Python Module End Project (DSML)

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

In [3]:
```
df=pd.read_csv("myexcel.csv")
df
```

Out[3]:

	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

Preprocessing:

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. (1 mark)

In [13]:
```
df["Height"]=np.random.randint(150,180,len(df))
```

In [14]:
```
df
```

Out[14]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	171	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	175	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	163	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	170	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	170	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	169	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	167	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	159	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	167	231	Kansas	947276.0
457	Priyanka	Utah Jazz	34	C	25	151	231	Kansas	947276.0

458 rows × 9 columns

In [7]:

```
dfc=df.copy()
dfc
```

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

In [16]:

```
dfc.isnull().sum()
```

Out[16]:

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

In [17]:

```
dfc.dropna()
```

Out[17]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	171	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	175	235	Marquette	6796117.0
3	R.J. Hunter	Boston Celtics	28	SG	22	170	185	Georgia State	1148640.0
6	Jordan Mickey	Boston Celtics	55	PF	21	178	235	LSU	1170960.0
7	Kelly Olynyk	Boston Celtics	41	C	25	172	238	Gonzaga	2165160.0
...	...	...	...	...	...	...	...	...	...
451	Chris Johnson	Utah Jazz	23	SF	26	157	206	Dayton	981348.0
452	Trey Lyles	Utah Jazz	41	PF	20	156	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	PG	26	169	203	Butler	2433333.0
456	Jeff Withey	Utah Jazz	24	C	26	167	231	Kansas	947276.0
457	Priyanka	Utah Jazz	34	C	25	151	231	Kansas	947276.0

365 rows × 9 columns

In [18]:
```
dfc.isnull().sum()
```

Out[18]:
```
Name         0
Team         0
Number       0
Position     0
Age          0
Height       0
Weight       0
College     84
Salary      11
dtype: int64
```

In [19]:
```
dfc.replace(["", " ", "None", "null"],np.nan,inplace=True)
```

In [23]:
```
dfc.Salary.isnull().sum()
```

Out[23]:
```
11
```

In [9]:
```
# Replace null values in categorical columns with the column mode
dfc['College'] = dfc['College'].fillna(dfc['College'].mode()[0])
```

In [11]:
```
dfc.College.isnull().sum()
```

Out[11]:
```
0
```

In [15]:
```
# Replace null values in numerical columns with the column mean
dfc['Salary'] = dfc['Salary'].fillna(dfc['Salary'].mean())
```

In [25]:
```
dfc.Salary.isnull().sum()
```

Out[25]:
```
0
```

In [17]:
```
dfc.isnull().sum()
```

Out[17]:
```
Name        0
Team        0
Number      0
Position    0
Age         0
Height      0
Weight      0
College     0
Salary      0
dtype: int64
```

In [27]:
```
dfc.describe()
```

Out[27]:

	Number	Age	Height	Weight	Salary
count	458.000000	458.000000	458.000000	458.000000	4.580000e+02
mean	17.713974	26.934498	165.021834	221.543668	4.833970e+06
std	15.966837	4.400128	8.626683	26.343200	5.163335e+06
min	0.000000	19.000000	150.000000	161.000000	3.088800e+04
25%	5.000000	24.000000	157.000000	200.000000	1.100150e+06
50%	13.000000	26.000000	166.000000	220.000000	2.862190e+06
75%	25.000000	30.000000	172.750000	240.000000	6.323553e+06
max	99.000000	40.000000	179.000000	307.000000	2.500000e+07

In [28]:
```
dfc.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   374 non-null    object 
 8   Salary    458 non-null    float64
dtypes: float64(1), int32(1), int64(3), object(4)
memory usage: 30.5+ KB
```

1. Determine the distribution of Players across each team and calculate the percentage split relative to the total number of Players. (2 marks)

In [30]:
```
dfc
```

Out[30]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	171	180	Texas	7.730337e+06
1	Jae Crowder	Boston Celtics	99	SF	25	175	235	Marquette	6.796117e+06
2	John Holland	Boston Celtics	30	SG	27	163	205	Boston University	4.833970e+06
3	R.J. Hunter	Boston Celtics	28	SG	22	170	185	Georgia State	1.148640e+06
4	Jonas Jerebko	Boston Celtics	8	PF	29	170	231	NaN	5.000000e+06
...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	169	203	Butler	2.433333e+06
454	Raul Neto	Utah Jazz	25	PG	24	167	179	NaN	9.000000e+05
455	Tibor Pleiss	Utah Jazz	21	C	26	159	256	NaN	2.900000e+06
456	Jeff Withey	Utah Jazz	24	C	26	167	231	Kansas	9.472760e+05
457	Priyanka	Utah Jazz	34	C	25	151	231	Kansas	9.472760e+05

458 rows × 9 columns

In [31]:

```
T_count=dfc.groupby("Team")#grouping by team
T_count
```

Out[31]: <pandas.core.groupby.generic.DataFrameGroupBy object at 0x000002B6B0641050>

In [32]:

```
T_count.size().head()
```

Out[32]:

```
Team
Atlanta Hawks        15
Boston Celtics       15
Brooklyn Nets        15
Charlotte Hornets    15
Chicago Bulls        15
dtype: int64
```

