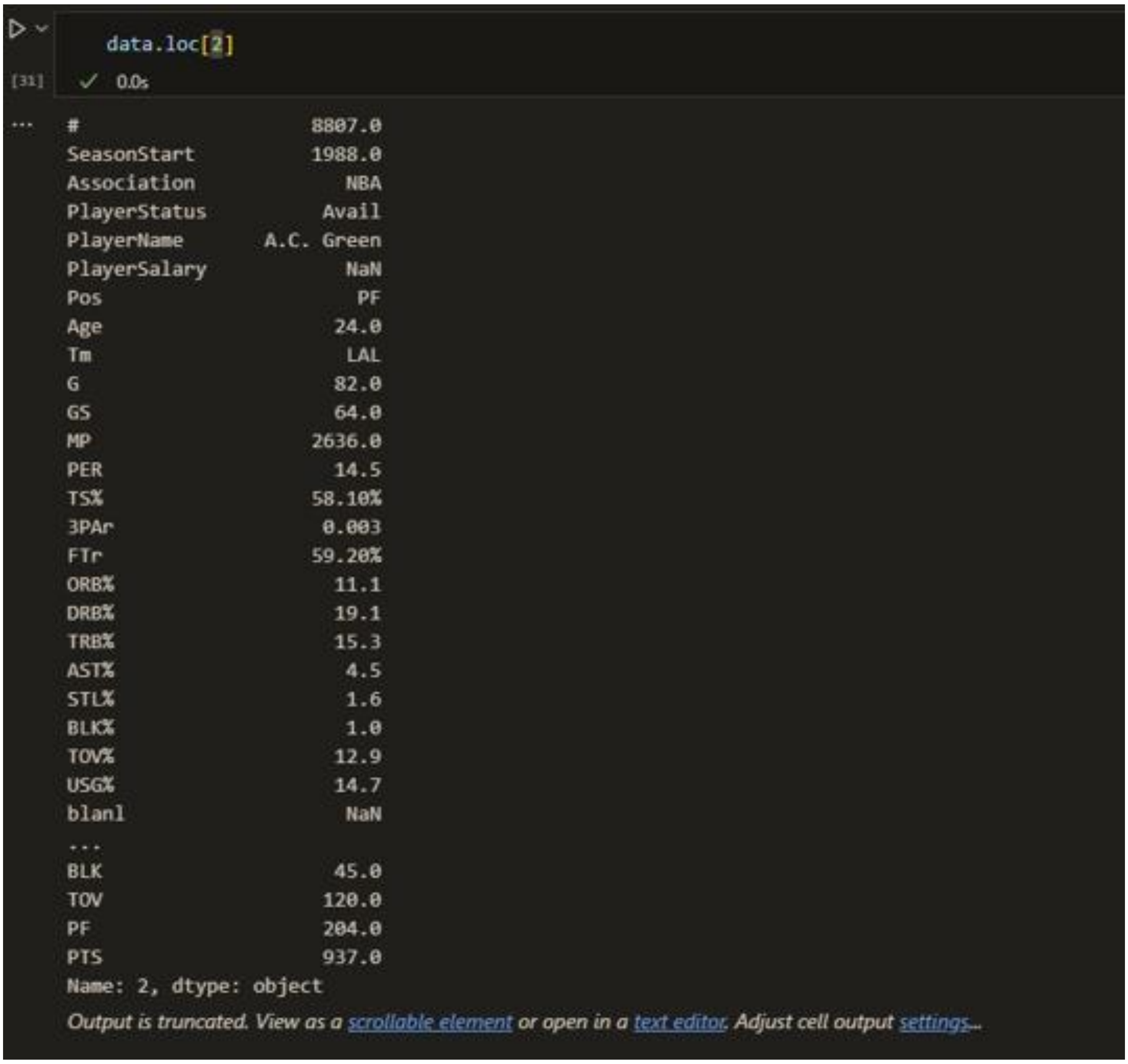
[29] ✓ 0.0s Python

```
18.0
```

```
data['Age'].max()
```

