Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

Gemini

```
[18] import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns

     #loading the dataset
     df = pd.read_csv('/content/myexcel - myexcel.csv.csv')

     #displaying first 5 rows of data
     df.head()
```

	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

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

Preprocessing

Correcting the data in the "height" column by replacing it with random numbers between 150 and 180.

```
[2] #replacing 'Height' column with random values between 150 and 180.
    df['Height'] = np.random.randint(150, 181, size=len(df))

    df
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary

0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

Gemini

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

Preprocessing

Correcting the data in the "height" column by replacing it with random numbers between 150 and 180.

```
#replacing 'Height' column with random values between 150 and 180.
df['Height'] = np.random.randint(150, 181, size=len(df))

df
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	170	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	174	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	152	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	160	231	NaN	5000000.0
...	...	...	...	...	...	...	...	...	...
453	Shelvin Mack	Utah Jazz	8	PG	26	164	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	151	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	178	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	158	231	Kansas	947276.0
457	Priyanka	Utah Jazz	34	C	25	175	231	Kansas	947276.0

458 rows × 9 columns

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

```
[3] #checking null values
```

0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

Gemini

```
[3] #checking null values
    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    int64
 6   Weight    458 non-null    int64
 7   College   374 non-null    object
 8   Salary    447 non-null    float64
dtypes: float64(1), int64(4), object(4)
memory usage: 32.3+ KB
```

```
[4] #checking the count of null values
    df.isna().sum()
```

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

dtype: int64

0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

Gemini

```
dtype: int64
```

here seems that the 'College' column has 84 null values and column 'Salary' has 11 null values. Rather than dropping null values, i'm going to replace it with 'Unknown' in 'College' column and median of salary in the 'Salary' Column where null values exists.

```
[5] #handling missing values in the 'College' column using 'Unknown'.
    df['College'] = df['College'].fillna('Unknown')

    #handling missing values in the 'Salary' column using median of salary.
    df['Salary'] = df['Salary'].fillna(df['Salary'].median())
```

```
#checking the dataframe for verifying missing or inconsistent values.
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    int64
 6   Weight    458 non-null    int64
 7   College   458 non-null    object
 8   Salary    458 non-null    float64
dtypes: float64(1), int64(4), object(4)
memory usage: 32.3+ KB
```

Analysis Tasks

Task 1 : Distribution of employees across each team and calculate the percentage split relative to the total number of employees.

0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

Gemini

Analysis Tasks

Task 1 : Distribution of employees across each team and calculate the percentage split relative to the total number of employees.

```
[7] #number of employees per team and percentage
    team_distribution = df['Team'].value_counts()
    team_percentage = (team_distribution / len(df)) * 100

    #displaying the results
    print('Employee Distribution by Team : ')
    print(team_distribution)
    print('\nPercentage Split by Team : ')
    print(team_percentage)
```

```
Employee Distribution by Team :
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
```

0s completed at 21:06

```
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

Percentage Split by Team :
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
```

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

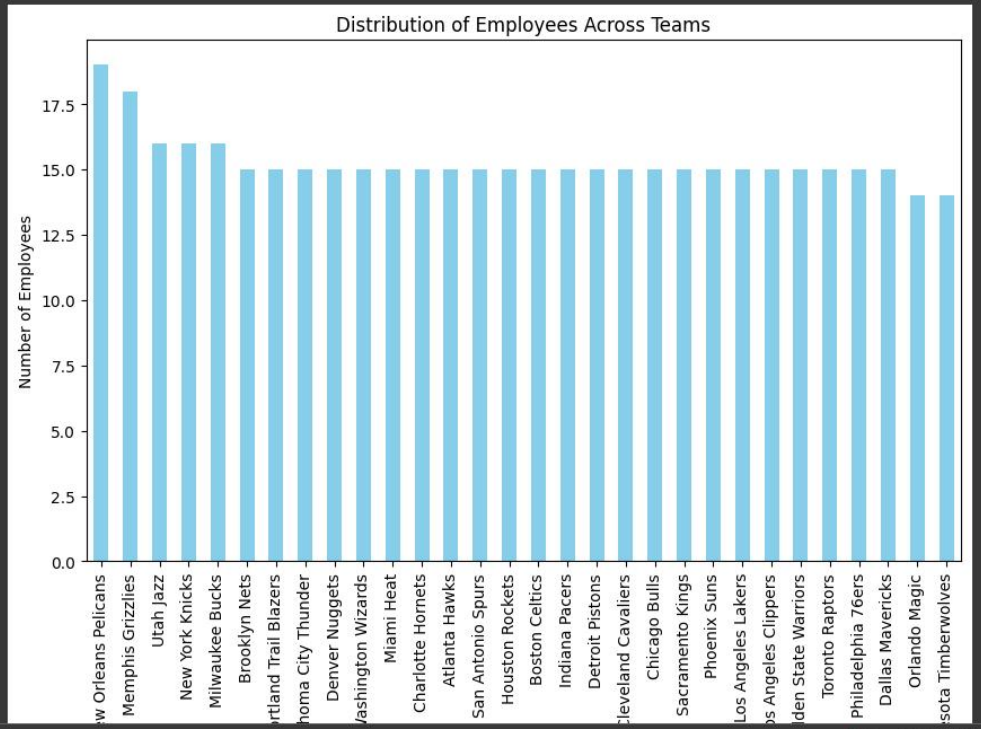
Share

+ Code + Text

RAM Disk

Gemini

```
#visualization of Distribution of Employees Across Teams.
team_distribution.plot(kind='bar', figsize=(10,6), color='skyblue')
plt.title('Distribution of Employees Across Teams')
plt.xlabel('Team')
plt.ylabel('Number of Employees')
plt.show()
```



0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

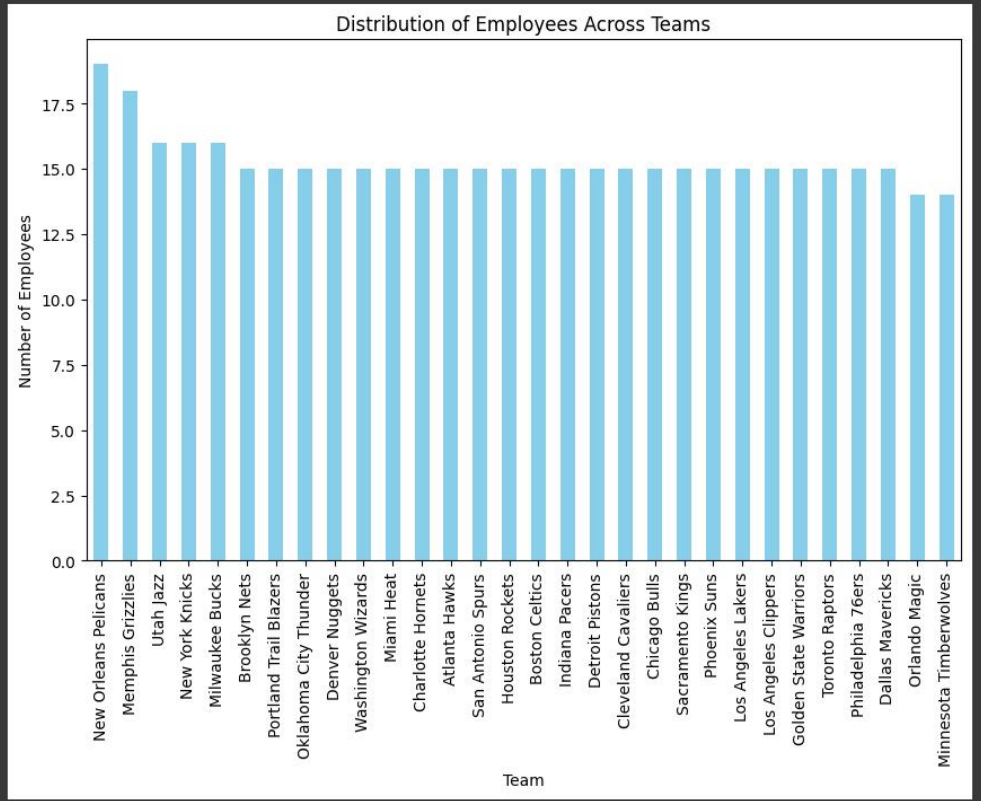
Share

+ Code + Text

RAM Disk

Gemini

```
plt.ylabel('Number of Employees')
plt.show()
```



0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

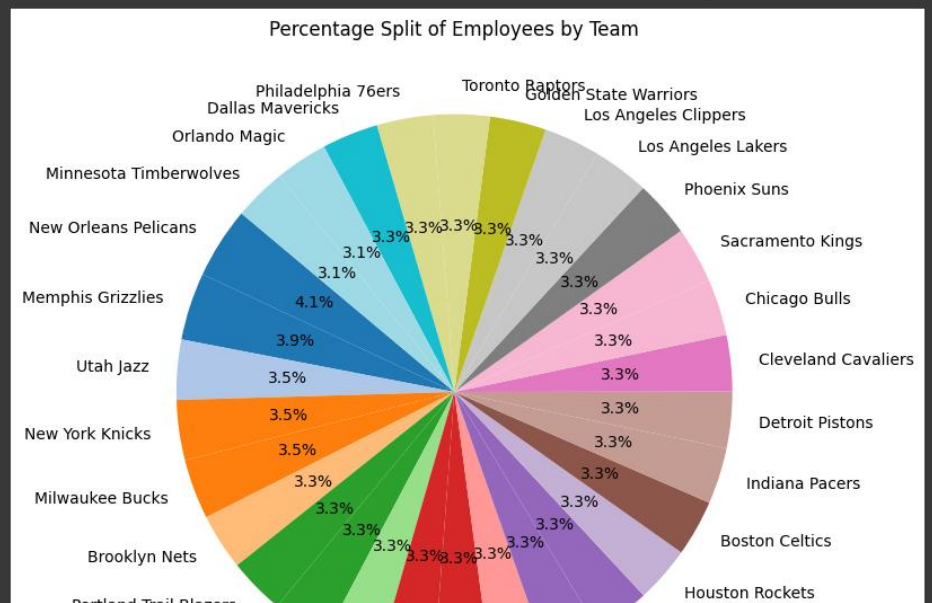
+ Code + Text

RAM Disk

Gemini

[8]


```
#visualization of percentage split.
plt.figure(figsize=(8,8))
team_percentage.plot(kind='pie', autopct='%1.1f%%', startangle=140, colormap='tab20')
plt.title('Percentage Split of Employees by Team')
plt.ylabel('')
plt.show()
```



