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

In [2]:
```
data = pd.read_excel('myexcel.xlsx')
data.head()
```

Out[2]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	2023-02-06 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	2023-05-06 00:00:00	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	2023-05-06 00:00:00	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	2023-10-06 00:00:00	231	NaN	5000000.0

In [5]:
```
np.random.seed(0)
data['Height'] = np.random.randint(150, 181, size=len(data))
data.head()
```

Out[5]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	162	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	165	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	171	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	150	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	153	231	NaN	5000000.0

In [4]:
```
data.isnull()
```

Out[4]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	True
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...
453	False	False	False	False	False	False	False	False	False
454	False	False	False	False	False	False	False	True	False
455	False	False	False	False	False	False	False	True	False
456	False	False	False	False	False	False	False	False	False
457	False	False	False	False	False	False	False	False	False

458 rows × 9 columns

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
Name         0
Team         0
Number       0
Position     0
Age          0
Height       0
Weight       0
College     84
Salary      11
dtype: int64
```

In [12]:
```
mode_college = data['College'].mode()[0]
mode_college
```

Out[12]:
```
'Kentucky'
```

In [14]:
```
data['College'] = data['College'].fillna(mode_college)
data.isnull().sum()
```

Out[14]:
```
Name         0
Team         0
Number       0
Position     0
Age          0
Height       0
Weight       0
College      0
Salary      11
dtype: int64
```

In [16]:
```
mean_Salary = data['Salary'].mean()
```

In [18]:
```
data['Salary'] = data['Salary'].fillna(mean_Salary)
```

In [20]:
```
data.isnull().sum()
```

Out[20]:
```
Name        0
Team        0
Number      0
Position    0
Age         0
Height      0
Weight      0
College     0
Salary      0
dtype: int64
```

In [22]:
```
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    int32  
 6   Weight    458 non-null    int64  
 7   College   458 non-null    object 
 8   Salary    458 non-null    float64
dtypes: float64(1), int32(1), int64(3), object(4)
memory usage: 30.5+ KB
```

In [24]:
```
data.describe()
```

Out[24]:

	Number	Age	Height	Weight	Salary
count	458.000000	458.000000	458.00000	458.000000	4.580000e+02
mean	17.713974	26.934498	164.60262	221.543668	4.833970e+06
std	15.966837	4.400128	9.13522	26.343200	5.163335e+06
min	0.000000	19.000000	150.00000	161.000000	3.088800e+04
25%	5.000000	24.000000	157.00000	200.000000	1.100150e+06
50%	13.000000	26.000000	165.00000	220.000000	2.862190e+06
75%	25.000000	30.000000	172.00000	240.000000	6.323553e+06
max	99.000000	40.000000	180.00000	307.000000	2.500000e+07

In [26]:
```
data
```

Out[26]:

	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	Kentucky	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	Kentucky	9.000000e+05
455	Tibor Pleiss	Utah Jazz	21	C	26	157	256	Kentucky	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

In [28]:
```
duplicate = data.duplicated().sum()
```

