```
import pandas as pd
import numpy as np
```

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

```
data=pd.read_csv('/content/project_data - myexcel.csv.csv')
data
```

	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

Next steps: Generate code with `data` View recommended plots New interactive sheet

checking for any null values

```
data.isnull().sum()
```

	0
Name	0
Team	0
Number	0
Position	0
Age	0
Height	0
Weight	0
College	84
Salary	11

dtype: int64

```
data.dropna ( inplace = True)
```

```
data.isnull().sum()
```

	0
Name	0
Team	0
Number	0
Position	0
Age	0
Height	0
Weight	0
College	0
Salary	0

dtype: int64

REPLACE THE HEIGHT COLUMN WITH RANDOM VALUES BETWEEN 150 and 180

```
data['Height'] = np.random.uniform(150,180,size = len(data))
data
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	168.684734	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	164.101445	235	Marquette	6796117.0
3	R.J. Hunter	Boston Celtics	28	SG	22	178.327763	185	Georgia State	1148640.0
6	Jordan Mickey	Boston Celtics	55	PF	21	169.926498	235	LSU	1170960.0
7	Kelly Olynyk	Boston Celtics	41	C	25	157.662968	238	Gonzaga	2165160.0
...	...	...	...	...	...	...	...	...	...
451	Chris Johnson	Utah Jazz	23	SF	26	164.215846	206	Dayton	981348.0
452	Trey Lyles	Utah Jazz	41	PF	20	169.886309	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	PG	26	178.683299	203	Butler	2433333.0
456	Jeff Withey	Utah Jazz	24	C	26	151.440309	231	Kansas	947276.0
457	Priyanka	Utah Jazz	34	C	25	171.033950	231	Kansas	947276.0

365 rows × 9 columns

Next steps: Generate code with data | View recommended plots | New interactive sheet

Checking for duplicate rows

```
data.drop_duplicates(inplace = True)
data
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	168.684734	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	164.101445	235	Marquette	6796117.0
3	R.J. Hunter	Boston Celtics	28	SG	22	178.327763	185	Georgia State	1148640.0
6	Jordan Mickey	Boston Celtics	55	PF	21	169.926498	235	LSU	1170960.0
7	Kelly Olynyk	Boston Celtics	41	C	25	157.662968	238	Gonzaga	2165160.0
...	...	...	...	...	...	...	...	...	...
451	Chris Johnson	Utah Jazz	23	SF	26	164.215846	206	Dayton	981348.0
452	Trey Lyles	Utah Jazz	41	PF	20	169.886309	234	Kentucky	2239800.0
453	Shelvin Mack	Utah Jazz	8	PG	26	178.683299	203	Butler	2433333.0
456	Jeff Withey	Utah Jazz	24	C	26	151.440309	231	Kansas	947276.0
457	Priyanka	Utah Jazz	34	C	25	171.033950	231	Kansas	947276.0

365 rows × 9 columns

Next steps: Generate code with `data` View recommended plots New interactive sheet

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

```
team_count = data['Team'].value_counts()
team_count
```

	count
Team	
New Orleans Pelicans	16
Portland Trail Blazers	15
Detroit Pistons	15
Milwaukee Bucks	14
Philadelphia 76ers	14
Oklahoma City Thunder	14
Los Angeles Clippers	14
Washington Wizards	13
Charlotte Hornets	13
Phoenix Suns	13
Sacramento Kings	13
Memphis Grizzlies	13
Brooklyn Nets	13
Boston Celtics	12
Dallas Mavericks	12
Indiana Pacers	12
Chicago Bulls	12
Los Angeles Lakers	12
Golden State Warriors	12
Houston Rockets	11
Cleveland Cavaliers	11
San Antonio Spurs	11
Atlanta Hawks	11
New York Knicks	11
Utah Jazz	11
Miami Heat	10
Orlando Magic	10
Toronto Raptors	10
Denver Nuggets	9
Minnesota Timberwolves	8

dtype: int64

percentage split relative to the total number of employees(round to 2 decimal points)

