Analysing employee details in company ABC

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

#reading the excel data into dataframe
df = pd.read_excel("C:\DSML Downloads\Python Module End Project\Employees_A

#displaying top rows
df.head()
```

Out[1]:

	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 [3]:
```
#details about columns
df.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    object
 6   Weight    458 non-null    int64
 7   College   374 non-null    object
 8   Salary    447 non-null    float64
dtypes: float64(1), int64(3), object(5)
memory usage: 32.3+ KB
```

Data Preprocessing

In [30]:
```
#changing value of column Height as per requirement

df['Height'] = np.random.randint(150,190, df.shape[0])
df.head(10)
```

Out[30]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	172	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	158	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	174	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	178	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	180	231	NaN	5000000.0
5	Amir Johnson	Boston Celtics	90	PF	29	154	240	NaN	12000000.0
6	Jordan Mickey	Boston Celtics	55	PF	21	155	235	LSU	1170960.0
7	Kelly Olynyk	Boston Celtics	41	C	25	184	238	Gonzaga	2165160.0
8	Terry Rozier	Boston Celtics	12	PG	22	180	190	Louisville	1824360.0
9	Marcus Smart	Boston Celtics	36	PG	22	183	220	Oklahoma State	3431040.0

In [32]:
```
# dropping unwanted columns
df = df.drop(['Number','Height','Weight'], axis = 1)
df.head()
```

Out[32]:

	Name	Team	Position	Age	College	Salary
0	Avery Bradley	Boston Celtics	PG	25	Texas	7730337.0
1	Jae Crowder	Boston Celtics	SF	25	Marquette	6796117.0
2	John Holland	Boston Celtics	SG	27	Boston University	NaN
3	R.J. Hunter	Boston Celtics	SG	22	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	PF	29	NaN	5000000.0

In [33]:
```
#replacing NaN salary values
df['Salary'] = df['Salary'].fillna(0)
df.head()
```

Out[33]:

	Name	Team	Position	Age	College	Salary
0	Avery Bradley	Boston Celtics	PG	25	Texas	7730337.0
1	Jae Crowder	Boston Celtics	SF	25	Marquette	6796117.0
2	John Holland	Boston Celtics	SG	27	Boston University	0.0
3	R.J. Hunter	Boston Celtics	SG	22	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	PF	29	NaN	5000000.0

Statistical Analysis

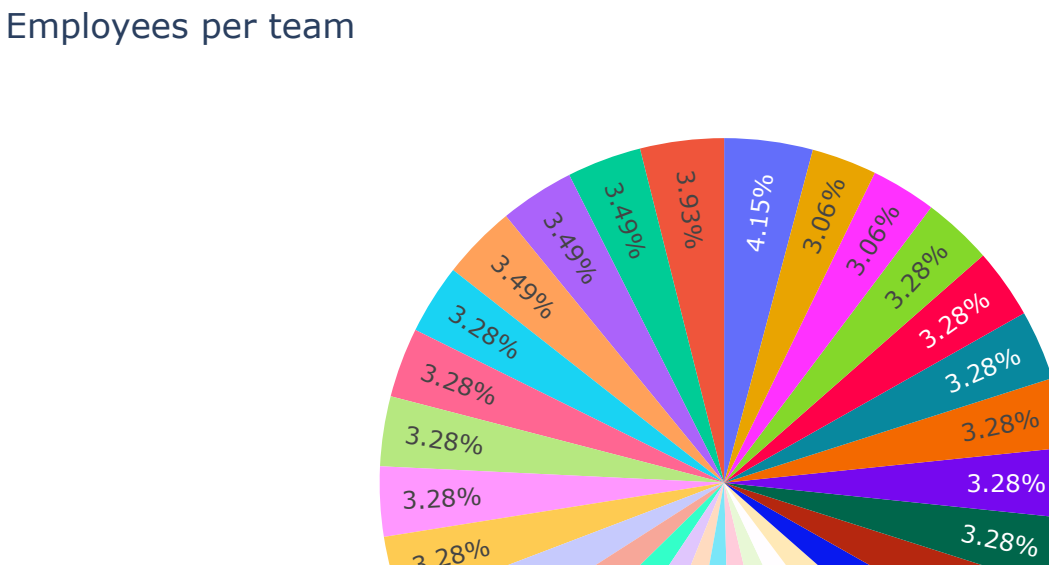
In [3]:
```
df.describe()
```

Out[3]:

	Number	Age	Weight	Salary
count	458.000000	458.000000	458.000000	4.470000e+02
mean	17.713974	26.934498	221.543668	4.833970e+06
std	15.966837	4.400128	26.343200	5.226620e+06
min	0.000000	19.000000	161.000000	3.088800e+04
25%	5.000000	24.000000	200.000000	1.025210e+06
50%	13.000000	26.000000	220.000000	2.836186e+06
75%	25.000000	30.000000	240.000000	6.500000e+06
max	99.000000	40.000000	307.000000	2.500000e+07

1. Finding number of employees in each Team and the percentage splitting with respect to the total employees.

In [2]:
```
import plotly.express as px

#categorizing employees based on team
team_count = df['Team'].value_counts()
emp = team_count.values.tolist()
team = team_count.index.tolist()

#plotting the values in a pie chart
fig = px.pie(values=emp, names=team, title = 'Employees per team', )
fig.show()
```



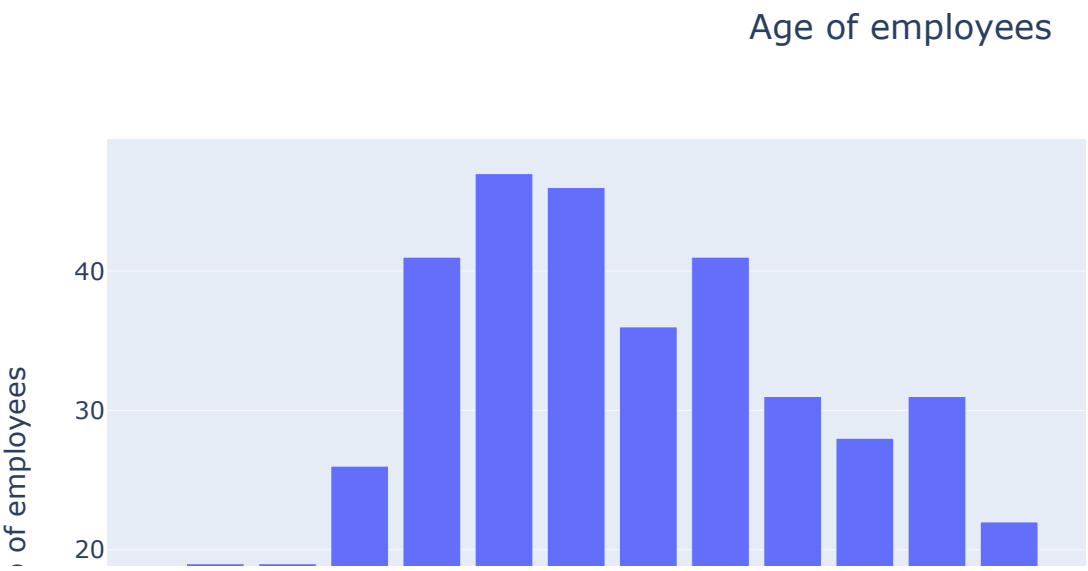
2. Classifying employees with respect to positions

In [16]:
```
#categorizing employees based on position
pos_split = df['Position'].value_counts()

#setting size of the plot
plt.figure(figsize =(5,3))
#creatin plot, assigning x axis, y axis and title values
sns.barplot(x = pos_split.index, y = pos_split.values)
plt.xlabel("Position")
plt.ylabel("No of Employees")
plt.title("Employees in different positions")
#displaying the plot
plt.show()
```

Most employees are in SG

3. Classifying according to age group

In [3]:
```
#categorizing based on age
age_counts = df['Age'].value_counts()

#plotting age and count
fig = px.bar(age_counts, title="Age of employees")
fig.update_layout(xaxis_title = "Age", yaxis_title = "No of employees", tit
fig.show()
```



Most employees are in the age 24

4. Highest paid Team and Position

In [30]:
```
df_max = df.sort_values(by = ['Salary'], ascending=False).iloc[0]
print("The team and position with maximum salary are: ",df_max['Team'],"and
print("Maximum salary is ",df_max['Salary'])
```

```
The team and position with maximum salary are:  Los Angeles Lakers and  SF
Maximum salary is  25000000.0
```

5. Correlation between age and salary

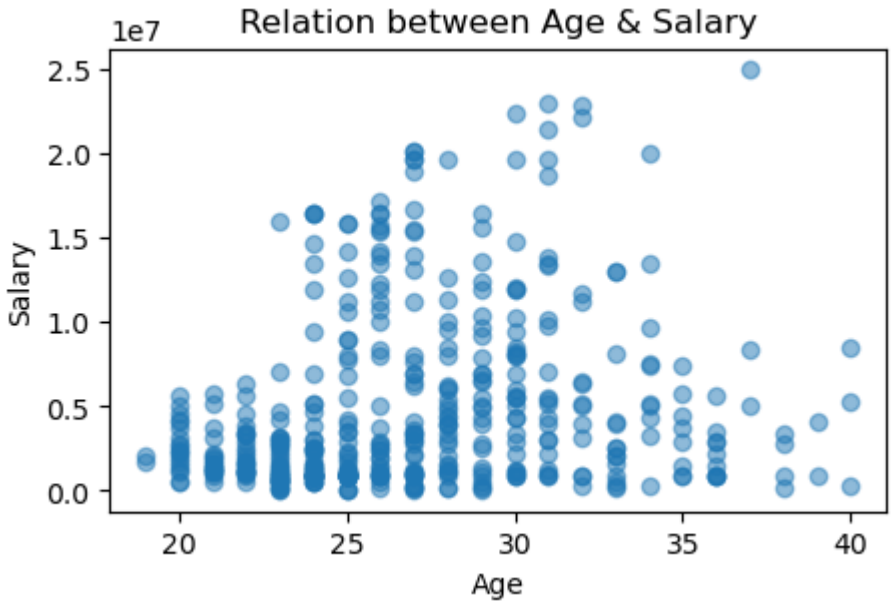
In [8]:
```
age_array = df['Age']
salary_array = df['Salary']
#calculating correlation value between age and salary
print(age_array.corr(salary_array))
print(salary_array.corr(age_array))
```

```
0.21400941226570974
0.21400941226570977
```

The correlation value shows that there is a weak relation between age and salary

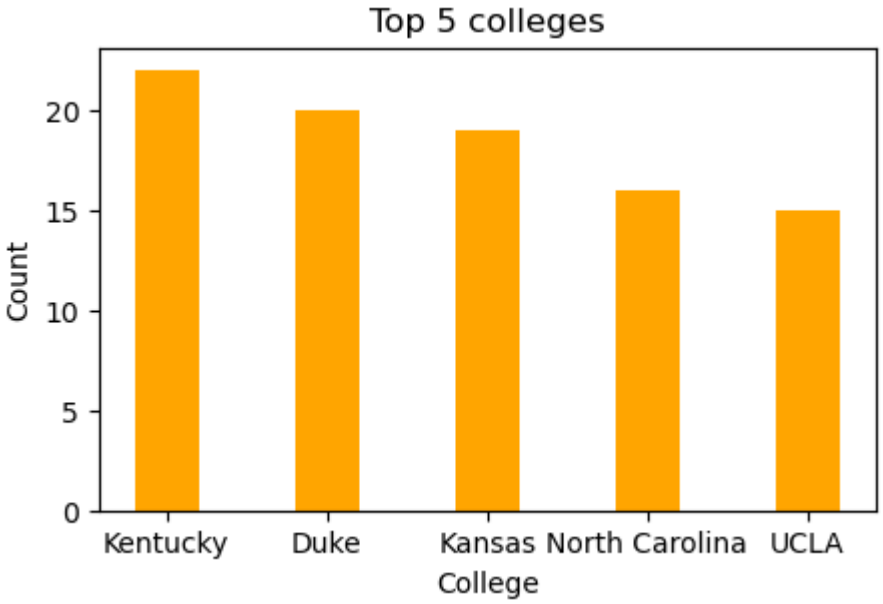
The representation is as below

In [15]:
```
plt.figure(figsize =(5,3))
plt.scatter(age_array, salary_array, alpha=0.5)
plt.xlabel('Age')
plt.ylabel('Salary')
plt.title("Relation between Age & Salary")
plt.show()
```



Top 5 colleges from where employees are hired

In [25]:
```
#categorizing based on college and taking the top 5 values
top_colg = df['College'].value_counts().head(5)
plt.figure(figsize=(5,3))
plt.bar(top_colg.index, top_colg.values, color='orange', width=0.4)
plt.xlabel('College')
plt.ylabel('Count')
plt.title("Top 5 colleges")
plt.show()
```



In [ ]:
```

```