Finding the percentage split

In [34]:

```
t_count=dfc['Team'].value_counts()
total_count=len(dfc)
percentage=round((t_count/total_count)*100,2)  #Calculating the percentage split
percentage.head()
```

Out[34]:
```
Team
New Orleans Pelicans    4.15
Memphis Grizzlies       3.93
Utah Jazz               3.49
New York Knicks         3.49
Milwaukee Bucks         3.49
Name: count, dtype: float64
```

In [35]:
```
resultantdf= pd.DataFrame({
    'Team': t_count.index,
    'Employee count':t_count.values,
    'Percentage Split':percentage
})
resultantdf.head()
```

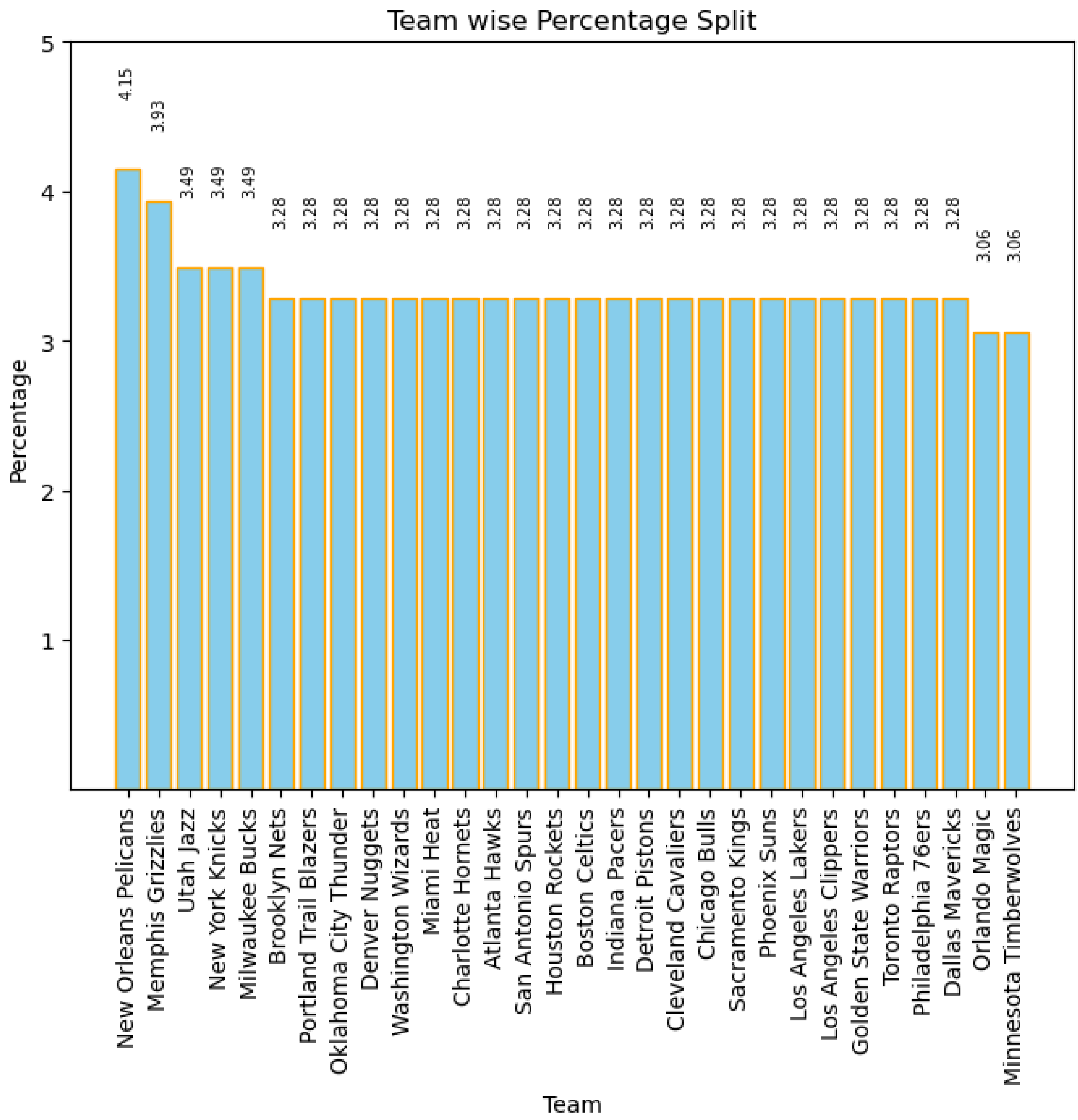
Out[35]:

	Team	Employee count	Percentage Split
Team			
New Orleans Pelicans	New Orleans Pelicans	19	4.15
Memphis Grizzlies	Memphis Grizzlies	18	3.93
Utah Jazz	Utah Jazz	16	3.49
New York Knicks	New York Knicks	16	3.49
Milwaukee Bucks	Milwaukee Bucks	16	3.49

Visualising percentage split of Each Team

In [37]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [38]:
```
plt.figure(figsize=(8,6))
plt.bar(resultantdf['Team'] ,resultantdf['Percentage Split'],color='skyblue',edgeco
plt.title("Team wise Percentage Split")
plt.xlabel('Team')
plt.ylabel('Percentage')
plt.yticks([1,2,3,4,5,])
plt.xticks(rotation=90)
# Add data labels
for i, v in enumerate(resultantdf['Percentage Split']):
    plt.text(i, v + 0.5, str(v), ha='center',rotation=90,fontsize=7)
plt.show()
```



Inference : "New Orleans Pelicans" has the highest percentage of players.

In [118…

```
# plt.figure(figsize=(8,6))
# plt.pie(resultantdf['Percentage Split'],labels=resultantdf['Team'],autopct='%1.0f
# plt.show()
```

2. Segregate Players based on their positions. (2 marks)

In [42]:

```
dfc.head()
```

Out[42]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	171	180	Texas	7.730337e+06
1	Jae Crowder	Boston Celtics	99	SF	25	175	235	Marquette	6.796117e+06
2	John Holland	Boston Celtics	30	SG	27	163	205	Boston University	4.833970e+06
3	R.J. Hunter	Boston Celtics	28	SG	22	170	185	Georgia State	1.148640e+06
4	Jonas Jerebko	Boston Celtics	8	PF	29	170	231	NaN	5.000000e+06

In [43]:

```
P_count=dfc['Position'].value_counts()
P_count
```

Out[43]:

```
Position
SG    102
PF    100
PG     92
SF     85
C      79
Name: count, dtype: int64
```

Segregated dataset

In [154…

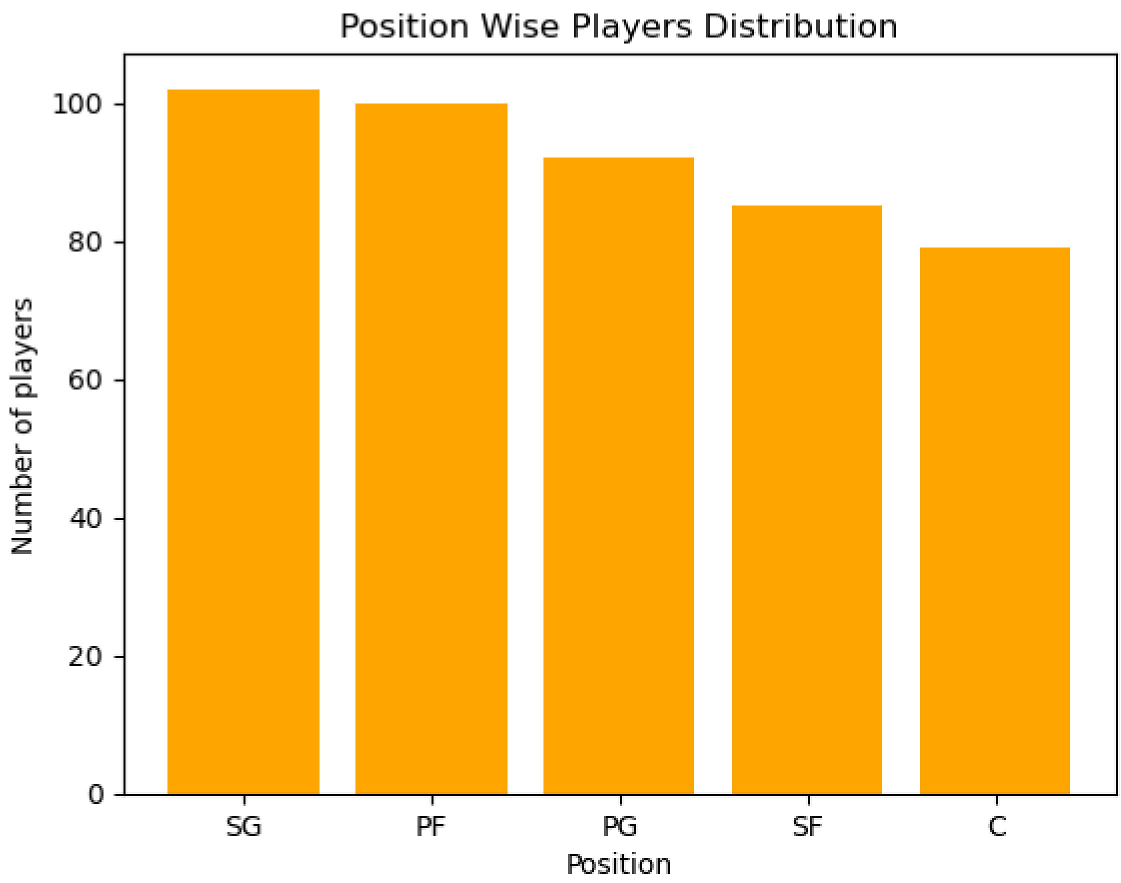
```
Position=pd.DataFrame({
    'Position':P_count.index,
    'Number of players':P_count.values
})
Position
```

Out[154]:

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

In [160…

```
plt.bar(Position['Position'] ,Position['Number of players'],color='Orange')
plt.title("Position Wise Players Distribution")
plt.xlabel('Position')
plt.ylabel('Number of players')
plt.show()
```



Inference : Position 'SG' has the highest number of players and Position 'C' has the lowest number of players.

3. Identify the predominant age group among the Players. (2 marks)

In [47]:
```
dfc.head()
```

Out[47]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	171	180	Texas	7.730337e+06
1	Jae Crowder	Boston Celtics	99	SF	25	175	235	Marquette	6.796117e+06
2	John Holland	Boston Celtics	30	SG	27	163	205	Boston University	4.833970e+06
3	R.J. Hunter	Boston Celtics	28	SG	22	170	185	Georgia State	1.148640e+06
4	Jonas Jerebko	Boston Celtics	8	PF	29	170	231	NaN	5.000000e+06

In [48]:
```
Agegroup=dfc.groupby('Age')
Agegroup
```

Out[48]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000002B6B39FBF10>
```

In [49]:
```
Agegroup.size()
```

Out[49]:
```
Age
19     2
20    19
21    19
22    26
23    41
24    47
25    46
26    36
27    41
28    31
29    28
30    31
31    22
32    13
33    14
34    10
35     9
36    10
37     4
38     4
39     2
40     3
dtype: int64
```

Identifying the predominant age group.

In [50]:
```
Agegroup.size().max() # Predominant Age
```

Out[50]:
```
47
```

In [51]:
```
Age_group=pd.DataFrame({
    'Name':dfc['Name'],
    'Age':dfc['Age']
})
Age_group.head()
```

Out[51]:

	Name	Age
0	Avery Bradley	25
1	Jae Crowder	25
2	John Holland	27
3	R.J. Hunter	22
4	Jonas Jerebko	29

Creating groups for age.

In [52]:
```
bins=[10,15,20,25,30,35,40,45,50]
labels=['10-15','15-20','20-25','25-30','30-35','35-40','40-45','45-50']
Age_group['Age_Group'] = pd.cut(Age_group['Age'], bins=bins, labels=labels, right=F
Age_group
```

Out[52]:

	Name	Age	Age_Group
0	Avery Bradley	25	25-30
1	Jae Crowder	25	25-30
2	John Holland	27	25-30
3	R.J. Hunter	22	20-25
4	Jonas Jerebko	29	25-30
...	...	...	...
453	Shelvin Mack	26	25-30
454	Raul Neto	24	20-25
455	Tibor Pleiss	26	25-30
456	Jeff Withey	26	25-30
457	Priyanka	25	25-30

458 rows × 3 columns

In [53]:

```
A_group=Age_group.groupby('Age_Group')
A_group
```

Out[53]: <pandas.core.groupby.generic.DataFrameGroupBy object at 0x000002B6B3A48810>

In [54]:

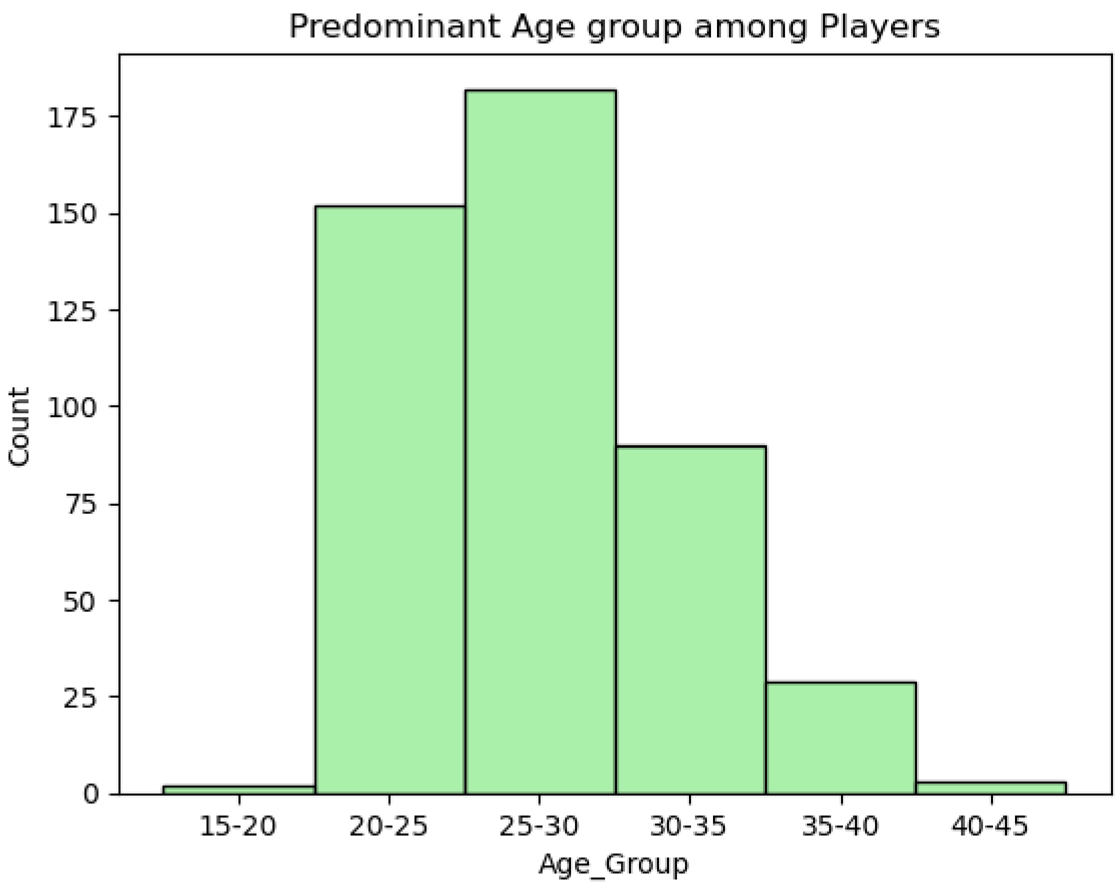
```
A_group.size()# finding the predominant age group
```

Out[54]:

```
Age_Group
10-15      0
15-20      2
20-25    152
25-30    182
30-35     90
35-40     29
40-45      3
45-50      0
dtype: int64
```

In [158…

```
sns.histplot(Age_group['Age_Group'], bins=9, color='lightgreen', edgecolor='black')
plt.title("Predominant Age group among Players")
plt.show()
```



Found that "25 to 30 " is the predominant age group.

4. Discover which team and position have the highest salary expenditure. (2 marks)

Finding the position wise mean of salaries to find which position has highest salary expenditure.

In [58]:
```
Position=dfc.groupby('Position')['Salary'].mean()
Position
```

Out[58]:
```
Position
C     5.903511e+06
PF    4.570628e+06
PG    5.067227e+06
SF    4.857117e+06
SG    4.034100e+06
Name: Salary, dtype: float64
```

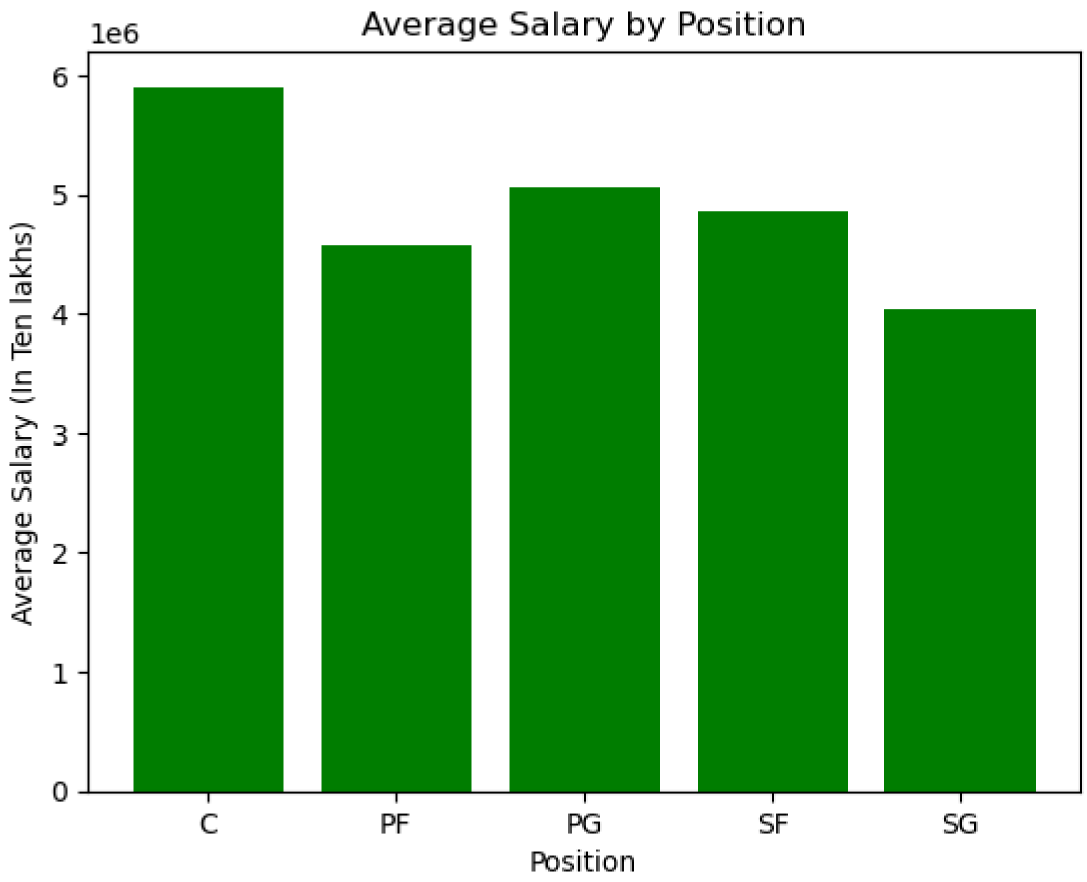
In [59]:
```
Position.max()
```

Out[59]:
```
5903510.53164557
```

In [60]:
```
dfc['Position'].unique()
```

Out[60]:
```
array(['PG', 'SF', 'SG', 'PF', 'C'], dtype=object)
```

In [61]:
```
plt.bar(Position.index,Position.values,color='green')
plt.xlabel('Position')
plt.ylabel('Average Salary (In Ten lakhs)')
plt.title('Average Salary by Position')
plt.show()
```



Inference : Highest salary is for 'C' position .

In [163…
```
team=dfc.groupby('Team')['Salary'].mean()
team
```

Out[163]:

```
Team
Atlanta Hawks             4.860197e+06
Boston Celtics            4.225003e+06
Brooklyn Nets             3.501898e+06
Charlotte Hornets         5.222728e+06
Chicago Bulls             5.785559e+06
Cleveland Cavaliers       7.454844e+06
Dallas Mavericks          4.746582e+06
Denver Nuggets            4.330393e+06
Detroit Pistons           4.477884e+06
Golden State Warriors     5.924600e+06
Houston Rockets           5.018868e+06
Indiana Pacers            4.450122e+06
Los Angeles Clippers      6.323643e+06
Los Angeles Lakers        4.784695e+06
Memphis Grizzlies         5.327042e+06
Miami Heat                6.145574e+06
Milwaukee Bucks           4.350220e+06
Minnesota Timberwolves    4.610262e+06
New Orleans Pelicans      4.355304e+06
New York Knicks           4.581494e+06
Oklahoma City Thunder     6.251020e+06
Orlando Magic             4.297248e+06
Philadelphia 76ers        2.388458e+06
Phoenix Suns              4.229676e+06
Portland Trail Blazers    3.220121e+06
Sacramento Kings          4.778911e+06
San Antonio Spurs         5.629516e+06
Toronto Raptors           4.741174e+06
Utah Jazz                 4.000460e+06
Washington Wizards        5.088576e+06
Name: Salary, dtype: float64
```

In [171…

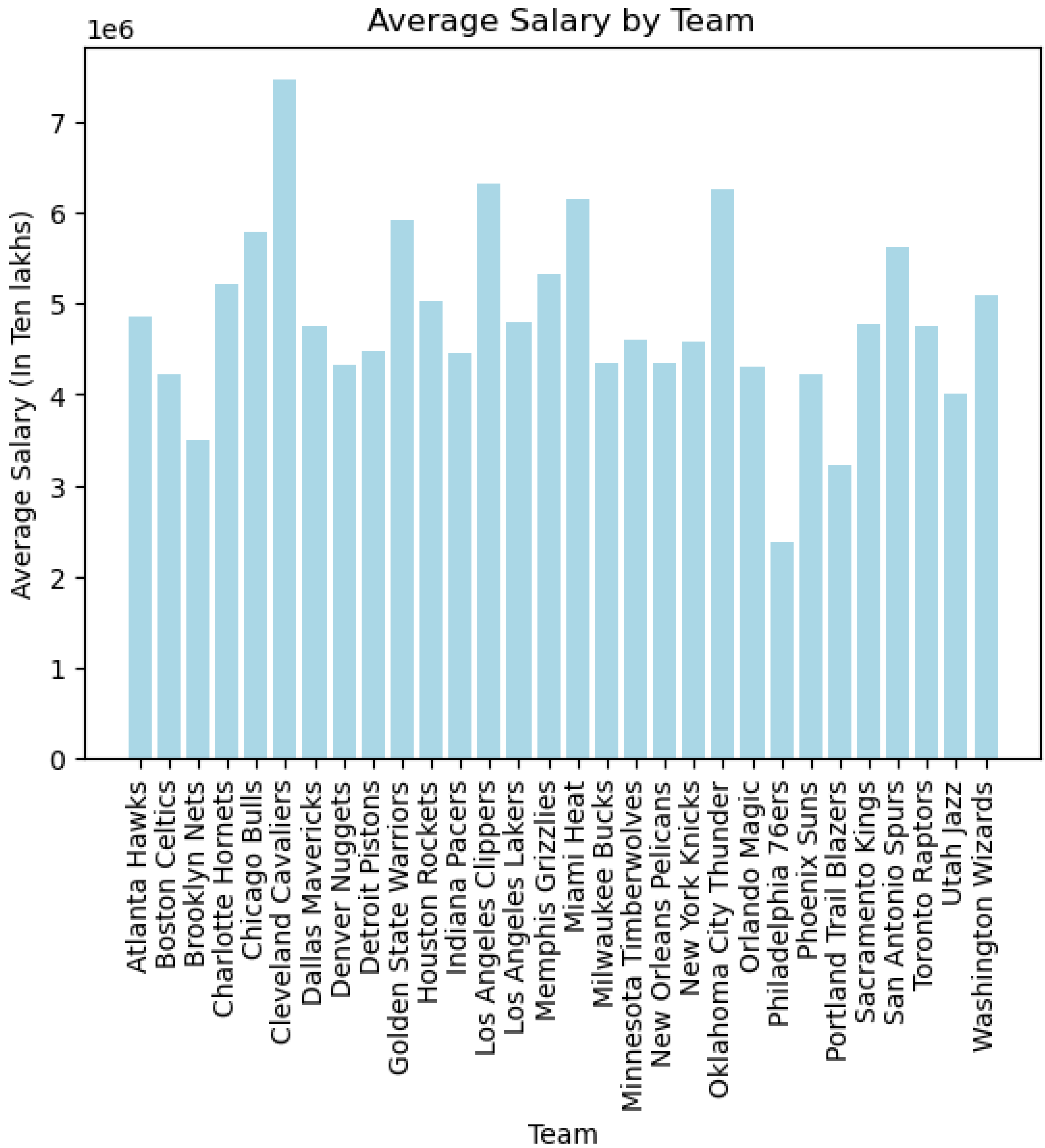
```
team_sorted = team.sort_values(ascending=False)
team_sorted
```

Out[171]:

```
Team
Cleveland Cavaliers       7.454844e+06
Los Angeles Clippers      6.323643e+06
Oklahoma City Thunder     6.251020e+06
Miami Heat                6.145574e+06
Golden State Warriors     5.924600e+06
Chicago Bulls             5.785559e+06
San Antonio Spurs         5.629516e+06
Memphis Grizzlies         5.327042e+06
Charlotte Hornets         5.222728e+06
Washington Wizards        5.088576e+06
Houston Rockets           5.018868e+06
Atlanta Hawks             4.860197e+06
Los Angeles Lakers        4.784695e+06
Sacramento Kings          4.778911e+06
Dallas Mavericks          4.746582e+06
Toronto Raptors           4.741174e+06
Minnesota Timberwolves    4.610262e+06
New York Knicks           4.581494e+06
Detroit Pistons           4.477884e+06
Indiana Pacers            4.450122e+06
New Orleans Pelicans      4.355304e+06
Milwaukee Bucks           4.350220e+06
Denver Nuggets            4.330393e+06
Orlando Magic             4.297248e+06
Phoenix Suns              4.229676e+06
Boston Celtics            4.225003e+06
Utah Jazz                 4.000460e+06
Brooklyn Nets             3.501898e+06
Portland Trail Blazers    3.220121e+06
Philadelphia 76ers        2.388458e+06
Name: Salary, dtype: float64
```

In [183…

```
plt.bar(team.index,team.values,color='lightblue')
plt.xlabel('Team')
plt.ylabel('Average Salary (In Ten lakhs)')
plt.title('Average Salary by Team')
plt.xticks(rotation=90)
plt.show()
```



Inference : Team 'Cleveland Cavaliers ' have the highest salary.

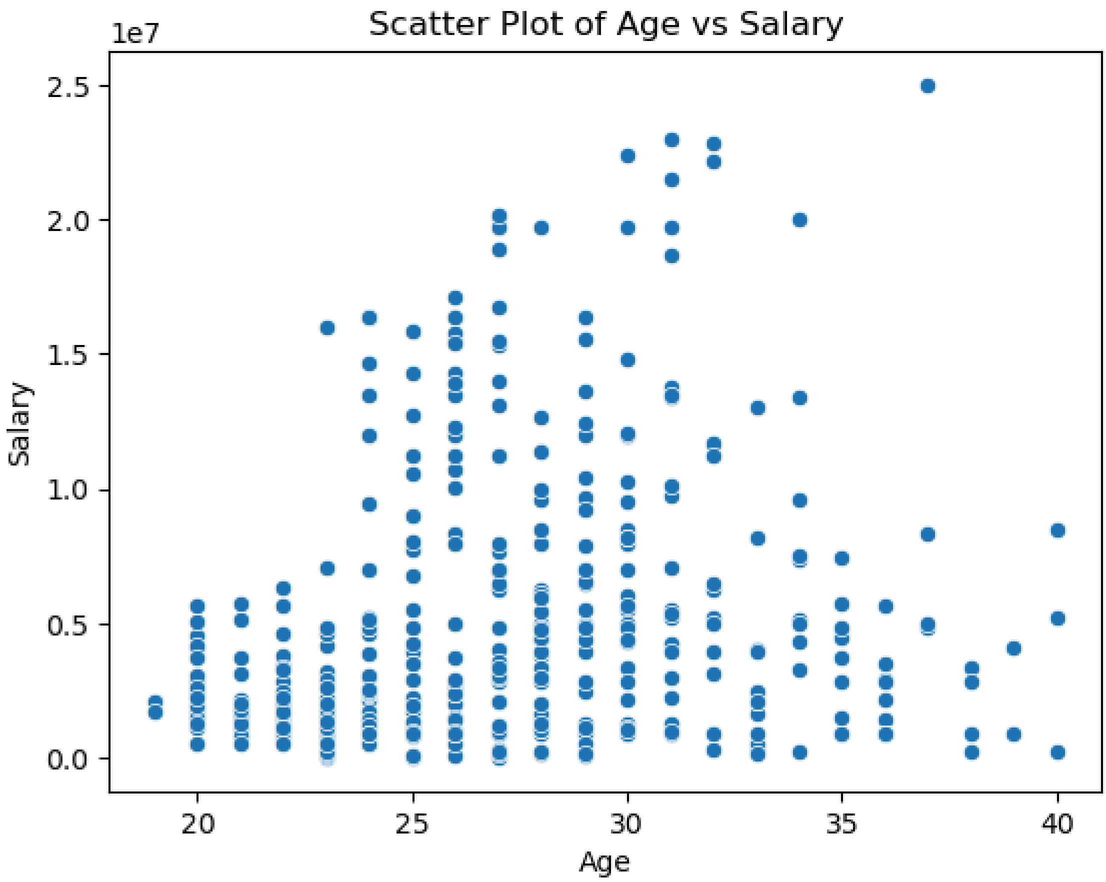
5. Investigate if there's any correlation between age and salary, and represent it visually. (2 marks)

In [64]:
```
# Pearson correlation (default)
correlation = dfc['Age'].corr(df['Salary'])
print(f"Pearson Correlation: {correlation:.2f}") #This is a method for finding the
```

```
Pearson Correlation: 0.21
```

Inference : The correlation between Age and Salary is slightly positive, that is When age increases the salary will increase in a slight manner.

In [66]:
```
# Scatter plot
sns.scatterplot(x='Age', y='Salary', data=dfc)
plt.title("Scatter Plot of Age vs Salary")
plt.xlabel("Age")
plt.ylabel("Salary")
plt.show()
```

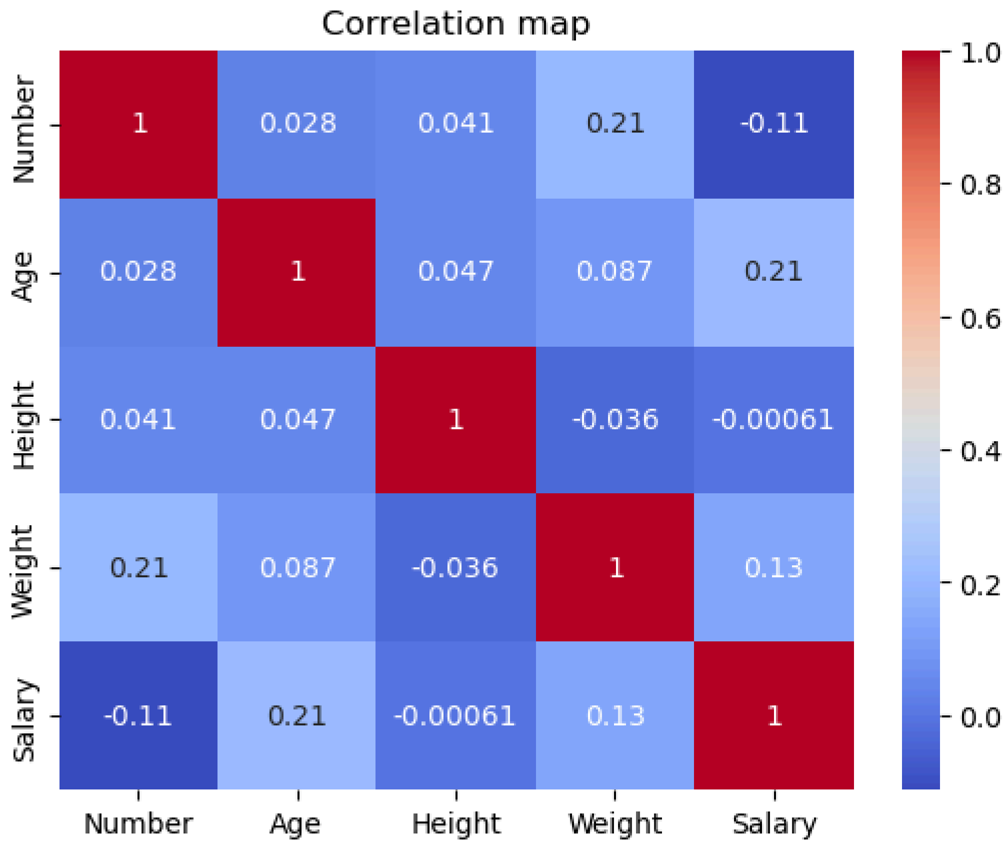


Detailed Analysis:

- General Trend: The plot still suggests a weak positive correlation, as the salaries tend to slightly increase with age, particularly between ages 20 to 30.
- Clusters: A significant number of data points are clustered between ages 20 and 30, with salaries mostly below 1e7. There are fewer data points for individuals above age 35, indicating either fewer data entries or less representation in the dataset.
- Outliers: Some individuals, especially between ages 25 and 30, have exceptionally high salaries (above 2e7). These outliers could be key executives, entrepreneurs, or anomalies in the data.
- Plateau or Decline: For ages above 30, the data points seem to flatten or spread more, with salaries not increasing significantly or even decreasing in some cases.

A heat map also ploting for much more relations .

In [69]:
```
numeric_df = dfc.select_dtypes(include=['number'])
sns.heatmap(numeric_df.corr(),annot=True, cmap='coolwarm')
plt.title('Correlation map')
plt.show()
```



Different correlation is visible here.

Inferences :

- "New Orleans Pelicans" has the highest percentage of players.
- Position 'SG' has the highest number of players and Position 'C' has the lowest number of players.
- Found that "25 to 30 " is the predominant age group.
- Highest salary is for 'C' position .
- The correlation between Age and Salary is slightly positive, that is When age increases the salary will increase in a slight manner.
- Team 'Cleveland Cavaliers ' have the highest salary.

```
In [ ]:
```