0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

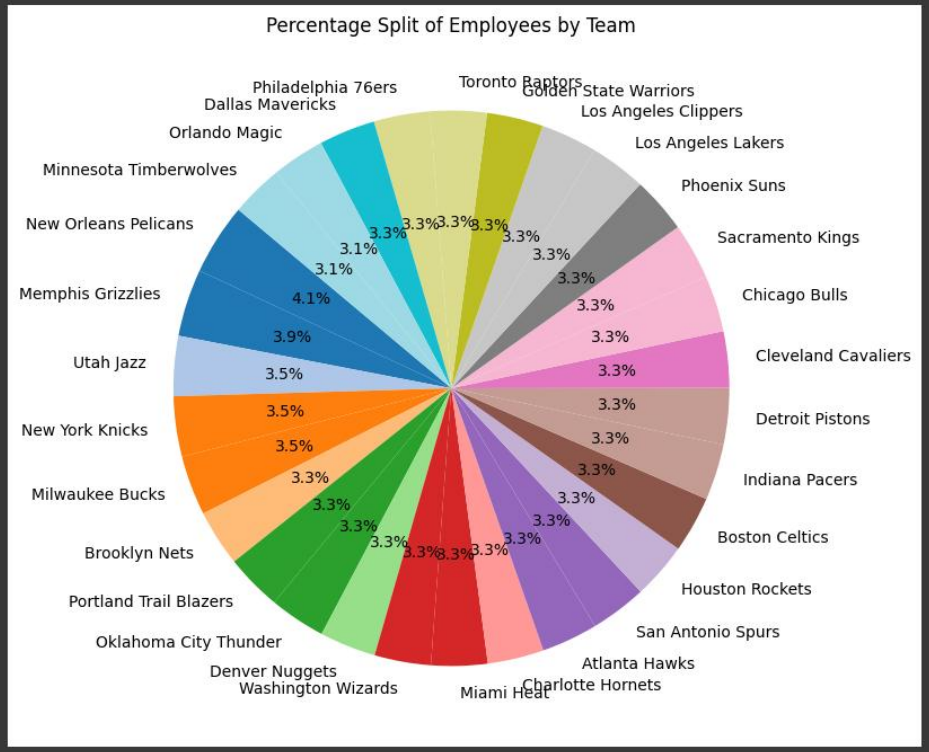
Share

+ Code + Text

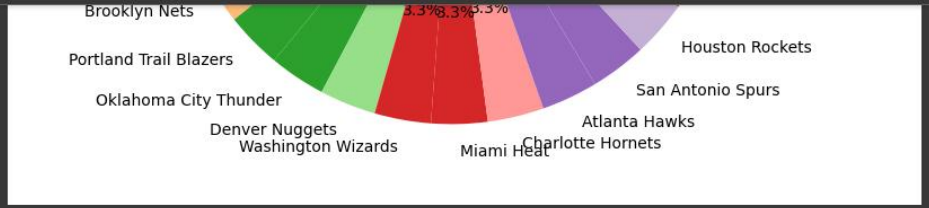
RAM Disk

Gemini

```
team_percentage.plot(kind='pie', autopct='%1.1f%%', startangle=140, colormap='tab20')
plt.title('Percentage Split of Employees by Team')
plt.ylabel('')
plt.show()
```



0s completed at 21:06

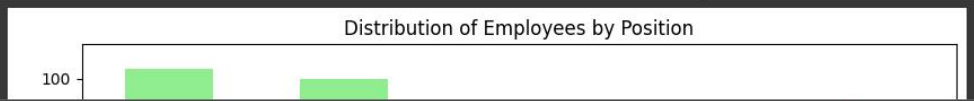


Task 2 : Segregate employees based on their positions with company.

```
#number of employees per position
position_distribution = df['Position'].value_counts()

#displaying the results
print("Employee Distribution by Position")
print(position_distribution)
```

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

```
[11] #visualization
position_distribution.plot(kind='bar', figsize=(10, 6), color='lightgreen')
plt.title('Distribution of Employees by Position')
plt.xlabel('Position')
plt.ylabel('Number of Employees')
plt.show()
```



Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

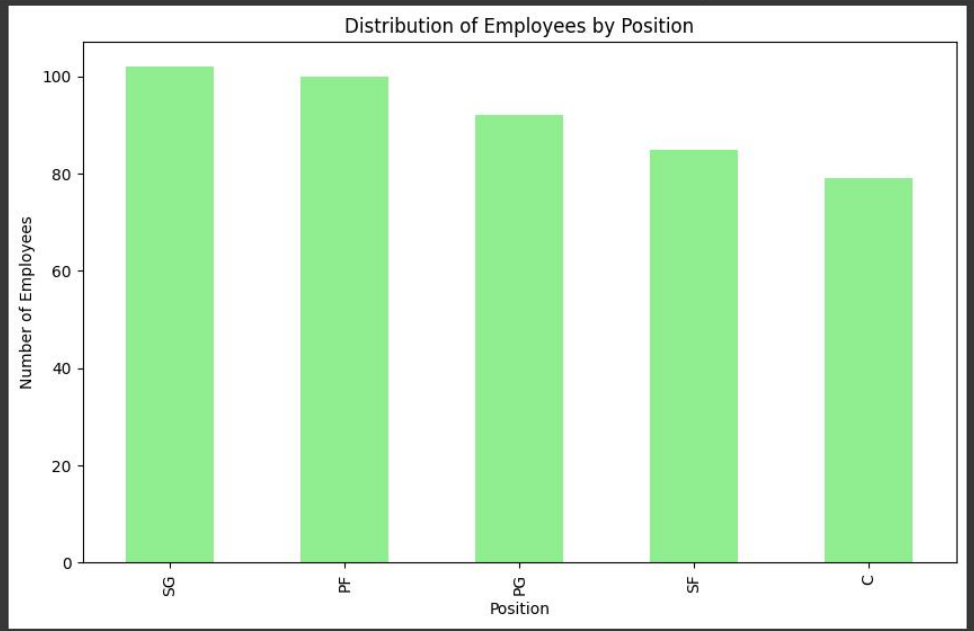
Share

+ Code + Text

RAM Disk

Gemini

```
[11] #visualization
     position_distribution.plot(kind='bar', figsize=(10, 6), color='lightgreen')
     plt.title('Distribution of Employees by Position')
     plt.xlabel('Position')
     plt.ylabel('Number of Employees')
     plt.show()
```



Task 3 : Predominant age group among employees.

0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

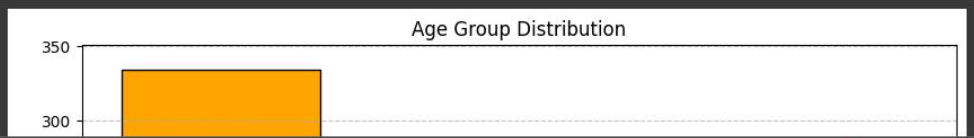
Gemini

Task 3 : Predominant age group among employees.

```
[12] #defining age groups
     bins = [20, 30, 40, 50, 60]
     labels = ['20-29', '30-39', '40-49', '50-59']
     df['Age_Group'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)

     #calculating the age group distribution
     age_group_distribution = df['Age_Group'].value_counts()

     #dsiplaying the results
     print("Age Group Distribution")
     print(age_group_distribution)
```

```
Age Group Distribution
Age_Group
20-29    334
30-39    119
40-49      3
50-59      0
Name: count, dtype: int64
```

```
#visualization

plt.figure(figsize=(10, 6))
plt.hist(df['Age'], bins=bins, color='orange', edgecolor='black')
plt.title("Age Group Distribution")
plt.xlabel("Age")
plt.ylabel("Number of Employees")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

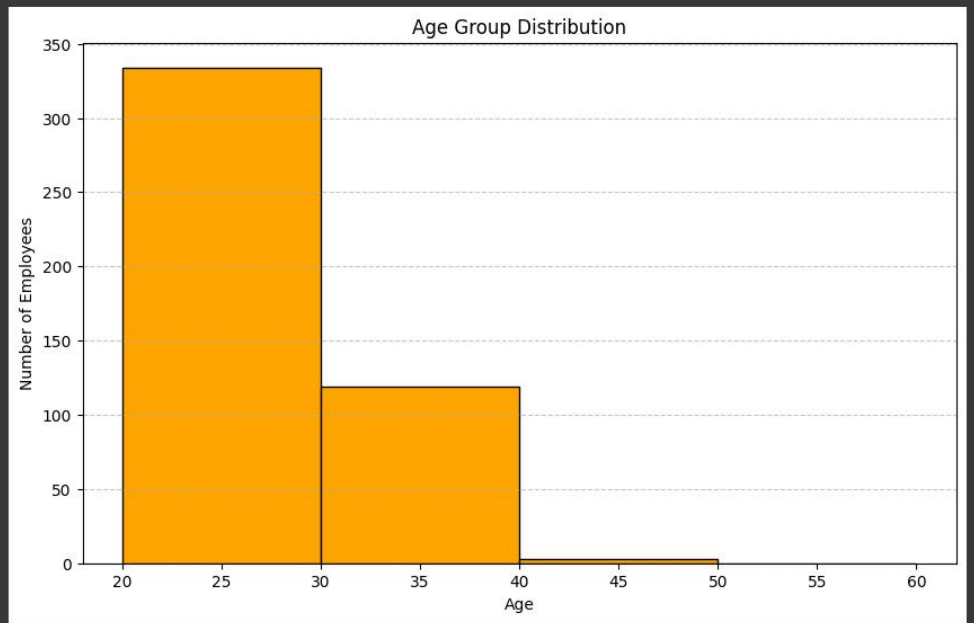
Share

+ Code + Text

RAM Disk

Gemini

[27]
```
plt.figure(figsize=(10, 6))
plt.hist(df['Age'], bins=bins, color='orange', edgecolor='black')
plt.title("Age Group Distribution")
plt.xlabel("Age")
plt.ylabel("Number of Employees")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



Task 4 : Team and Position having highest salary expenditure.