```
duplicate
```

Out[28]:
```
0
```

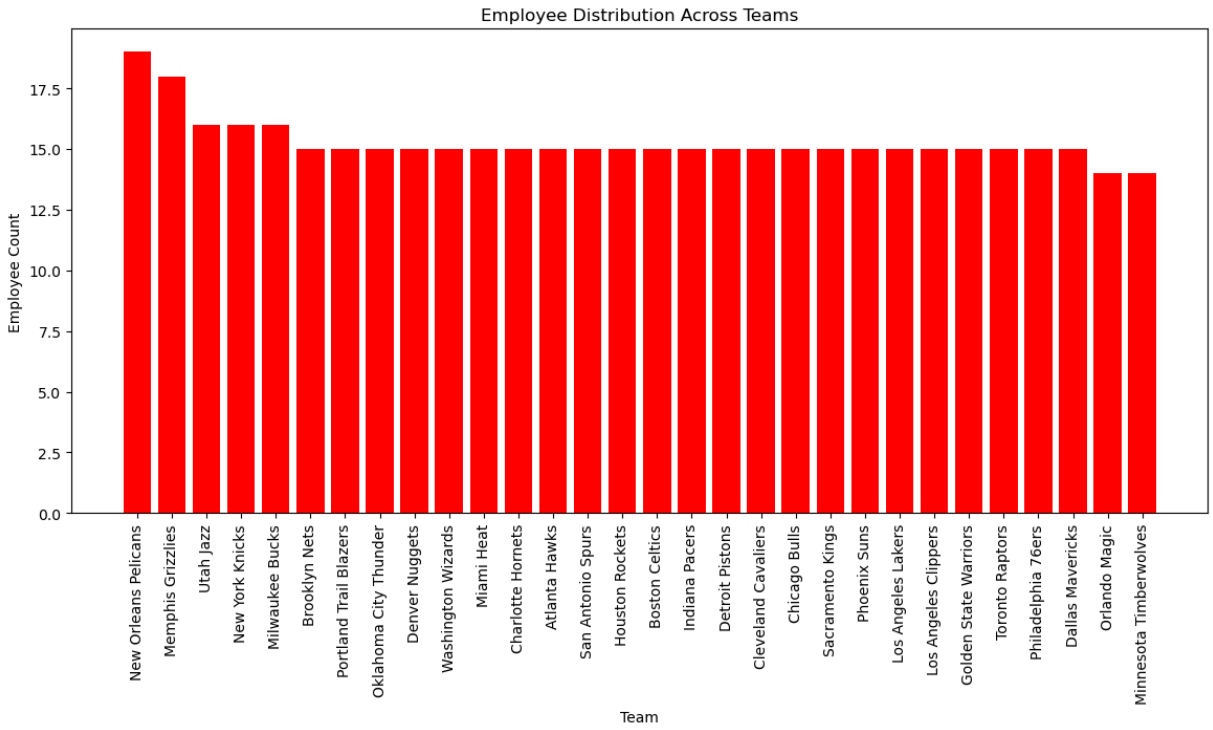
In [ ]:
```
1. Determine the distribution of employees across each team and calculate the perce
```

In [30]:
```
# Grouping by 'Team' and counting the number of employees (players) in each team
team_distribution = data['Team'].value_counts().reset_index()
team_distribution.columns = ['Team', 'EmployeeCount']

# Calculating the total number of employees
total_employees = data['Team'].count()

# Adding a percentage split column
team_distribution['PercentageSplit'] = (team_distribution['EmployeeCount'] / total_

# Displaying the result
team_distribution
```

Out[30]:

	Team	EmployeeCount	PercentageSplit
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 [32]:
```
# Bar chart
plt.figure(figsize=(14, 6))
plt.bar(team_distribution['Team'], team_distribution['EmployeeCount'], color='red')
plt.title('Employee Distribution Across Teams')
plt.xlabel('Team')
plt.ylabel('Employee Count')
plt.xticks(rotation=90)
plt.show()
```



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

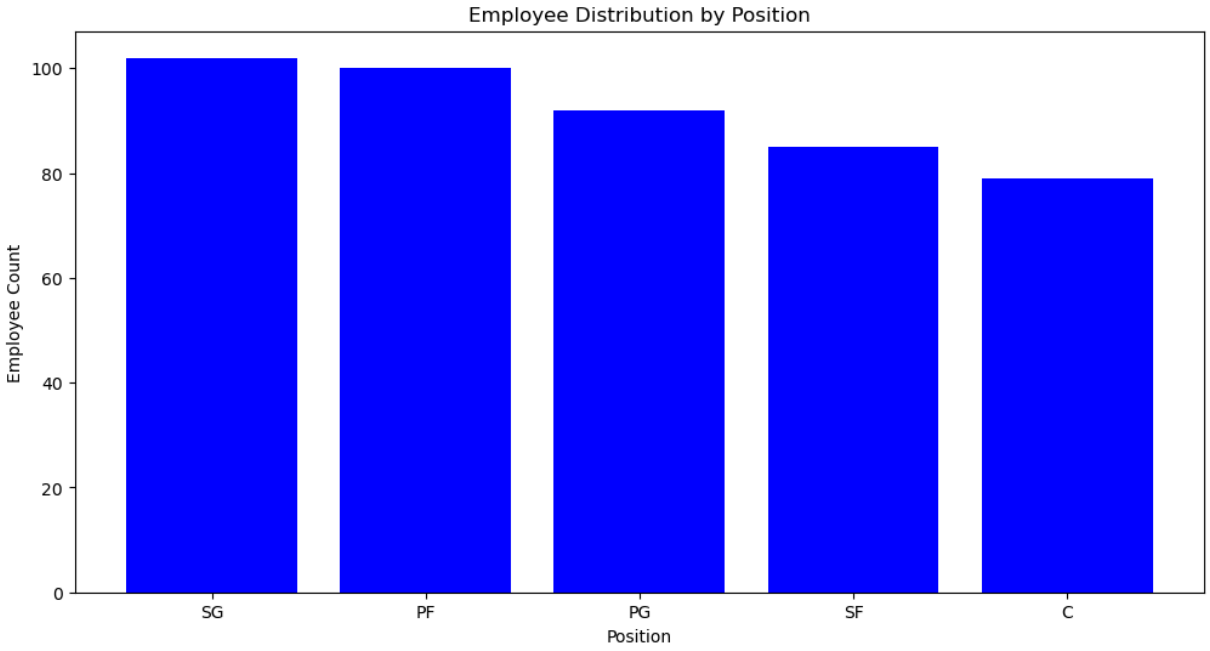
In [34]:
```
position_distribution = data['Position'].value_counts().reset_index()
position_distribution.columns = ['Position', 'EmployeeCount']

position_distribution['PercentageSplit'] = (position_distribution['EmployeeCount']

# Displaying the result
position_distribution
```

Out[34]:

	Position	EmployeeCount	PercentageSplit
0	SG	102	22.270742
1	PF	100	21.834061
2	PG	92	20.087336
3	SF	85	18.558952
4	C	79	17.248908

In [36]:
```
# Bar Chart for Position Distribution
plt.figure(figsize=(12, 6))
```

```
plt.bar(position_distribution['Position'], position_distribution['EmployeeCount'],
plt.title('Employee Distribution by Position')
plt.xlabel('Position')
plt.ylabel('Employee Count')
plt.show()
```



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

In [38]:
```
age_bins = [0, 20, 30, 40, 50, 60, 100]
age_labels = ['<20', '20-29', '30-39', '40-49', '50-69', '60+']

# Categorize employees into age groups
data['AgeGroup'] = pd.cut(data['Age'], bins=age_bins, labels=age_labels, right=Fals

# Count employees in each age group
age_group_distribution = data['AgeGroup'].value_counts().reset_index()
age_group_distribution.columns = ['AgeGroup', 'EmployeeCount']

# Calculate percentage split for each age group
total_employees = data['Age'].count()
age_group_distribution['PercentageSplit'] = (age_group_distribution['EmployeeCount'

# Display result
age_group_distribution.head()
```

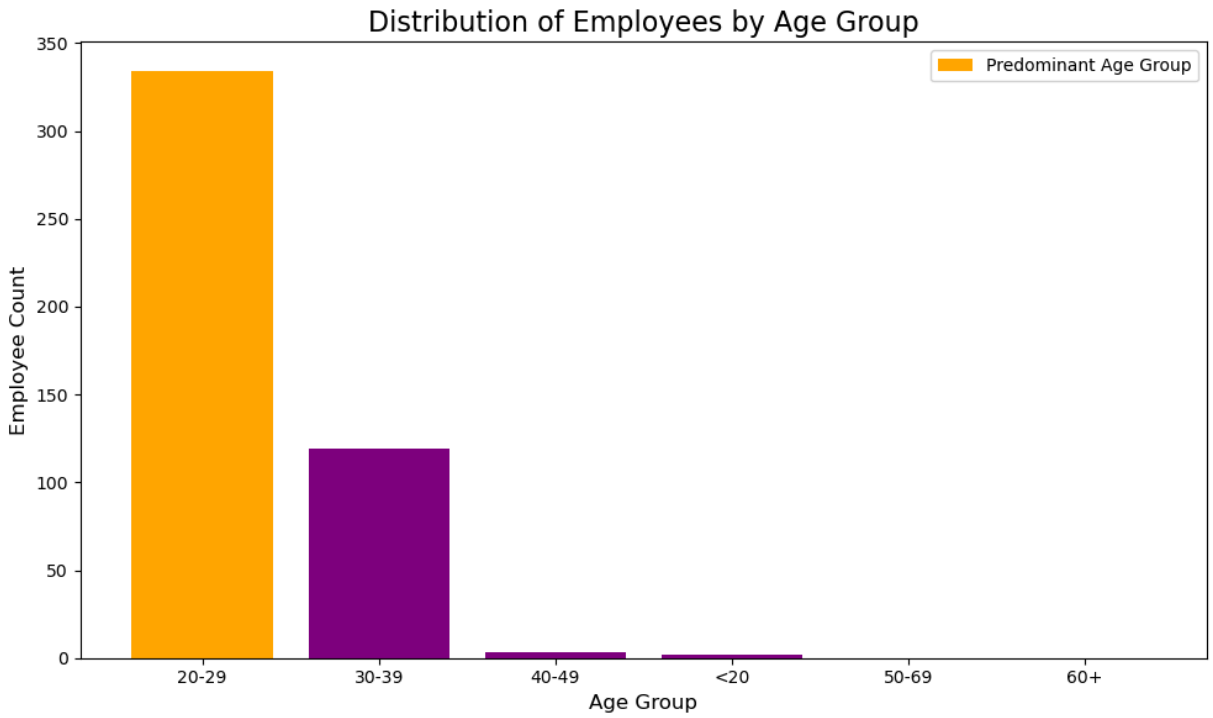
Out[38]:

	AgeGroup	EmployeeCount	PercentageSplit
0	20-29	334	72.925764
1	30-39	119	25.982533
2	40-49	3	0.655022
3	<20	2	0.436681
4	50-69	0	0.000000

In [40]:
```
# Find the predominant age group
predominant_age_group = age_group_distribution.loc[age_group_distribution['Employee

# Display the result
predominant_age_group.head()
```

Out[40]:
```
AgeGroup               20-29
EmployeeCount            334
PercentageSplit    72.925764
Name: 0, dtype: object
```

In [100…
```
# Bar Chart for Age Group Distribution
plt.figure(figsize=(10, 6))
plt.bar(age_group_distribution['AgeGroup'], age_group_distribution['EmployeeCount']
plt.title('Distribution of Employees by Age Group', fontsize=16)
plt.xlabel('Age Group', fontsize=12)
plt.ylabel('Employee Count', fontsize=12)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)

# Highlight the predominant age group
predominant_group = age_group_distribution.loc[age_group_distribution['EmployeeCoun
plt.bar(predominant_group['AgeGroup'], predominant_group['EmployeeCount'], color='o
plt.legend()

plt.tight_layout()
plt.show()
```



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

In [42]:
```
team_exp = data.groupby('Team')['Salary'].sum()
team_exp
```

Out[42]:
```
Team
Atlanta Hawks             7.290295e+07
Boston Celtics            6.337504e+07
Brooklyn Nets             5.252848e+07
Charlotte Hornets         7.834092e+07
Chicago Bulls             8.678338e+07
Cleveland Cavaliers       1.118227e+08
Dallas Mavericks          7.119873e+07
Denver Nuggets            6.495590e+07
Detroit Pistons           6.716826e+07
Golden State Warriors     8.886900e+07
Houston Rockets           7.528302e+07
Indiana Pacers            6.675183e+07
Los Angeles Clippers      9.485464e+07
Los Angeles Lakers        7.177043e+07
Memphis Grizzlies         9.588676e+07
Miami Heat                9.218361e+07
Milwaukee Bucks           6.960352e+07
Minnesota Timberwolves    6.454367e+07
New Orleans Pelicans      8.275077e+07
New York Knicks           7.330390e+07
Oklahoma City Thunder     9.376530e+07
Orlando Magic             6.016147e+07
Philadelphia 76ers        3.582686e+07
Phoenix Suns              6.344514e+07
Portland Trail Blazers    4.830182e+07
Sacramento Kings          7.168367e+07
San Antonio Spurs         8.444273e+07
Toronto Raptors           7.111761e+07
Utah Jazz                 6.400737e+07
Washington Wizards        7.632864e+07
Name: Salary, dtype: float64
```

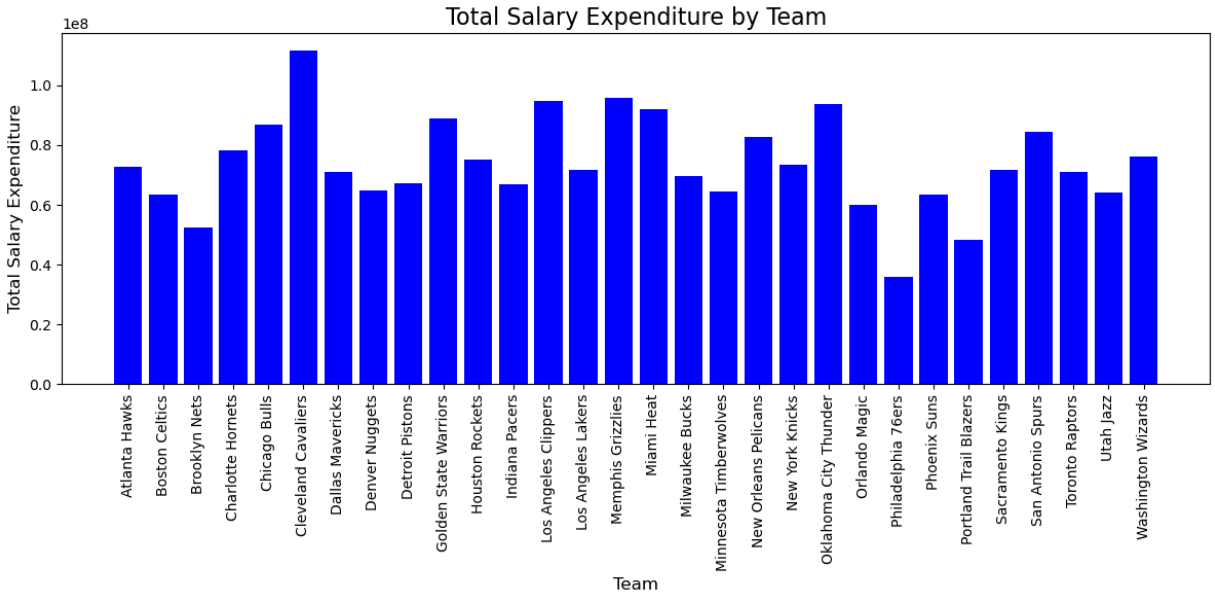
In [44]:
```
team_high_exp = team_exp.idxmax()
team_high_exp
```

Out[44]:
```
'Cleveland Cavaliers'
```

In [46]:
```
highest_exp = team_exp.max()
highest_exp
```

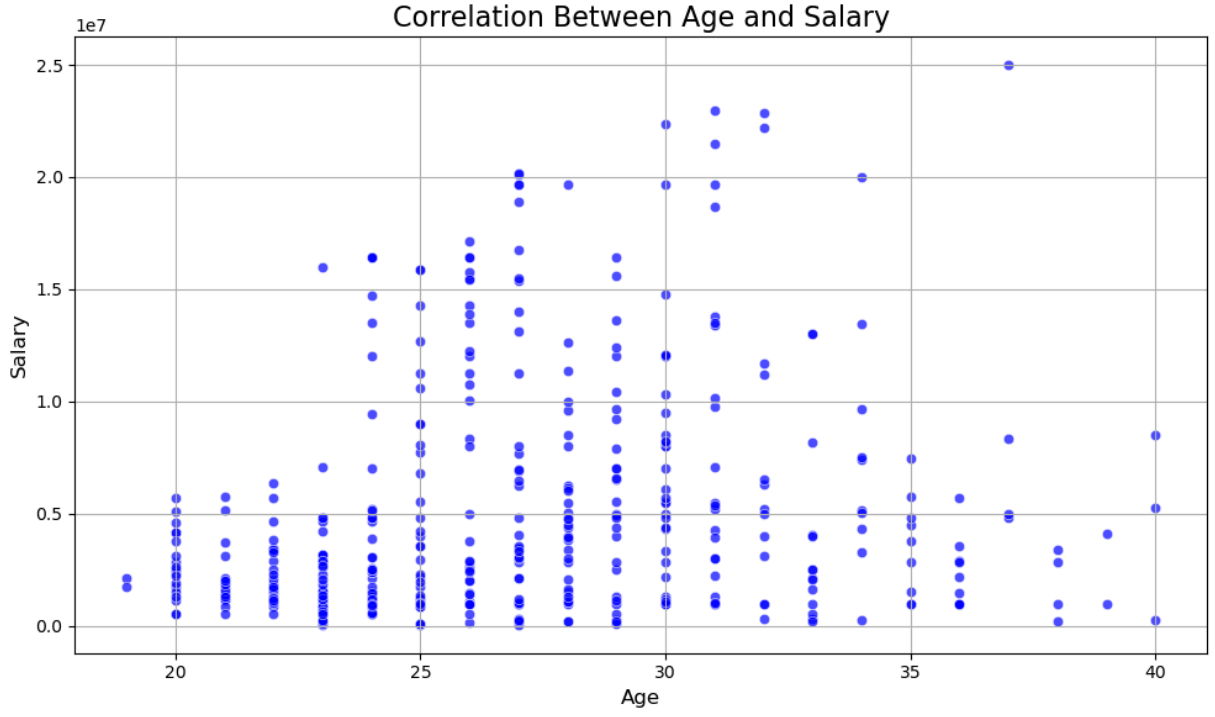
Out[46]:
```
111822658.5458613
```

In [48]:
```
# Group by Team and calculate total salary expenditure
team_exp = data.groupby('Team')['Salary'].sum()

# Bar Chart
plt.figure(figsize=(12, 6))
plt.bar(team_exp.index, team_exp.values, color='blue')
plt.title('Total Salary Expenditure by Team', fontsize=16)
plt.xlabel('Team', fontsize=12)
plt.ylabel('Total Salary Expenditure', fontsize=12)
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



In [ ]:
```
#Investigate if there's any correlation between age and salary, and represent it vi
```

In [50]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='Age', y='Salary', color='blue', alpha=0.7)
plt.title('Correlation Between Age and Salary', fontsize=16)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Salary', fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()
```



In [56]:
```
correlation_coefficient = data['Age'].corr(data['Salary'])
print(f"Correlation Coefficient between Age and Salary: {correlation_coefficient:.2
```

```
Correlation Coefficient between Age and Salary: 0.21
```

In [ ]:
```

```