

```
In [1]: import pandas as pd
import numpy as np
data=pd.read_csv("https://media.geeksforgeeks.org/wp-content/uploads/nba.csv")

#data=data.replace("?",np.NaN) #v
#data.head()
```

```
In [2]: perc=[.20,.40,.60,.80]
include=['object','float','int']
desc=data.describe(percentiles=perc,include=include)
desc
```

Out[2]:

	Name	Team	Number	Position	Age	Height	Weight	College	
count	457	457	457.000000	457	457.000000	457	457.000000	373	4.460000
unique	457	30	NaN	5	NaN	18	NaN	118	
top	Avery Bradley	New Orleans Pelicans	NaN	SG	NaN	6-9	NaN	Kentucky	
freq	1	19	NaN	102	NaN	59	NaN	22	
mean	NaN	NaN	17.678337	NaN	26.938731	NaN	221.522976	NaN	4.842681
std	NaN	NaN	15.966090	NaN	4.404016	NaN	26.368343	NaN	5.229211
min	NaN	NaN	0.000000	NaN	19.000000	NaN	161.000000	NaN	3.088800
20%	NaN	NaN	4.000000	NaN	23.000000	NaN	195.600000	NaN	9.472760
40%	NaN	NaN	10.000000	NaN	25.000000	NaN	213.400000	NaN	1.938840
50%	NaN	NaN	13.000000	NaN	26.000000	NaN	220.000000	NaN	2.839000
60%	NaN	NaN	18.600000	NaN	27.000000	NaN	230.000000	NaN	3.815000
80%	NaN	NaN	30.000000	NaN	30.000000	NaN	245.000000	NaN	8.042800
max	NaN	NaN	99.000000	NaN	40.000000	NaN	307.000000	NaN	2.500000

```
In [3]: data.isnull().any().any()
data.isnull().sum()
avg_age = data['Age'].astype("float").mean()
avg_age
```

Out[3]: 26.938730853391686

```
In [4]: data["Age"].replace(np.nan ,avg_age, inplace=True)
data["Age"]
desc
```

Out[4]:

	Name	Team	Number	Position	Age	Height	Weight	College	
<b>count</b>	457	457	457.000000	457	457.000000	457	457.000000	373	4.460000
<b>unique</b>	457	30	NaN	5	NaN	18	NaN	118	
<b>top</b>	Avery Bradley	New Orleans Pelicans	NaN	SG	NaN	6-9	NaN	Kentucky	
<b>freq</b>	1	19	NaN	102	NaN	59	NaN	22	
<b>mean</b>	NaN	NaN	17.678337	NaN	26.938731	NaN	221.522976	NaN	4.84268
<b>std</b>	NaN	NaN	15.966090	NaN	4.404016	NaN	26.368343	NaN	5.22925
<b>min</b>	NaN	NaN	0.000000	NaN	19.000000	NaN	161.000000	NaN	3.08880
<b>20%</b>	NaN	NaN	4.000000	NaN	23.000000	NaN	195.600000	NaN	9.47276
<b>40%</b>	NaN	NaN	10.000000	NaN	25.000000	NaN	213.400000	NaN	1.93884
<b>50%</b>	NaN	NaN	13.000000	NaN	26.000000	NaN	220.000000	NaN	2.83907
<b>60%</b>	NaN	NaN	18.600000	NaN	27.000000	NaN	230.000000	NaN	3.81500
<b>80%</b>	NaN	NaN	30.000000	NaN	30.000000	NaN	245.000000	NaN	8.04289
<b>max</b>	NaN	NaN	99.000000	NaN	40.000000	NaN	307.000000	NaN	2.50000

```
In [5]: data.isnull().sum()
data.dtypes #1
```

```
Out[5]: Name      object
Team      object
Number    float64
Position  object
Age       float64
Height    object
Weight    float64
College   object
Salary    float64
dtype: object
```

```
In [6]: data[["Name"]]=data[["Name"]].astype("str")
data.dtypes
```

```
Out[6]: Name          object
Team            object
Number         float64
Position       object
Age            float64
Height         object
Weight         float64
College        object
Salary         float64
dtype: object
```

```
In [7]: data[["Team"]]=data[["Team"]].astype("str")
data.dtypes      #a
```

```
Out[7]: Name          object
Team            object
Number         float64
Position       object
Age            float64
Height         object
Weight         float64
College        object
Salary         float64
dtype: object
```

```
In [8]: data.isnull().sum()
data["Name"].replace(np.nan, 'New Name', inplace=True)
data["Name"]
```

```
Out[8]: 0      Avery Bradley
1      Jae Crowder
2      John Holland
3      R.J. Hunter
4      Jonas Jerebko
...
453    Shelvin Mack
454      Raul Neto
455    Tibor Pleiss
456    Jeff Withey
457              nan
Name: Name, Length: 458, dtype: object
```

```
In [9]: data.isnull().sum()  
desc
```

Out[9]:

	Name	Team	Number	Position	Age	Height	Weight	College	
count	457	457	457.000000	457	457.000000	457	457.000000	373	4.460000
unique	457	30	NaN	5	NaN	18	NaN	118	
top	Avery Bradley	New Orleans Pelicans	NaN	SG	NaN	6-9	NaN	Kentucky	
freq	1	19	NaN	102	NaN	59	NaN	22	
mean	NaN	NaN	17.678337	NaN	26.938731	NaN	221.522976	NaN	4.842600
std	NaN	NaN	15.966090	NaN	4.404016	NaN	26.368343	NaN	5.229200
min	NaN	NaN	0.000000	NaN	19.000000	NaN	161.000000	NaN	3.088800
20%	NaN	NaN	4.000000	NaN	23.000000	NaN	195.600000	NaN	9.472700
40%	NaN	NaN	10.000000	NaN	25.000000	NaN	213.400000	NaN	1.938800
50%	NaN	NaN	13.000000	NaN	26.000000	NaN	220.000000	NaN	2.839000
60%	NaN	NaN	18.600000	NaN	27.000000	NaN	230.000000	NaN	3.815000
80%	NaN	NaN	30.000000	NaN	30.000000	NaN	245.000000	NaN	8.042800
max	NaN	NaN	99.000000	NaN	40.000000	NaN	307.000000	NaN	2.500000

```
In [10]: data["Team"].replace(np.nan, 'NEW TEAM', inplace=True)  
data["Team"]  
data.isnull().sum()  
#data[["Number"]]=data[["Number"]].astype("int")    #c  
data.dtypes
```

Out[10]:

Name	object
Team	object
Number	float64
Position	object
Age	float64
Height	object
Weight	float64
College	object
Salary	float64
dtype:	object

```
In [11]: data["Number"].replace(np.nan,12,inplace=True)
data["Number"]
data.isnull().sum()
data.dtypes
desc
```

Out[11]:

	Name	Team	Number	Position	Age	Height	Weight	College	
count	457	457	457.000000	457	457.000000	457	457.000000	373	4.460000
unique	457	30	NaN	5	NaN	18	NaN	118	
top	Avery Bradley	New Orleans Pelicans	NaN	SG	NaN	6-9	NaN	Kentucky	
freq	1	19	NaN	102	NaN	59	NaN	22	
mean	NaN	NaN	17.678337	NaN	26.938731	NaN	221.522976	NaN	4.842600
std	NaN	NaN	15.966090	NaN	4.404016	NaN	26.368343	NaN	5.229200
min	NaN	NaN	0.000000	NaN	19.000000	NaN	161.000000	NaN	3.088800
20%	NaN	NaN	4.000000	NaN	23.000000	NaN	195.600000	NaN	9.472700
40%	NaN	NaN	10.000000	NaN	25.000000	NaN	213.400000	NaN	1.938800
50%	NaN	NaN	13.000000	NaN	26.000000	NaN	220.000000	NaN	2.839000
60%	NaN	NaN	18.600000	NaN	27.000000	NaN	230.000000	NaN	3.815000
80%	NaN	NaN	30.000000	NaN	30.000000	NaN	245.000000	NaN	8.042800
max	NaN	NaN	99.000000	NaN	40.000000	NaN	307.000000	NaN	2.500000

```
In [12]: data.dtypes
data[["Position"]]=data[["Position"]].astype("str")
data.dtypes
data.isnull().sum()
data["Position"].replace(np.nan,'RP',inplace=True)
data["Position"]
data.dtypes
#hieght=data[["Height"]].astype("float") #k
data.dtypes
avg_age
```

Out[12]: 26.938730853391686

```
In [13]: data[["Height"]] = data[["Height"]].astype("str")
data[["Height"]].replace(np.nan,"7.8",inplace=True)
data.dtypes
data[["Height"]]
data.isnull().sum()
#Weight
data[["Weight"]]
```

```
Out[13]: 0      180.0
1      235.0
2      205.0
3      185.0
4      231.0
...
453    203.0
454    179.0
455    256.0
456    231.0
457      NaN
Name: Weight, Length: 458, dtype: float64
```

```
In [14]: data[["Weight"]] = data[["Weight"]].astype("float")
wt=data[["Weight"]].mean()
data[["Weight"]].replace(np.nan,wt,inplace=True)
data[["Weight"]]
```

```
Out[14]: 0      180.000000
1      235.000000
2      205.000000
3      185.000000
4      231.000000
...
453    203.000000
454    179.000000
455    256.000000
456    231.000000
457    221.522976
Name: Weight, Length: 458, dtype: float64
```

```
In [15]: data.isnull().sum()
data.dtypes
data[["College"]]=data[["College"]].astype("str")
data.dtypes
data.isnull().sum()
k=85
for i in range (k):

    data["College"].replace(np.nan,"NEW COLLAGE",inplace=True)
data.isnull().sum()
```

```
Out[15]: Name          0
         Team          0
         Number        0
         Position      0
         Age           0
         Height        0
         Weight        0
         College       0
         Salary       12
         dtype: int64
```

```
In [16]: data[["Salary"]] = data[["Salary"]].astype("float")
data["Salary"]
data.isnull().sum()
j=12
s=data["Salary"].mean()
s
```

```
Out[16]: 4842684.105381166
```

```
In [17]: for i in range (j):
          data["Salary"].replace(np.nan,s,inplace=True)
          data["Salary"]
          #data.isnull().sum()
          #s=4842684.105381166
          df=data["Age"]
          df2=df.copy()
          df2=pd.get_dummies(df2,"Age")
          df2
```

Out[17]:

	Age_19.0	Age_20.0	Age_21.0	Age_22.0	Age_23.0	Age_24.0	Age_25.0	Age_26.0	Age_26.0
0	0	0	0	0	0	0	1	0	
1	0	0	0	0	0	0	1	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	1	0	0	0	0	
4	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
453	0	0	0	0	0	0	0	1	
454	0	0	0	0	0	1	0	0	
455	0	0	0	0	0	0	0	1	
456	0	0	0	0	0	0	0	1	
457	0	0	0	0	0	0	0	0	

458 rows × 23 columns