[30] ✓ 0.0s Python

```
44.0
```



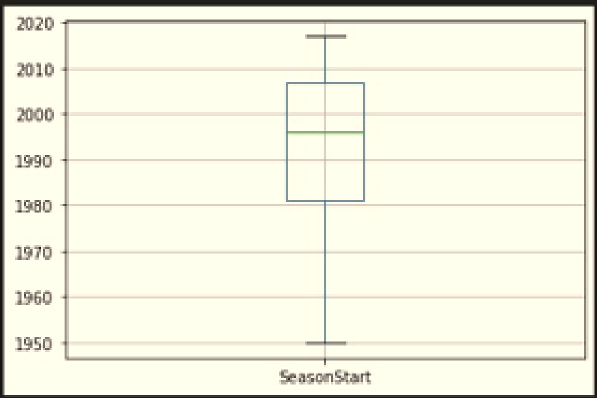
```
data.loc[2]
[31]  ✓ 0.0s
```

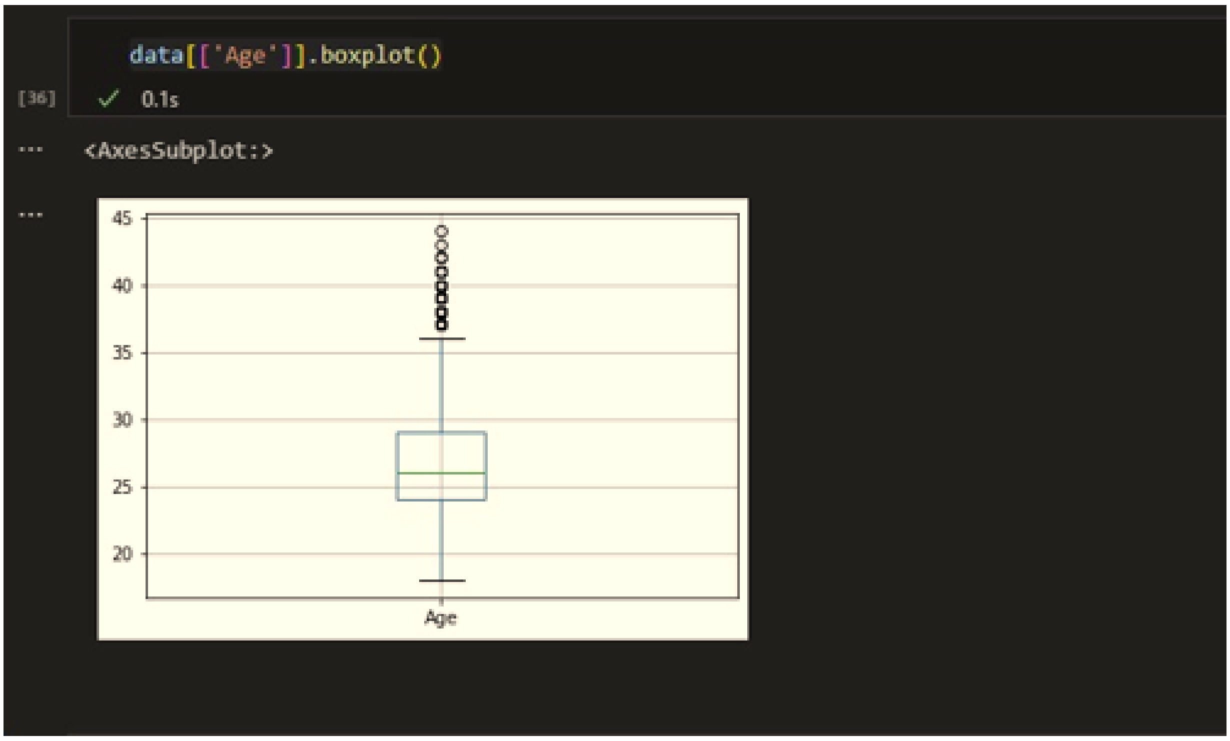
```
#                      8807.0
SeasonStart            1988.0
Association               NBA
PlayerStatus            Avail
PlayerName         A.C. Green
PlayerSalary              NaN
Pos                        PF
Age                      24.0
Tm                        LAL
G                        82.0
GS                       64.0
MP                     2636.0
PER                      14.5
TS%                    58.10%
3PAr                    0.003
FTr                    59.20%
ORB%                     11.1
DRB%                     19.1
TRB%                     15.3
AST%                      4.5
STL%                      1.6
BLK%                      1.0
TOV%                     12.9
USG%                     14.7
blanl                     NaN
                      ...
BLK                      45.0
TOV                     120.0
PF                      204.0
PTS                     937.0
Name: 2, dtype: object
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

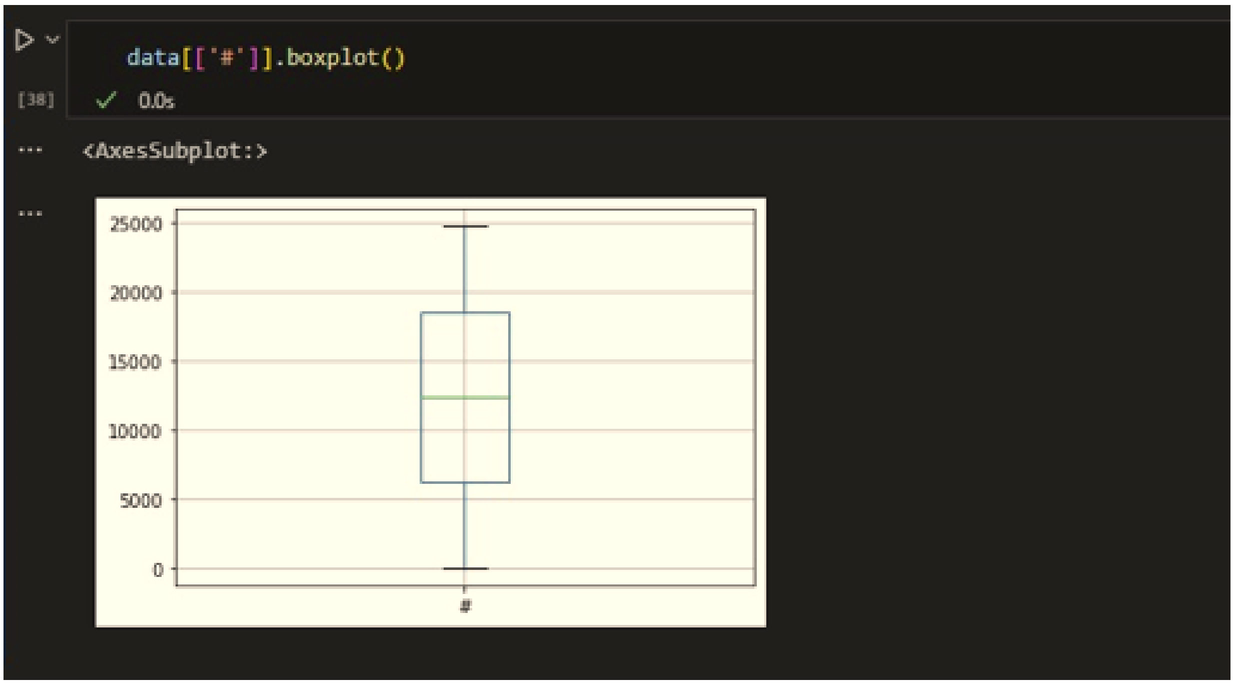
```
data[['SeasonStart']].boxplot()
[35]  ✓ 3.4s
```

```
<AxesSubplot:>
```

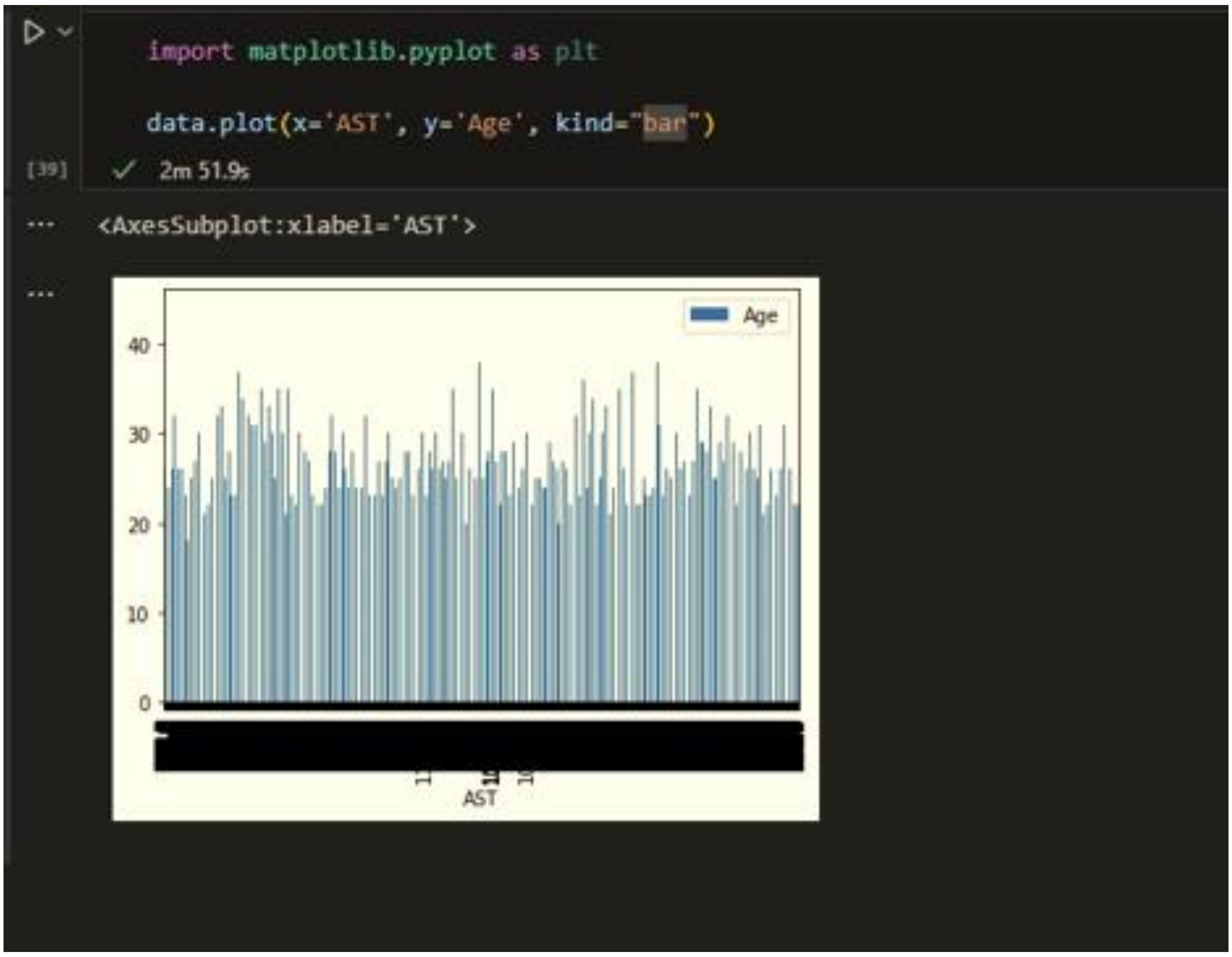




```
data[['Age']].boxplot()
[36]  ✓ 0.1s
<AxesSubplot:>
```



```
data[['#']].boxplot()
[38]  ✓ 0.0s
<AxesSubplot:>
```



```
import matplotlib.pyplot as plt

data.plot(x='AST', y='Age', kind="bar")
```

```
[39]  ✓  2m 51.9s
```

```
<AxesSubplot:xlabel='AST'>
```

- ***<u>CONCLUSION:</u>*** **Hence, we have successfully implemented program on demonstrating Data Series and Data Frames using Pandas; LO 1.**