0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

Gemini

Task 4 : Team and Position having highest salary expenditure.

```
#salary expenditure by team
team_salary = df.groupby('Team')['Salary'].sum().sort_values(ascending=False)

#salary expenditure by position
position_salary = df.groupby('Position')['Salary'].sum().sort_values(ascending=False)

#displaying the results
print("Total Salary by Team : ")
print(team_salary)
print("\nTotal Salary by Position : ")
print(position_salary)
```

```
Total Salary by Team :
Team
Cleveland Cavaliers       109824875.0
Los Angeles Clippers       94854640.0
Oklahoma City Thunder      93765298.0
Golden State Warriors      88868997.0
Miami Heat                 88188045.0
Memphis Grizzlies          87895624.0
Chicago Bulls              86783378.0
San Antonio Spurs          84442733.0
New Orleans Pelicans       82750774.0
Charlotte Hornets          78340920.0
Washington Wizards         76328636.0
Houston Rockets            75283021.0
New York Knicks            73303898.0
Atlanta Hawks              72902950.0
Los Angeles Lakers         71770431.0
Sacramento Kings           71683666.0
Dallas Mavericks           71198732.0
Toronto Raptors            71117611.0
Milwaukee Bucks            69603517.0
Detroit Pistons            67168263.0
Indiana Pacers             66751826.0
Utah Jazz                  64007367.0
Phoenix Suns               63445135.0
Denver Nuggets             62958116.0
Minnesota Timberwolves     62545883.0
Boston Celtics             61377254.0
```

0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

Gemini

[14]
```
Golden State Warriors     88868997.0
Miami Heat                88188045.0
Memphis Grizzlies         87895624.0
Chicago Bulls             86783378.0
San Antonio Spurs         84442733.0
New Orleans Pelicans      82750774.0
Charlotte Hornets         78340920.0
Washington Wizards        76328636.0
Houston Rockets           75283021.0
New York Knicks           73303898.0
Atlanta Hawks             72902950.0
Los Angeles Lakers        71770431.0
Sacramento Kings          71683666.0
Dallas Mavericks          71198732.0
Toronto Raptors           71117611.0
Milwaukee Bucks           69603517.0
Detroit Pistons           67168263.0
Indiana Pacers            66751826.0
Utah Jazz                 64007367.0
Phoenix Suns              63445135.0
Denver Nuggets            62958116.0
Minnesota Timberwolves    62545883.0
Boston Celtics            61377254.0
Orlando Magic             60161470.0
Brooklyn Nets             52528475.0
Portland Trail Blazers    48301818.0
Philadelphia 76ers        33829080.0
Name: Salary, dtype: float64

Total Salary by Position :
Position
C     466377332.0
PG    458193715.0
PF    451069408.0
SF    410857162.0
SG    405484816.0
Name: Salary, dtype: float64
```

[24]
```
# Group by Team and Position to get the sum of salaries
salary_data = df.groupby(['Team', 'Position'])['Salary'].sum().unstack()

# Stacked Bar Chart
salary_data.plot(kind='bar', stacked=True, figsize=(12, 8), colormap='tab20')
plt.title('Salary Expenditure by Team and Position')
```

