In [2]:
```
import os
os.getcwd()
```

Out[2]:
```
'C:\\Users\\admin'
```

In [15]:
```
import pandas as pd
df= pd.read_csv('Documents/nba.csv')
```

In [16]:
```
df.shape
```

Out[16]:
```
(300, 8)
```

In [17]:
```
df.head()
```

Out[17]:

	Name	Team	Number	Position	Age	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	NaN	SG	27	NaN	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28.0	NaN	22	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29	231.0	NaN	5000000.0

In [18]:
```
df.tail()
```

Out[18]:

	Name	Team	Number	Position	Age	Weight	College	Salary
295	Kyle Anderson	San Antonio Spurs	1.0	SF	22	230.0	UCLA	1142880.0
296	Matt Bonner	San Antonio Spurs	15.0	C	36	235.0	Florida	947276.0
297	Boris Diaw	San Antonio Spurs	33.0	C	34	250.0	NaN	7500000.0
298	Tim Duncan	San Antonio Spurs	21.0	C	40	250.0	Wake Forest	5250000.0
299	Manu Ginobili	San Antonio Spurs	20.0	SG	38	205.0	NaN	2814000.0

In [19]:
```
df.count()
```

Out[19]:
```
Name        300
Team        300
Number      298
Position    297
Age         300
Weight      297
College     251
Salary      292
dtype: int64
```

