Preprocessing:

In [1]:

```
import pandas as pd
import numpy as np

# Load the dataset
file_path = 'C:/Users/gokul/Documents/DATA SCIENCE/Python Project/myexcel - myexcel.cs
data = pd.read_csv(file_path)
```

In [2]:

```
data
```

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]:

```
np.random.seed(0)
data['Height'] = np.random.randint(150, 181, size=data.shape[0])
```

In [4]:

```
missing_values = data.isnull().sum()
```

In [5]:

```
data['Salary'].fillna(data['Salary'].mean(), inplace=True)
```

In [6]:

```
data_types = data.dtypes
```

In [7]:

```
data.to_csv('preprocessed_data.csv', index=False)
print("Missing values:\n", missing_values)
print("\nData types:\n", data_types)
print("\nFirst few rows of the dataset:\n", data.head())
```

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

Data types:
 Name         object
Team         object
Number        int64
Position     object
Age           int64
Height        int32
Weight        int64
College      object
Salary      float64
dtype: object

First few rows of the dataset:
             Name            Team  Number Position  Age  Height  Weight  \
0  Avery Bradley  Boston Celtics       0       PG   25     162     180   
1    Jae Crowder  Boston Celtics      99       SF   25     165     235   
2   John Holland  Boston Celtics      30       SG   27     171     205   
3    R.J. Hunter  Boston Celtics      28       SG   22     150     185   
4  Jonas Jerebko  Boston Celtics       8       PF   29     153     231   

             College        Salary  
0              Texas  7.730337e+06  
1          Marquette  6.796117e+06  
2  Boston University  4.833970e+06  
3      Georgia State  1.148640e+06  
4                NaN  5.000000e+06  
```

In [10]:

```
data
```

Out[10]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	162	180	Texas	7.730337e+06
1	Jae Crowder	Boston Celtics	99	SF	25	165	235	Marquette	6.796117e+06
2	John Holland	Boston Celtics	30	SG	27	171	205	Boston University	4.833970e+06
3	R.J. Hunter	Boston Celtics	28	SG	22	150	185	Georgia State	1.148640e+06
4	Jonas Jerebko	Boston Celtics	8	PF	29	153	231	NaN	5.000000e+06
...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	176	203	Butler	2.433333e+06
454	Raul Neto	Utah Jazz	25	PG	24	169	179	NaN	9.000000e+05
455	Tibor Pleiss	Utah Jazz	21	C	26	157	256	NaN	2.900000e+06
456	Jeff Withey	Utah Jazz	24	C	26	158	231	Kansas	9.472760e+05
457	Priyanka	Utah Jazz	34	C	25	179	231	Kansas	9.472760e+05

458 rows × 9 columns

1. Determine the distribution of employees across each team and calculate the percentage split relative to the total number of employees.

In [11]:

```
team_distribution = data['Team'].value_counts()
```

In [12]:

```
team_percentage = (team_distribution / len(data)) * 100
```

In [13]:

```
distribution_df = pd.DataFrame({'Team': team_distribution.index,
                                'Number of Employees': team_distribution.values,
                                'Percentage': team_percentage.values})
```

In [14]:

```
print(distribution_df)
```

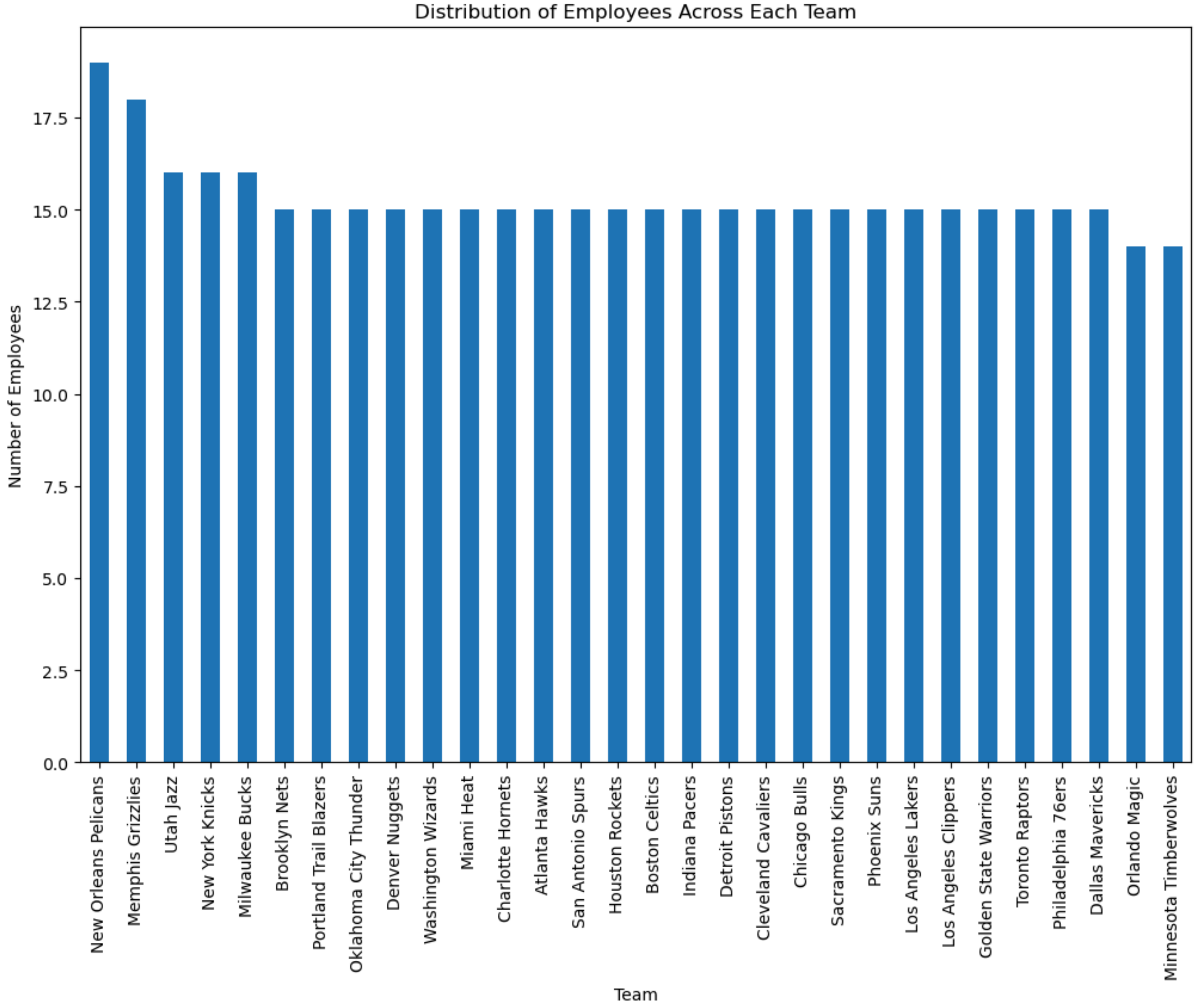
```
                      Team  Number of Employees  Percentage
0     New Orleans Pelicans                   19    4.148472
1        Memphis Grizzlies                   18    3.930131
2                Utah Jazz                   16    3.493450
3          New York Knicks                   16    3.493450
4          Milwaukee Bucks                   16    3.493450
5            Brooklyn Nets                   15    3.275109
6   Portland Trail Blazers                   15    3.275109
7    Oklahoma City Thunder                   15    3.275109
8           Denver Nuggets                   15    3.275109
9       Washington Wizards                   15    3.275109
10              Miami Heat                   15    3.275109
11       Charlotte Hornets                   15    3.275109
12           Atlanta Hawks                   15    3.275109
13       San Antonio Spurs                   15    3.275109
14         Houston Rockets                   15    3.275109
15          Boston Celtics                   15    3.275109
16          Indiana Pacers                   15    3.275109
17         Detroit Pistons                   15    3.275109
18     Cleveland Cavaliers                   15    3.275109
19           Chicago Bulls                   15    3.275109
20        Sacramento Kings                   15    3.275109
21            Phoenix Suns                   15    3.275109
22      Los Angeles Lakers                   15    3.275109
23    Los Angeles Clippers                   15    3.275109
24   Golden State Warriors                   15    3.275109
25         Toronto Raptors                   15    3.275109
26      Philadelphia 76ers                   15    3.275109
27        Dallas Mavericks                   15    3.275109
28           Orlando Magic                   14    3.056769
29  Minnesota Timberwolves                   14    3.056769
```

In [ ]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

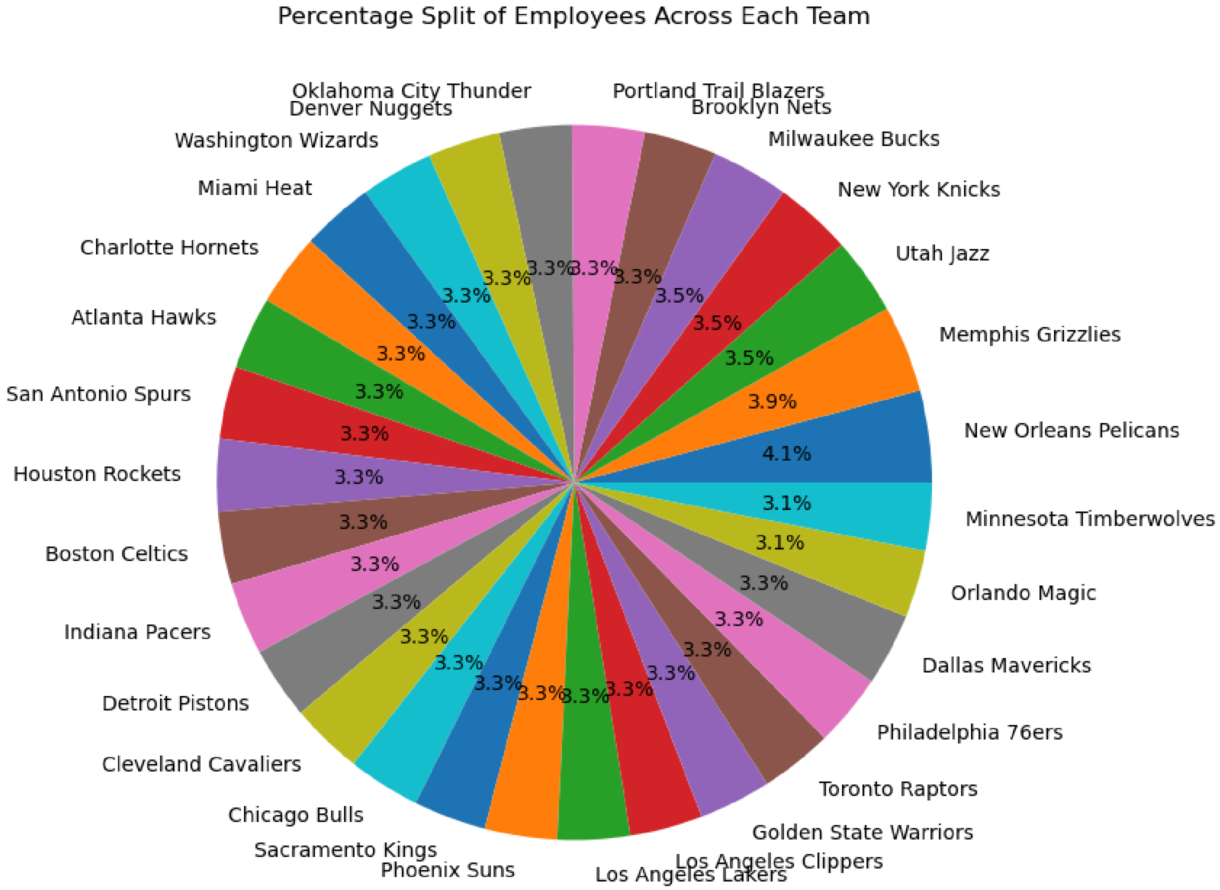
In [39]:

```
plt.figure(figsize=(12, 8))
team_distribution.plot(kind='bar')
plt.title('Distribution of Employees Across Each Team')
plt.xlabel('Team')
plt.ylabel('Number of Employees')
plt.show()
```



In [40]:

```
plt.figure(figsize=(12, 8))
team_percentage.plot(kind='pie', autopct='%1.1f%%')
plt.title('Percentage Split of Employees Across Each Team')
plt.ylabel('')
plt.show()
```



Write a code to Segregate employees based on their positions within the company.

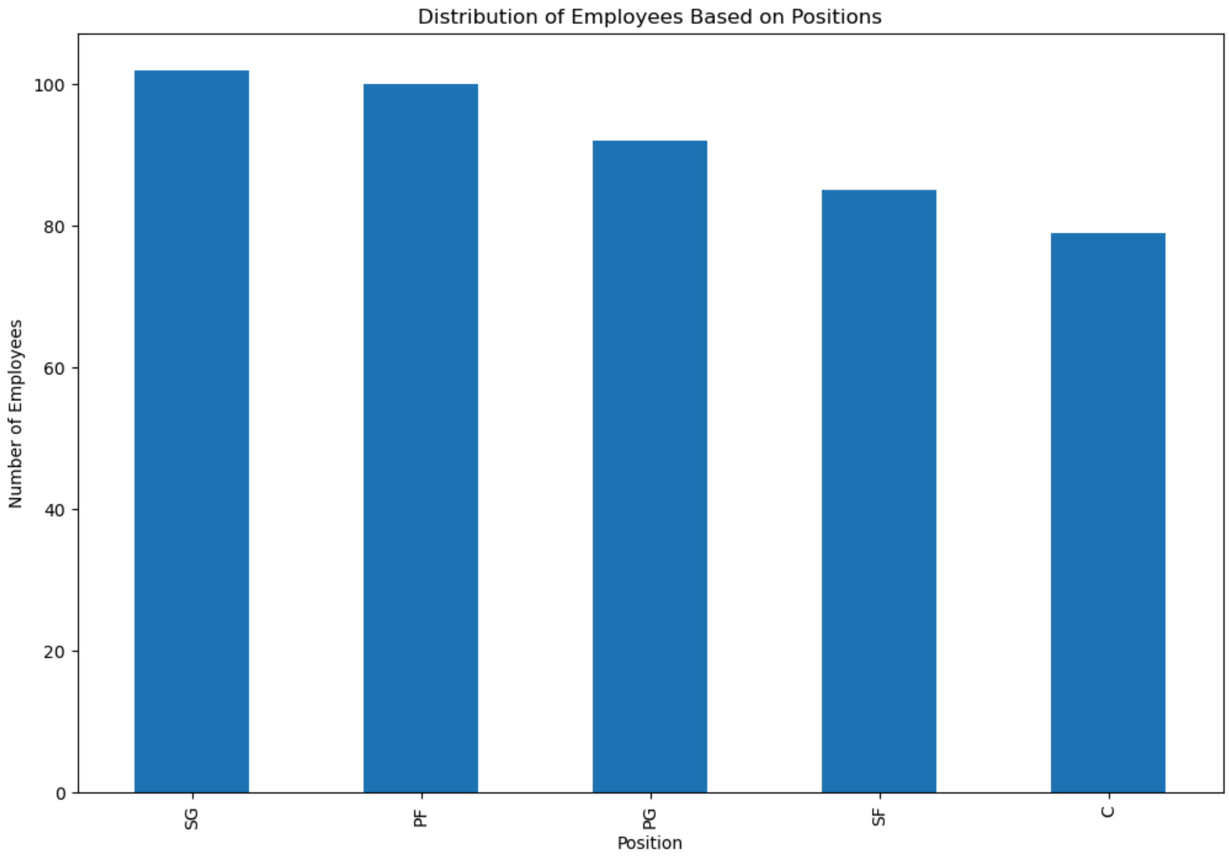
In [15]:
```
position_distribution = data['Position'].value_counts()
```

In [16]:
```
position_distribution_df = pd.DataFrame({'Position': position_distribution.index,
                                         'Number of Employees': position_distribution.
```

In [17]:
```
print(position_distribution_df)
```

```
  Position  Number of Employees
0       SG                  102
1       PF                  100
2       PG                   92
3       SF                   85
4        C                   79
```

In [41]:
```
plt.figure(figsize=(12, 8))
position_distribution.plot(kind='bar')
plt.title('Distribution of Employees Based on Positions')
plt.xlabel('Position')
plt.ylabel('Number of Employees')
plt.show()
```



Identify the predominant age group among employees.

In [18]:

```
bins = [20, 25, 30, 35, 40, 45, 50]
labels = ['20-24', '25-29', '30-34', '35-39', '40-44', '45-49']
data['Age Group'] = pd.cut(data['Age'], bins=bins, labels=labels, right=False)
```

In [19]:

```
age_group_distribution = data['Age Group'].value_counts().sort_index()
```

In [20]:

```
age_group_distribution_df = pd.DataFrame({'Age Group': age_group_distribution.index,
                                          'Number of Employees': age_group_distributic
```

In [21]:

```
print(age_group_distribution_df)
```

```
  Age Group  Number of Employees
0     20-24                  152
1     25-29                  182
2     30-34                   90
3     35-39                   29
4     40-44                    3
5     45-49                    0
```

In [22]:

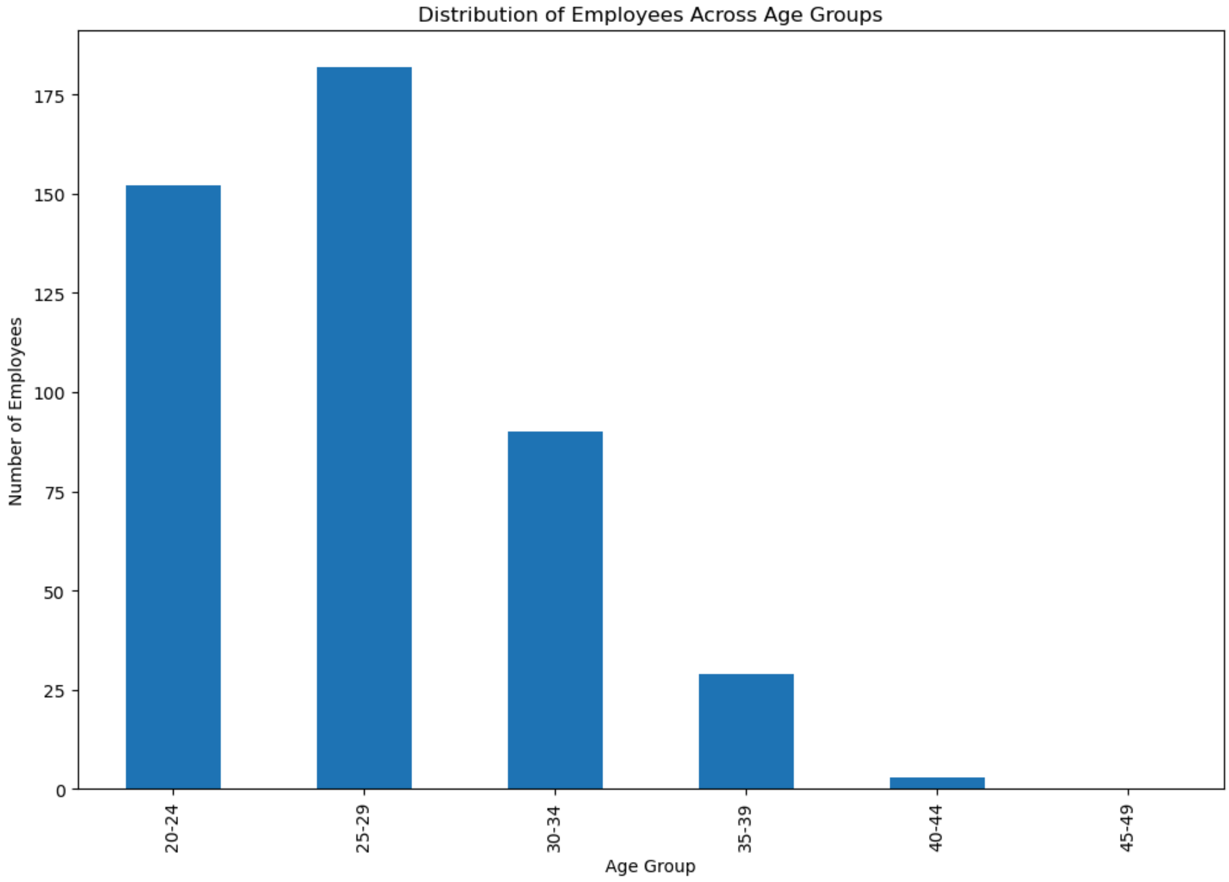
```
predominant_age_group = age_group_distribution.idxmax()
predominant_count = age_group_distribution.max()
```

In [23]:

```
print(f"The predominant age group is {predominant_age_group} with {predominant_count}
```

```
The predominant age group is 25-29 with 182 employees.
```

In [42]:
```
plt.figure(figsize=(12, 8))
age_group_distribution.plot(kind='bar')
plt.title('Distribution of Employees Across Age Groups')
plt.xlabel('Age Group')
plt.ylabel('Number of Employees')
plt.show()
```



Discover which team and position have the highest salary expenditure.

In [24]:
```
team_salary_expenditure = data.groupby('Team')['Salary'].sum().sort_values(ascending=F
```

In [25]:
```
position_salary_expenditure = data.groupby('Position')['Salary'].sum().sort_values(asc
```

In [26]:
```
highest_salary_team = team_salary_expenditure.idxmax()
highest_salary_team_amount = team_salary_expenditure.max()
```

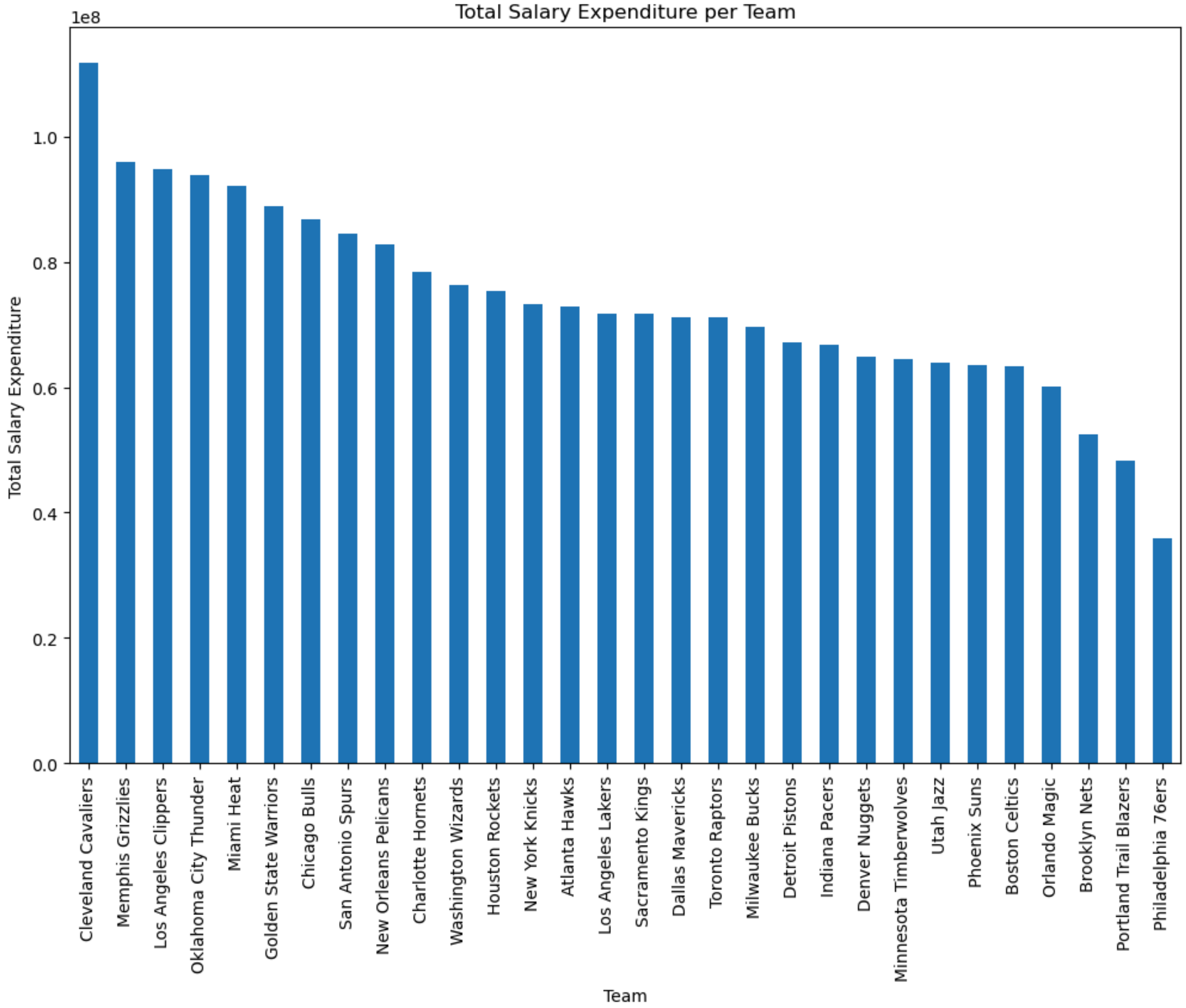
In [27]:
```
highest_salary_position = position_salary_expenditure.idxmax()
highest_salary_position_amount = position_salary_expenditure.max()
```

In [28]:
```
print(f"The team with the highest salary expenditure is {highest_salary_team} with a t
print(f"The position with the highest salary expenditure is {highest_salary_position}
```

```
The team with the highest salary expenditure is Cleveland Cavaliers with a total of
$111822658.55.
The position with the highest salary expenditure is C with a total of $466377332.00.
```

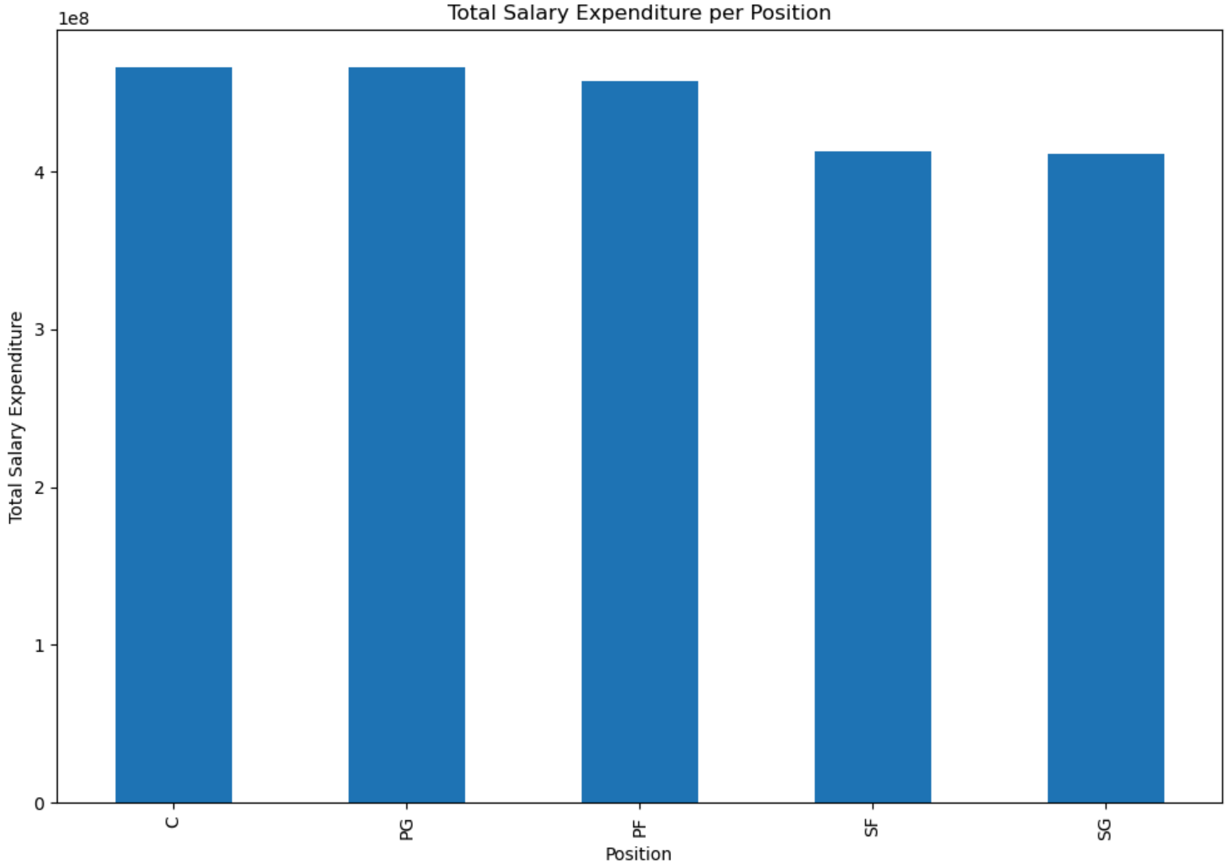
In [43]:

```
plt.figure(figsize=(12, 8))
team_salary_expenditure.plot(kind='bar')
plt.title('Total Salary Expenditure per Team')
plt.xlabel('Team')
plt.ylabel('Total Salary Expenditure')
plt.show()
```



In [44]:

```
plt.figure(figsize=(12, 8))
position_salary_expenditure.plot(kind='bar')
plt.title('Total Salary Expenditure per Position')
plt.xlabel('Position')
plt.ylabel('Total Salary Expenditure')
plt.show()
```

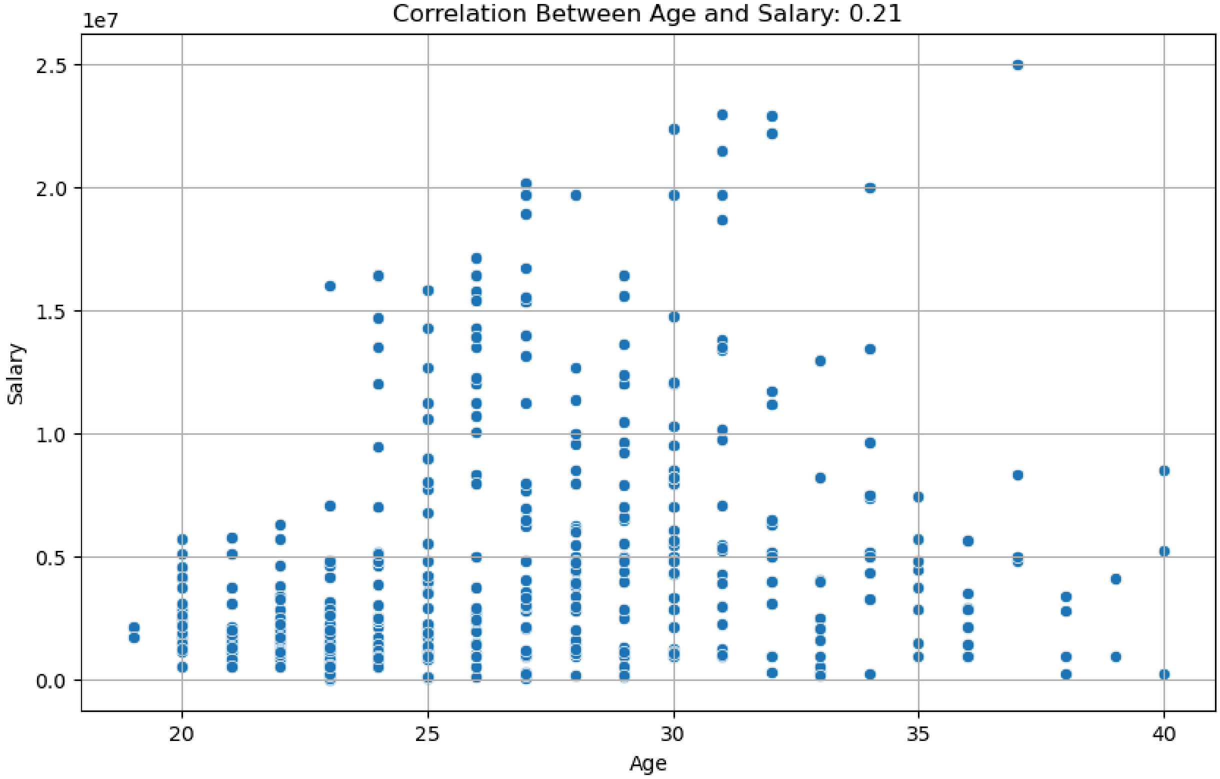


Investigate if there's any correlation between age and salary, and represent it visually.

In [29]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [30]:
```
correlation_coefficient = data['Age'].corr(data['Salary'])
```

In [31]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Age', y='Salary', data=data)
plt.title(f'Correlation Between Age and Salary: {correlation_coefficient:.2f}')
plt.xlabel('Age')
plt.ylabel('Salary')
plt.grid(True)
plt.show()
```



```
In [32]: print(f"The correlation coefficient between age and salary is {correlation_coefficient
```

```
The correlation coefficient between age and salary is 0.21.
```

```
In [ ]:
```