In [5]:
```
import pandas as pd
import numpy as np
```

In [6]:
```
data=pd.read_csv("C:\\Users\\YOGA\\downloads\\PYTHON EDA DATA SET.csv")
```

In [7]:
```
data
```

Out[7]:

	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

replace the values in 'height' by random numbers in between 150 and 180

In [8]:
```
data['Height'] = np.random.randint(150, 181, size=len(data))
```

In [22]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 447 entries, 0 to 457
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype   
---  ------    --------------  -----   
 0   Name      447 non-null    object  
 1   Team      447 non-null    object  
 2   Number    447 non-null    int64   
 3   Position  447 non-null    object  
 4   Age       447 non-null    int64   
 5   Height    447 non-null    int32   
 6   Weight    447 non-null    int64   
 7   College   365 non-null    object  
 8   Salary    447 non-null    float64 
 9   AgeGroup  415 non-null    category
dtypes: category(1), float64(1), int32(1), int64(3), object(4)
memory usage: 33.7+ KB
```

In [23]:
```
data.isnull().sum()
```

Out[23]:
```
Name         0
Team         0
Number       0
Position     0
Age          0
Height       0
Weight       0
College     82
Salary       0
AgeGroup    32
dtype: int64
```

In [28]:
```
data.shape[0]
```

Out[28]:
```
447
```

In [35]:
```
s=(data.isnull().sum()/data.shape[0])*100
round(s,2)
```

Out[35]:
```
Name         0.00
Team         0.00
Number       0.00
Position     0.00
Age          0.00
Height       0.00
Weight       0.00
College     18.34
Salary       0.00
AgeGroup     7.16
dtype: float64
```

In [38]:
```
data.shape
```

Out[38]: (447, 10)

In [39]:
```
d1=data.drop(columns='College')
```

In [40]:
```
d1
```

Out[40]:

	Name	Team	Number	Position	Age	Height	Weight	Salary	AgeGroup
0	Avery Bradley	Boston Celtics	0	PG	25	165	180	7730337.0	25-29
1	Jae Crowder	Boston Celtics	99	SF	25	158	235	6796117.0	25-29
3	R.J. Hunter	Boston Celtics	28	SG	22	168	185	1148640.0	20-24
4	Jonas Jerebko	Boston Celtics	8	PF	29	151	231	5000000.0	25-29
5	Amir Johnson	Boston Celtics	90	PF	29	168	240	12000000.0	25-29
...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	178	203	2433333.0	25-29
454	Raul Neto	Utah Jazz	25	PG	24	176	179	900000.0	20-24
455	Tibor Pleiss	Utah Jazz	21	C	26	172	256	2900000.0	25-29
456	Jeff Withey	Utah Jazz	24	C	26	177	231	947276.0	25-29
457	Priyanka	Utah Jazz	34	C	25	177	231	947276.0	25-29

447 rows × 9 columns

In [42]:
```
d1.describe()
```

Out[42]:

	Number	Age	Height	Weight	Salary
count	447.000000	447.000000	447.000000	447.000000	4.470000e+02
mean	17.718121	26.914989	166.053691	221.774049	4.833970e+06
std	16.026218	4.394955	8.826010	26.132217	5.226620e+06
min	0.000000	19.000000	150.000000	161.000000	3.088800e+04
25%	5.000000	24.000000	159.000000	200.000000	1.025210e+06
50%	13.000000	26.000000	167.000000	220.000000	2.836186e+06
75%	25.000000	30.000000	174.000000	240.000000	6.500000e+06
max	99.000000	40.000000	180.000000	307.000000	2.500000e+07

In [41]:
```
d1.duplicated().sum()
```

Out[41]: 0

In [52]:
```
d1.shape
```

Out[52]: (447, 9)

1.Distribution of employees

In [43]:
```
import pandas as pd

d1= pd.DataFrame(d1)

# Group by 'Team' and count the number of employees in each team
team_counts = d1['Team'].value_counts()

# Calculate the percentage split
total_employees = len(d1)
team_percentages = (team_counts / total_employees) * 100
team_distribution = pd.DataFrame({'Count': team_counts, 'Percentage': tear

print(team_distribution)
```

Team	Count	Percentage
New Orleans Pelicans	19	4.250559
Utah Jazz	16	3.579418
New York Knicks	16	3.579418
Milwaukee Bucks	16	3.579418
Indiana Pacers	15	3.355705
Portland Trail Blazers	15	3.355705
Oklahoma City Thunder	15	3.355705
Washington Wizards	15	3.355705
Charlotte Hornets	15	3.355705
Atlanta Hawks	15	3.355705
San Antonio Spurs	15	3.355705
Houston Rockets	15	3.355705
Brooklyn Nets	15	3.355705
Dallas Mavericks	15	3.355705
Detroit Pistons	15	3.355705
Chicago Bulls	15	3.355705
Sacramento Kings	15	3.355705
Phoenix Suns	15	3.355705
Los Angeles Lakers	15	3.355705
Los Angeles Clippers	15	3.355705
Golden State Warriors	15	3.355705
Toronto Raptors	15	3.355705
Cleveland Cavaliers	14	3.131991
Memphis Grizzlies	14	3.131991
Orlando Magic	14	3.131991
Denver Nuggets	14	3.131991
Philadelphia 76ers	14	3.131991
Boston Celtics	14	3.131991
Miami Heat	13	2.908277
Minnesota Timberwolves	13	2.908277

2.Employees and their positions

In [ ]:

```
```

3.predominant age group among employees.

In [44]:

```
import pandas as pd
d1= pd.DataFrame(d1)

# Creating age grouo
bins = [20, 25, 30, 35]
labels = ['20-24', '25-29', '30-34']
d1['AgeGroup'] = pd.cut(d1['Age'], bins=bins, labels=labels, right=False)

age_group_counts = d1['AgeGroup'].value_counts()

print(age_group_counts)
```

```
AgeGroup
25-29    178
20-24    148
30-34     89
Name: count, dtype: int64
```

4.the team position which has highest salary expenditure

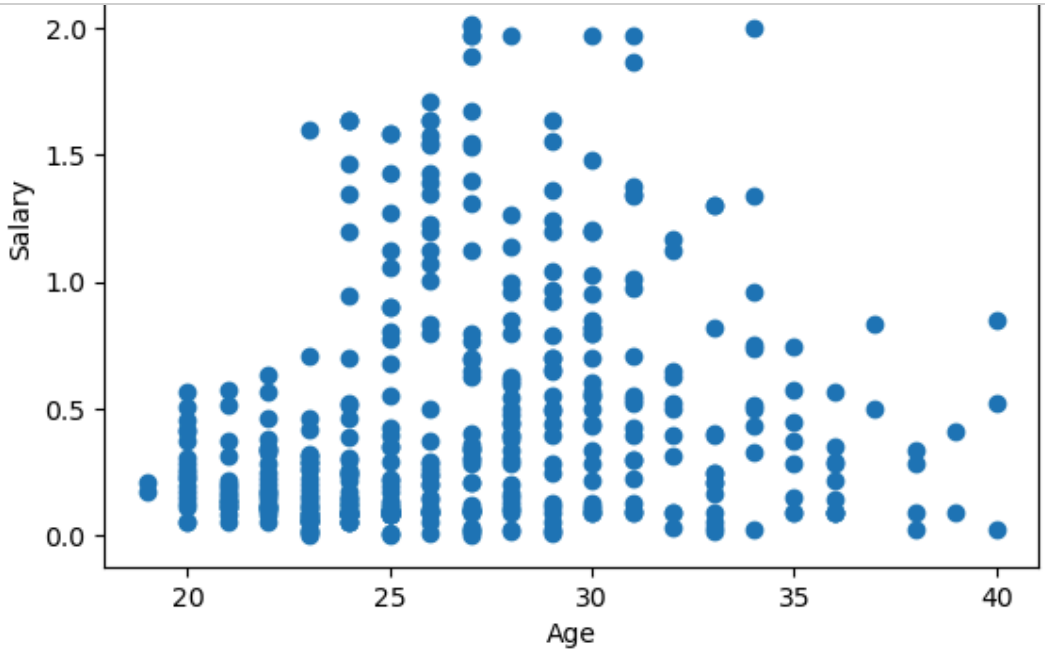
In [45]:

```
import pandas as pd
d1= pd.DataFrame(d1)

team_salary = d1.groupby('Team')['Salary'].sum()
position_salary = d1.groupby('Position')['Salary'].sum()
highest_team_salary = team_salary.idxmax()
highest_position_salary = position_salary.idxmax()

print(f"Team with the highest salary expenditure: {highest_team_salary}")
print(f"Position with the highest salary expenditure: {highest_position_sa
```

```
Team with the highest salary expenditure: Cleveland Cavaliers
Position with the highest salary expenditure: C
```

5.relation between age and salary

In [46]:
```
import pandas as pd
import matplotlib.pyplot as plt

d1= pd.DataFrame(d1)

# Drop rows where the salary is missing
d1= d1.dropna(subset=['Salary'])

# Plot the data
plt.scatter(d1['Age'], d1['Salary'])
plt.xlabel('Age')
plt.ylabel('Salary')
plt.title('Age and Salary')
plt.show()
```



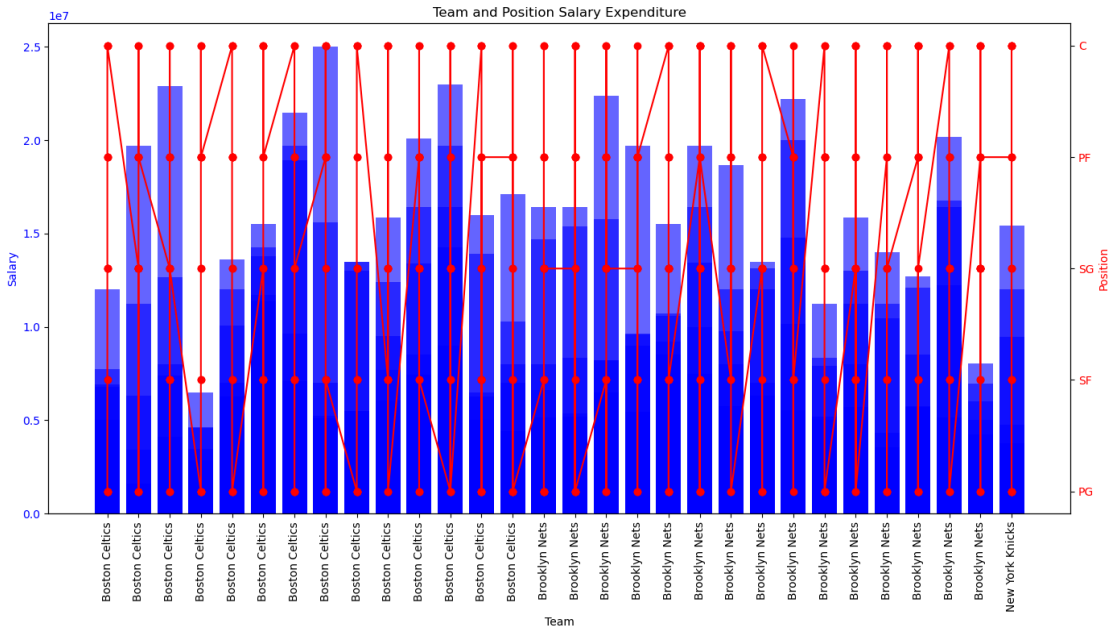
In [ ]:
```

```

GRAPHICAL REPRESENTATIONS OF ANALYSIS

1

In [48]:
```
import pandas as pd
import matplotlib.pyplot as plt