In [20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Name      300 non-null    object
 1   Team      300 non-null    object
 2   Number    298 non-null    float64
 3   Position  297 non-null    object
 4   Age       300 non-null    int64
 5   Weight    297 non-null    float64
 6   College   251 non-null    object
 7   Salary    292 non-null    float64
dtypes: float64(3), int64(1), object(4)
memory usage: 18.9+ KB
```

In [21]:
```
df.isnull()
```

Out[21]:

	Name	Team	Number	Position	Age	Weight	College	Salary
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	True	False	False	True	False	True
3	False	False	False	True	False	False	False	False
4	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...
295	False	False	False	False	False	False	False	False
296	False	False	False	False	False	False	False	False
297	False	False	False	False	False	False	True	False
298	False	False	False	False	False	False	False	False
299	False	False	False	False	False	False	True	False

300 rows × 8 columns

In [22]:
```
df.isnull().sum()
```

Out[22]:
```
Name          0
Team          0
Number        2
Position      3
Age           0
Weight        3
College      49
Salary        8
dtype: int64
```

In [23]:
```
df.dropna()
```

Out[23]:

	Name	Team	Number	Position	Age	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25	235.0	Marquette	6796117.0
7	Kelly Olynyk	Boston Celtics	41.0	C	25	238.0	Gonzaga	2165160.0
9	Marcus Smart	Boston Celtics	36.0	PG	22	220.0	Oklahoma State	3431040.0
11	Isaiah Thomas	Boston Celtics	4.0	PG	27	185.0	Washington	6912869.0
...	...	...	...	...	...	...	...	...
293	Quincy Pondexter	New Orleans Pelicans	20.0	SF	28	220.0	Washington	3382023.0
294	LaMarcus Aldridge	San Antonio Spurs	12.0	PF	30	240.0	Texas	19689000.0
295	Kyle Anderson	San Antonio Spurs	1.0	SF	22	230.0	UCLA	1142880.0
296	Matt Bonner	San Antonio Spurs	15.0	C	36	235.0	Florida	947276.0
298	Tim Duncan	San Antonio Spurs	21.0	C	40	250.0	Wake Forest	5250000.0

239 rows × 8 columns

In [24]:
```
df.fillna(0)
```

Out[24]:

	Name	Team	Number	Position	Age	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	0.0	SG	27	0.0	Boston University	0.0
3	R.J. Hunter	Boston Celtics	28.0	0	22	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29	231.0	0	5000000.0
...	...	...	...	...	...	...	...	...
295	Kyle Anderson	San Antonio Spurs	1.0	SF	22	230.0	UCLA	1142880.0
296	Matt Bonner	San Antonio Spurs	15.0	C	36	235.0	Florida	947276.0
297	Boris Diaw	San Antonio Spurs	33.0	C	34	250.0	0	7500000.0
298	Tim Duncan	San Antonio Spurs	21.0	C	40	250.0	Wake Forest	5250000.0
299	Manu Ginobili	San Antonio Spurs	20.0	SG	38	205.0	0	2814000.0

300 rows × 8 columns

In [25]:
```
#only using class column
df['Position'].fillna('None')
```

Out[25]:
```
0        PG
1        SF
2        SG
3      None
4        PF
       ...
295      SF
296       C
297       C
298       C
299      SG
Name: Position, Length: 300, dtype: object
```

In [27]:
```
df['Weight'].fillna(df['Weight'].mean())
```

Out[27]:
```
0      180.000000
1      235.000000
2      222.205387
3      185.000000
4      231.000000
          ...
295    230.000000
296    235.000000
297    250.000000
298    250.000000
299    205.000000
Name: Weight, Length: 300, dtype: float64
```

In [28]:
```
df['Age'].fillna(df['Age'].median())
```

Out[28]:
```
0      25
1      25
2      27
3      22
4      29
       ..
295    22
296    36
297    34
298    40
299    38
Name: Age, Length: 300, dtype: int64
```

In [31]:
```
df['Position'].value_counts()
```

Out[31]:
```
Position
PF    66
SG    63
PG    62
SF    57
C     49
Name: count, dtype: int64
```

In [33]:
```
df['Age'].fillna(df['Age'].mode()[0])
```

Out[33]:
```
0      25
1      25
2      27
3      22
4      29
       ..
295    22
296    36
297    34
298    40
299    38
Name: Age, Length: 300, dtype: int64
```

In [34]:
```
df.fillna(method='backfill')
```

Out[34]:

	Name	Team	Number	Position	Age	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	28.0	SG	27	185.0	Boston University	1148640.0
3	R.J. Hunter	Boston Celtics	28.0	PF	22	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29	231.0	LSU	5000000.0
...	...	...	...	...	...	...	...	...
295	Kyle Anderson	San Antonio Spurs	1.0	SF	22	230.0	UCLA	1142880.0
296	Matt Bonner	San Antonio Spurs	15.0	C	36	235.0	Florida	947276.0
297	Boris Diaw	San Antonio Spurs	33.0	C	34	250.0	Wake Forest	7500000.0
298	Tim Duncan	San Antonio Spurs	21.0	C	40	250.0	Wake Forest	5250000.0
299	Manu Ginobili	San Antonio Spurs	20.0	SG	38	205.0	NaN	2814000.0

300 rows × 8 columns

In [35]:
```
df.fillna(method='pad')
```

Out[35]:

	Name	Team	Number	Position	Age	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	99.0	SG	27	235.0	Boston University	6796117.0
3	R.J. Hunter	Boston Celtics	28.0	SG	22	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29	231.0	Georgia State	5000000.0
...	...	...	...	...	...	...	...	...
295	Kyle Anderson	San Antonio Spurs	1.0	SF	22	230.0	UCLA	1142880.0
296	Matt Bonner	San Antonio Spurs	15.0	C	36	235.0	Florida	947276.0
297	Boris Diaw	San Antonio Spurs	33.0	C	34	250.0	Florida	7500000.0
298	Tim Duncan	San Antonio Spurs	21.0	C	40	250.0	Wake Forest	5250000.0
299	Manu Ginobili	San Antonio Spurs	20.0	SG	38	205.0	Wake Forest	2814000.0

300 rows × 8 columns

In [36]:
```
df.describe()
```

Out[36]:

	Number	Age	Weight	Salary
count	298.000000	300.000000	297.000000	2.920000e+02
mean	18.164430	27.050000	222.205387	4.845753e+06
std	16.013351	4.354773	26.353142	5.385895e+06
min	0.000000	19.000000	161.000000	3.088800e+04
25%	6.000000	24.000000	200.000000	1.005270e+06
50%	14.000000	27.000000	220.000000	2.814000e+06
75%	27.750000	30.000000	242.000000	6.649029e+06
max	99.000000	40.000000	279.000000	2.500000e+07

In [39]:
```
import numpy as np
x= np.array([5,6,3,4,7,5,2,3,89,90,99])
```

In [41]:
```
np.mean(x)
```

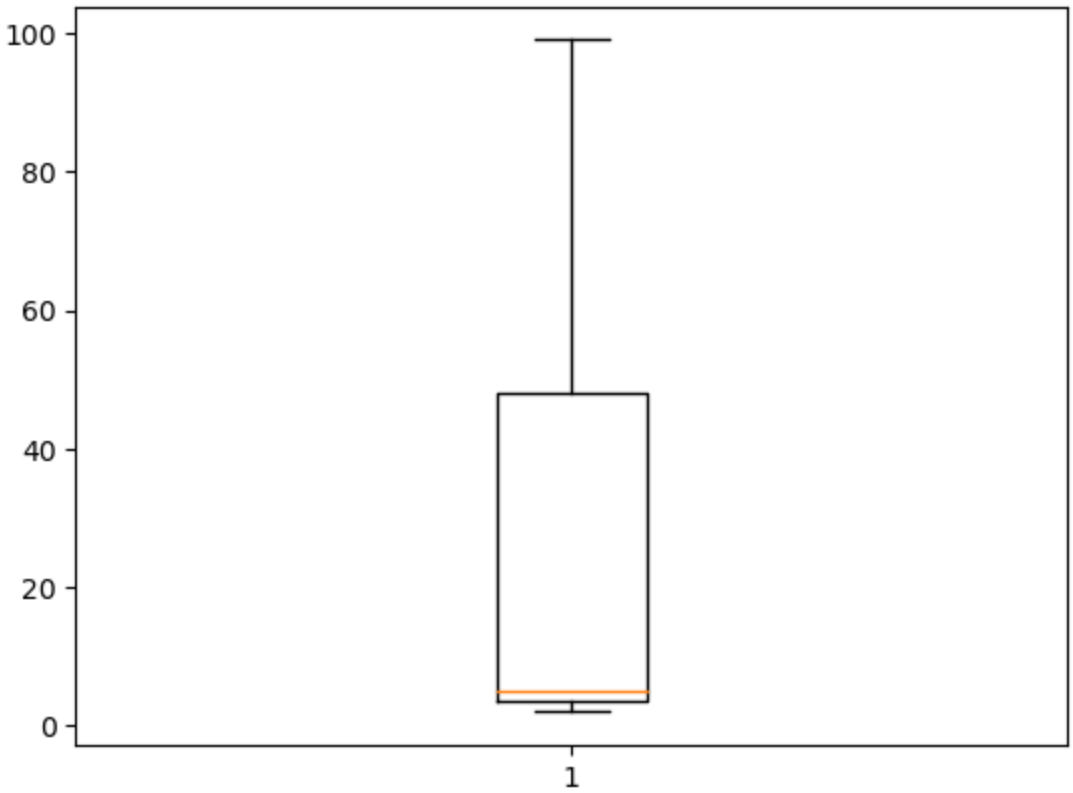
Out[41]: 28.454545454545453

In [42]:
```
np.median(x)
```

Out[42]: 5.0

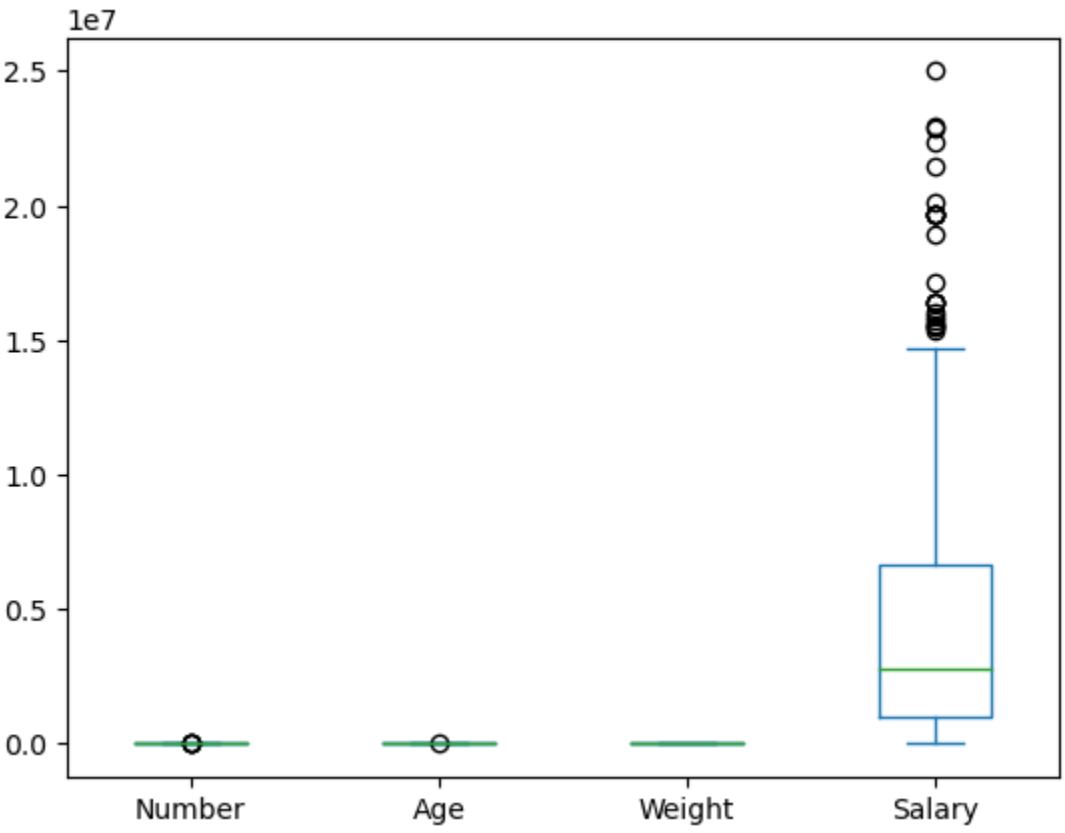
In [44]:
```
import matplotlib.pyplot as plt
```

In [45]:
```
plt.boxplot(x);
```



In [46]:
```
df.plot.box()
```

Out[46]: <Axes: >

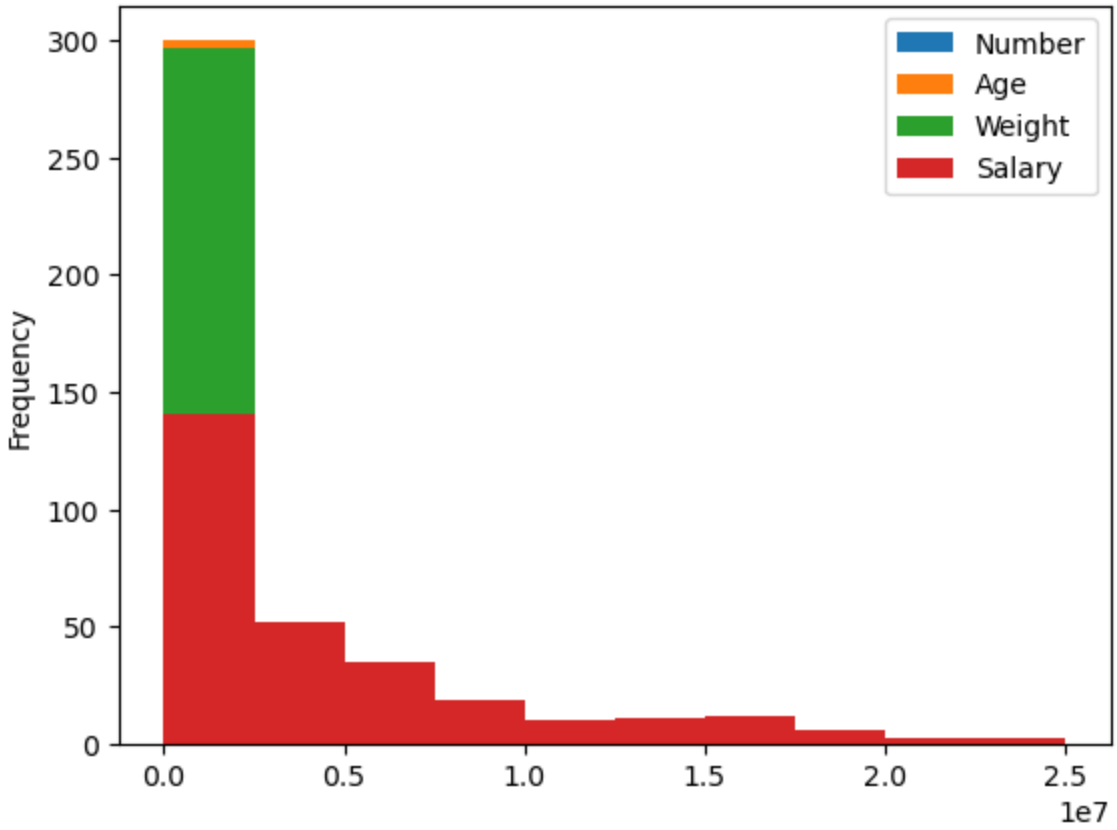


In [47]:
```
df.loc[6,'Age']
```

Out[47]: 21

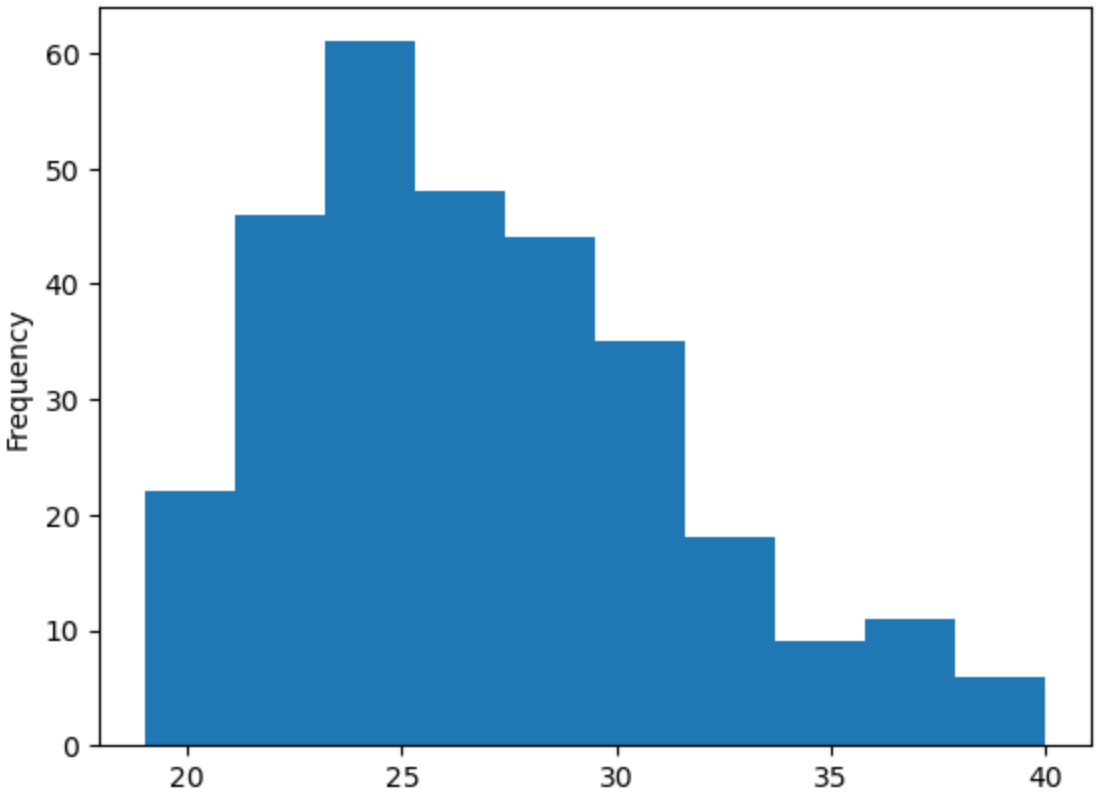
In [48]:
```
df.plot.hist()
```

Out[48]: <Axes: ylabel='Frequency'>



In [49]:
```
df['Age'].plot.hist()
```

Out[49]: <Axes: ylabel='Frequency'>



In [51]:
```
x= df[['Age','Salary']]
x.describe()
```

Out[51]:

	Age	Salary
count	300.000000	2.920000e+02
mean	27.050000	4.845753e+06
std	4.354773	5.385895e+06
min	19.000000	3.088800e+04
25%	24.000000	1.005270e+06
50%	27.000000	2.814000e+06
75%	30.000000	6.649029e+06
max	40.000000	2.500000e+07

In [52]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x_scaled = scaler.fit_transform(x)
```

In [53]:
```
pd.DataFrame(x_scaled).describe()
```

Out[53]:

	0	1
count	300.000000	292.000000
mean	0.383333	0.192833
std	0.207370	0.215702
min	0.000000	0.000000
25%	0.238095	0.039023
50%	0.380952	0.111462
75%	0.523810	0.265053
max	1.000000	1.000000

In [54]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
x_scaled = scaler.fit_transform(x)
```

In [55]:
```
pd.DataFrame(x_scaled).describe()
```

Out[55]:

	0	1
count	3.000000e+02	2.920000e+02
mean	-1.598721e-16	-6.083414e-17
std	1.001671e+00	1.001717e+00
min	-1.851635e+00	-8.955114e-01
25%	-7.015512e-01	-7.142873e-01
50%	-1.150084e-02	-3.778835e-01
75%	6.785495e-01	3.353894e-01
max	2.978717e+00	3.748467e+00