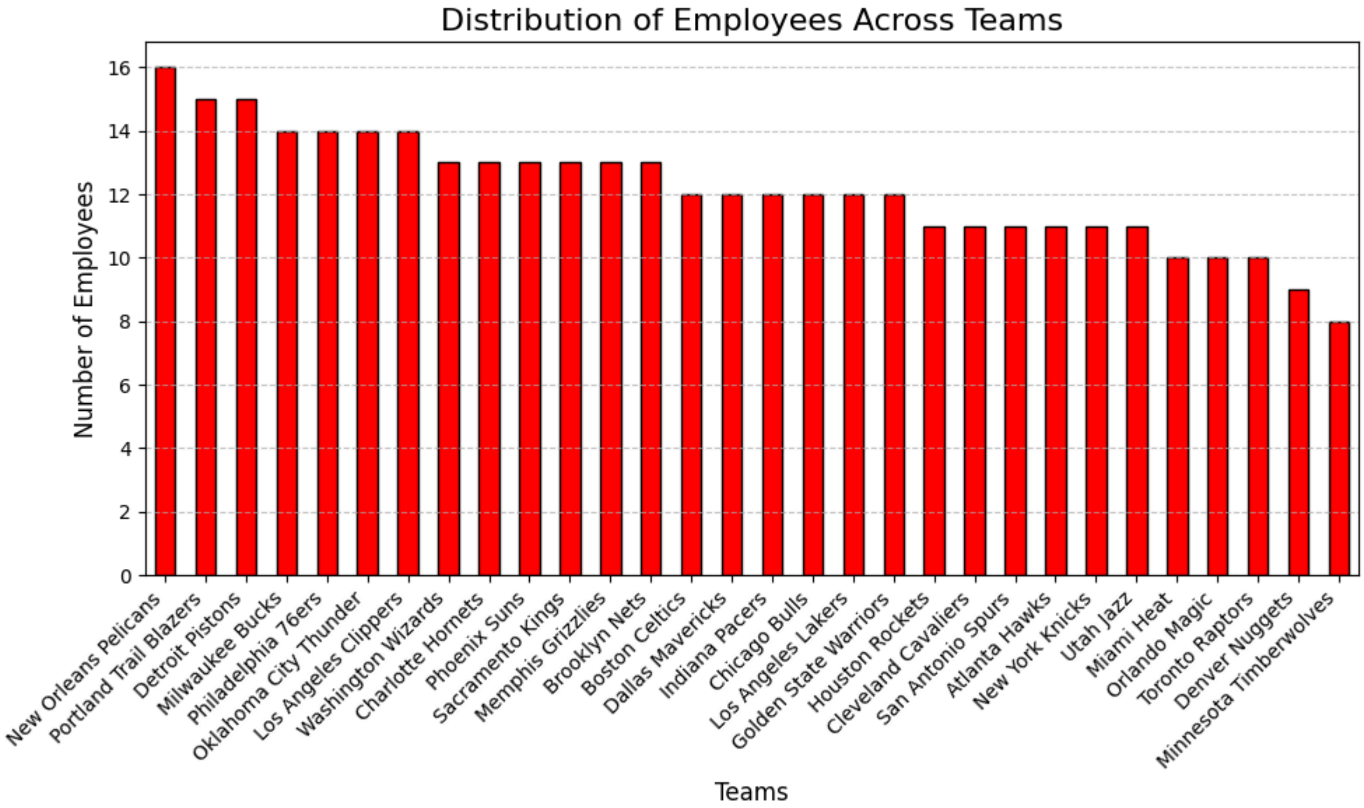
```
percent_team_count = team_count/len(data)*100
percent_team_count.round(2)
```

Team	count
New Orleans Pelicans	4.38
Portland Trail Blazers	4.11
Detroit Pistons	4.11
Milwaukee Bucks	3.84
Philadelphia 76ers	3.84
Oklahoma City Thunder	3.84
Los Angeles Clippers	3.84
Washington Wizards	3.56
Charlotte Hornets	3.56
Phoenix Suns	3.56
Sacramento Kings	3.56
Memphis Grizzlies	3.56
Brooklyn Nets	3.56
Boston Celtics	3.29
Dallas Mavericks	3.29
Indiana Pacers	3.29
Chicago Bulls	3.29
Los Angeles Lakers	3.29
Golden State Warriors	3.29
Houston Rockets	3.01
Cleveland Cavaliers	3.01
San Antonio Spurs	3.01
Atlanta Hawks	3.01
New York Knicks	3.01
Utah Jazz	3.01
Miami Heat	2.74
Orlando Magic	2.74
Toronto Raptors	2.74
Denver Nuggets	2.47
Minnesota Timberwolves	2.19

dtype: float64

```
# graphical representation of employees in each team

plt.figure(figsize=(10, 6))
team_count.plot(kind='bar', color='red', edgecolor='black')

# Customize the plot
plt.title("Distribution of Employees Across Teams", fontsize=16)
plt.xlabel("Teams", fontsize=12)
plt.ylabel("Number of Employees", fontsize=12)
plt.xticks(rotation=45,ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Show the plot
```

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



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

```
employees = data.groupby('Position')['Name'].apply(list)
for Position, Name in employees.items():
    print(f"Employees in {Position} Position:")
    for name in Name:
     print(name)
    print("\n")
```

```
Ramon Sessions
John Wall
D.J. Augustin
Jameer Nelson
Tyus Jones
Zach LaVine
Cameron Payne
Russell Westbrook
Damian Lillard
Brian Roberts
Trey Burke
Shelvin Mack


Employees in SF Position:
Jae Crowder
Carmelo Anthony
Cleanthony Early
Lance Thomas
Robert Covington
Jerami Grant
DeMarre Carroll
Terrence Ross
Harrison Barnes
Andre Iguodala
Kevon Looney
James Michael McAdoo
Brandon Rush
Branden Dawson
Jeff Green
Wesley Johnson
Paul Pierce
Anthony Brown
Metta World Peace
```

3.Identify the predominant age group among employees.

```
data['Age Group'] = data['Age'].apply(lambda age:'20-25' if 20 <= age <= 25 else ('26-30' if 26 <= age <= 30 else ('31-35' if

data
```

```
              Name           Team  Number  Position  Age      Height  Weight        College     Salary  Age Group
  0  Avery Bradley  Boston Celtics       0        PG   25  168.684734     180          Texas  7730337.0      20-25
  1    Jae Crowder  Boston Celtics      99        SF   25  164.101445     235      Marquette  6796117.0      20-25
  3    R.J. Hunter  Boston Celtics      28        SG   22  178.327763     185  Georgia State  1148640.0      20-25
  6  Jordan Mickey  Boston Celtics      55        PF   21  169.926498     235            LSU  1170960.0      20-25
  7   Kelly Olynyk  Boston Celtics      41         C   25  157.662968     238        Gonzaga  2165160.0      20-25
...            ...            ...     ...       ...  ...         ...     ...            ...        ...        ...
451  Chris Johnson      Utah Jazz      23        SF   26  164.215846     206         Dayton   981348.0      26-30
452     Trey Lyles      Utah Jazz      41        PF   20  169.886309     234       Kentucky  2239800.0      20-25
453   Shelvin Mack      Utah Jazz       8        PG   26  178.683299     203         Butler  2433333.0      26-30
456    Jeff Withey      Utah Jazz      24         C   26  151.440309     231         Kansas   947276.0      26-30
457       Priyanka      Utah Jazz      34         C   25  171.033950     231         Kansas   947276.0      20-25

365 rows × 10 columns
```

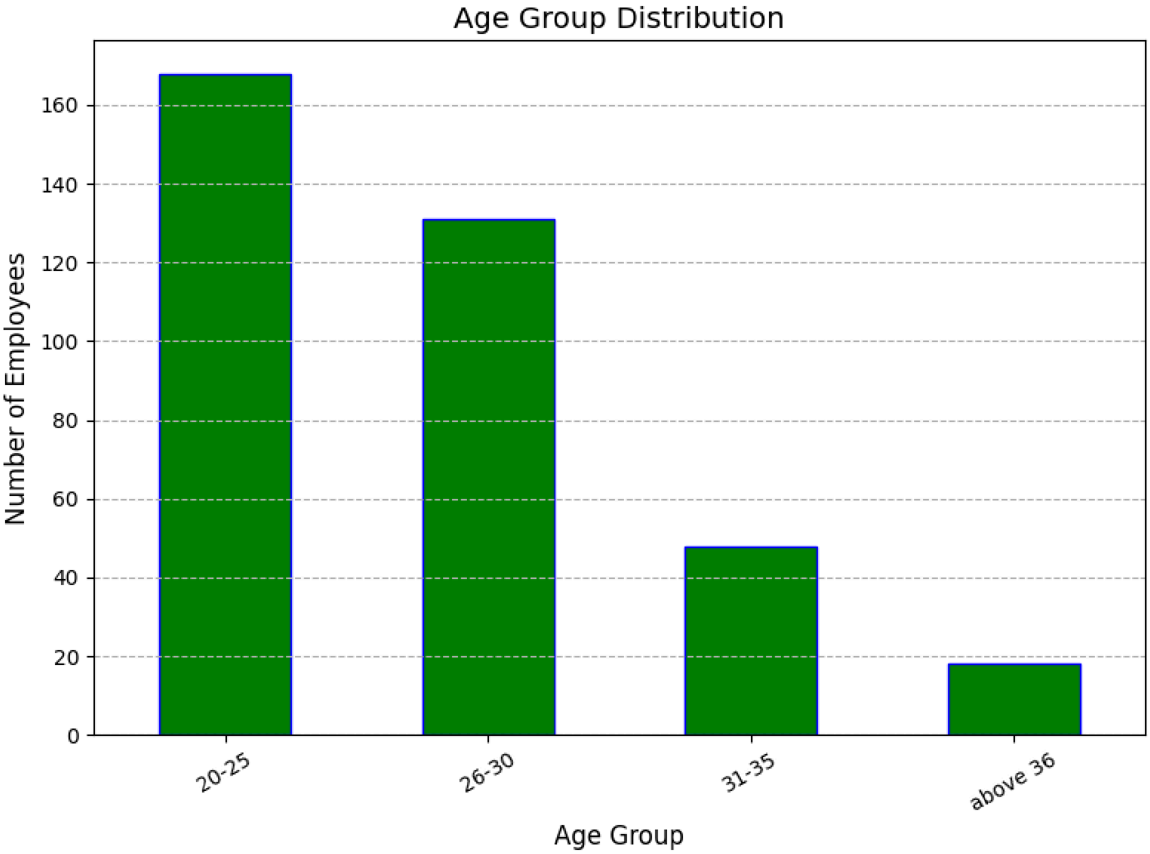
Next steps: Generate code with data View recommended plots New interactive sheet

```
age_counts = data['Age Group'].value_counts()
predominant_age_group = age_counts.idxmax()
predominant_count = age_counts.max()
```

```
print("Age group distribution:\n", age_counts)
print("\nPredominant age group is", predominant_age_group, "with", predominant_count, "employees")
```

```
Age group distribution:
 Age Group
20-25       168
26-30       131
31-35        48
above 36     18
Name: count, dtype: int64

Predominant age group is 20-25 with 168 employees
```

```
# Plotting age group distribution

plt.figure(figsize=(8, 6))
age_counts.plot(kind='bar', color='green', edgecolor='blue')

plt.title("Age Group Distribution", fontsize=14)
plt.xlabel("Age Group", fontsize=12)
plt.ylabel("Number of Employees", fontsize=12)
plt.xticks(rotation=30)
plt.grid(axis='y', linestyle='--')

# Show the plot
plt.tight_layout()
plt.show()
```



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

```
salary_spend = data.groupby(['Team','Position'])['Salary'].sum()
salary_spend.idxmax()
print("\nThe highest salary expenditure is for",salary_spend.idxmax())
```

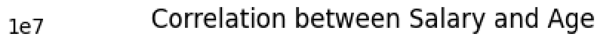
```
The highest salary expenditure is for ('Miami Heat', 'PF')
```

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

```
correlation = data['Salary'].corr(data['Age'])
print("THE CORRELATION Between Salary and Age is:",correlation)
```

```
THE CORRELATION Between Salary and Age is: 0.15999189342806172
```

```
# visual representation
sns.scatterplot(x="Age" ,y= "Salary",data= data)
plt.ylabel("Salary")
plt.xlabel("Age")
plt.title("Correlation between Salary and Age")
plt.show()
```


Correlation between Salary and Age