d1 = pd.DataFrame(d1)
fig, ax1 = plt.subplots(figsize=(14, 8))

ax1.bar(d1['Team'], d1['Salary'], color='b', alpha=0.6, label='Salary')
ax1.set_xlabel('Team')
ax1.set_ylabel('Salary', color='b')
ax1.tick_params(axis='y', labelcolor='b')
ax1.set_xticklabels(d1['Team'], rotation=90)

ax2 = ax1.twinx()
ax2.plot(d1['Team'], d1['Position'], color='r', marker='o', label='Percent
ax2.set_ylabel('Position', color='r')
ax2.tick_params(axis='y', labelcolor='r')

plt.title('Team and Position Salary Expenditure')
fig.tight_layout()
plt.show()
```

```
C:\Users\YOGA\AppData\Local\Temp\ipykernel_18204\724937870.py:11: UserWa
rning: FixedFormatter should only be used together with FixedLocator
  ax1.set_xticklabels(d1['Team'], rotation=90)
```

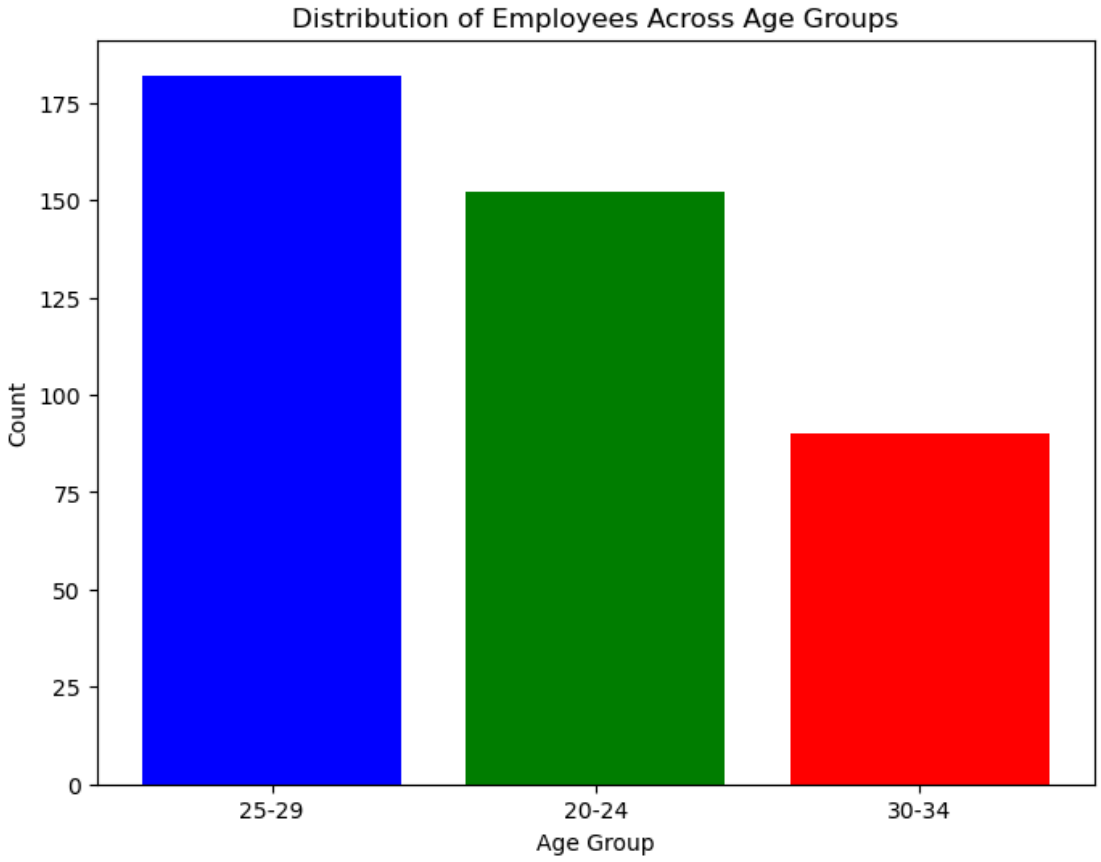


In [ ]:
```
2.
```

In [ ]:
```

```

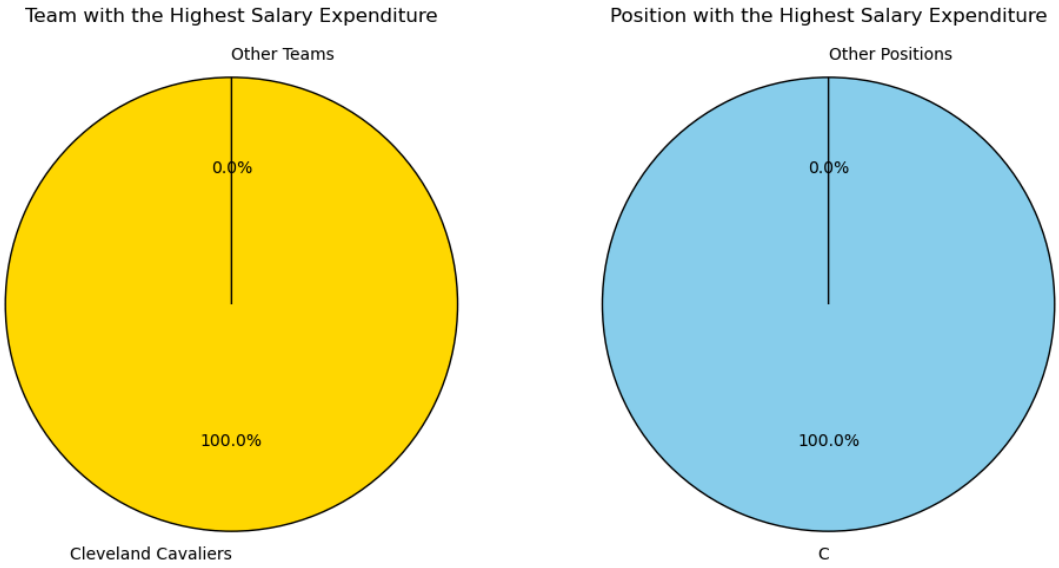
3

In [51]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Data
data = {
    'AgeGroup': ['25-29', '20-24', '30-34'],
    'Count': [182, 152, 90]
}

# Create DataFrame
df = pd.DataFrame(data)

# Plot
plt.figure(figsize=(8, 6))
plt.bar(df['AgeGroup'], df['Count'], color=['blue', 'green', 'red'])
plt.xlabel('Age Group')
plt.ylabel('Count')
plt.title('Distribution of Employees Across Age Groups')
plt.show()
```



4

In [105]:
```
import matplotlib.pyplot as plt

# Data
labels = ['Cleveland Cavaliers', 'Other Teams']
sizes = [1, 0]
colors = ['gold', 'lightgrey']

labels_position = ['C', 'Other Positions']
sizes_position = [1, 0]
colors_position = ['skyblue', 'lightgrey']

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))
ax1.pie(sizes, labels=labels, colors=colors, startangle=90, autopct='%1.1f
ax1.axis('equal')
ax1.set_title('Team with the Highest Salary Expenditure')

ax2.pie(sizes_position, labels=labels_position, colors=colors_position, st
ax2.axis('equal')
ax2.set_title('Position with the Highest Salary Expenditure')

plt.show()
```



5.

In [111]:

```
import pandas as pd
import matplotlib.pyplot as plt

data = data
df = pd.DataFrame(data)
df = df.dropna(subset=['Salary'])

plt.scatter(df['Age'], df['Salary'])
plt.xlabel('Age')
plt.ylabel('Salary')
plt.title('Age vs. Salary')
plt.show()
```