0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

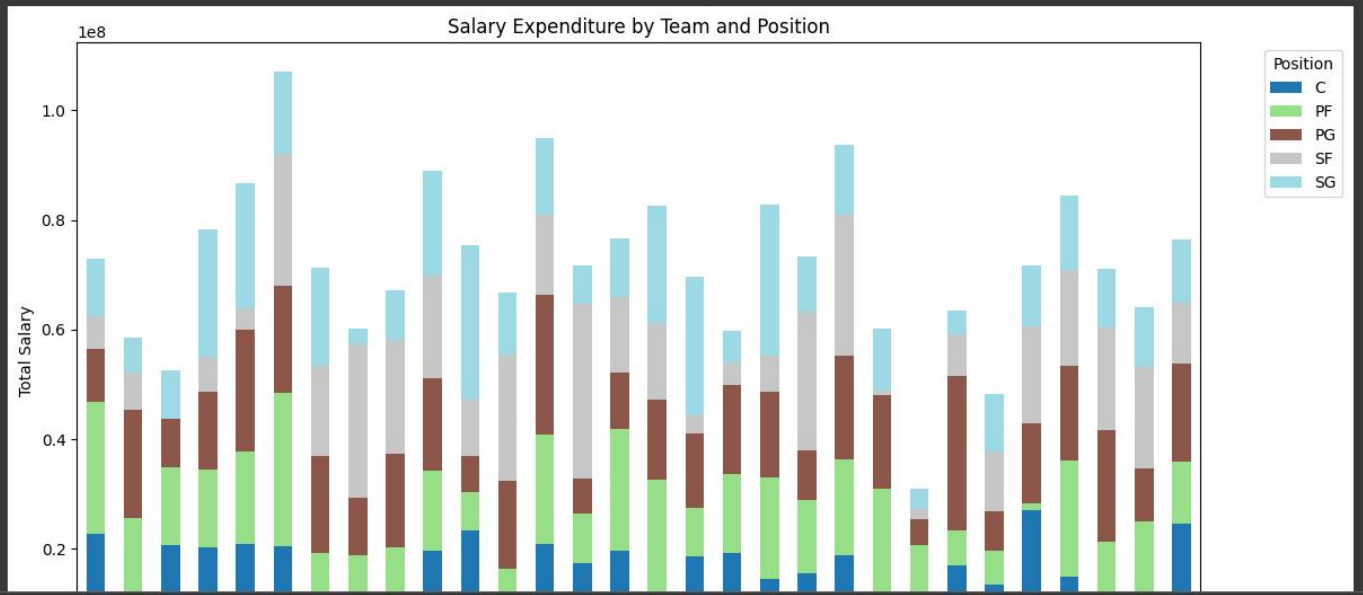
Share

+ Code + Text

RAM Disk

Gemini

```python
# Group by Team and Position to get the sum of salaries
salary_data = df.groupby(['Team', 'Position'])['Salary'].sum().unstack()

# Stacked Bar Chart
salary_data.plot(kind='bar', stacked=True, figsize=(12, 8), colormap='tab20')
plt.title('Salary Expenditure by Team and Position')
plt.xlabel('Team')
plt.ylabel('Total Salary')
plt.legend(title='Position', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

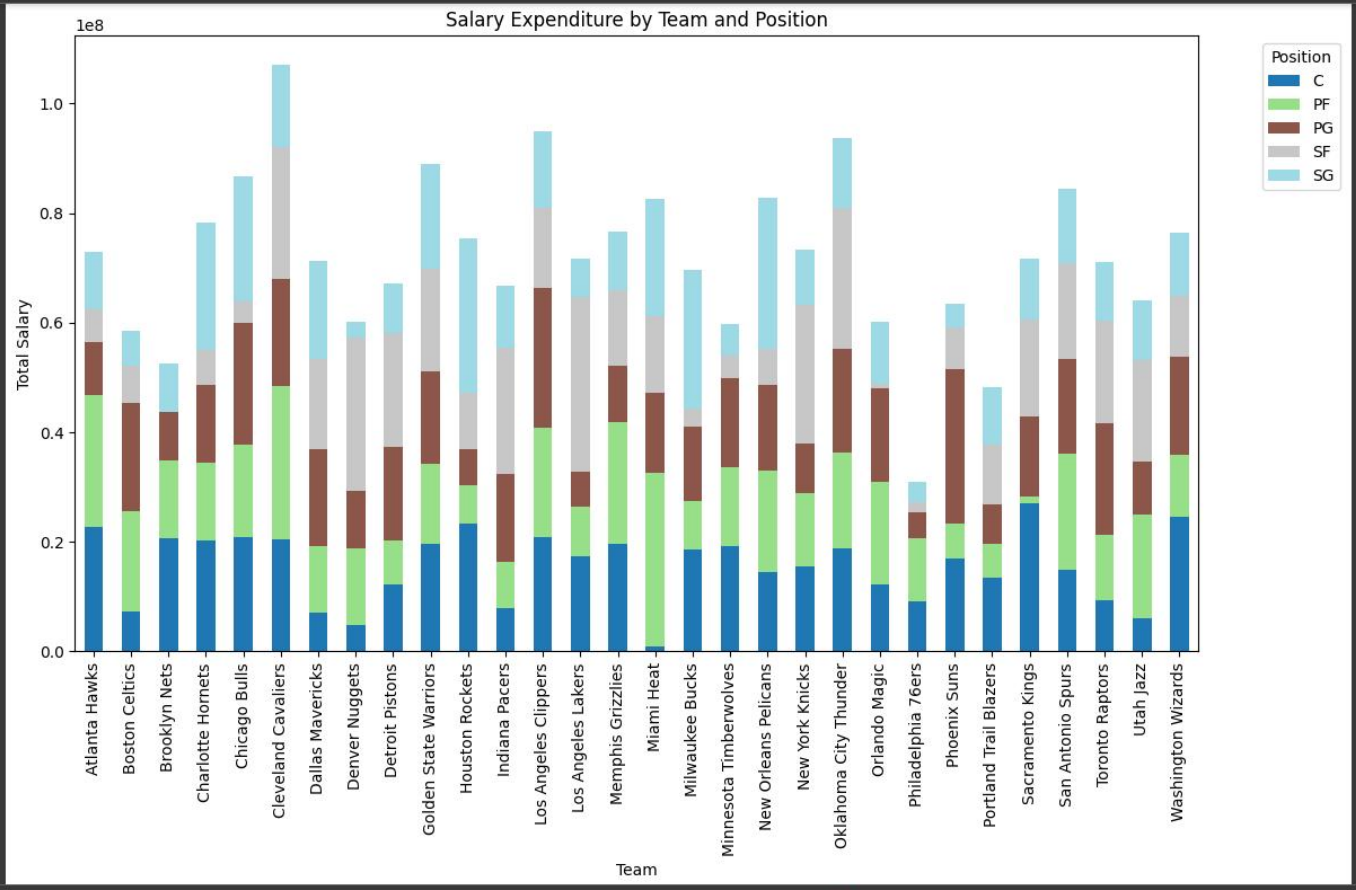
Share

+ Code + Text

RAM Disk

Gemini

[24]



0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

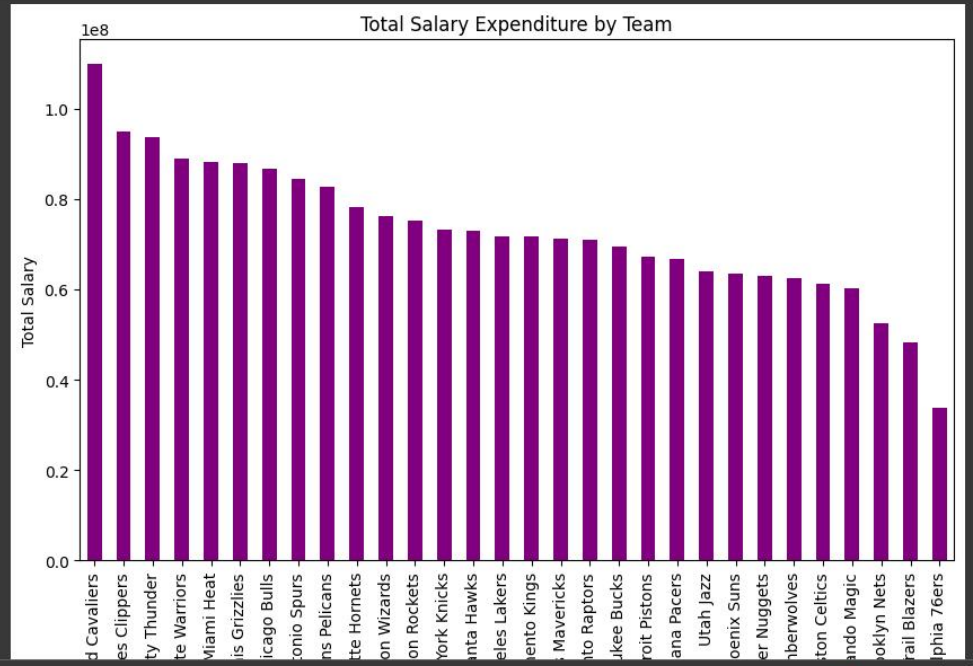
RAM Disk

Gemini

[24]

Team

```
team_salary.plot(kind='bar', figsize=(10, 6), color='purple')
plt.title('Total Salary Expenditure by Team')
plt.xlabel('Team')
plt.ylabel('Total Salary')
plt.show()
```



0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

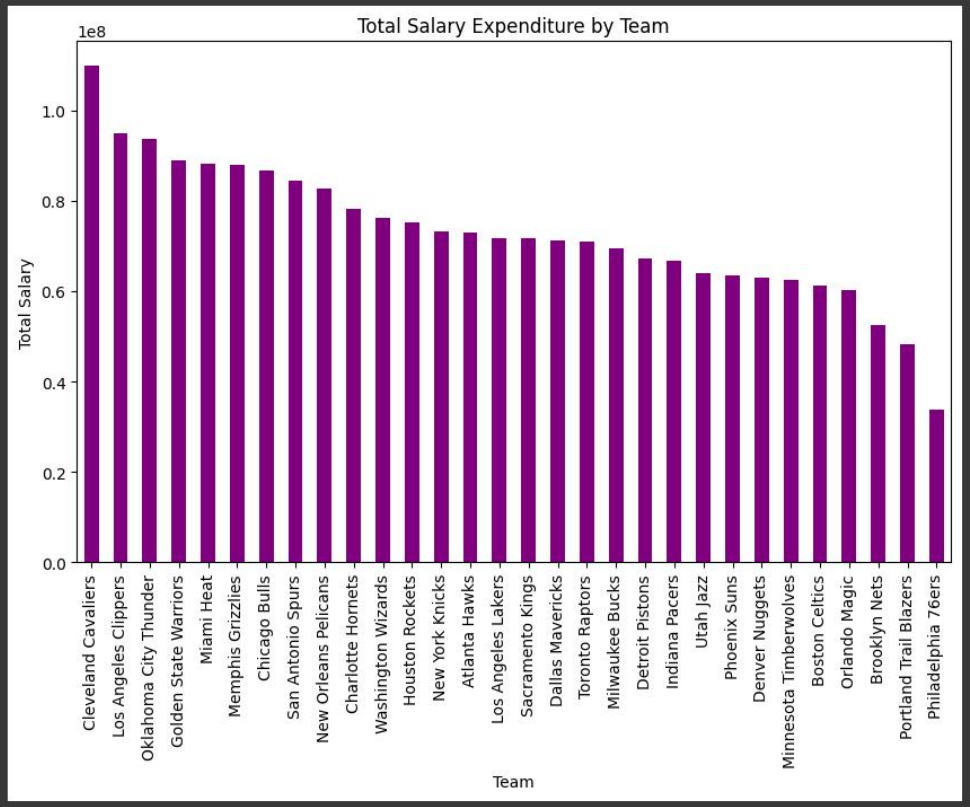
Share

+ Code + Text

RAM Disk

Gemini

```
plt.show()
```



0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

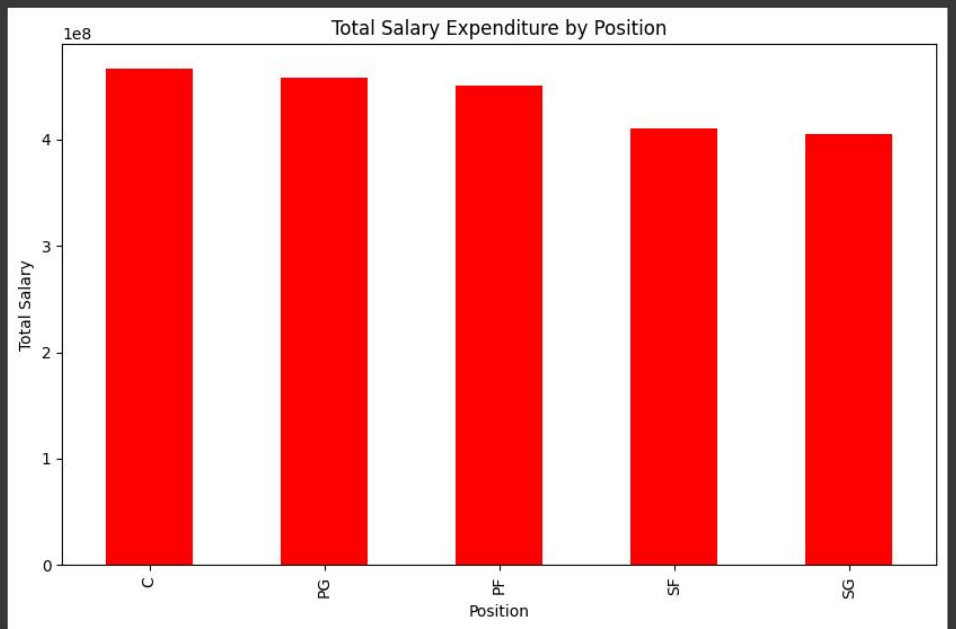
+ Code + Text

RAM Disk

Gemini

Team

```
position_salary.plot(kind='bar', figsize=(10, 6), color='red')
plt.title('Total Salary Expenditure by Position')
plt.xlabel("Position")
plt.ylabel("Total Salary")
plt.show()
```



Task 5 : Correlation between Age and Salary

0s completed at 21:06

Python_End_Project.ipynb

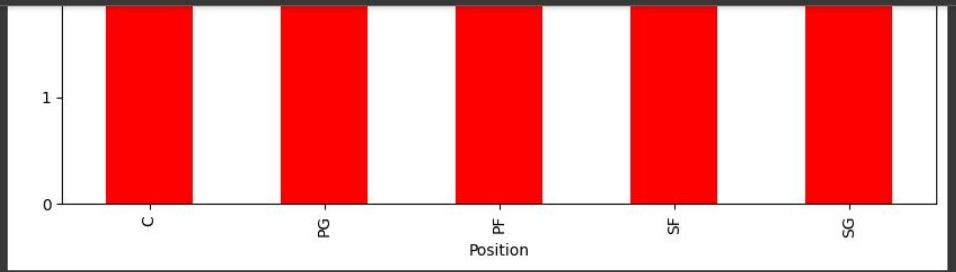
File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

Gemini



Task 5 : Correlation between Age and Salary.

```
[17] #correlation
     correlation = df['Age'].corr(df['Salary'])
     print(f"Correlation between Age and Salary : {correlation:.2f}")
```

```
Correlation between Age and Salary : 0.21
```

```
[19] #visualization with trendline
     plt.figure(figsize=(10, 6))
     sns.scatterplot(data=df, x='Age', y='Salary', color='teal')
     sns.regplot(data=df, x='Age', y='Salary', scatter=False, color='red')
     plt.title("Correlation between Age and Salary")
     plt.xlabel("Age")
     plt.ylabel("Salary")
     plt.show()
```

0s completed at 21:06

Python_End_Project.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Share

+ Code + Text

RAM Disk

Gemini

```
#visualization with trendline
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Age', y='Salary', color='teal')
sns.regplot(data=df, x='Age', y='Salary', scatter=False, color='red')
plt.title("Correlation between Age and Salary")
plt.xlabel("Age")
plt.ylabel("Salary")
plt.show()
```

0s completed at 21:06

Data Story

1. The New Orleans Pelicans is having the largest number of employees, totaling 19, making them the most staffed team. On the other hand, the Orlando Magic and Minnesota Timberwolves have the smallest teams, each employing only 14 members.

2. The majority of employees hold the positions of 'SG' and 'PF', with 102 and 100 employees respectively. The position with the least number of employees is 'C', which has 79 members.

3. The predominant age group among employees is 20-29 years, encompassing 334 individuals, followed by 30-39 years with 119 individuals. The age group of 40-49 years is the least represented, with only 3 employees.

4. The Cleveland Cavaliers lead in total salary expenditure, with a staggering 109,824,875.0, highlighting their investment in talent. In contrast, the Philadelphia 76ers have the lowest salary expenditure at 33,829,080.0, suggesting a different strategic approach.

5. When examining positions, the 'C' role incurs the highest total salary at 466,377,332.0, reflecting the importance and value placed on this position. Conversely, the 'SG' position has the lowest total salary expenditure at 405,484,816.0.

6. There is a positive correlation (0.21) between age and salary. This indicates that older employees generally earn more, though the correlation is moderate, suggesting that while age influences salary, other factors also play a significant role.