In [1]:

```
# ##integer
# int8
# int16
# int32
# int64

# ##Unsigned
# uint8
# uint16
# uint32
# uint64

# ##Floating point
# float16
# float32
# float64

##Mixed datatpes=Object

##Pure text values-Category
```

In [8]:

```
import numpy as np
np.iinfo(np.uint8)              ##Int info
```

Out[8]:

```
iinfo(min=0, max=255, dtype=uint8)
```

In [12]:

```
np.finfo(np.float16)                ##float info
```

Out[12]:

```
finfo(resolution=0.001, min=-6.55040e+04, max=6.55040e+04, dtype=float16)
```

Changin datatype of column

In [18]:

```
import pandas as pd
df = pd.read_excel("nba.xlsx")
df.describe()
```

Out[18]:

	Number	Age	Weight	Salary
count	457.000000	457.000000	457.000000	4.460000e+02
mean	17.678337	26.938731	221.522976	4.842684e+06
std	15.966090	4.404016	26.368343	5.229238e+06
min	0.000000	19.000000	161.000000	3.088800e+04
25%	5.000000	24.000000	200.000000	1.044792e+06
50%	13.000000	26.000000	220.000000	2.839073e+06
75%	25.000000	30.000000	240.000000	6.500000e+06
max	99.000000	40.000000	307.000000	2.500000e+07

In [27]:

```
df = pd.read_csv("Titanic - Titanic.csv")
```

```
df.describe()
```

Out[27]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [23]:

```
df["PassengerId"].min(),df["PassengerId"].max()
```

Out[23]:

```
(1, 891)
```

In [43]:

```
print(df["PassengerId"].astype("uint16"))            ##Convert datatype to unsignedint16
df.columns
df["Age"].min(),df["Age"].max()
```

```
0        1
1        2
2        3
3        4
4        5
      ...
886    887
887    888
888    889
889    890
890    891
Name: PassengerId, Length: 891, dtype: uint16
```

Out[43]:

```
(0.42, 80.0)
```

In [49]:

```
schema = {'PassengerId': np.uint16,
           'Survived' : np.uint8,
          'Pclass': np.uint8,
          'Sex': "category",
          'name': "category",
          'Age': np.float16,
          'SibSp': np.uint8,
          'Parch': np.uint8,
          'Ticket':'object',
          'Fare': np.float16,
          'Cabin':"object",
          'Embarked':'category'
         }
schema
```

Out[49]:

```
{'PassengerId': numpy.uint16,
 'Survived': numpy.uint8,
 'Pclass': numpy.uint8,
```

```
 'Sex': 'category',
 'name': 'category',
 'Age': numpy.float16,
 'SibSp': numpy.uint8,
 'Parch': numpy.uint8,
 'Ticket': 'object',
 'Fare': numpy.float16,
 'Cabin': 'object',
 'Embarked': 'category'}
```

In [53]:

```
df = pd.read_csv("Titanic - Titanic.csv",dtype = schema)
df
```

Out[53]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250000	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.312500	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925781	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.093750	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.046875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.000000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.000000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.453125	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.000000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.750000	NaN	Q

891 rows × 12 columns

In [67]:

```
df.memory_usage(deep=True)
df
```

Out[67]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	2023-06-02 00:00:00	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	2023-06-06 00:00:00	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	2023-06-05 00:00:00	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	2023-06-05 00:00:00	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	2023-06-10 00:00:00	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
452	Trey Lyles	Utah Jazz	41	PF	20	2023-06-10 00:00:00	234	Kentucky	2239800.0

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
453	Shelvin Mack	Utah Jazz	8	PG	26	2023-06-03 00:00:00	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	2023-06-01 00:00:00	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	2023-07-03 00:00:00	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0

457 rows × 9 columns

In [146]:

```
df = pd.read_excel("nba.xlsx")
print(df.memory_usage(deep=True))
df.describe()
df["Height"].astype("category")
df.memory_usage(deep=True)
df
```

```
Index          128
Name         32008
Team         33405
Number        3656
Position     26885
Age           3656
Height       25740
Weight        3656
College      27236
Salary        3656
dtype: int64
```

Out[146]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	2023-06-02 00:00:00	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	2023-06-06 00:00:00	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	2023-06-05 00:00:00	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	2023-06-05 00:00:00	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	2023-06-10 00:00:00	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
452	Trey Lyles	Utah Jazz	41	PF	20	2023-06-10 00:00:00	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	PG	26	2023-06-03 00:00:00	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	2023-06-01 00:00:00	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	2023-07-03 00:00:00	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0

457 rows × 9 columns

In [128]:

```
df.describe()
```

Out[128]:

	Number	Age	Weight	Salary
count	457.000000	457.000000	457.000000	4.460000e+02
mean	17.678337	26.938731	221.522976	4.842684e+06
std	15.966090	4.404016	26.368343	5.229238e+06
min	0.000000	19.000000	161.000000	3.088800e+04
25%	5.000000	24.000000	200.000000	1.044792e+06
50%	13.000000	26.000000	220.000000	2.839073e+06

	Number	Age	Weight	Salary
75%	25.000000	30.000000	240.000000	6.500000e+06
max	99.000000	40.000000	307.000000	2.500000e+07

In [160]:

```
np.finfo(np.float32)
df.dtypes
```

Out[160]:

```
Name          object
Team          object
Number         int64
Position      object
Age            int64
Height        object
Weight         int64
College       object
Salary       float64
dtype: object
```

In [162]:

```
df["Name"].astype("category")
df["Team"].astype("category")
df["Position"].astype("category")
df["Height"].astype("category")
df["College"].astype("category")
df["Number"].astype("uint8")
df["Weight"].astype("uint16")
df["Salary"].astype("float32")
df.memory_usage()
df.info()
df.dtypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 457 entries, 0 to 456
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Name      457 non-null    object
 1   Team      457 non-null    object
 2   Number    457 non-null    int64
 3   Position  457 non-null    object
 4   Age       457 non-null    int64
 5   Height    457 non-null    object
 6   Weight    457 non-null    int64
 7   College   373 non-null    object
 8   Salary    446 non-null    float64
dtypes: float64(1), int64(3), object(5)
memory usage: 32.3+ KB
```

Out[162]:

```
Name          object
Team          object
Number         int64
Position      object
Age            int64
Height        object
Weight         int64
College       object
Salary       float64
dtype: object
```

DROP AND FIND MISSING VALUES IN DATASET

In [176]:

```
# Drop NA And IS Na
```

```
df = pd.read_csv("Titanic - Titanic.csv")
df.isna().sum()
```

Out[176]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [179]:

```
df.dropna(how = "all")
df.isna().sum()
```

Out[179]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [178]:

```
df = pd.read_csv("Titanic - Titanic.csv")
df
```

Out[178]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
				Johnston, Miss.					W./C.			

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
888	889	0	3	Catherine Helen "Carrie"	female	NaN	1	2	6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [273]:

```
df = pd.read_excel("nba.xlsx")
```

In [262]:

```
df = pd.read_excel("nba.xlsx")
df
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 457 entries, 0 to 456
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Name      457 non-null    object 
 1   Team      457 non-null    object 
 2   Number    457 non-null    int64  
 3   Position  457 non-null    object 
 4   Age       457 non-null    int64  
 5   Height    457 non-null    object 
 6   Weight    457 non-null    int64  
 7   College   373 non-null    object 
 8   Salary    446 non-null    float64
dtypes: float64(1), int64(3), object(5)
memory usage: 32.3+ KB
```

In [263]:

```
df.isna().sum()
df.dropna(how="any",inplace=True)
df
```

Out[263]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	2023-06-02 00:00:00	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	2023-06-06 00:00:00	235	Marquette	6796117.0
3	R.J. Hunter	Boston Celtics	28	SG	22	2023-06-05 00:00:00	185	Georgia State	1148640.0
6	Jordan Mickey	Boston Celtics	55	PF	21	2023-06-08 00:00:00	235	LSU	1170960.0
7	Kelly Olynyk	Boston Celtics	41	C	25	7-0	238	Gonzaga	2165160.0
...	...	...	...	...	...	...	...	...	...
449	Rodney Hood	Utah Jazz	5	SG	23	2023-06-08 00:00:00	206	Duke	1348440.0
451	Chris Johnson	Utah Jazz	23	SF	26	2023-06-06 00:00:00	206	Dayton	981348.0
452	Trey Lyles	Utah Jazz	41	PF	20	2023-06-10 00:00:00	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	PG	26	2023-06-03 00:00:00	203	Butler	2433333.0
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0

364 rows × 9 columns

In [249]:

```
df.drop
```

Out[249]:

Out[249]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	2023-06-02 00:00:00	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	2023-06-06 00:00:00	235	Marquette	6796117.0
3	R.J. Hunter	Boston Celtics	28	SG	22	2023-06-05 00:00:00	185	Georgia State	1148640.0
6	Jordan Mickey	Boston Celtics	55	PF	21	2023-06-08 00:00:00	235	LSU	1170960.0
7	Kelly Olynyk	Boston Celtics	41	C	25	7-0	238	Gonzaga	2165160.0
...	...	...	...	...	...	...	...	...	...
449	Rodney Hood	Utah Jazz	5	SG	23	2023-06-08 00:00:00	206	Duke	1348440.0
451	Chris Johnson	Utah Jazz	23	SF	26	2023-06-06 00:00:00	206	Dayton	981348.0
452	Trey Lyles	Utah Jazz	41	PF	20	2023-06-10 00:00:00	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	PG	26	2023-06-03 00:00:00	203	Butler	2433333.0
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0

364 rows × 9 columns

In [196]:

```
df.info()
df.memory_usage()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 364 entries, 0 to 456
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Name      364 non-null    object
 1   Team      364 non-null    object
 2   Number    364 non-null    int64
 3   Position  364 non-null    object
 4   Age       364 non-null    int64
 5   Height    364 non-null    object
 6   Weight    364 non-null    int64
 7   College   364 non-null    object
 8   Salary    364 non-null    float64
dtypes: float64(1), int64(3), object(5)
memory usage: 28.4+ KB
```

Out[196]:

```
Index       2912
Name        2912
Team        2912
Number      2912
Position    2912
Age         2912
Height      2912
Weight      2912
College     2912
Salary      2912
dtype: int64
```

In [204]:

```
##Changing the datatypes:-----
df["Name"]= df["Name"].astype("category")
df["Team"] = df["Team"].astype("category")
df["Position"] =df["Position"].astype("category")
df["Height"]=df["Height"].astype("category")
df["College"]=df["College"].astype("category")
df["Number"]=df["Number"].astype("uint8")
df["Weight"]=df["Weight"].astype("uint16")
df["Salary"]=df["Salary"].astype("float32")
```

In [246]:

```
df.dtypes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 457 entries, 0 to 456
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Name      457 non-null    object 
 1   Team      457 non-null    object 
 2   Number    457 non-null    int64  
 3   Position  457 non-null    object 
 4   Age       457 non-null    int64  
 5   Height    457 non-null    object 
 6   Weight    457 non-null    int64  
 7   College   373 non-null    object 
 8   Salary    446 non-null    float64
dtypes: float64(1), int64(3), object(5)
memory usage: 32.3+ KB
```

In [217]:

```
df.memory_usage(deep = True)
df.dtypes
```

Out[217]:

```
Name         category
Team         category
Number          uint8
Position     category
Age             int64
Height       category
Weight         uint16
College      category
Salary        float32
dtype: object
```

Drop Rows and Column

In [232]:

```
df = pd.read_csv("Titanic - Titanic.csv")
df.drop([0])
df.drop([0,4])
df.drop(["PassengerId","Pclass"],axis = 1)
# or
df.drop(columns = ["PassengerId","Pclass"])
```

Out[232]:

	Survived	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	0	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	1	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	0	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...
886	0	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S

	Survived	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
888	0	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	0	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 10 columns

In [1]:

```
# Drop rows (25,72,63) from the NBA dataset. What is the average age of
# the players after dropping rows?

# Note: Please do not change the original dataset while solving this assignment.

df.drop([25,63])
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 6
      1 # Drop rows (25,72,63) from the NBA dataset. What is the average age of
      2 # the players after dropping rows?
      3
      4 # Note: Please do not change the original dataset while solving this assignment.
----> 6 df.drop([25,63])

NameError: name 'df' is not defined
```

In [299]:

```
df.rename(columns = {"Name": "Chest_Number",
            "Weight":"Weight(lbs)"},inplace= True)
df
```

Out[299]:

	Chest_Number	Team	Number	Position	Age	Height	Weight(lbs)	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	2023-06-02 00:00:00	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	2023-06-06 00:00:00	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	2023-06-05 00:00:00	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	2023-06-05 00:00:00	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	2023-06-10 00:00:00	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
452	Trey Lyles	Utah Jazz	41	PF	20	2023-06-10 00:00:00	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	PG	26	2023-06-03 00:00:00	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	2023-06-01 00:00:00	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	2023-07-03 00:00:00	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0

457 rows × 9 columns

In [3]:

```
df.isna().sum()
df.dropna(how="any",inplace=True)

# Question 3
# 0/1 point (graded)
# Drop rows (25,72,63) from the NBA dataset.
# What is the average age of the players after dropping rows?

drop = df.drop([25,63])
avg = drop["Age"].mean()
```

```
print(avg)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[3], line 1
----> 1 df.isna().sum()
      2 df.dropna(how="any",inplace=True)
      4 # Question 3
      5 # 0/1 point (graded)
      6 # Drop rows (25,72,63) from the NBA dataset.
      7 # What is the average age of the players after dropping rows?

NameError: name 'df' is not defined
```

In [22]:

```
import pandas as pd
df = pd.read_excel("nba.xlsx")
```

In [50]:

```
# df.drop([25,72,63])
dropped = df.drop([26,72,63])
avg = dropped["Age"].mean()
print("Average age after dropping rows :", avg)
```

```
Average age after dropping rows : 26.91850220264317
```

In [43]:

```
dropped = df.drop([24,71,62])
avg = dropped["Age"].mean()
print("Average age after dropping rows :", avg)
```

```
Average age after dropping rows : 26.97136563876652
```

In [47]:

```
nba_df = pd.read_excel("nba.xlsx")
nba_df.head()
```

Out[47]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	2023-06-02 00:00:00	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	2023-06-06 00:00:00	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	2023-06-05 00:00:00	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	2023-06-05 00:00:00	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	2023-06-10 00:00:00	231	NaN	5000000.0

In [45]:

```
df1 = df.drop([5,6])
df1.head(10)
```

Out[45]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	2023-06-02 00:00:00	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	2023-06-06 00:00:00	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	2023-06-05 00:00:00	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	2023-06-05 00:00:00	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	2023-06-10 00:00:00	231	NaN	5000000.0
7	Kelly Olynyk	Boston Celtics	41	C	25	7-0	238	Gonzaga	2165160.0

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
8	Terry Rozier	Boston Celtics	12	PG	22	2023-06-02 00:00:00	190	Louisville	1824360.0
9	Marcus Smart	Boston Celtics	36	PG	22	2023-06-04 00:00:00	220	Oklahoma State	3431040.0
10	Jared Sullinger	Boston Celtics	7	C	24	2023-06-09 00:00:00	260	Ohio State	2569260.0
11	Isaiah Thomas	Boston Celtics	4	PG	27	2023-05-09 00:00:00	185	Washington	6912869.0

In [49]:

```
# Create a copy of the DataFrame
nba_df_copy = nba_df.copy()

# Drop rows 25, 72, and 63 from the copy
nba_df_copy = nba_df_copy.drop(index=[25, 72, 63])

# Calculate the average age of players after dropping rows
average_age = nba_df_copy['Age'].mean()

# Print the result
print(f'Average age of players after dropping rows: {average_age:.2f}')
```

```
Average age of players after dropping rows: 26.92
```

In [ ]: