In [1]:
```
import pandas as pd

# Load the dataset
dataset = "https://docs.google.com/spreadsheets/d/1VP9BE_eI2yl6uUHSm4mGiiw
data = pd.read_csv(dataset)


print(data)
```

```
              Name            Team  Number Position  Age  Height  Weight  \
0    Avery Bradley  Boston Celtics       0       PG   25  06-Feb     180
1      Jae Crowder  Boston Celtics      99       SF   25  06-Jun     235
2     John Holland  Boston Celtics      30       SG   27  06-May     205
3      R.J. Hunter  Boston Celtics      28       SG   22  06-May     185
4    Jonas Jerebko  Boston Celtics       8       PF   29  06-Oct     231
..             ...             ...     ...      ...  ...     ...     ...
453   Shelvin Mack       Utah Jazz       8       PG   26  06-Mar     203
454      Raul Neto       Utah Jazz      25       PG   24  06-Jan     179
455   Tibor Pleiss       Utah Jazz      21        C   26  07-Mar     256
456    Jeff Withey       Utah Jazz      24        C   26     7-0     231
457       Priyanka       Utah Jazz      34        C   25  07-Mar     231

               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 x 9 columns]
```

In [3]:
```
import numpy as np
data.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    object 
 6   Weight    458 non-null    int64  
 7   College   374 non-null    object 
 8   Salary    447 non-null    float64
dtypes: float64(1), int64(3), object(5)
memory usage: 32.3+ KB
```

In [3]:
```
data.isnull().sum()
```

Out[3]:
```
Name         0
Team         0
Number       0
Position     0
Age          0
Height       0
Weight       0
College     84
Salary      11
dtype: int64
```

In [4]:
```
data.count
```

Out[4]:
```
<bound method DataFrame.count of              Name            Team  Num
ber Position  Age  Height  Weight  \
0    Avery Bradley  Boston Celtics       0       PG   25  06-Feb     180
1      Jae Crowder  Boston Celtics      99       SF   25  06-Jun     235
2     John Holland  Boston Celtics      30       SG   27  06-May     205
3      R.J. Hunter  Boston Celtics      28       SG   22  06-May     185
4    Jonas Jerebko  Boston Celtics       8       PF   29  06-Oct     231
..             ...             ...     ...      ...  ...     ...     ...
453   Shelvin Mack       Utah Jazz       8       PG   26  06-Mar     203
454      Raul Neto       Utah Jazz      25       PG   24  06-Jan     179
455   Tibor Pleiss       Utah Jazz      21        C   26  07-Mar     256
456    Jeff Withey       Utah Jazz      24        C   26     7-0     231
457       Priyanka       Utah Jazz      34        C   25  07-Mar     231

               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 x 9 columns]>
```

In [5]:
```
data.describe()
```

Out[5]:

	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 [3]:
```
data.duplicated().sum()
```

Out[3]: 0

In [6]:
```
data.drop_duplicates()
```

Out[6]:

	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 [7]:
```
data.index
```

Out[7]:
```
RangeIndex(start=0, stop=458, step=1)
```

In [9]:

```
# Replace height values with random numbers between 150 and 180
data['height'] = np.random.randint(150, 181, size=len(data))data
```

In [10]:

```
data
```

Out[10]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary	height
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7730337.0	179
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6796117.0	156
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN	177
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1148640.0	176
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	NaN	5000000.0	167
...	...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	06-Mar	203	Butler	2433333.0	176
454	Raul Neto	Utah Jazz	25	PG	24	06-Jan	179	NaN	900000.0	152
455	Tibor Pleiss	Utah Jazz	21	C	26	07-Mar	256	NaN	2900000.0	160
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0	167
457	Priyanka	Utah Jazz	34	C	25	07-Mar	231	Kansas	947276.0	155

458 rows × 10 columns

In [11]:

```
#1. Determine the distribution of employees across each team and calculate
```

In [6]:

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

In [17]:
```
data["Team"].value_counts()
```

Out[17]:
```
Team
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: count, dtype: int64
```

In [19]:
```
# % spliting with respect to the total employees
data['Team'].value_counts()/len(data)*100
```

Out[19]:
```
Team
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: count, dtype: float64
```

In [6]:
```
# 2. Segregate employees based on their positions within the company.
employees=data.groupby('Position') ['Name'].apply(list)
for Position, Names in employees.items():
    print(f"employees in {Position} positions:")
    for name in Names:
        print("\n",name)
```

```
employees in C positions:

 Kelly Olynyk

 Jared Sullinger

 Tyler Zeller

 Brook Lopez

 Henry Sims

 Robin Lopez

 Kevin Seraphin

 Joel Embiid

 Jahlil Okafor
```

In [7]:
```
#3. Identify the  predominantage group among employees
```

In [4]:
```
data['Age Group']=data['Age'].apply(lambda age:'20-29' if 20 <= age < 30 e
```

In [5]:
```
data
```

Out[5]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary	Age Group
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7730337.0	20-29
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6796117.0	20-29
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN	20-29
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1148640.0	20-29
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	NaN	5000000.0	20-29
...	...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	06-Mar	203	Butler	2433333.0	20-29
454	Raul Neto	Utah Jazz	25	PG	24	06-Jan	179	NaN	900000.0	20-29
455	Tibor Pleiss	Utah Jazz	21	C	26	07-Mar	256	NaN	2900000.0	20-29
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0	20-29
457	Priyanka	Utah Jazz	34	C	25	07-Mar	231	Kansas	947276.0	20-29

458 rows × 10 columns

In [6]:
```
data['Age Group'].value_counts()
```

Out[6]:
```
Age Group
20-29    334
30-39    119
40-49      3
50+        2
Name: count, dtype: int64
```

In [3]:
```
#4 Discover which term and position have the highest salary expenditure
spending_salary=data.groupby(['Team','Position'])['Salary'].sum()
spending_salary.idxmax()
```

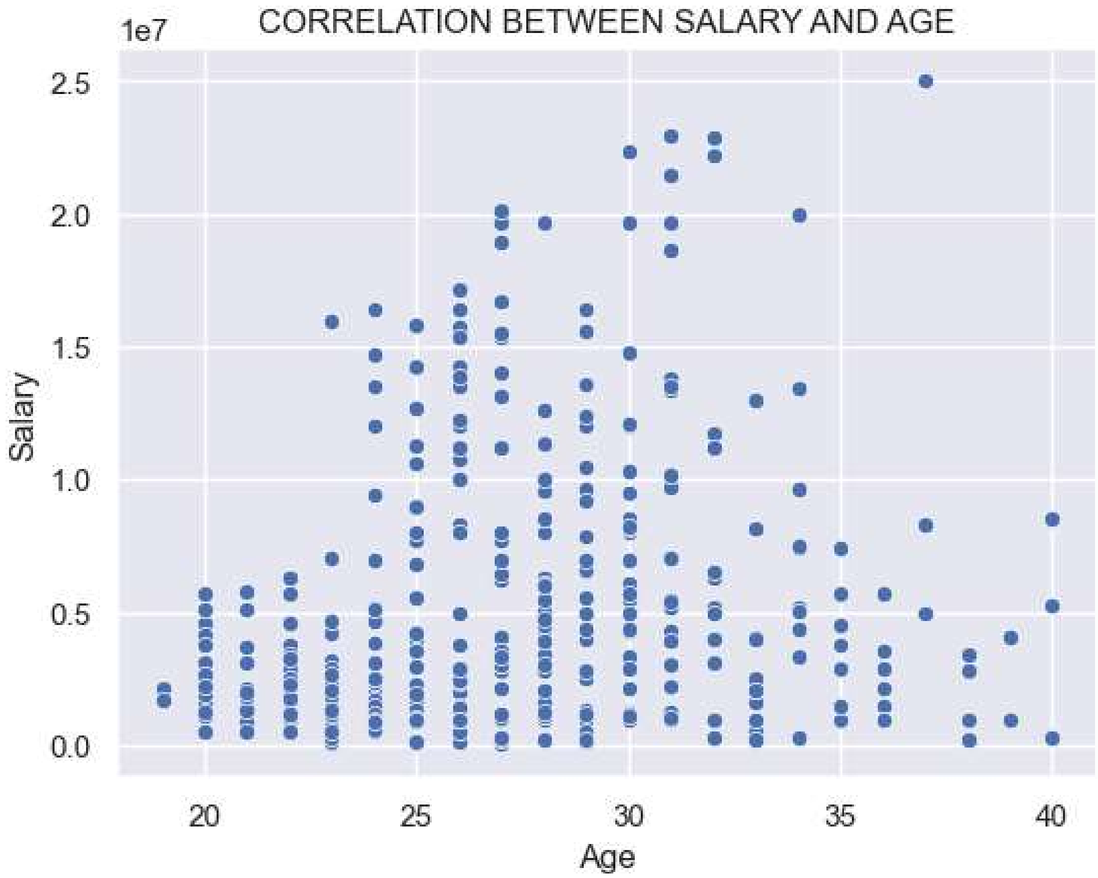
Out[3]:
```
('Los Angeles Lakers', 'SF')
```

In [9]:
```
#5.Investigate if there's any correlation between age and salary and repre
correction = data['Salary'].corr(data['Age'])
print("The correlation between Salary and age is")
```

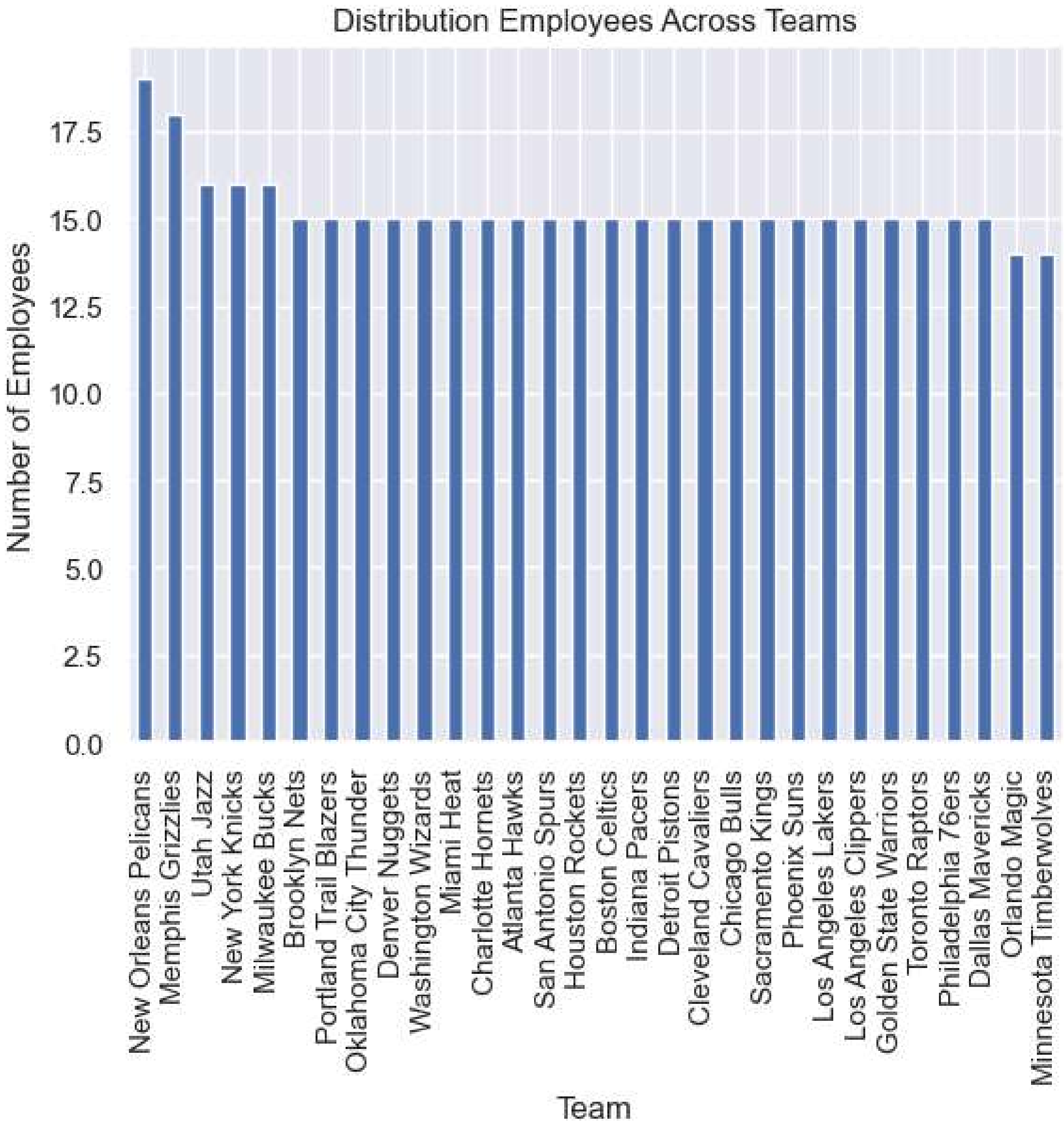
```
The correlation between Salary and age is
```

In [11]:

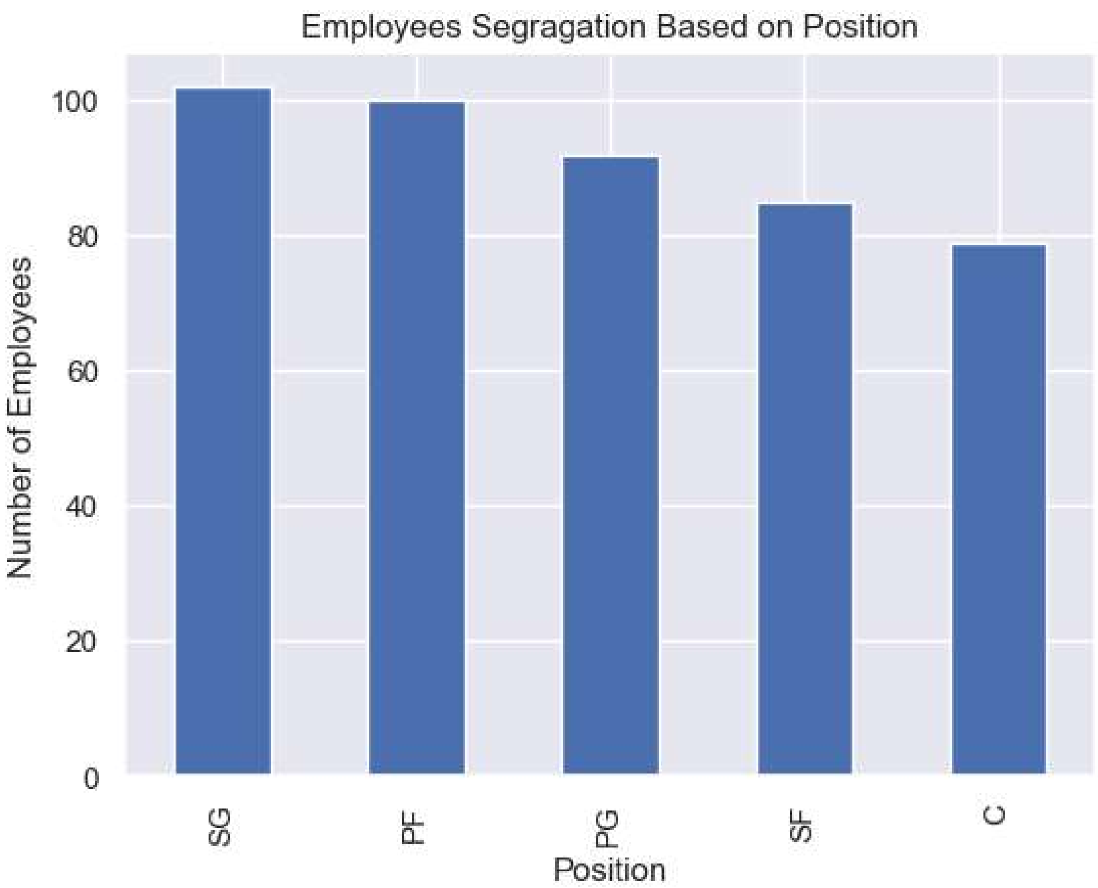
```
sns.scatterplot (x="Age", y= "Salary",data=data)
plt.ylabel("Salary")
plt.xlabel("Age")
plt.title("CORRELATION BETWEEN SALARY AND AGE")
plt.show()
```



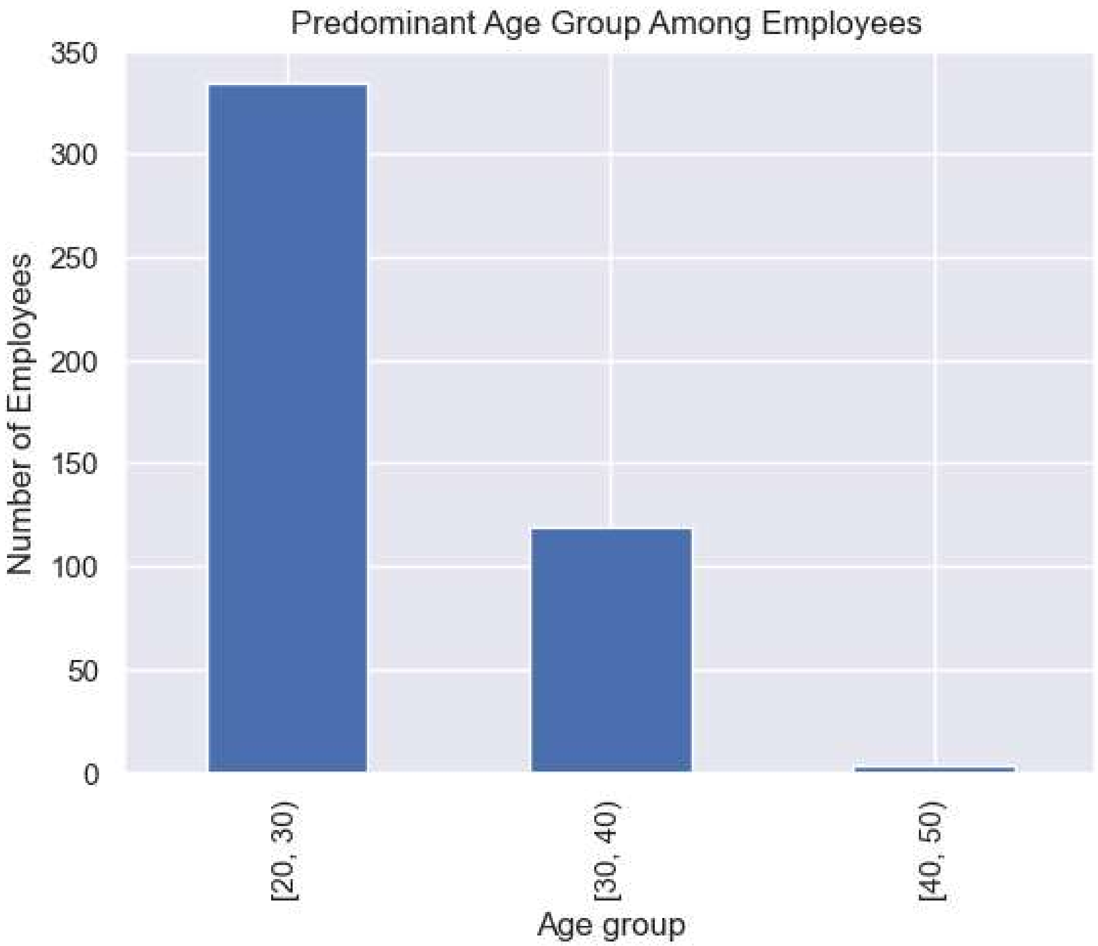
In [14]:
```
#1.Determine the distribution of employees across each team
data['Team'].value_counts().plot (kind="bar")
plt.title('Distribution Employees Across Teams')
plt.xlabel("Team")
plt.ylabel("Number of Employees")
plt.show()
```



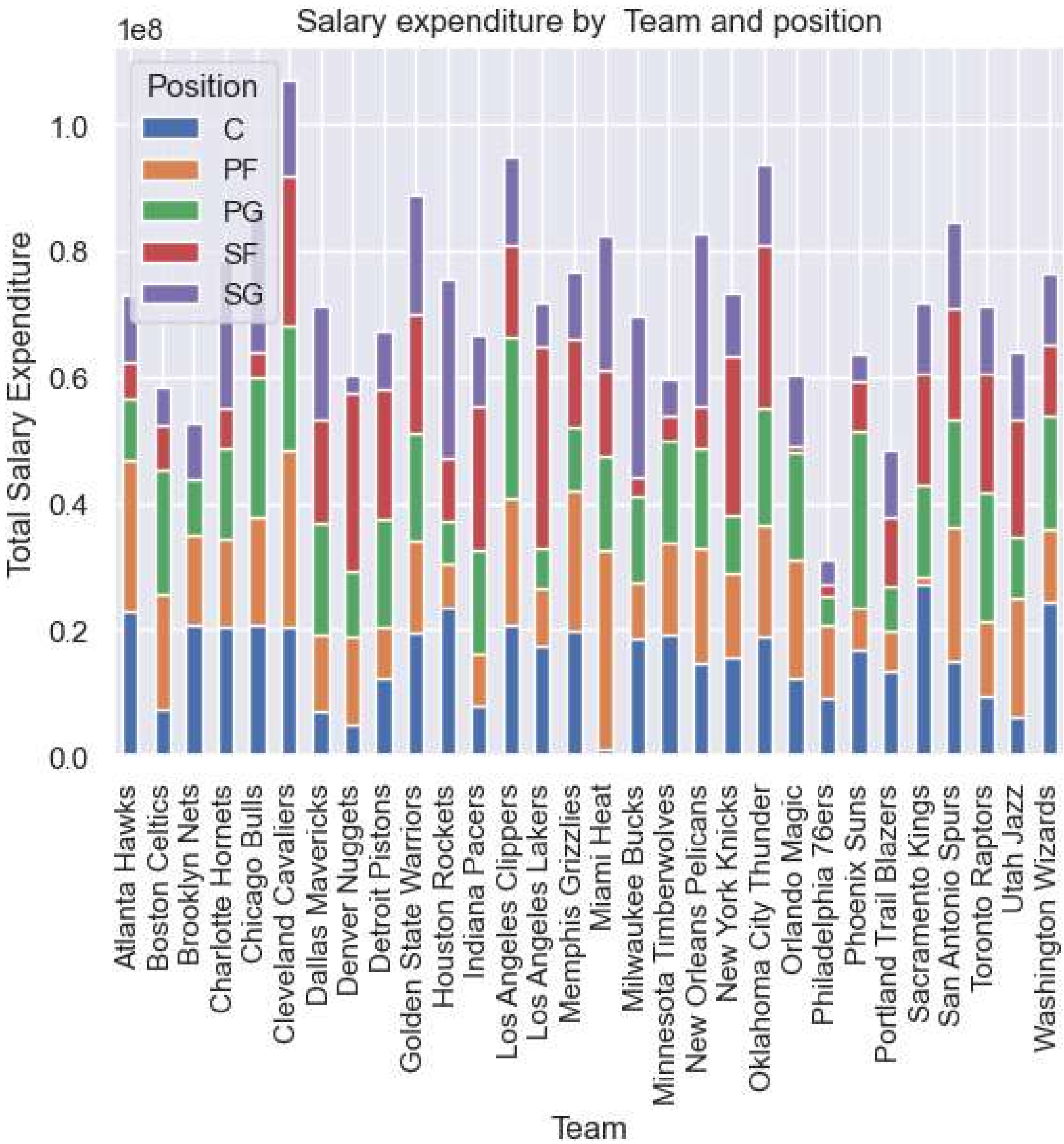
In [15]:
```
# segragate employees based on their positions.
position_distribution= data['Position'].value_counts()
position_distribution.plot (kind="bar")
plt.title('Employees Segragation Based on Position')
plt.xlabel("Position")
plt.ylabel("Number of Employees")
plt.show()
```



In [17]:
```
# identify the predominant age group among emplogees.
age_groups= pd.cut(data['Age'], bins=[20,30,40,50],right= False)
age_group_distribution=age_groups.value_counts()
age_group_distribution.plot (kind="bar")
plt.title('Predominant Age Group Among Employees ')
plt.xlabel("Age group")
plt.ylabel("Number of Employees")
plt.show()
```



In [18]:
```
#4. Discover which teamand position have the highest salary expenditure.
spending_salary.unstack().plot (kind="bar",stacked= True)
plt.title('Salary expenditure by  Team and position')
plt.xlabel("Team")
plt.ylabel("Total Salary Expenditure")
plt.show()
```



In [21]:
```
sns.regplot(x="Age", y="Salary",data= data)
plt.title('correlation between Salary and Age')
plt.xlabel("Age")
plt.ylabel("Salary")
plt.show()
```

data