In [1]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

In [2]:
```
 abc=pd.read_csv("C:\\Users\\SHONIMA S\\Downloads\\myexcel - myexcel.csv.csv")
abc
```

Out[2]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	06-Mar	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	06-Jan	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	07-Mar	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0
457	Priyanka	Utah Jazz	34	C	25	07-Mar	231	Kansas	947276.0

458 rows × 9 columns

In [3]:
```
print(abc.describe())
```

```
           Number         Age      Weight        Salary
count  458.000000  458.000000  458.000000  4.470000e+02
mean    17.713974   26.934498  221.543668  4.833970e+06
std     15.966837    4.400128   26.343200  5.226620e+06
min      0.000000   19.000000  161.000000  3.088800e+04
25%      5.000000   24.000000  200.000000  1.025210e+06
50%     13.000000   26.000000  220.000000  2.836186e+06
75%     25.000000   30.000000  240.000000  6.500000e+06
max     99.000000   40.000000  307.000000  2.500000e+07
```

In [4]:
```
print(abc.isnull().sum())
```

```
Name         0
Team         0
Number       0
Position     0
Age          0
Height       0
Weight       0
College     84
Salary      11
dtype: int64
```

In [5]:
```
abc.duplicated().sum()
```

Out[5]:
```
0
```

In [ ]:
```
Preprocessing:
Correct the data in the "height" column by replacing it with random numbers between 150 and 180. Ensure data consistency and integrity before proceeding with analysis.
```

In [6]:
```
abc["Height"]=abc["Height"].replace(["06-Sep"],167)
abc["Height"]=abc["Height"].replace(["06-Oct"],170)
abc["Height"]=abc["Height"].replace(["06-Jul"],172)
abc["Height"]=abc["Height"].replace(["06-Aug"],157)
abc["Height"]=abc["Height"].replace(["06-Jun"],162)
abc["Height"]=abc["Height"].replace(["06-Nov"],174)
abc["Height"]=abc["Height"].replace(["06-Mar"],155)
abc["Height"]=abc["Height"].replace(["06-May"],160)
abc["Height"]=abc["Height"].replace(["06-Apr"],164)
abc["Height"]=abc["Height"].replace(["7-0"],178)
abc["Height"]=abc["Height"].replace(["06-Jan"],166)
abc["Height"]=abc["Height"].replace(["06-Feb"],169)
abc["Height"]=abc["Height"].replace(["6-0"],159)
abc["Height"]=abc["Height"].replace(["07-Jan"],177)
abc["Height"]=abc["Height"].replace(["07-Mar"],168)
abc["Height"]=abc["Height"].replace(["05-Nov"],175)
abc["Height"]=abc["Height"].replace(["07-Feb"],173)
abc["Height"]=abc["Height"].replace(["05-Sep"],179)
```

In [7]:
```
abc["Height"].value_counts()
```

Out[7]:
```
167    59
170    47
172    45
157    43
162    42
174    40
155    33
160    32
164    29
178    27
166    16
169    16
159    10
177     7
168     5
175     3
173     3
179     1
Name: Height, dtype: int64
```

In [ ]:
```
Analysis Tasks:
1. Determine the distribution of employees across each team and calculate the percentage split relative to the total number of employees.
```

In [8]:

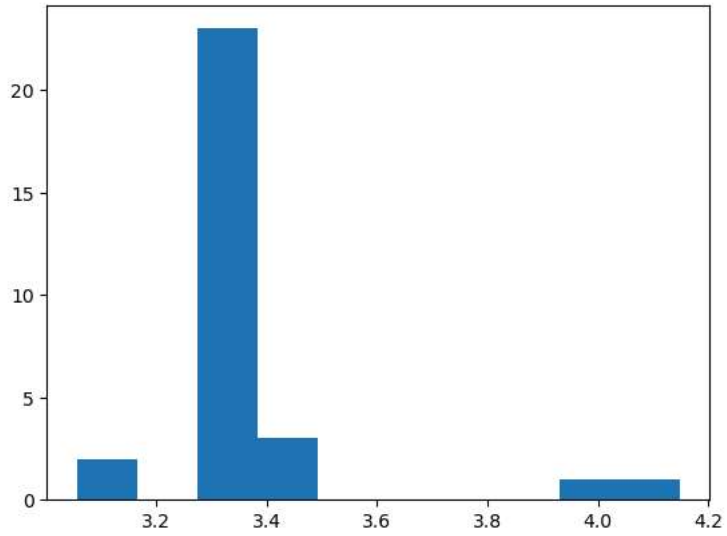
```
emp=abc["Team"].value_counts()
print(emp)
tot=emp.sum()
print(tot)
percentage=(emp/tot)*100
print(percentage)
```

```
New Orleans Pelicans      19
Memphis Grizzlies         18
Utah Jazz                 16
New York Knicks           16
Milwaukee Bucks           16
Brooklyn Nets             15
Portland Trail Blazers    15
Oklahoma City Thunder     15
Denver Nuggets            15
Washington Wizards        15
Miami Heat                15
Charlotte Hornets         15
Atlanta Hawks             15
San Antonio Spurs         15
Houston Rockets           15
Boston Celtics            15
Indiana Pacers            15
Detroit Pistons           15
Cleveland Cavaliers       15
Chicago Bulls             15
Sacramento Kings          15
Phoenix Suns              15
Los Angeles Lakers        15
Los Angeles Clippers      15
Golden State Warriors     15
Toronto Raptors           15
Philadelphia 76ers        15
Dallas Mavericks          15
Orlando Magic             14
Minnesota Timberwolves    14
Name: Team, dtype: int64
458
New Orleans Pelicans      4.148472
Memphis Grizzlies         3.930131
Utah Jazz                 3.493450
New York Knicks           3.493450
Milwaukee Bucks           3.493450
Brooklyn Nets             3.275109
Portland Trail Blazers    3.275109
Oklahoma City Thunder     3.275109
Denver Nuggets            3.275109
Washington Wizards        3.275109
Miami Heat                3.275109
Charlotte Hornets         3.275109
Atlanta Hawks             3.275109
San Antonio Spurs         3.275109
Houston Rockets           3.275109
Boston Celtics            3.275109
Indiana Pacers            3.275109
Detroit Pistons           3.275109
Cleveland Cavaliers       3.275109
Chicago Bulls             3.275109
Sacramento Kings          3.275109
Phoenix Suns              3.275109
Los Angeles Lakers        3.275109
Los Angeles Clippers      3.275109
Golden State Warriors     3.275109
Toronto Raptors           3.275109
Philadelphia 76ers        3.275109
Dallas Mavericks          3.275109
Orlando Magic             3.056769
Minnesota Timberwolves    3.056769
Name: Team, dtype: float64
```

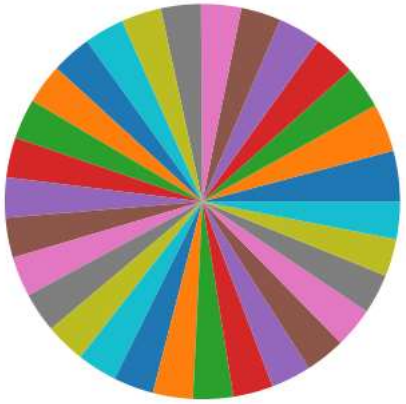
In [9]:
```
percentage.value_counts()
```

Out[9]:
```
3.275109    23
3.493450     3
3.056769     2
4.148472     1
3.930131     1
Name: Team, dtype: int64
```

In [10]:
```
plt.hist(percentage)
plt.show()
```



In [11]:
```
plt.pie(percentage)
plt.show()
```



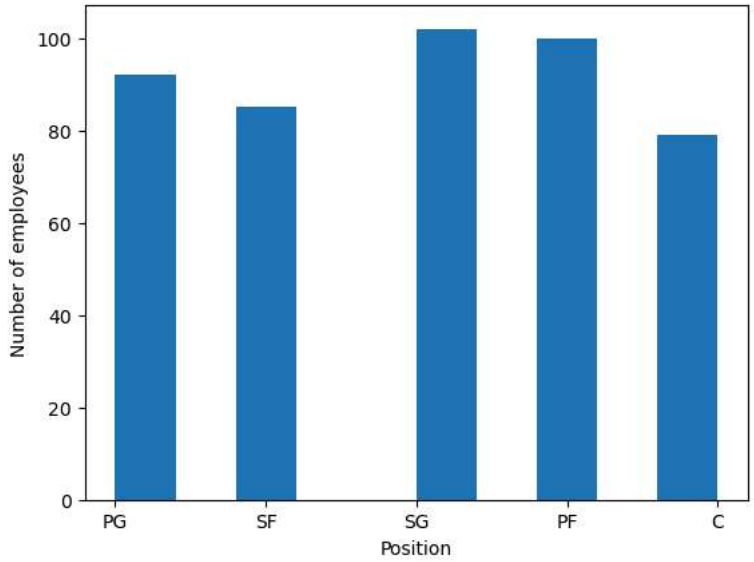
In [ ]:
```
insight:team New Orleans Pelicans have most number of employees with 19 employees and their percentage split is 4.148472%.twenty three teams have the same percentage split.So
```

In [ ]:
```
2. Segregate employees based on their positions within the company.
```

In [12]:
```
t1=abc["Position"].value_counts()
print(t1)
```

```
SG    102
PF    100
PG     92
SF     85
C      79
Name: Position, dtype: int64
```

In [13]:
```
pos=abc["Position"]
age=abc["Age"]
salary=abc["Salary"]
plt.hist(pos)
plt.xlabel("Position")
plt.ylabel("Number of employees")
plt.show()
```



In [14]:
```
abc[abc["Position"]=="SG"]
```

Out[14]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
2	John Holland	Boston Celtics	30	SG	27	160	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	160	185	Georgia State	1148640.0
12	Evan Turner	Boston Celtics	11	SG	27	172	220	Ohio State	3425510.0
13	James Young	Boston Celtics	13	SG	20	162	215	Kentucky	1749840.0
15	Bojan Bogdanovic	Brooklyn Nets	44	SG	27	157	216	NaN	3425510.0
...	...	...	...	...	...	...	...	...	...
433	Gerald Henderson	Portland Trail Blazers	9	SG	28	160	215	Duke	6000000.0
437	C.J. McCollum	Portland Trail Blazers	3	SG	24	164	200	Lehigh	2525160.0
438	Luis Montero	Portland Trail Blazers	44	SG	23	172	185	Westchester CC	525093.0
444	Alec Burks	Utah Jazz	10	SG	24	162	214	Colorado	9463484.0
449	Rodney Hood	Utah Jazz	5	SG	23	157	206	Duke	1348440.0

102 rows × 9 columns

In [15]:
```
abc[abc["Position"]=="C"]
```

Out[15]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
7	Kelly Olynyk	Boston Celtics	41	C	25	178	238	Gonzaga	2165160.0
10	Jared Sullinger	Boston Celtics	7	C	24	167	260	Ohio State	2569260.0
14	Tyler Zeller	Boston Celtics	44	C	26	178	253	North Carolina	2616975.0
23	Brook Lopez	Brooklyn Nets	11	C	28	178	275	Stanford	19689000.0
27	Henry Sims	Brooklyn Nets	14	C	26	170	248	Georgetown	947276.0
...	...	...	...	...	...	...	...	...	...
439	Mason Plumlee	Portland Trail Blazers	24	C	26	174	235	Duke	1415520.0
447	Rudy Gobert	Utah Jazz	27	C	23	177	245	NaN	1175880.0
455	Tibor Pleiss	Utah Jazz	21	C	26	168	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	178	231	Kansas	947276.0
457	Priyanka	Utah Jazz	34	C	25	168	231	Kansas	947276.0

79 rows × 9 columns

In [16]:
```
abc[abc["Position"]=="SF"]
```

Out[16]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
1	Jae Crowder	Boston Celtics	99	SF	25	162	235	Marquette	6796117.0
32	Thanasis Antetokounmpo	New York Knicks	43	SF	23	172	205	NaN	30888.0
33	Carmelo Anthony	New York Knicks	7	SF	32	157	240	Syracuse	22875000.0
35	Cleanthony Early	New York Knicks	11	SF	25	157	210	Wichita State	845059.0
42	Lance Thomas	New York Knicks	42	SF	28	157	235	Duke	1636842.0
...	...	...	...	...	...	...	...	...	...
428	Al-Farouq Aminu	Portland Trail Blazers	8	SF	25	167	215	Wake Forest	8042895.0
432	Maurice Harkless	Portland Trail Blazers	4	SF	23	167	215	St. John's	2894059.0
448	Gordon Hayward	Utah Jazz	20	SF	26	157	226	Butler	15409570.0
450	Joe Ingles	Utah Jazz	2	SF	28	157	226	NaN	2050000.0
451	Chris Johnson	Utah Jazz	23	SF	26	162	206	Dayton	981348.0

85 rows × 9 columns

In [17]:
```
abc[abc["Position"]=="PG"]
```

Out[17]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	169	180	Texas	7730337.0
8	Terry Rozier	Boston Celtics	12	PG	22	169	190	Louisville	1824360.0
9	Marcus Smart	Boston Celtics	36	PG	22	164	220	Oklahoma State	3431040.0
11	Isaiah Thomas	Boston Celtics	4	PG	27	179	185	Washington	6912869.0
19	Jarrett Jack	Brooklyn Nets	2	PG	32	155	200	Georgia Tech	6300000.0
...	...	...	...	...	...	...	...	...	...
440	Brian Roberts	Portland Trail Blazers	2	PG	30	166	173	Dayton	2854940.0
443	Trey Burke	Utah Jazz	3	PG	23	166	191	Michigan	2658240.0
445	Dante Exum	Utah Jazz	11	PG	20	162	190	NaN	3777720.0
453	Shelvin Mack	Utah Jazz	8	PG	26	155	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	166	179	NaN	900000.0

92 rows × 9 columns

In [18]:
```
abc[abc["Position"]=="PF"]
```

Out[18]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
4	Jonas Jerebko	Boston Celtics	8	PF	29	170	231	NaN	5000000.0
5	Amir Johnson	Boston Celtics	90	PF	29	167	240	NaN	12000000.0
6	Jordan Mickey	Boston Celtics	55	PF	21	157	235	LSU	1170960.0
24	Chris McCullough	Brooklyn Nets	1	PF	21	174	200	Syracuse	1140240.0
25	Willie Reed	Brooklyn Nets	33	PF	26	170	220	Saint Louis	947276.0
...	...	...	...	...	...	...	...	...	...
435	Meyers Leonard	Portland Trail Blazers	11	PF	24	177	245	Illinois	3075880.0
441	Noah Vonleh	Portland Trail Blazers	21	PF	20	167	240	Indiana	2637720.0
442	Trevor Booker	Utah Jazz	33	PF	28	157	228	Clemson	4775000.0
446	Derrick Favors	Utah Jazz	15	PF	24	170	265	Georgia Tech	12000000.0
452	Trey Lyles	Utah Jazz	41	PF	20	170	234	Kentucky	2239800.0

100 rows × 9 columns

In [ ]:
```
insight:position SG has most number of employees with 102 employees,secondmost employees are in PF position with 100 employees,PG position has 92 employees,SF has 85 and C has
```

In [ ]:
```
3. Identify the predominant age group among employees.
```

In [19]:
```
abc["Age"].value_counts()
```

Out[19]:
```
24    47
25    46
27    41
23    41
26    36
28    31
30    31
29    28
22    26
31    22
20    19
21    19
33    14
32    13
34    10
36    10
35     9
37     4
38     4
40     3
39     2
19     2
Name: Age, dtype: int64
```

In [20]:
```
sns.distplot(age)
plt.show()
```

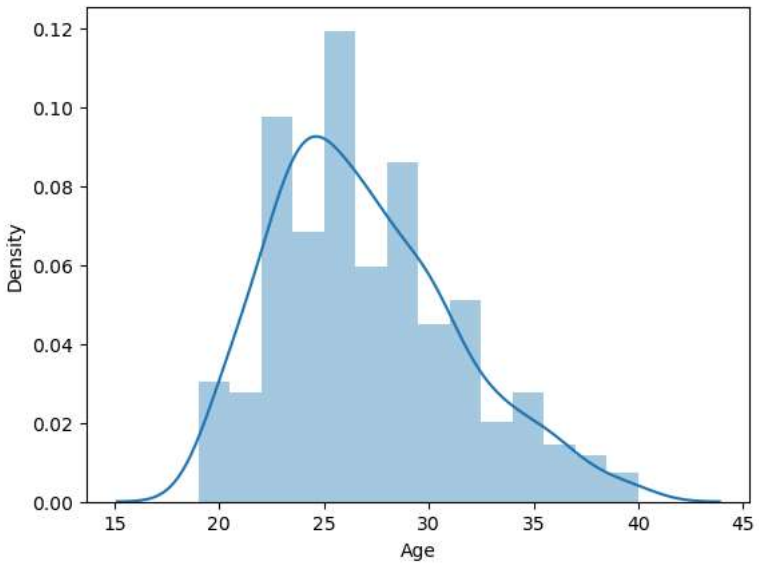
```
C:\Users\SHONIMA S\AppData\Local\Temp\ipykernel_15316\2912825138.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(age)
```



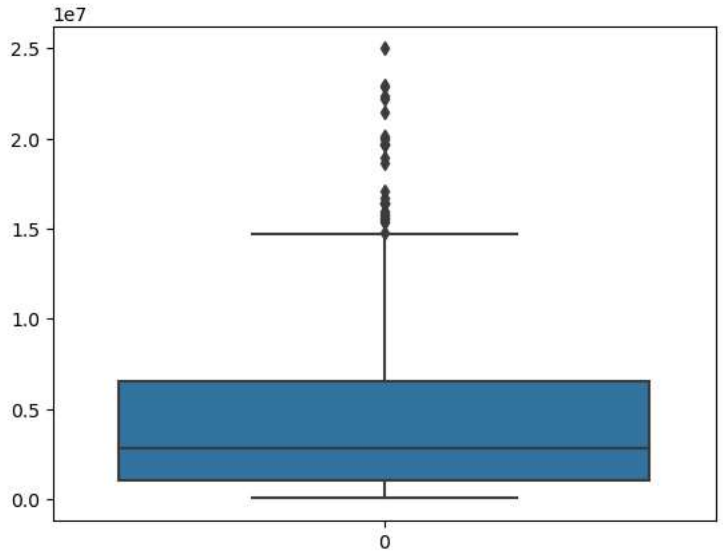
In [ ]:
```
insight: predominant age group is 24.the number of employees under 30 years of age are more.It shows that most of the employees are young people aged between 20 to 33.
```

In [ ]:
```
4. Discover which team and position have the highest salary expenditure.
```

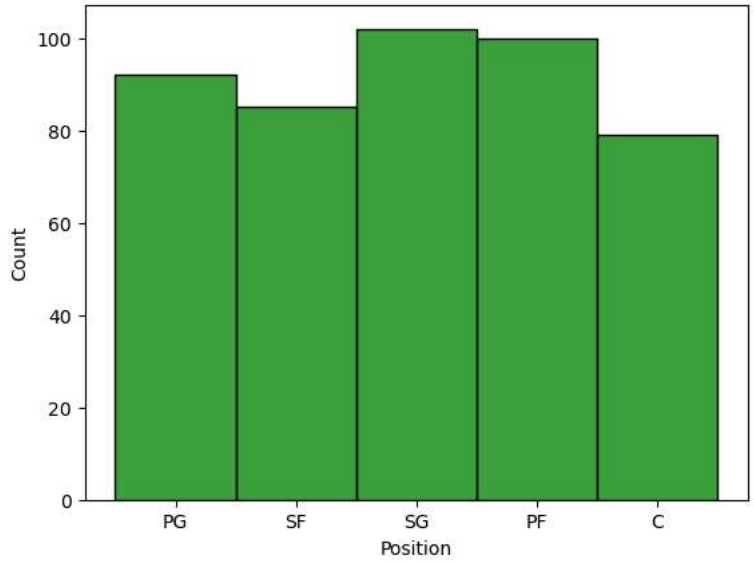
In [21]:
```
abc[["Salary","Position","Team"]].max()
```

Out[21]:
```
Salary              25000000.0
Position                    SG
Team        Washington Wizards
dtype: object
```

In [22]:
```
sns.boxplot(salary)
plt.show()
```



In [23]:
```
sns.histplot(pos,color="g")
plt.show()
```



In [24]:
```
abc["Position"].value_counts()
```

Out[24]:
```
SG    102
PF    100
PG     92
SF     85
C      79
Name: Position, dtype: int64
```

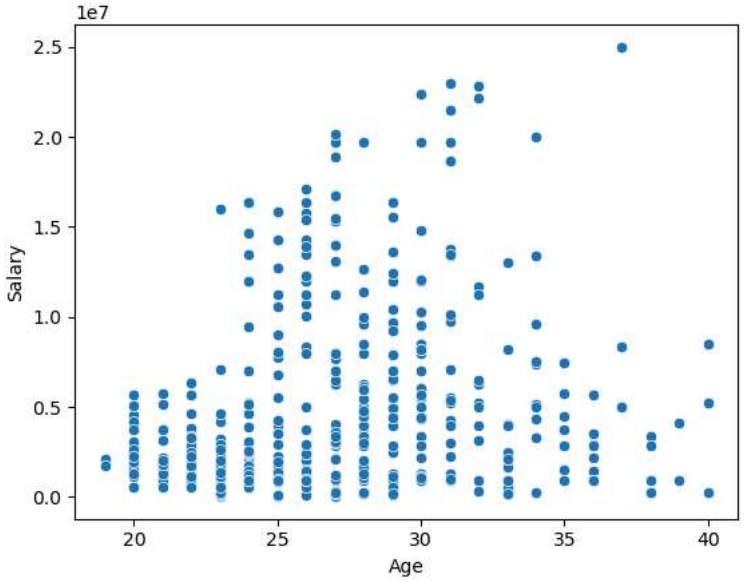
In [ ]:
```
insight:Washington Wizards has highest salary expenditure and the most salary expenditure is for the position "SG"
```

In [ ]:
```
5. Investigate if there's any correlation between age and salary, and represent it visually.
```

In [25]:
```
print(abc.describe())
```

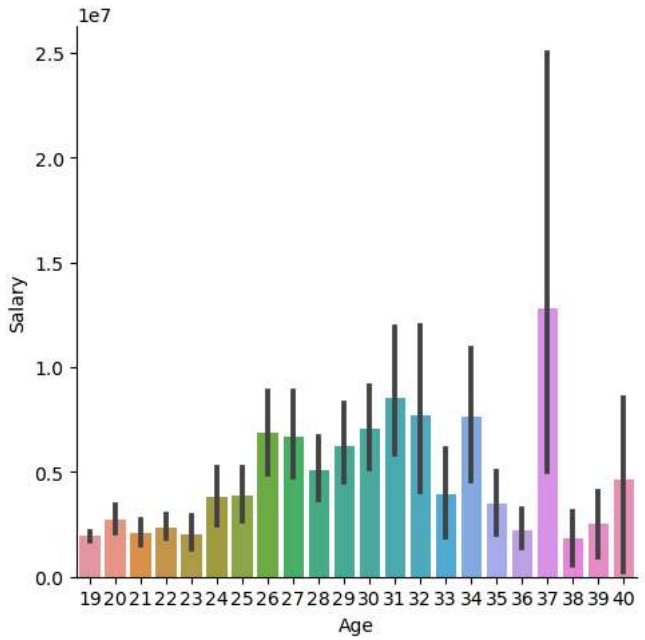
```
           Number         Age      Height      Weight        Salary
count  458.000000  458.000000  458.000000  458.000000  4.470000e+02
mean    17.713974   26.934498  166.259825  221.543668  4.833970e+06
std     15.966837    4.400128    6.668795   26.343200  5.226620e+06
min      0.000000   19.000000  155.000000  161.000000  3.088800e+04
25%      5.000000   24.000000  160.000000  200.000000  1.025210e+06
50%     13.000000   26.000000  167.000000  220.000000  2.836186e+06
75%     25.000000   30.000000  172.000000  240.000000  6.500000e+06
max     99.000000   40.000000  179.000000  307.000000  2.500000e+07
```

In [26]:
```
sns.scatterplot(x="Age",y="Salary",data=abc)
plt.show()
```



In [27]:
```
sns.catplot(x="Age",y="Salary",data=abc,kind="bar")
plt.show()
```

```
C:\ProgramData\anaconda3\lib\site-packages\seaborn\algorithms.py:98: RuntimeWarning: Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
```



In [ ]:
```
insight:age group belonging to 26-34 have comparatively higher salary than rest of the age group.One exception is the employees having age 37 who have the highest salary than
```

In [28]:
```
abc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 458 entries, 0 to 457
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Name      458 non-null    object
 1   Team      458 non-null    object
 2   Number    458 non-null    int64
 3   Position  458 non-null    object
 4   Age       458 non-null    int64
 5   Height    458 non-null    int64
 6   Weight    458 non-null    int64
 7   College   374 non-null    object
 8   Salary    447 non-null    float64
dtypes: float64(1), int64(4), object(4)
memory usage: 32.3+ KB
```

In [29]:
```
abc.loc[:,abc.isnull().any()]
```

Out[29]:

	College	Salary
0	Texas	7730337.0
1	Marquette	6796117.0
2	Boston University	NaN
3	Georgia State	1148640.0
4	NaN	5000000.0
...	...	...
453	Butler	2433333.0
454	NaN	900000.0
455	NaN	2900000.0
456	Kansas	947276.0
457	Kansas	947276.0

458 rows × 2 columns

In [ ]: