Python - COMPREHENSIVE ASSESMENT - (Topic: EDA)

Data Analysis:

In [44]:

```
import pandas as pd

file_path = 'C:\\Users\\Admin\\Downloads\\preprocessed_abc_company_dataset.
df = pd.read_csv(file_path)
df
```

Out[44]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary	height
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7730337.0	154
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6796117.0	179
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN	165
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1148640.0	177
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	NaN	5000000.0	156
...	...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	06-Mar	203	Butler	2433333.0	178
454	Raul Neto	Utah Jazz	25	PG	24	06-Jan	179	NaN	900000.0	173
455	Tibor Pleiss	Utah Jazz	21	C	26	07-Mar	256	NaN	2900000.0	153
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0	170
457	Priyanka	Utah Jazz	34	C	25	07-Mar	231	Kansas	947276.0	166

458 rows × 10 columns

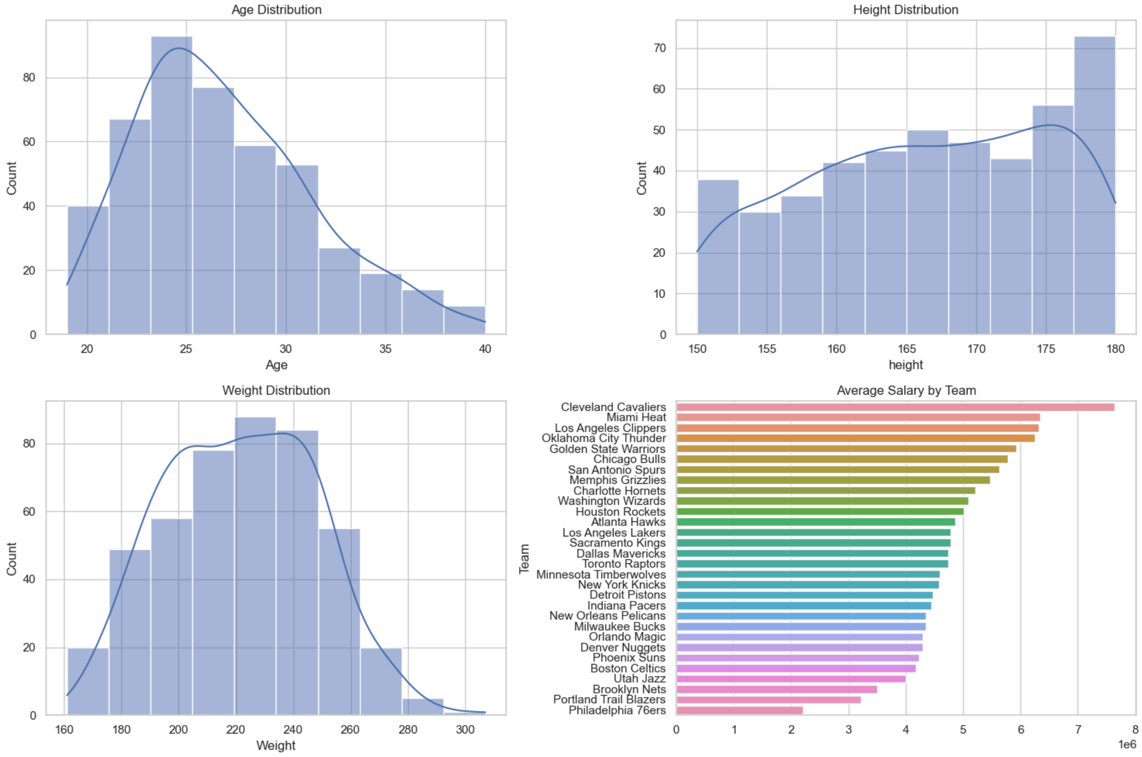
In [ ]:

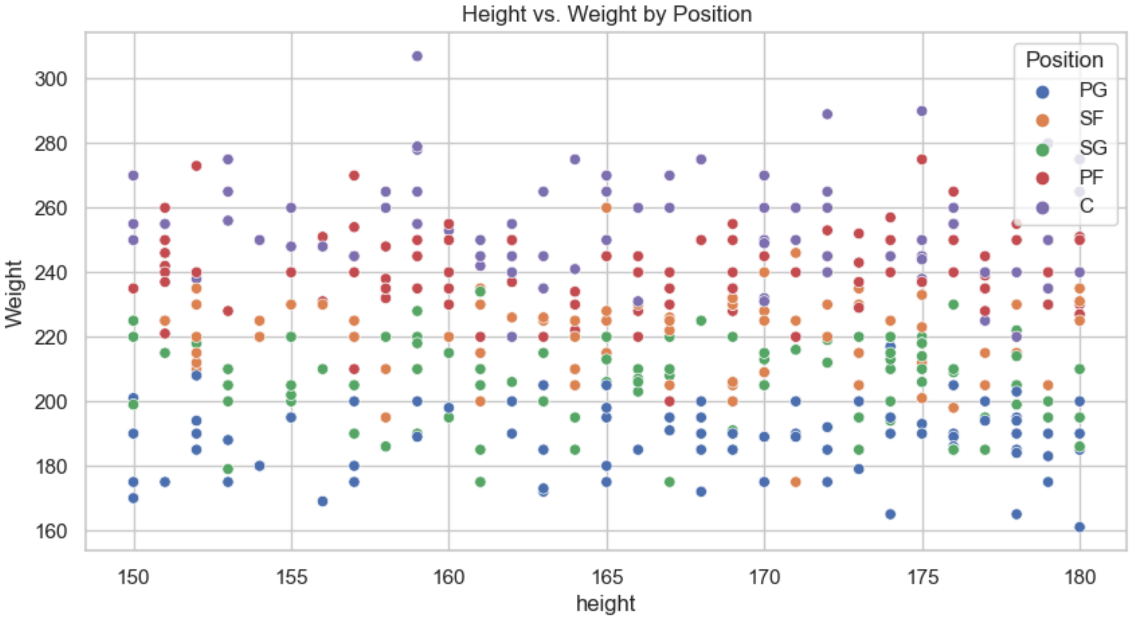
```
```

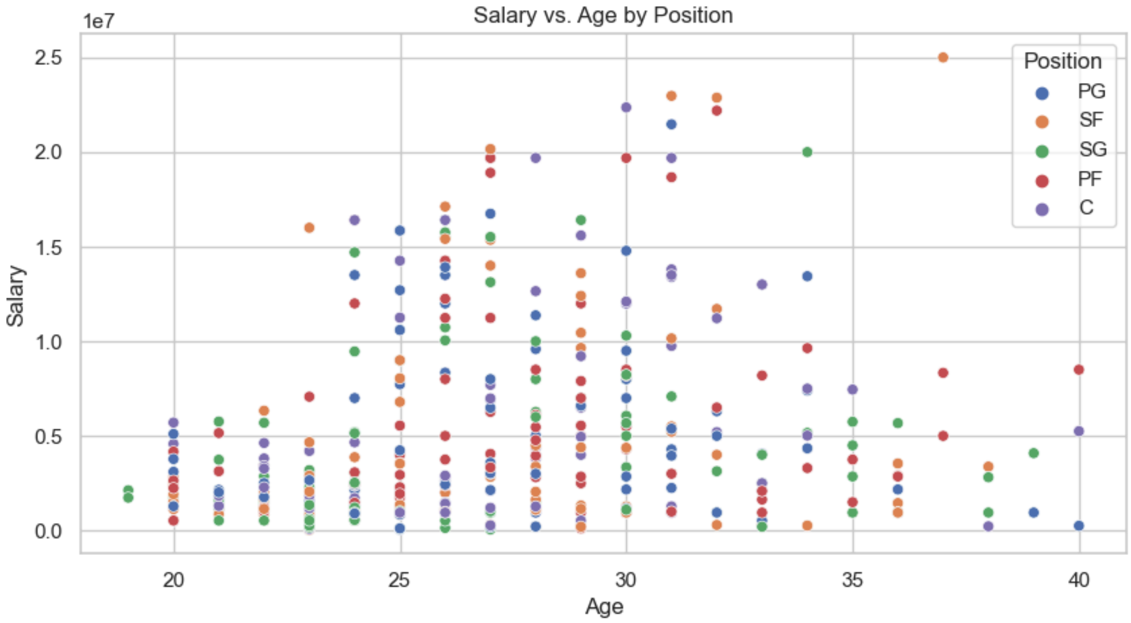
Data Visualization:

In [45]:

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

df = pd.read_csv("C:\\Users\\Admin\\Downloads\\preprocessed_abc_company_data


# Set up the visualisation style
sns.set(style="whitegrid")

# Basic statistical analysis
summary_stats = df.describe()

# Plotting distributions
plt.figure(figsize=(15, 10))

# Histogram for Age
plt.subplot(2, 2, 1)
sns.histplot(df['Age'], bins=10, kde=True)
plt.title('Age Distribution')

# Histogram for Height
plt.subplot(2, 2, 2)
sns.histplot(df['height'], bins=10, kde=True)
plt.title('Height Distribution')

# Histogram for Weight
plt.subplot(2, 2, 3)
sns.histplot(df['Weight'], bins=10, kde=True)
plt.title('Weight Distribution')

# Bar chart for average salary by team
plt.subplot(2, 2, 4)
avg_salary_by_team = df.groupby('Team')['Salary'].mean().sort_values(ascendi
sns.barplot(x=avg_salary_by_team.values, y=avg_salary_by_team.index)
plt.title('Average Salary by Team')

plt.tight_layout()
plt.show()

# Position-wise Analysis
position_summary = df.groupby('Position').agg({
    'Age': ['mean', 'median'],
    'height': ['mean', 'median'],
    'Weight': ['mean', 'median'],
    'Salary': ['mean', 'median']
}).reset_index()

# Correlation Matrix
correlation_matrix = df[['Age', 'height', 'Weight', 'Salary']].corr()

# Scatter Plot for Height vs. Weight
plt.figure(figsize=(10, 5))
sns.scatterplot(x='height', y='Weight', hue='Position', data=df)
plt.title('Height vs. Weight by Position')
plt.show()

# Scatter Plot for Salary vs. Age
plt.figure(figsize=(10, 5))
sns.scatterplot(x='Age', y='Salary', hue='Position', data=df)
plt.title('Salary vs. Age by Position')
plt.show()
```


Age Distribution
Count
0
20
40
60
80
20
25
30
35
40
Age
Height Distribution
Count
0
10
20
30
40
50
60
70
150
155
160
165
170
175
180
height
Weight Distribution
Count
0
20
40
60
80
160
180
200
220
240
260
280
300
Weight
Average Salary by Team
Team
Cleveland Cavaliers
Miami Heat
Los Angeles Clippers
Oklahoma City Thunder
Golden State Warriors
Chicago Bulls
San Antonio Spurs
Memphis Grizzlies
Charlotte Hornets
Washington Wizards
Houston Rockets
Atlanta Hawks
Los Angeles Lakers
Sacramento Kings
Dallas Mavericks
Toronto Raptors
Minnesota Timberwolves
New York Knicks
Detroit Pistons
Indiana Pacers
New Orleans Pelicans
Milwaukee Bucks
Orlando Magic
Denver Nuggets
Phoenix Suns
Boston Celtics
Utah Jazz
Brooklyn Nets
Portland Trail Blazers
Philadelphia 76ers
0
1
2
3
4
5
6
7
8
1e6


Height vs. Weight by Position
Weight
160
180
200
220
240
260
280
300
150
155
160
165
170
175
180
height
Position
PG
SF
SG
PF
C



In [ ]:

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.

Program Code :

In [9]:

```
import pandas as pd
import numpy as np

# Load the dataset
file_path = 'C:\\Users\\Admin\\Downloads\\preprocessed_abc_company_dataset.
df = pd.read_csv(file_path)

# Replace the "height" column with random numbers between 150 and 180
df['height'] = np.random.randint(150, 181, size=len(df))

# Verify the changes
df
```

Out[9]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary	height
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7730337.0	160
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6796117.0	169
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN	160
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1148640.0	173
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	NaN	5000000.0	178
...	...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	06-Mar	203	Butler	2433333.0	152
454	Raul Neto	Utah Jazz	25	PG	24	06-Jan	179	NaN	900000.0	153
455	Tibor Pleiss	Utah Jazz	21	C	26	07-Mar	256	NaN	2900000.0	172
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0	155
457	Priyanka	Utah Jazz	34	C	25	07-Mar	231	Kansas	947276.0	167

458 rows × 10 columns

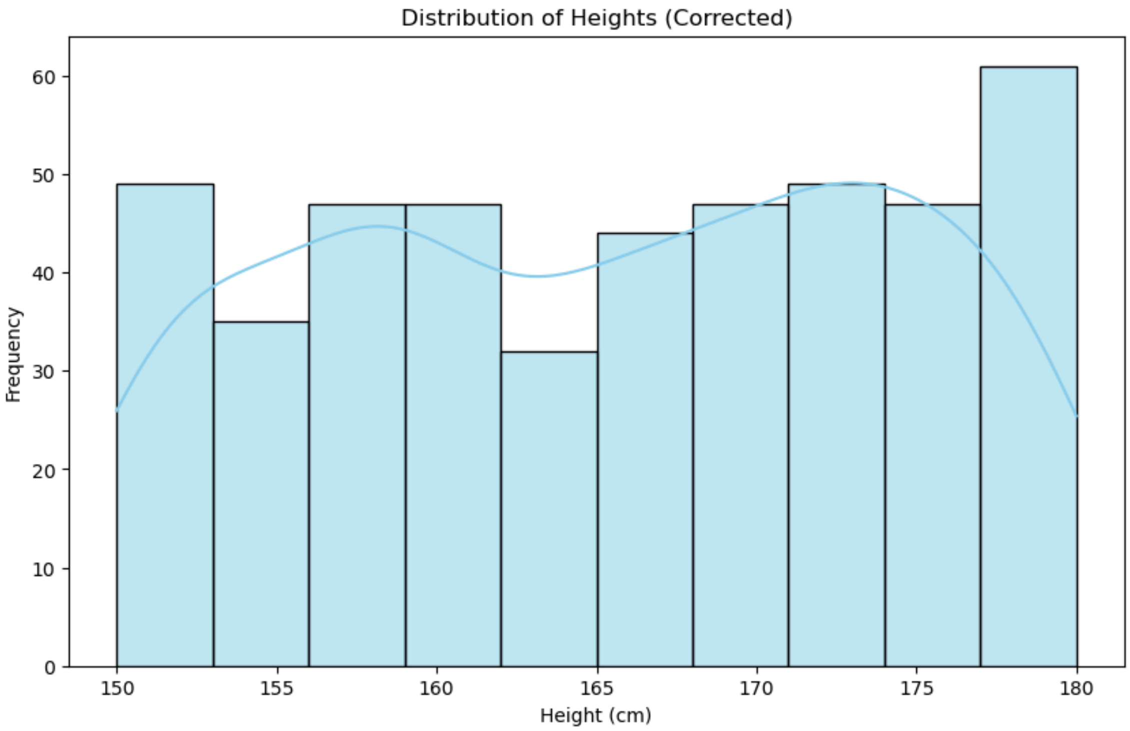
In [ ]:

Graphical Representation :

In [11]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
file_path = 'C:\\Users\\Admin\\Downloads\\preprocessed_abc_company_dataset.
df = pd.read_csv(file_path)

# Replace the "height" column with random numbers between 150 and 180
df['height'] = np.random.randint(150, 181, size=len(df))

# Verify the changes
print(df['height'])

# Plotting the new height distribution
plt.figure(figsize=(10, 6))
sns.histplot(df['height'], bins=10, kde=True, color='skyblue')
plt.xlabel('Height (cm)')
plt.ylabel('Frequency')
plt.title('Distribution of Heights (Corrected)')
plt.show()
```

```
0      176
1      180
2      173
3      151
4      180
      ...
453    158
454    152
455    177
456    161
457    167
Name: height, Length: 458, dtype: int32
```



In [ ]:

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

In [ ]:

In [6]:

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

# Load the dataset
file_path =  'C:\\Users\\Admin\\Downloads\\preprocessed_abc_company_dataset
df = pd.read_csv(file_path)

# Calculate the distribution of employees across each team
team_distribution = df['Team'].value_counts()

# Calculate the percentage split relative to the total number of employees
team_percentage = (team_distribution / len(df)) * 100

team_distribution, team_percentage
```

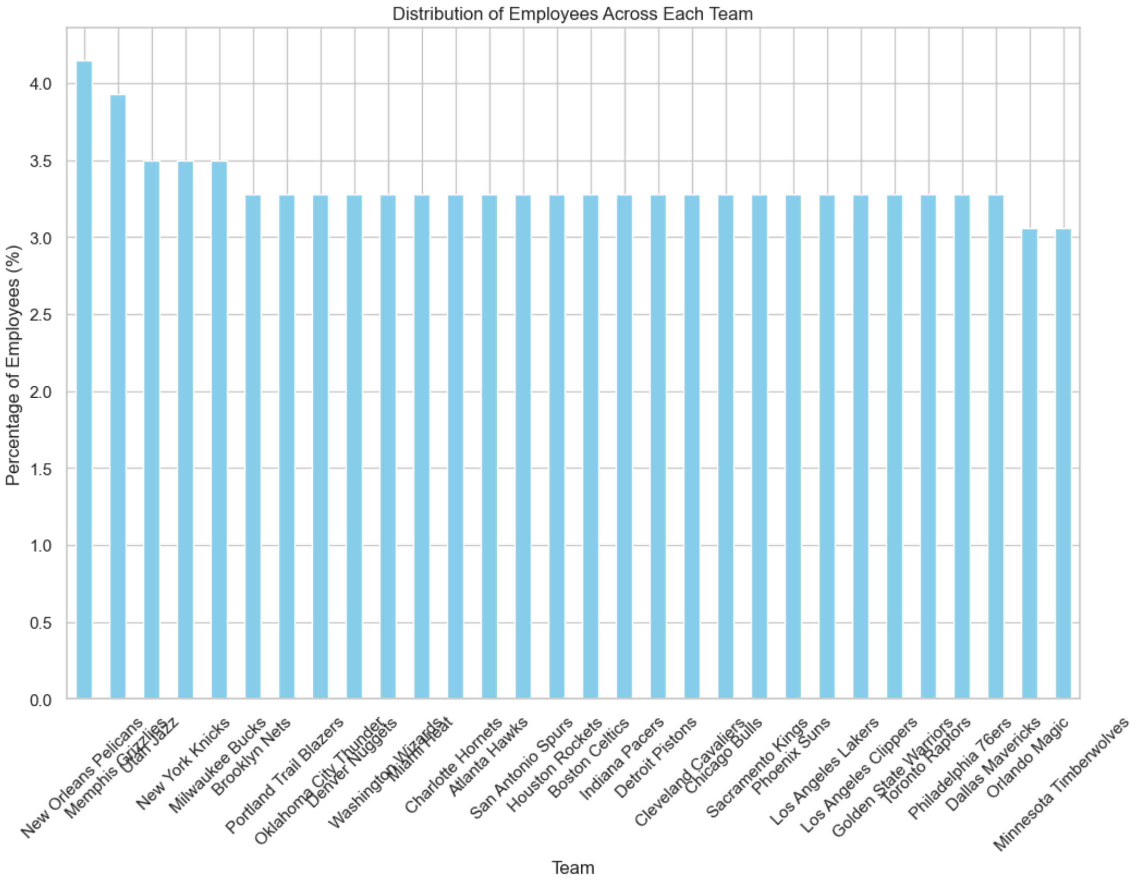
```
Out[6]: (Team
         New Orleans Pelicans      19
         Memphis Grizzlies         18
         Utah Jazz                 16
         New York Knicks           16
         Milwaukee Bucks           16
         Brooklyn Nets             15
         Portland Trail Blazers    15
         Oklahoma 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,
         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 [ ]:

Graphical Representation

In [20]:

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

# Re-load the dataset
file_path = 'C:\\Users\\Admin\\Downloads\\preprocessed_abc_company_dataset.
df = pd.read_csv(file_path)

# Calculate the distribution of employees across each team
team_distribution = df['Team'].value_counts()

# Calculate the percentage split relative to the total number of employees
team_percentage = (team_distribution / len(df)) * 100

# Plotting the distribution of employees across each team
plt.figure(figsize=(12, 8))
team_percentage.plot(kind='bar', color='skyblue')
plt.xlabel('Team')
plt.ylabel('Percentage of Employees (%)')
plt.title('Distribution of Employees Across Each Team')
plt.xticks(rotation=45)
plt.show()
```



In [ ]:

2. Segregate employees based on their positions within the company.

Program Code

In [23]:

```python
import pandas as pd

# Load the dataset
file_path = 'C:\\Users\\Admin\\Downloads\\preprocessed_abc_company_dataset.
df = pd.read_csv(file_path)

# Segregate employees based on their positions
position_distribution = df['Position'].value_counts()

# Calculate the percentage split relative to the total number of employees
position_percentage = (position_distribution / len(df)) * 100

print(position_distribution)
print(position_percentage)
```

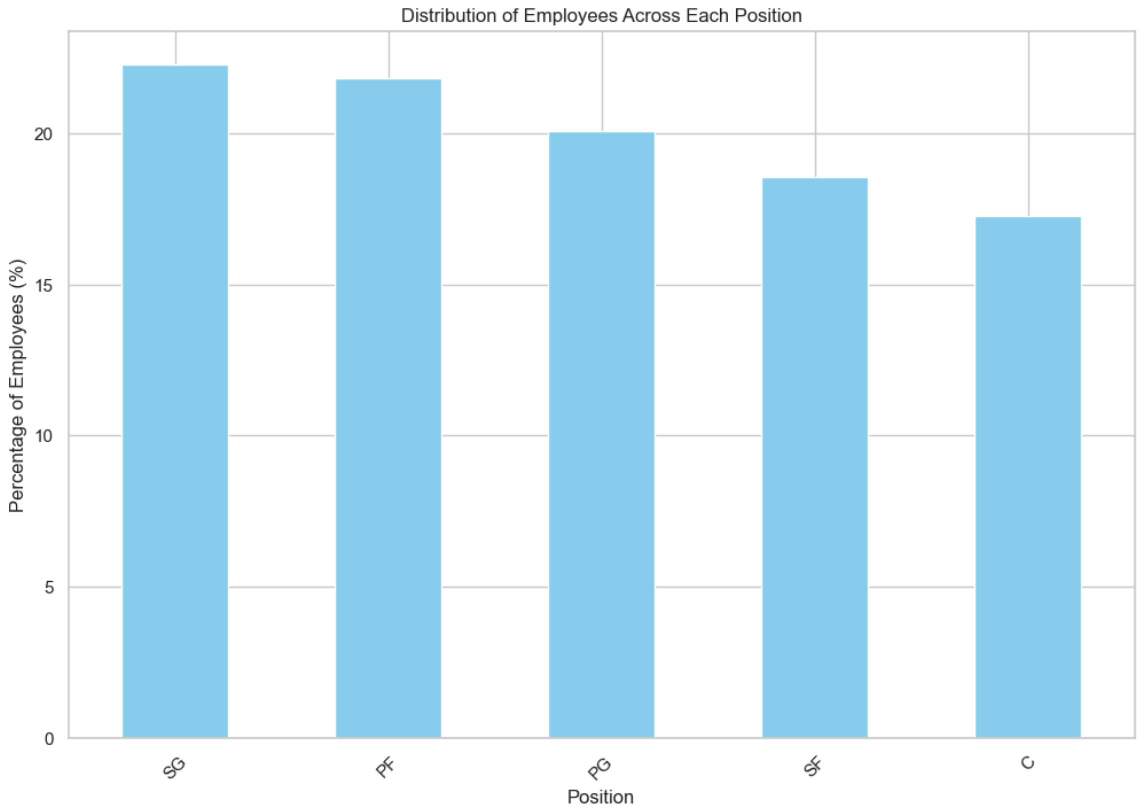
```
Position
SG    102
PF    100
PG     92
SF     85
C      79
Name: count, dtype: int64
Position
SG    22.270742
PF    21.834061
PG    20.087336
SF    18.558952
C     17.248908
Name: count, dtype: float64
```

In [ ]:

Graphical Representation

In [24]:

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

# Load the dataset
file_path = 'C:\\Users\\Admin\\Downloads\\preprocessed_abc_company_dataset.
df = pd.read_csv(file_path)

# Segregate employees based on their positions
position_distribution = df['Position'].value_counts()

# Calculate the percentage split relative to the total number of employees
position_percentage = (position_distribution / len(df)) * 100

# Plotting the distribution of employees across each position
plt.figure(figsize=(12, 8))
position_percentage.plot(kind='bar', color='skyblue')
plt.xlabel('Position')
plt.ylabel('Percentage of Employees (%)')
plt.title('Distribution of Employees Across Each Position')
plt.xticks(rotation=45)
plt.show()
```



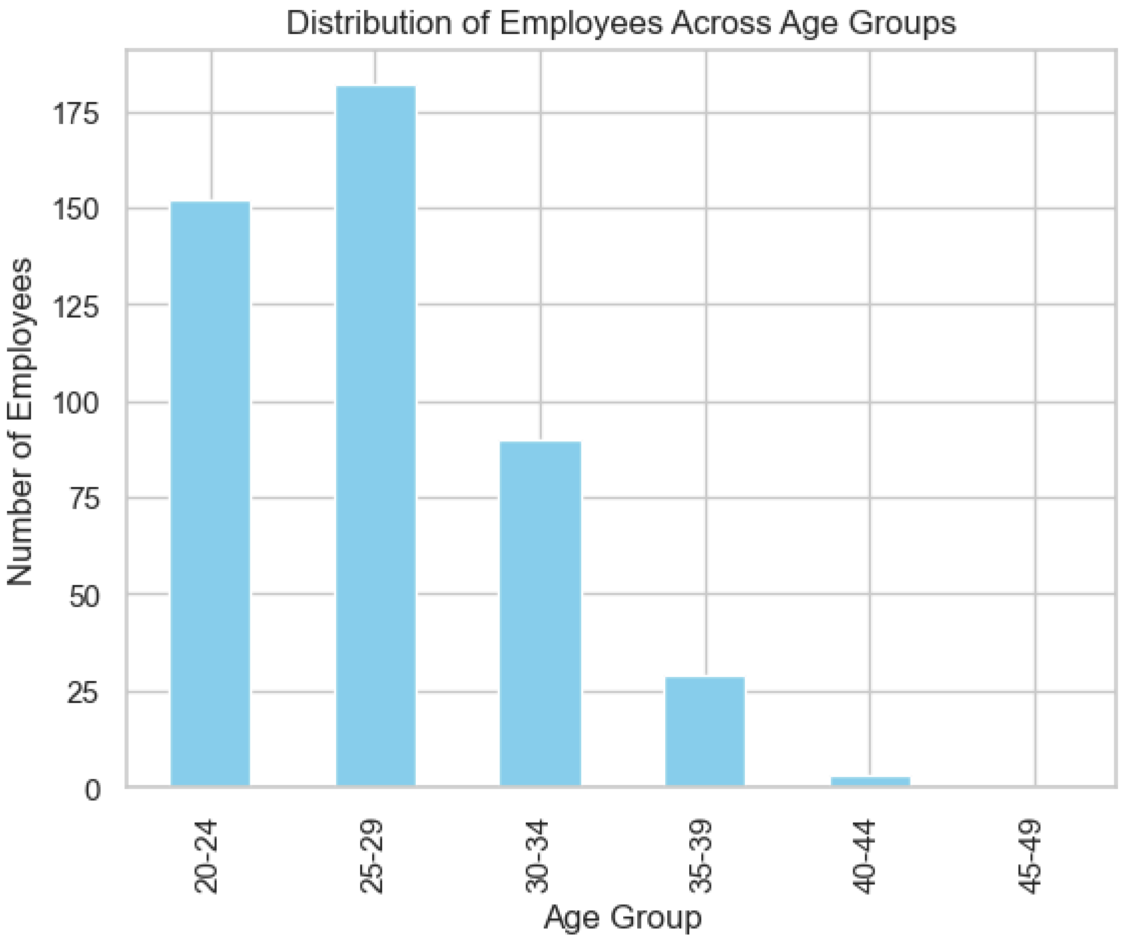
In [ ]:

3. Identify the predominant age group among employees.

In [29]:

```
import pandas as pd

# Load the dataset
file_path = 'C:\\Users\\Admin\\Desktop\Programs\\preprocessed_abc_company_da
df = pd.read_csv(file_path)

# Define age groups (bins)
bins = [20, 25, 30, 35, 40, 45, 50]
labels = ['20-24', '25-29', '30-34', '35-39', '40-44', '45-49']

# Create a new column 'AgeGroup' based on the bins
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)

# Group by 'AgeGroup' and count the number of employees in each group
age_group_distribution = df['AgeGroup'].value_counts().sort_index()

# Identify the predominant age group
predominant_age_group = age_group_distribution.idxmax()
predominant_age_group_count = age_group_distribution.max()

print("Predominant Age Group:", predominant_age_group)
print("Number of Employees in Predominant Age Group:", predominant_age_grou

# Plotting the age group distribution
age_group_distribution.plot(kind='bar', color='skyblue')
plt.xlabel('Age Group')
plt.ylabel('Number of Employees')
plt.title('Distribution of Employees Across Age Groups')
plt.show()
```

```
Predominant Age Group: 25-29
Number of Employees in Predominant Age Group: 182
```



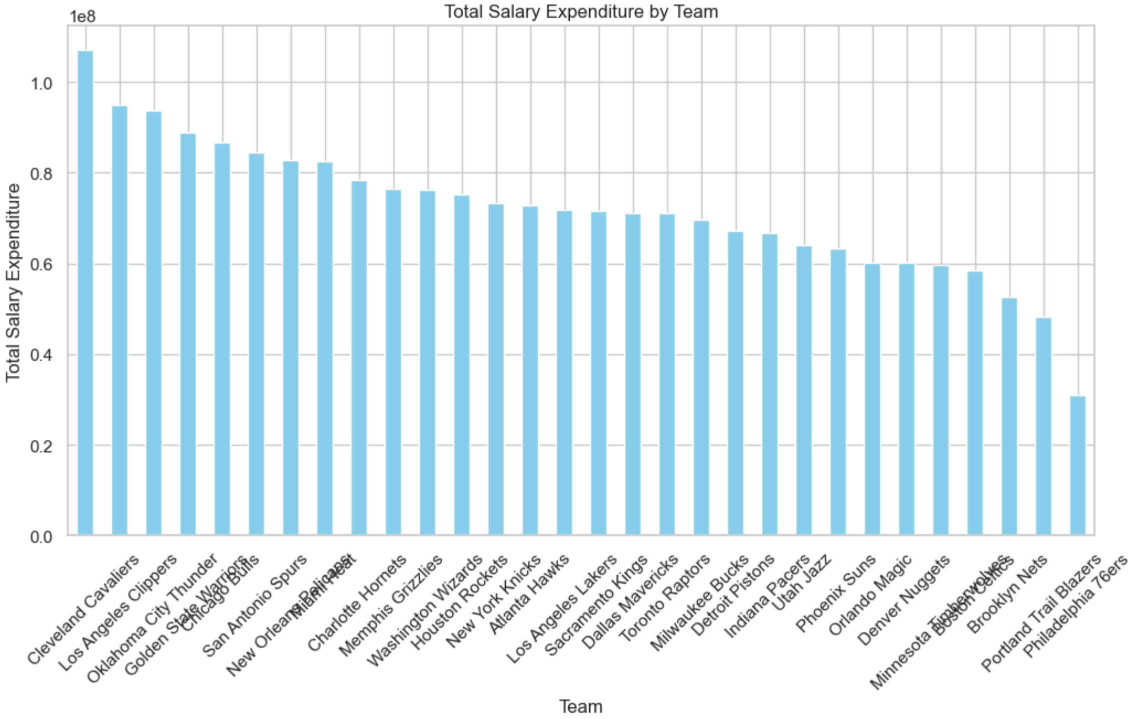
```
In [ ]:
```

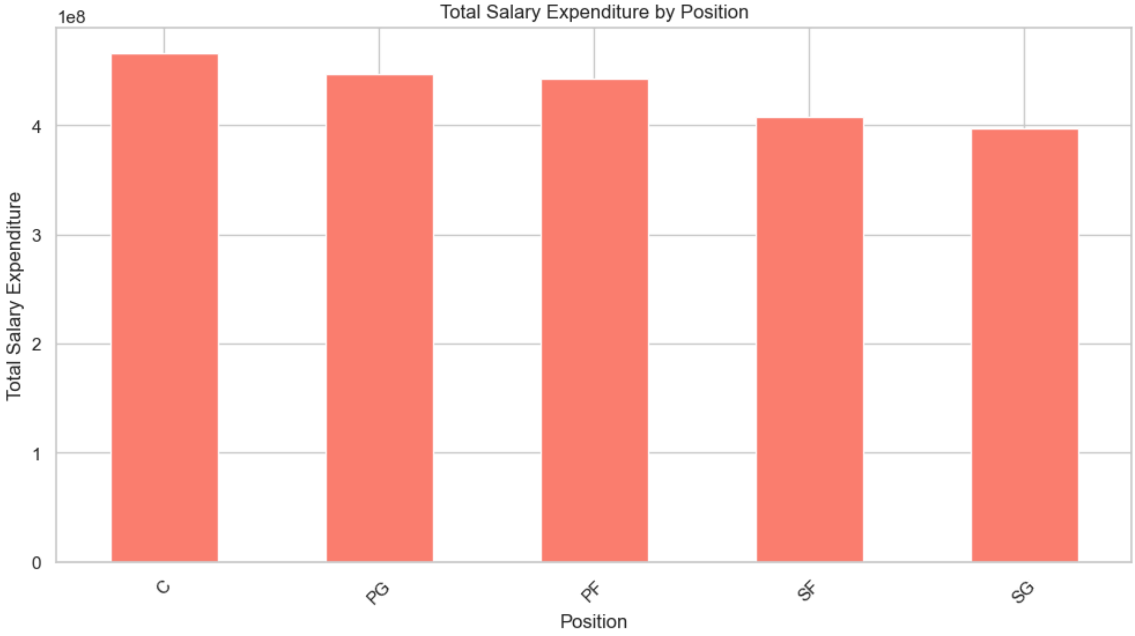
4. Discover which team and position have the highest salary expenditure.

In [35]:

```
import pandas as pd

# Load the dataset
file_path = 'C:\\Users\\Admin\\Downloads\\preprocessed_abc_company_dataset.
df = pd.read_csv(file_path)

# Group by Team and calculate total salary expenditure for each team
team_salary_expenditure = df.groupby('Team')['Salary'].sum().sort_values(as

# Group by Position and calculate total salary expenditure for each positio
position_salary_expenditure = df.groupby('Position')['Salary'].sum().sort_v

# Identify the team with the highest salary expenditure
highest_salary_team = team_salary_expenditure.idxmax()
highest_salary_team_expenditure = team_salary_expenditure.max()

# Identify the position with the highest salary expenditure
highest_salary_position = position_salary_expenditure.idxmax()
highest_salary_position_expenditure = position_salary_expenditure.max()

print("Team with the Highest Salary Expenditure:", highest_salary_team)
print("Highest Salary Expenditure by Team:", highest_salary_team_expenditur
print("\nPosition with the Highest Salary Expenditure:", highest_salary_pos
print("Highest Salary Expenditure by Position:", highest_salary_position_ex

# Plotting the total salary expenditure by team
plt.figure(figsize=(12, 6))
team_salary_expenditure.plot(kind='bar', color='skyblue')
plt.xlabel('Team')
plt.ylabel('Total Salary Expenditure')
plt.title('Total Salary Expenditure by Team')
plt.xticks(rotation=45)
plt.show()

# Plotting the total salary expenditure by position
plt.figure(figsize=(12, 6))
position_salary_expenditure.plot(kind='bar', color='salmon')
plt.xlabel('Position')
plt.ylabel('Total Salary Expenditure')
plt.title('Total Salary Expenditure by Position')
plt.xticks(rotation=45)
plt.show()
```

```
Team with the Highest Salary Expenditure: Cleveland Cavaliers
Highest Salary Expenditure by Team: 106988689.0

Position with the Highest Salary Expenditure: C
Highest Salary Expenditure by Position: 466377332.0
```



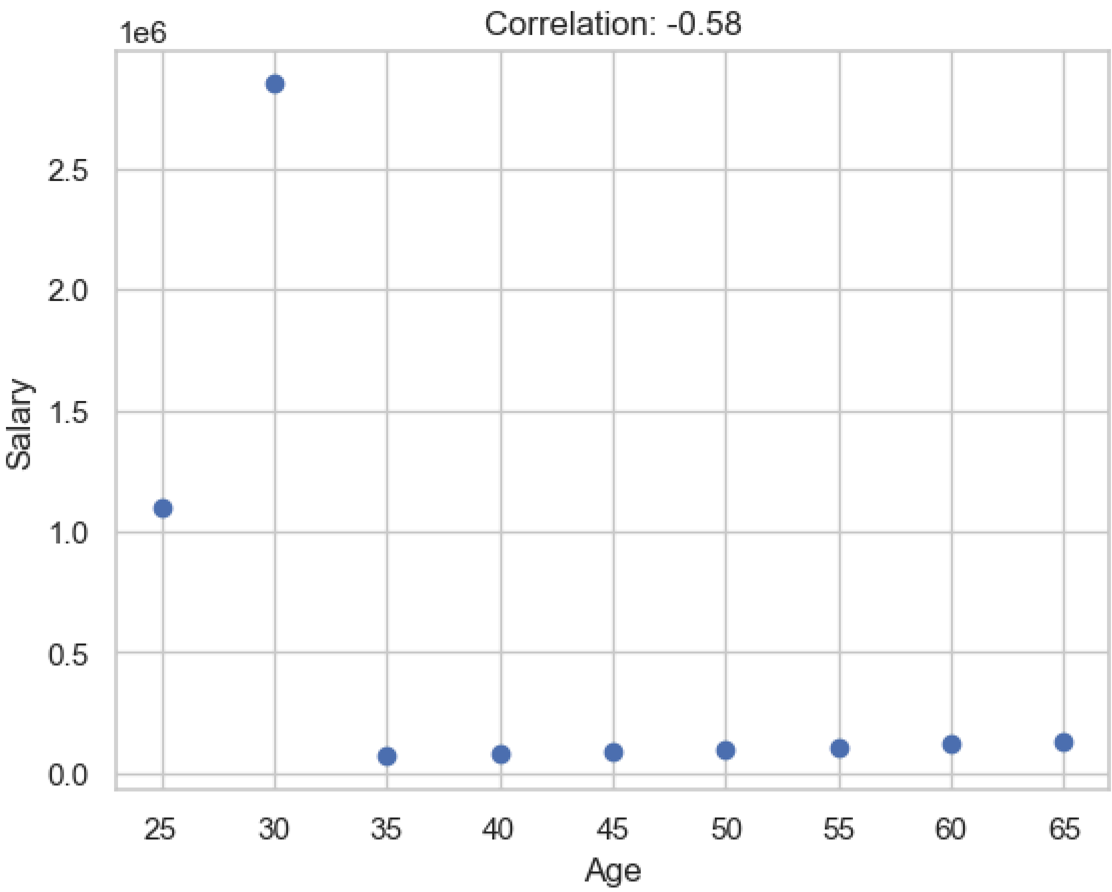


In [ ]:

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

In [40]:

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


data = {
    'Age': [25, 30, 35, 40, 45, 50, 55, 60, 65],
    'Salary': [1100602
, 2850000
, 70000, 80000, 90000, 100000, 110000, 120000, 130000]
}

# Create DataFrame
df = pd.DataFrame(data)

# Calculate correlation coefficient
correlation = df['Age'].corr(df['Salary'])

# Visual representation
plt.scatter(df['Age'], df['Salary'])
plt.xlabel('Age')
plt.ylabel('Salary')
plt.title(f'Correlation: {correlation:.2f}')
plt.grid(True)
plt.show()

print("Correlation coefficient:", correlation)
```



```
Correlation coefficient: -0.5840475876261246
```

In [ ]:

In [ ]: