

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

```
In [2]: df_employee=pd.read_csv(r"D:\Data Analysis\Ahmed Samier\PROJECT\Final project\Final
df_PerformanceRating=pd.read_csv(r"D:\Data Analysis\Ahmed Samier\PROJECT\Final proj
```

```
In [3]: #Display Table
df_employee
```

```
Out[3]:
```

|  | EmployeeID | FirstName | LastName | Gender | Age | BusinessTravel | Department |
|--|------------|-----------|----------|--------|-----|----------------|------------|
|--|------------|-----------|----------|--------|-----|----------------|------------|

|      |           |            |             |            |     |             |                 |
|------|-----------|------------|-------------|------------|-----|-------------|-----------------|
| 0    | 3012-1A41 | Leonelle   | Simco       | Female     | 30  | Some Travel | Sales           |
| 1    | CBCB-9C9D | Leonerd    | Aland       | Male       | 38  | Some Travel | Sales           |
| 2    | 95D7-1CE9 | Ahmed      | Sykes       | Male       | 43  | Some Travel | Human Resources |
| 3    | 47A0-559B | Ermentrude | Berrie      | Non-Binary | 39  | Some Travel | Technology      |
| 4    | 42CC-040A | Stace      | Savege      | Female     | 29  | Some Travel | Human Resources |
| ...  | ...       | ...        | ...         | ...        | ... | ...         | ...             |
| 1465 | 467E-977A | Jud        | Melanaphy   | Male       | 20  | Some Travel | Technology      |
| 1466 | 6FB9-A624 | Marc       | Calver      | Non-Binary | 27  | Some Travel | Technology      |
| 1467 | EBF4-5928 | Rudolph    | MacDearmont | Male       | 21  | Some Travel | Sales           |
| 1468 | 60E6-B1D9 | Merill     | Agg         | Male       | 21  | Some Travel | Technology      |
| 1469 | 84D4-D4C3 | Naoma      | Hebbard     | Female     | 20  | No Travel   | Technology      |

1470 rows × 24 columns

```
In [4]: # Display the first 5 rows
df_employee.head()
```

Out[4]:

|   | EmployeeID | FirstName  | LastName | Gender     | Age | BusinessTravel | Department      | Distance |
|---|------------|------------|----------|------------|-----|----------------|-----------------|----------|
| 0 | 3012-1A41  | Leonelle   | Simco    | Female     | 30  | Some Travel    | Sales           |          |
| 1 | CBCB-9C9D  | Leonerd    | Aland    | Male       | 38  | Some Travel    | Sales           |          |
| 2 | 95D7-1CE9  | Ahmed      | Sykes    | Male       | 43  | Some Travel    | Human Resources |          |
| 3 | 47A0-559B  | Ermentrude | Berrie   | Non-Binary | 39  | Some Travel    | Technology      |          |
| 4 | 42CC-040A  | Stace      | Savege   | Female     | 29  | Some Travel    | Human Resources |          |

5 rows × 24 columns

```
In [18]: # Display the Last 5 rows
df_employee.tail()
```

Out[18]:

|      | EmployeeID | FirstName | LastName    | Gender     | Age | BusinessTravel | Department | D |
|------|------------|-----------|-------------|------------|-----|----------------|------------|---|
| 1465 | 467E-977A  | Jud       | Melanaphy   | Male       | 20  | Some Travel    | Technology |   |
| 1466 | 6FB9-A624  | Marc      | Calver      | Non-Binary | 27  | Some Travel    | Technology |   |
| 1467 | EBF4-5928  | Rudolph   | MacDearmont | Male       | 21  | Some Travel    | Sales      |   |
| 1468 | 60E6-B1D9  | Merill    | Agg         | Male       | 21  | Some Travel    | Technology |   |
| 1469 | 84D4-D4C3  | Naoma     | Hebbard     | Female     | 20  | No Travel      | Technology |   |

5 rows × 24 columns

```
In [3]: ##Check the shape
df_employee.shape
```

Out[3]: (1470, 24)

```
In [13]: #Check for missing values
print("\nMissing values in Employee table:")
```

Missing values in Employee table:

```
In [10]: #check duplication
df_employee.duplicated().sum()
```

Out[10]: np.int64(0)

```
In [6]: #check isnull in table
print(df_employee.isnull().sum())
```

|                         |   |
|-------------------------|---|
| EmployeeID              | 0 |
| FirstName               | 0 |
| LastName                | 0 |
| Gender                  | 0 |
| Age                     | 0 |
| BusinessTravel          | 0 |
| Department              | 0 |
| DistanceFromHome (KM)   | 0 |
| State                   | 0 |
| Ethnicity               | 0 |
| Education               | 0 |
| EducationLevel          | 0 |
| EducationField          | 0 |
| JobRole                 | 0 |
| MaritalStatus           | 0 |
| Salary                  | 0 |
| StockOptionLevel        | 0 |
| Overtime                | 0 |
| HireDate                | 0 |
| Attrition               | 0 |
| YearsAtCompany          | 0 |
| YearsInMostRecentRole   | 0 |
| YearsSinceLastPromotion | 0 |
| YearsWithCurrManager    | 0 |

dtype: int64

```
In [11]: df_employee.isnull().sum()
```

```
Out[11]: EmployeeID      0
         FirstName      0
         LastName      0
         Gender      0
         Age      0
         BusinessTravel  0
         Department      0
         DistanceFromHome (KM)  0
         State      0
         Ethnicity      0
         Education      0
         EducationLevel  0
         EducationField  0
         JobRole      0
         MaritalStatus  0
         Salary      0
         StockOptionLevel  0
         OverTime      0
         HireDate      0
         Attrition      0
         YearsAtCompany  0
         YearsInMostRecentRole  0
         YearsSinceLastPromotion  0
         YearsWithCurrManager  0
         dtype: int64
```

```
In [7]: # show describe
        print(df_employee.describe())
```

|       | Age         | DistanceFromHome (KM) | Education   | Salary \      |
|-------|-------------|-----------------------|-------------|---------------|
| count | 1470.000000 | 1470.000000           | 1470.000000 | 1470.000000   |
| mean  | 28.989796   | 22.502721             | 2.912925    | 112956.497959 |
| std   | 7.993055    | 12.811124             | 1.024165    | 103342.889222 |
| min   | 18.000000   | 1.000000              | 1.000000    | 20387.000000  |
| 25%   | 23.000000   | 12.000000             | 2.000000    | 43580.500000  |
| 50%   | 26.000000   | 22.000000             | 3.000000    | 71199.500000  |
| 75%   | 34.000000   | 33.000000             | 4.000000    | 142055.750000 |
| max   | 51.000000   | 45.000000             | 5.000000    | 547204.000000 |

|       | StockOptionLevel | YearsAtCompany | YearsInMostRecentRole \ |
|-------|------------------|----------------|-------------------------|
| count | 1470.000000      | 1470.000000    | 1470.000000             |
| mean  | 0.793878         | 4.562585       | 2.293197                |
| std   | 0.852077         | 3.288048       | 2.539093                |
| min   | 0.000000         | 0.000000       | 0.000000                |
| 25%   | 0.000000         | 2.000000       | 0.000000                |
| 50%   | 1.000000         | 4.000000       | 1.000000                |
| 75%   | 1.000000         | 7.000000       | 4.000000                |
| max   | 3.000000         | 10.000000      | 10.000000               |

|       | YearsSinceLastPromotion | YearsWithCurrManager |
|-------|-------------------------|----------------------|
| count | 1470.000000             | 1470.000000          |
| mean  | 3.440816                | 2.239456             |
| std   | 2.945194                | 2.505774             |
| min   | 0.000000                | 0.000000             |
| 25%   | 1.000000                | 0.000000             |
| 50%   | 3.000000                | 1.000000             |
| 75%   | 6.000000                | 4.000000             |
| max   | 10.000000               | 10.000000            |

```
In [8]: # Check data types of each column
print(df_employee.dtypes)
```

```

EmployeeID      object
FirstName       object
LastName        object
Gender          object
Age            int64
BusinessTravel  object
Department      object
DistanceFromHome (KM)  int64
State           object
Ethnicity       object
Education       int64
EducationLevel  object
EducationField  object
JobRole         object
MaritalStatus   object
Salary          int64
StockOptionLevel int64
OverTime        object
HireDate        object
Attrition       object
YearsAtCompany  int64
YearsInMostRecentRole int64
YearsSinceLastPromotion int64
YearsWithCurrManager int64
dtype: object

```

```
In [11]: df_employee['HireDate'] = pd.to_datetime(df_employee['HireDate'], errors='coerce')
```

```
In [18]: # Employee per Department
print("\nDistribution of Employees Across Departments:")
department_counts = df_employee['Department'].value_counts()
print(department_counts)
```

Distribution of Employees Across Departments:

```

Department
Technology      961
Sales            446
Human Resources   63
Name: count, dtype: int64

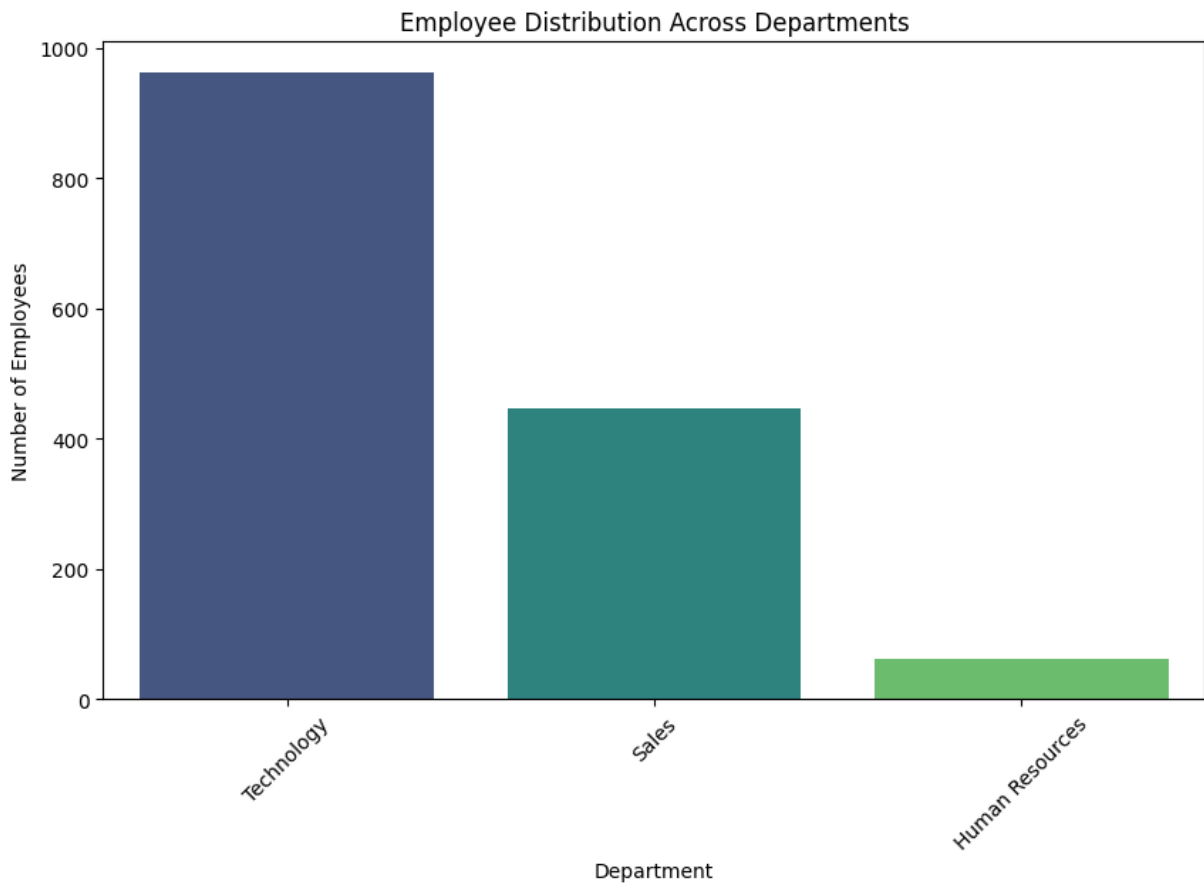
```

```
In [10]: plt.figure(figsize=(10, 6))
sns.barplot(x=department_counts.index, y=department_counts.values, palette='viridis')
plt.xlabel("Department")
plt.ylabel("Number of Employees")
plt.title("Employee Distribution Across Departments")
plt.xticks(rotation=45)
plt.show()
```

C:\Users\adm.e\AppData\Local\Temp\ipykernel\_36628\2854900584.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

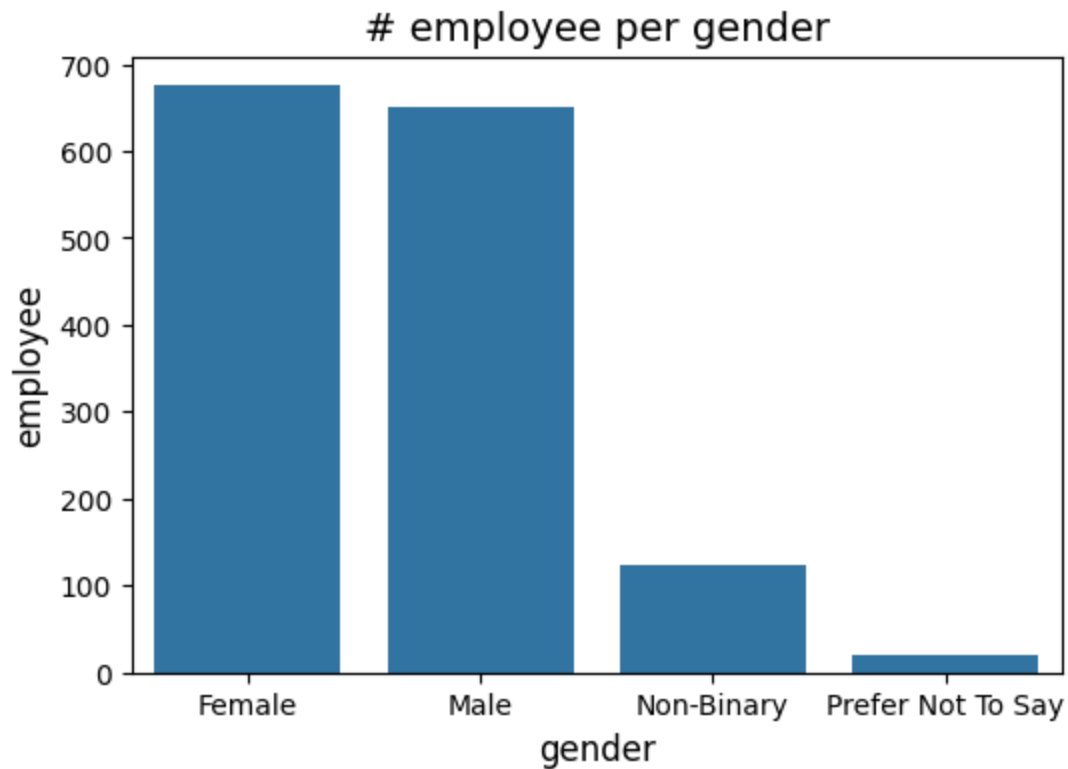
```
sns.barplot(x=department_counts.index, y=department_counts.values, palette='viridis')
```



```
In [19]: #Employee per Gender
gender_counts = df_employee["Gender"].value_counts()
print(gender_counts)
```

```
Gender
Female      675
Male        651
Non-Binary   124
Prefer Not To Say  20
Name: count, dtype: int64
```

```
In [13]: plt.figure(figsize=(6, 4))
sns.countplot(data=df_employee, x='Gender')
plt.title('# employee per gender', fontsize=14)
plt.xlabel('gender', fontsize=12)
plt.ylabel('employee', fontsize=12)
plt.show()
```

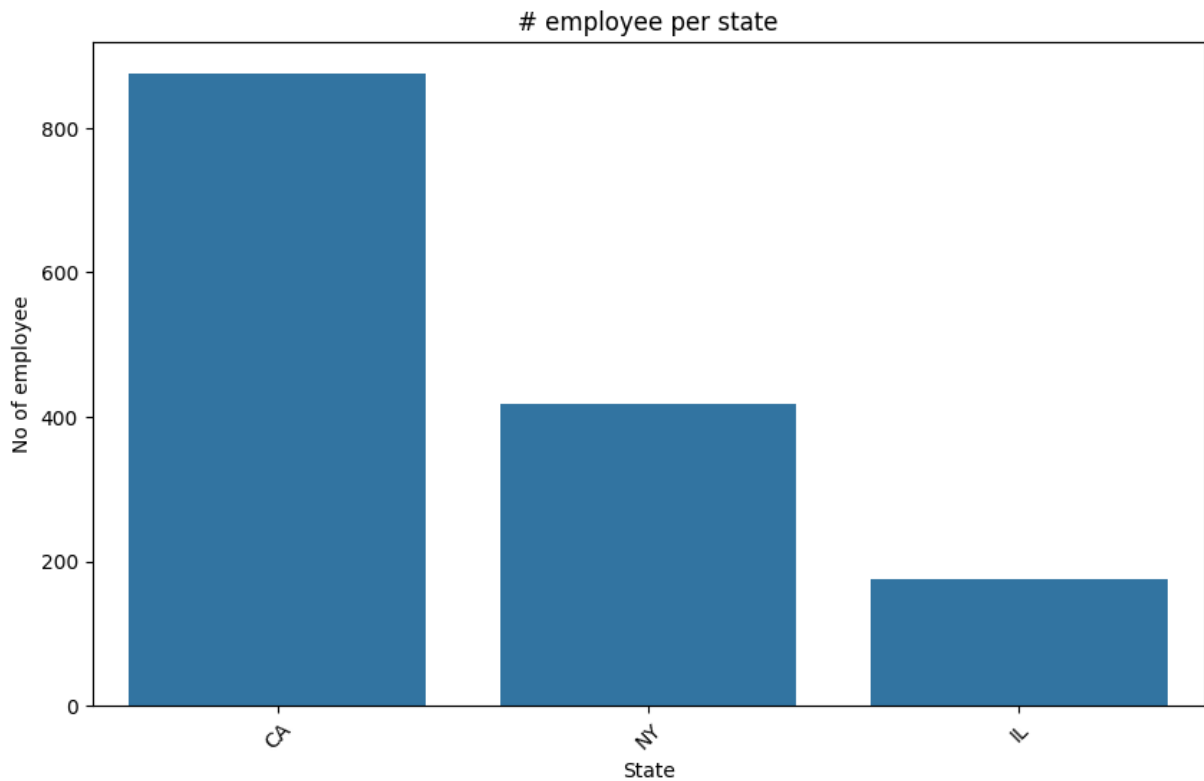


```
In [22]: #Employee per State
employee_count_by_state = df_employee['State'].value_counts()
print(employee_count_by_state)
```

```
State
CA    875
NY    419
IL    176
Name: count, dtype: int64
```

```
In [21]: plt.figure(figsize=(10, 6))
sns.countplot(data=df_employee, x='State', order=df_employee['State'].value_counts())
plt.title('# employee per state')
plt.xlabel('State')
plt.ylabel('No of employee')
plt.xticks(rotation=45)
plt.show()
```





```
In [23]: #Employee per job role
# عدد الموظفين في كل دور وظيفي
print("\nNumber of Employees in Each Job Role:")
job_role_counts = df_employee['JobRole'].value_counts()
print(job_role_counts)
```

Number of Employees in Each Job Role:

| JobRole                   | Count |
|---------------------------|-------|
| Sales Executive           | 327   |
| Software Engineer         | 294   |
| Data Scientist            | 261   |
| Machine Learning Engineer | 146   |
| Senior Software Engineer  | 132   |
| Sales Representative      | 83    |
| Engineering Manager       | 75    |
| Analytics Manager         | 52    |
| Manager                   | 37    |
| HR Executive              | 28    |
| Recruiter                 | 24    |
| HR Business Partner       | 7     |
| HR Manager                | 4     |

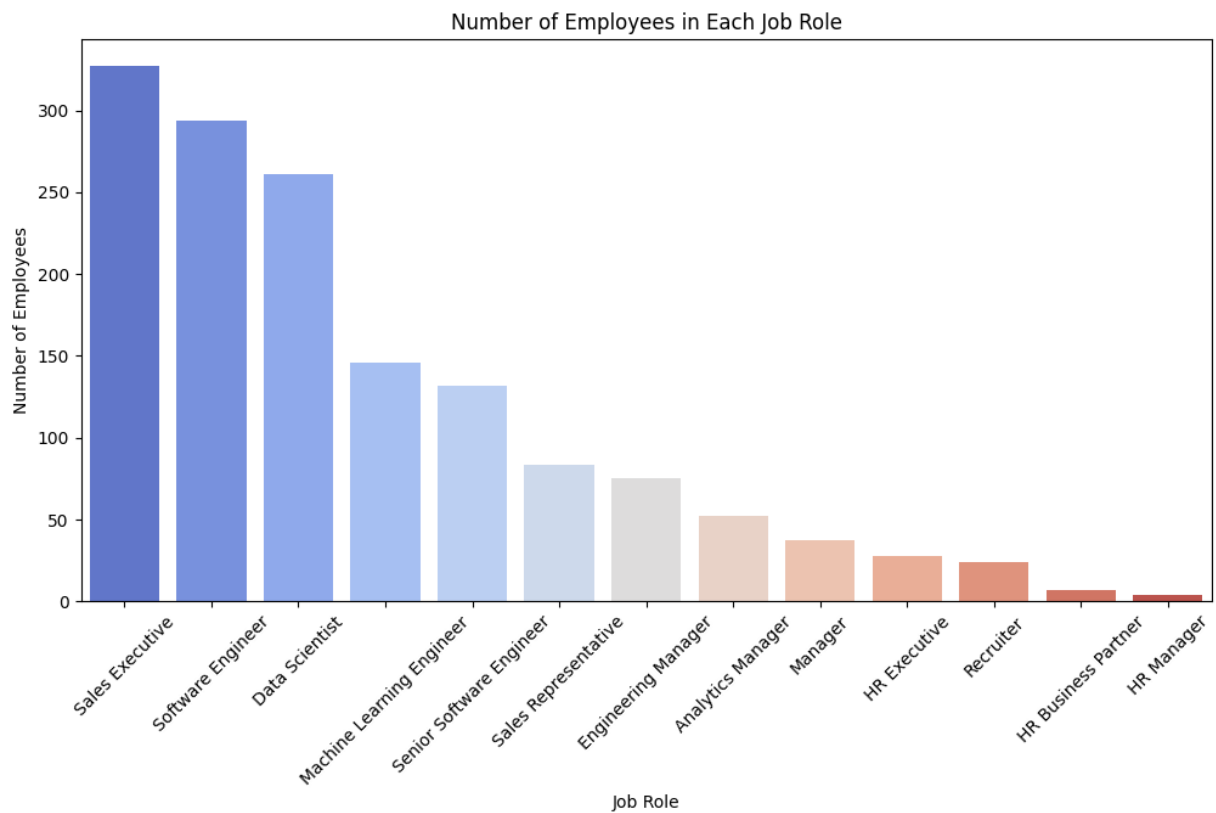
Name: count, dtype: int64

```
In [24]: plt.figure(figsize=(12, 6))
sns.barplot(x=job_role_counts.index, y=job_role_counts.values, palette='coolwarm')
plt.xlabel("Job Role")
plt.ylabel("Number of Employees")
plt.title("Number of Employees in Each Job Role")
plt.xticks(rotation=45)
plt.show()
```

C:\Users\adm.e\AppData\Local\Temp\ipykernel\_13604\3905543716.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=job_role_counts.index, y=job_role_counts.values, palette='coolwarm')
```



```
In [32]: # Employee per Hire Date
# Convert 'Hire Date' column to datetime
df_employee['HireDate'] = pd.to_datetime(df_employee['HireDate'])

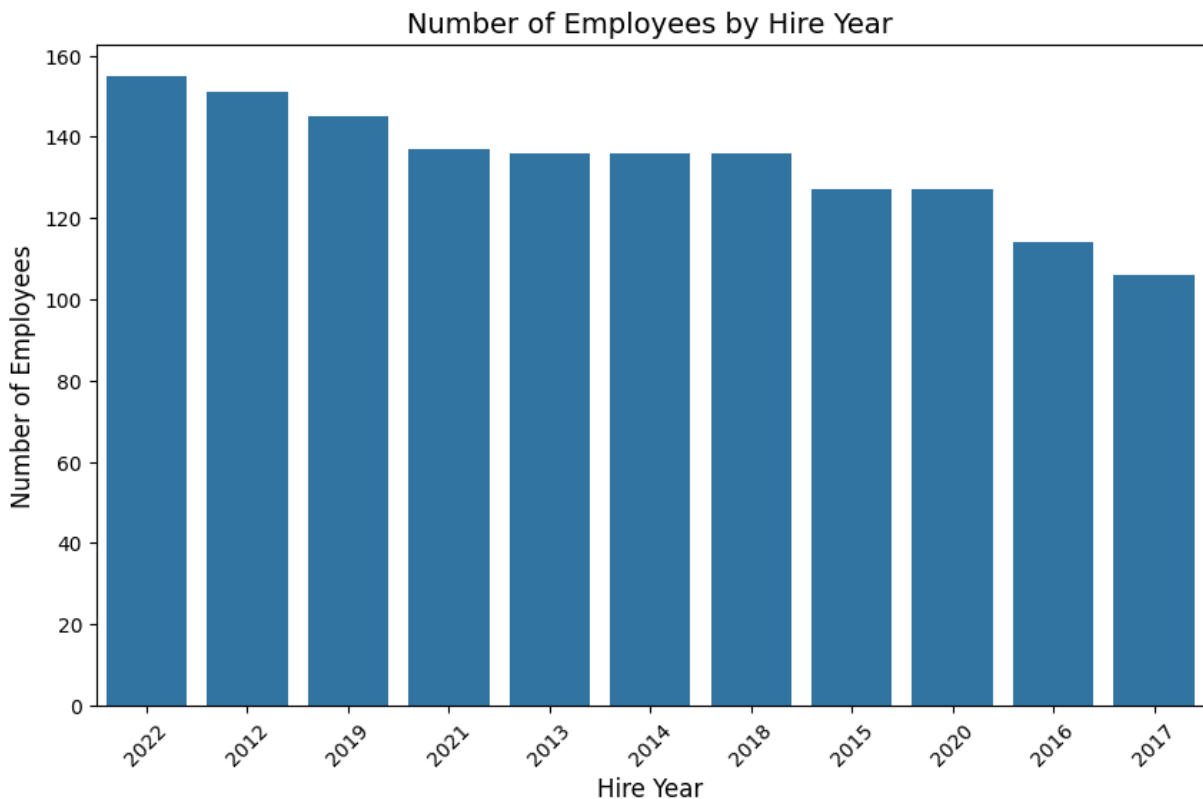
# Extract the year from the hire date
df_employee['Hire Year'] = df_employee['HireDate'].dt.year

# Count employees by hire year
employee_count_by_hire_year = df_employee['Hire Year'].value_counts().sort_index()

# Display the results
print(employee_count_by_hire_year)
```

```
Hire Year
2012    151
2013    136
2014    136
2015    127
2016    114
2017    106
2018    136
2019    145
2020    127
2021    137
2022    155
Name: count, dtype: int64
```

```
In [33]: # Plotting the number of employees by hire year
plt.figure(figsize=(10, 6))
sns.countplot(data=df_employee, x='Hire Year', order=df_employee['Hire Year'].value
plt.title('Number of Employees by Hire Year', fontsize=14)
plt.xlabel('Hire Year', fontsize=12)
plt.ylabel('Number of Employees', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



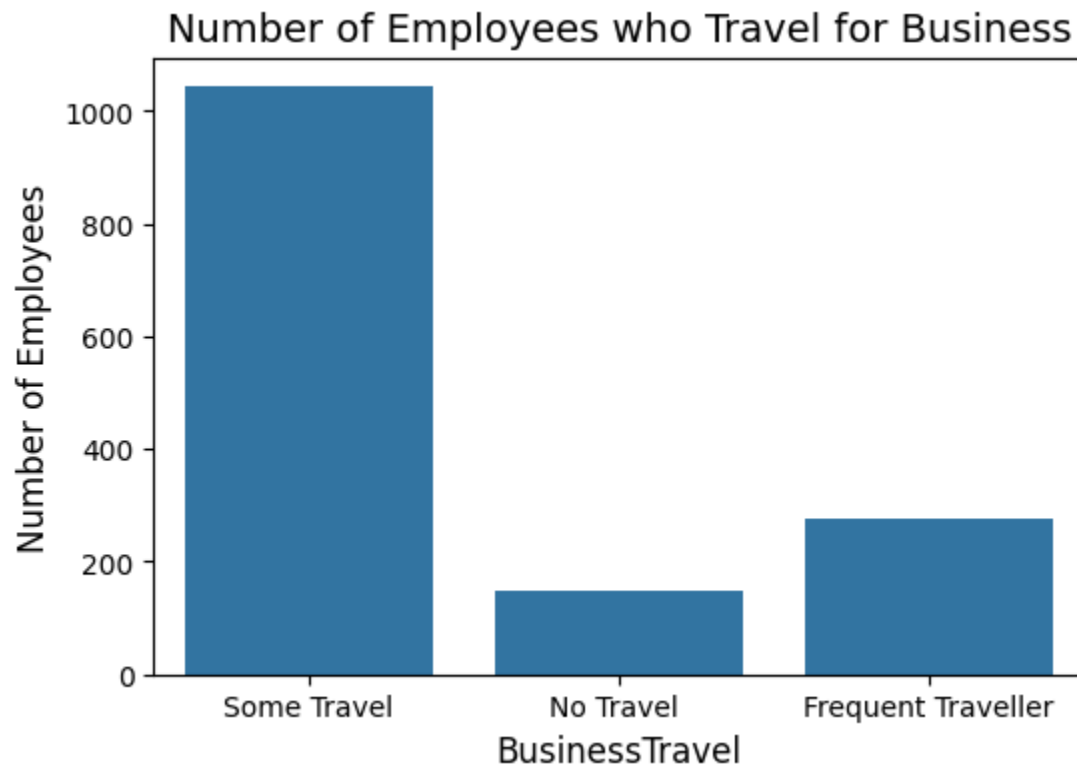
```
In [ ]:
```

```
In [34]: # Employee per Business Travel
# حساب عدد الموظفين الذين يسافرون للعمل
travel_counts = df_employee["BusinessTravel"].value_counts()
print(travel_counts)
```

```
Object `work` not found.
BusinessTravel
Some Travel      1043
Frequent Traveller  277
No Travel        150
Name: count, dtype: int64
```

In [ ]:

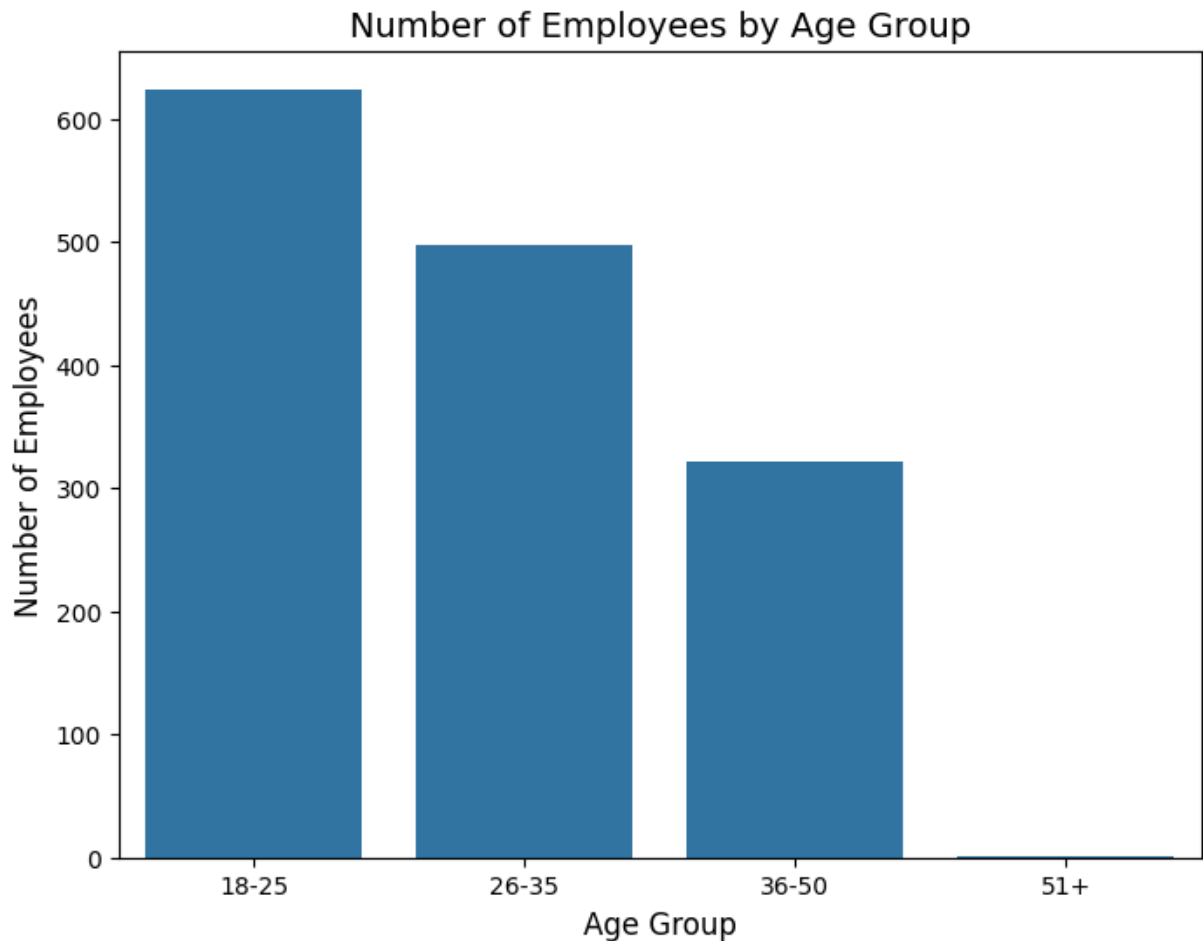
```
In [37]: plt.figure(figsize=(6, 4))
sns.countplot(data=df_employee, x='BusinessTravel')
plt.title('Number of Employees who Travel for Business', fontsize=14)
plt.xlabel('BusinessTravel', fontsize=12)
plt.ylabel('Number of Employees', fontsize=12)
plt.show()
```



```
In [47]: # Employee per age
bins = [18, 25, 35, 50, float('inf')] # تحديد الحدود العمرية
labels = ['18-25', '26-35', '36-50', '51+'] # تسمية الفئات العمرية
# إضافة عمود جديد للفئة العمرية
df_employee['Age Group'] = pd.cut(df_employee['Age'], bins=bins, labels=labels, right=False)
# حساب عدد الموظفين في كل فئة عمرية
age_group_count = df_employee['Age Group'].value_counts().sort_index()
print(age_group_count)
```

```
Age Group
18-25      624
26-35      497
36-50      322
51+         1
Name: count, dtype: int64
```

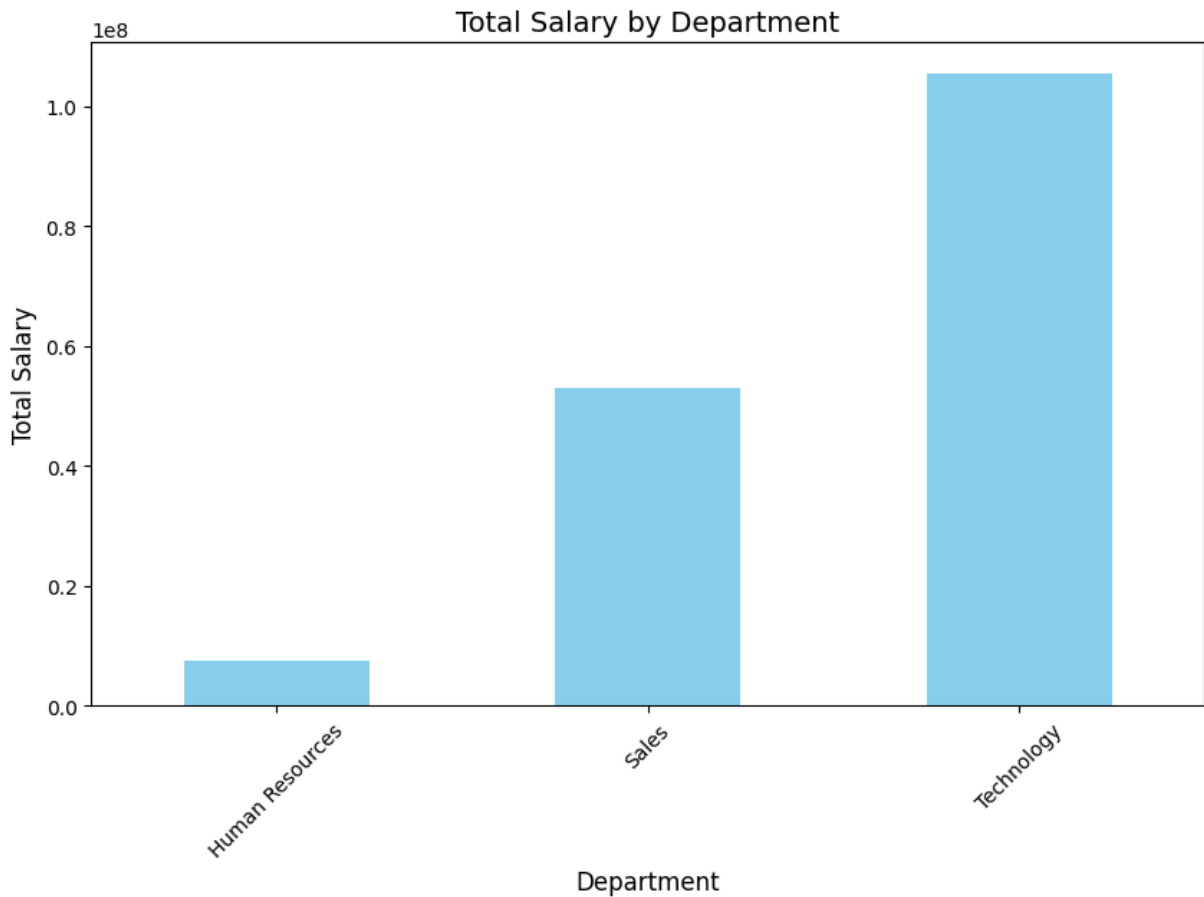
```
In [48]: plt.figure(figsize=(8, 6))
sns.countplot(data=df_employee, x='Age Group', order=df_employee['Age Group'].value
plt.title('Number of Employees by Age Group', fontsize=14)
plt.xlabel('Age Group', fontsize=12)
plt.ylabel('Number of Employees', fontsize=12)
plt.show()
```



```
In [50]: # Salary per Department
salary_by_department = df_employee.groupby('Department')['Salary'].sum()
print(salary_by_department)
```

```
Object `department` not found.
Department
Human Resources    7541025
Sales              53126454
Technology         105378573
Name: Salary, dtype: int64
```

```
In [51]: plt.figure(figsize=(10, 6))
salary_by_department.plot(kind='bar', color='skyblue')
plt.title('Total Salary by Department', fontsize=14)
plt.xlabel('Department', fontsize=12)
plt.ylabel('Total Salary', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



```
In [52]: # salary per Job Role
# حساب إجمالي الرواتب لكل دور وظيفي داخل كل قسم
salary_by_jobrole = df_employee.groupby('JobRole')['Salary'].sum()
print(salary_by_jobrole)
```

Object `department` not found.

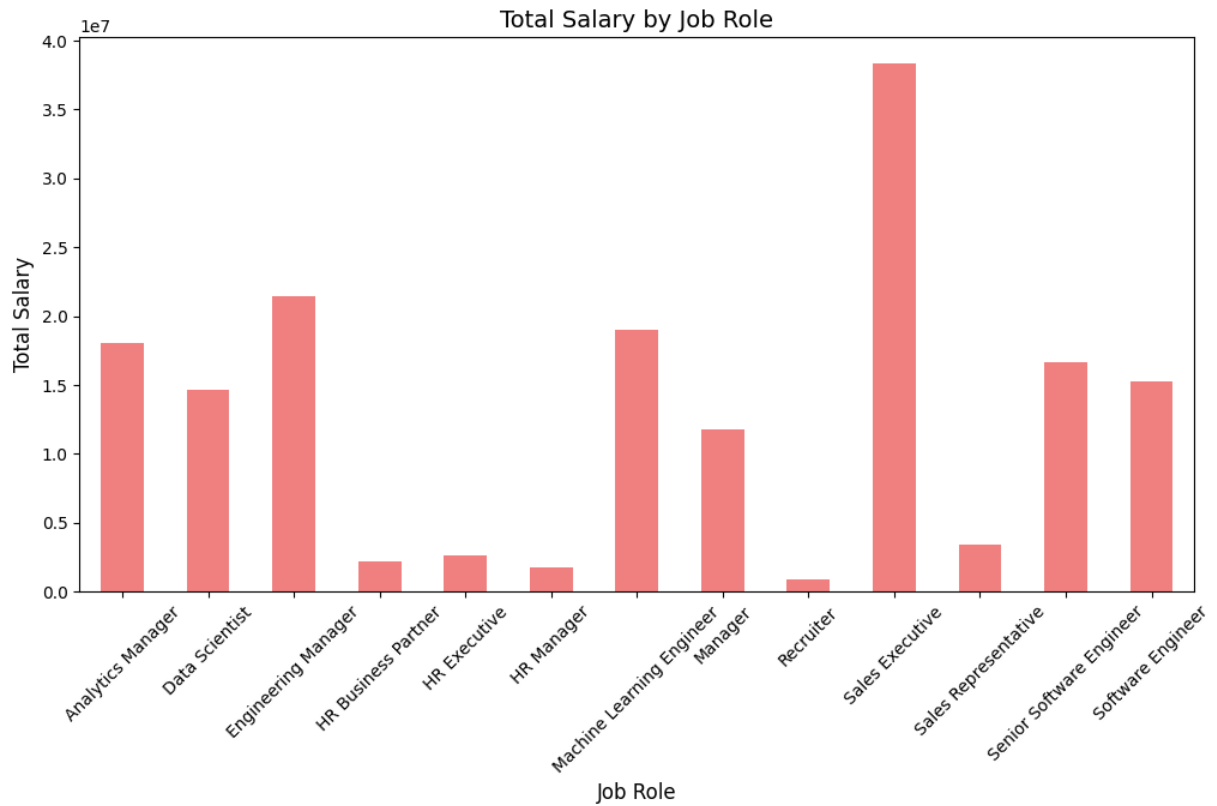
JobRole

|                           |          |
|---------------------------|----------|
| Analytics Manager         | 18017180 |
| Data Scientist            | 14636748 |
| Engineering Manager       | 21469388 |
| HR Business Partner       | 2198017  |
| HR Executive              | 2642145  |
| HR Manager                | 1797323  |
| Machine Learning Engineer | 19004034 |
| Manager                   | 11748649 |
| Recruiter                 | 903540   |
| Sales Executive           | 38322941 |
| Sales Representative      | 3374483  |
| Senior Software Engineer  | 16653291 |
| Software Engineer         | 15278313 |

Name: Salary, dtype: int64

```
In [53]: plt.figure(figsize=(12, 6))
salary_by_jobrole.plot(kind='bar', color='lightcoral')
plt.title('Total Salary by Job Role', fontsize=14)
plt.xlabel('Job Role', fontsize=12)
plt.ylabel('Total Salary', fontsize=12)
plt.xticks(rotation=45)
```

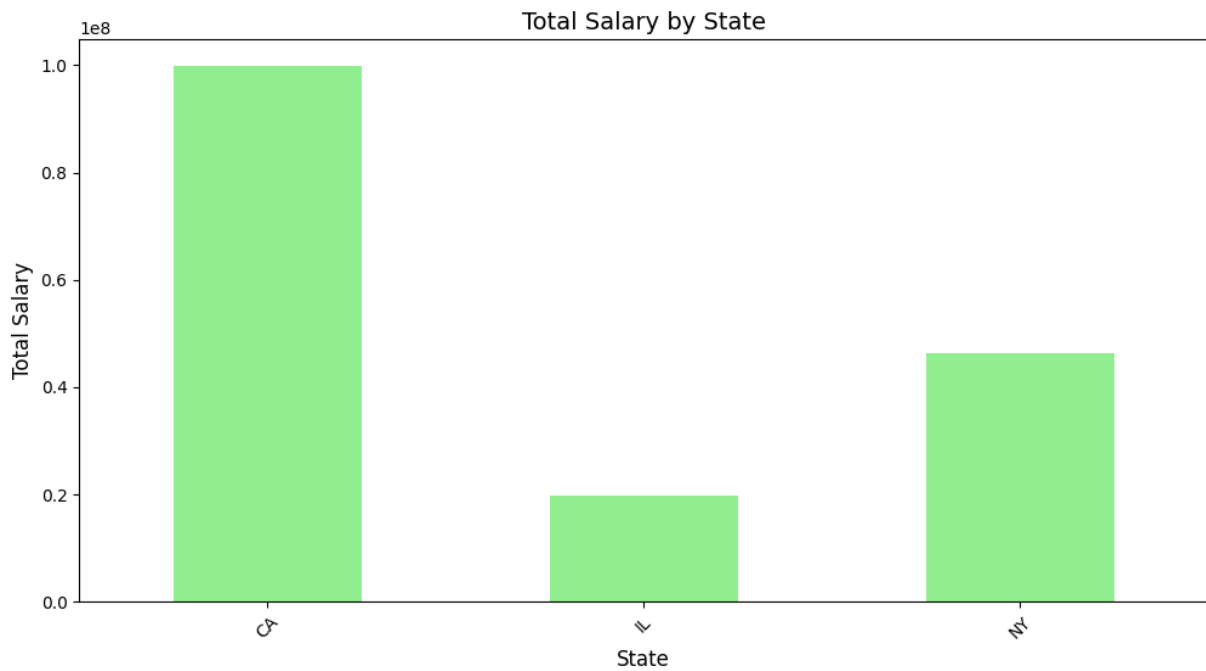
```
plt.show()
```



```
In [54]: # Salary per State
# حساب إجمالي المرتبات طبقاً للولاية
salary_by_state = df_employee.groupby('State')['Salary'].sum()
print(salary_by_state)
```

```
State
CA    99801324
IL    19831177
NY    46413551
Name: Salary, dtype: int64
```

```
In [55]: plt.figure(figsize=(12, 6))
salary_by_state.plot(kind='bar', color='lightgreen')
plt.title('Total Salary by State', fontsize=14)
plt.xlabel('State', fontsize=12)
plt.ylabel('Total Salary', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



```
In [56]: # Salary per gender
salary_by_gender = df_employee.groupby('Gender')['Salary'].sum()
print(salary_by_gender)
```

```
Gender
Female      76906592
Male        72764750
Non-Binary   13813043
Prefer Not To Say  2561667
Name: Salary, dtype: int64
```

```
In [57]: plt.figure(figsize=(8, 6))
salary_by_gender.plot(kind='bar', color='lightblue')
plt.title('Total Salary by Gender', fontsize=14)
plt.xlabel('Gender', fontsize=12)
plt.ylabel('Total Salary', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```

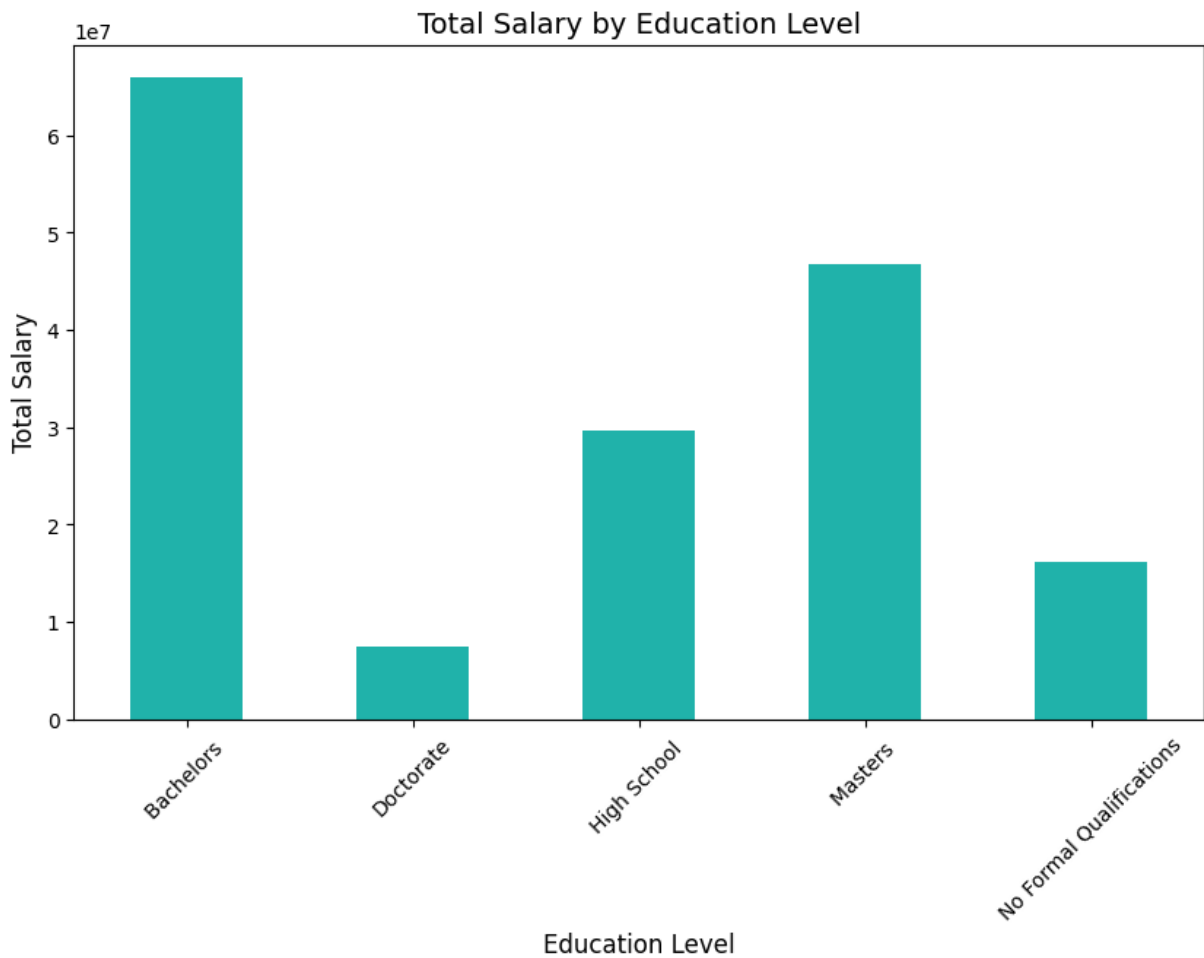




```
In [60]: # Salary Per Education Level
salary_by_education = df_employee.groupby('EducationLevel')['Salary'].sum()
print(salary_by_education)
```

```
EducationLevel
Bachelors          66011906
Doctorate           7404902
High School        29660911
Masters            46821141
No Formal Qualifications 16147192
Name: Salary, dtype: int64
```

```
In [61]: plt.figure(figsize=(10, 6))
salary_by_education.plot(kind='bar', color='lightseagreen')
plt.title('Total Salary by Education Level', fontsize=14)
plt.xlabel('Education Level', fontsize=12)
plt.ylabel('Total Salary', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



```
In [62]: # Salary per Age
bins = [18, 25, 35, 50, float('inf')] # تحديد الحدود العمرية
labels = ['18-25', '26-35', '36-50', '51+'] # تسمية الفئات العمرية
df_employee['Age Group'] = pd.cut(df_employee['Age'], bins=bins, labels=labels, right=True)
salary_by_age_group = df_employee.groupby('Age Group')['Salary'].sum()
print(salary_by_age_group)
```

```
Age Group
18-25    27110719
26-35    69406277
36-50    67480496
51+      2048560
Name: Salary, dtype: int64
```

C:\Users\adm.e\AppData\Local\Temp\ipykernel\_13604\117870652.py:4: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
salary_by_age_group = df_employee.groupby('Age Group')['Salary'].sum()
```

```
In [63]: plt.figure(figsize=(10, 6))
salary_by_age_group.plot(kind='bar', color='salmon')
plt.title('Total Salary by Age Group', fontsize=14)
plt.xlabel('Age Group', fontsize=12)
plt.ylabel('Total Salary', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



```
In [64]: # Salary per HireDate
df_employee['HireDate'] = pd.to_datetime(df_employee['HireDate'])
df_employee['HireYear'] = df_employee['HireDate'].dt.year
salary_by_hireyear = df_employee.groupby('HireYear')['Salary'].sum()
print(salary_by_hireyear)
```

```
HireYear
2012    20652863
2013    18656302
2014    16997091
2015    12090220
2016     9958220
2017    10189882
2018    13962424
2019    15051280
2020    16111236
2021    17361782
2022    15014752
Name: Salary, dtype: int64
```

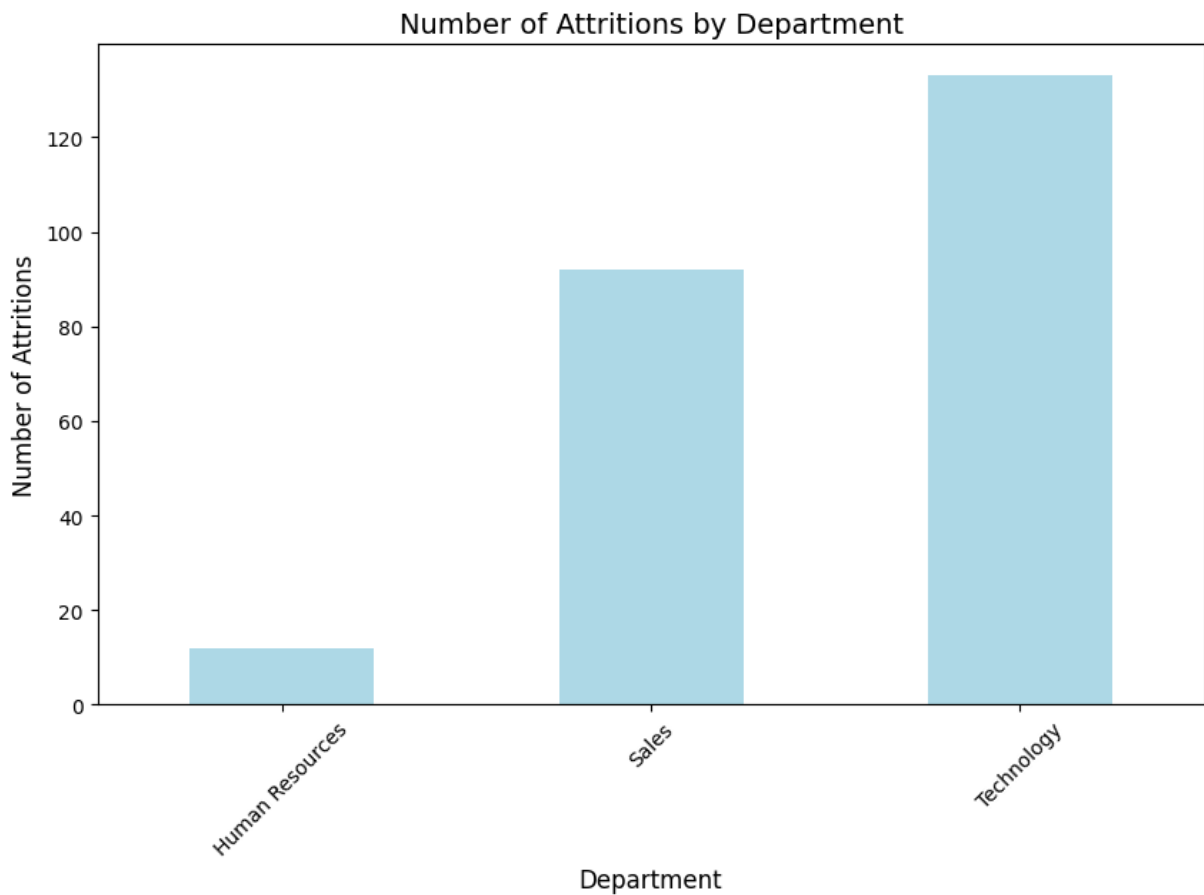
```
In [65]: plt.figure(figsize=(10, 6))
salary_by_hireyear.plot(kind='bar', color='lightcoral')
plt.title('Total Salary by Hire Year', fontsize=14)
plt.xlabel('Hire Year', fontsize=12)
plt.ylabel('Total Salary', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



```
In [66]: #Attrition Per Department
attrition_by_department = df_employee[df_employee['Attrition'] == 'Yes'].groupby('D')
print(attrition_by_department)
```

```
Department
Human Resources    12
Sales              92
Technology        133
dtype: int64
```

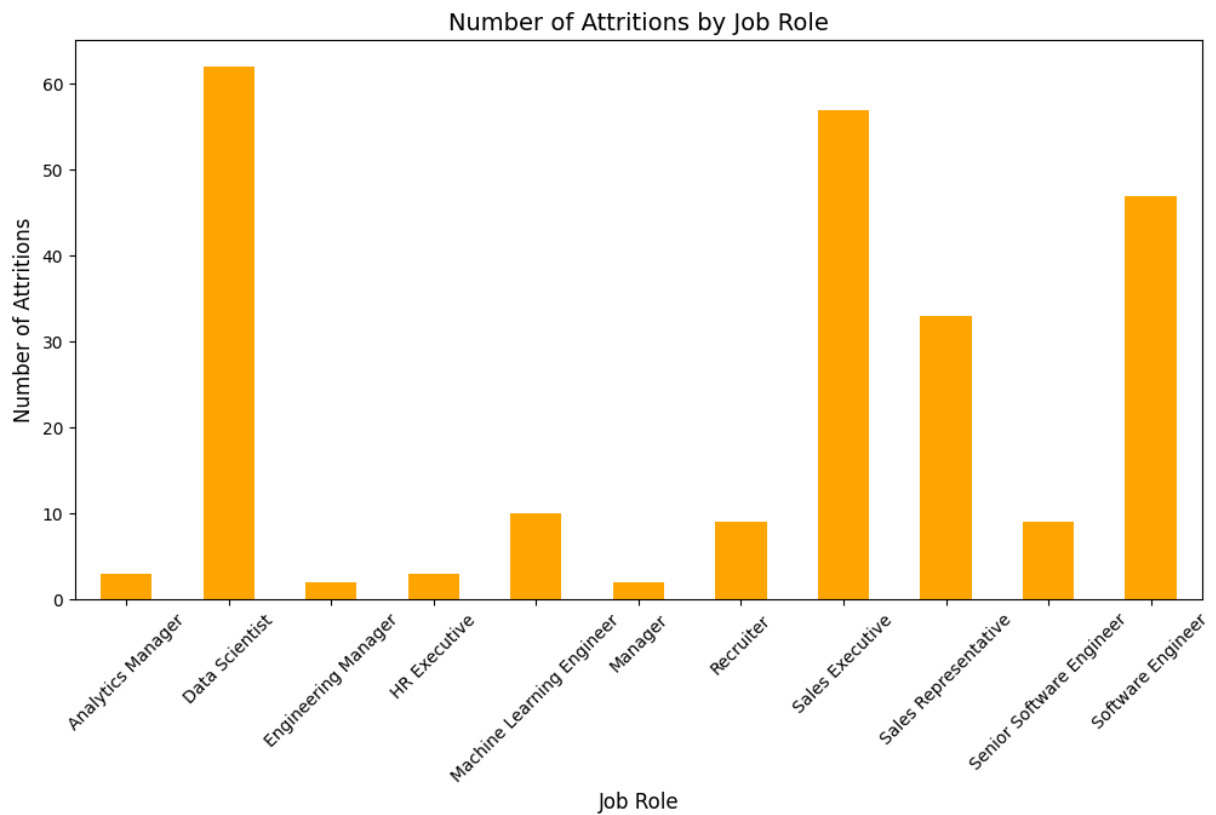
```
In [67]: plt.figure(figsize=(10, 6))
attrition_by_department.plot(kind='bar', color='lightblue')
plt.title('Number of Attritions by Department', fontsize=14)
plt.xlabel('Department', fontsize=12)
plt.ylabel('Number of Attritions', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



```
In [68]: #Attrition per Job Role
attrition_by_jobrole = df_employee[df_employee['Attrition'] == 'Yes'].groupby('JobR
print(attrition_by_jobrole)
```

```
JobRole
Analytics Manager      3
Data Scientist         62
Engineering Manager    2
HR Executive           3
Machine Learning Engineer  10
Manager                2
Recruiter              9
Sales Executive        57
Sales Representative    33
Senior Software Engineer  9
Software Engineer      47
dtype: int64
```

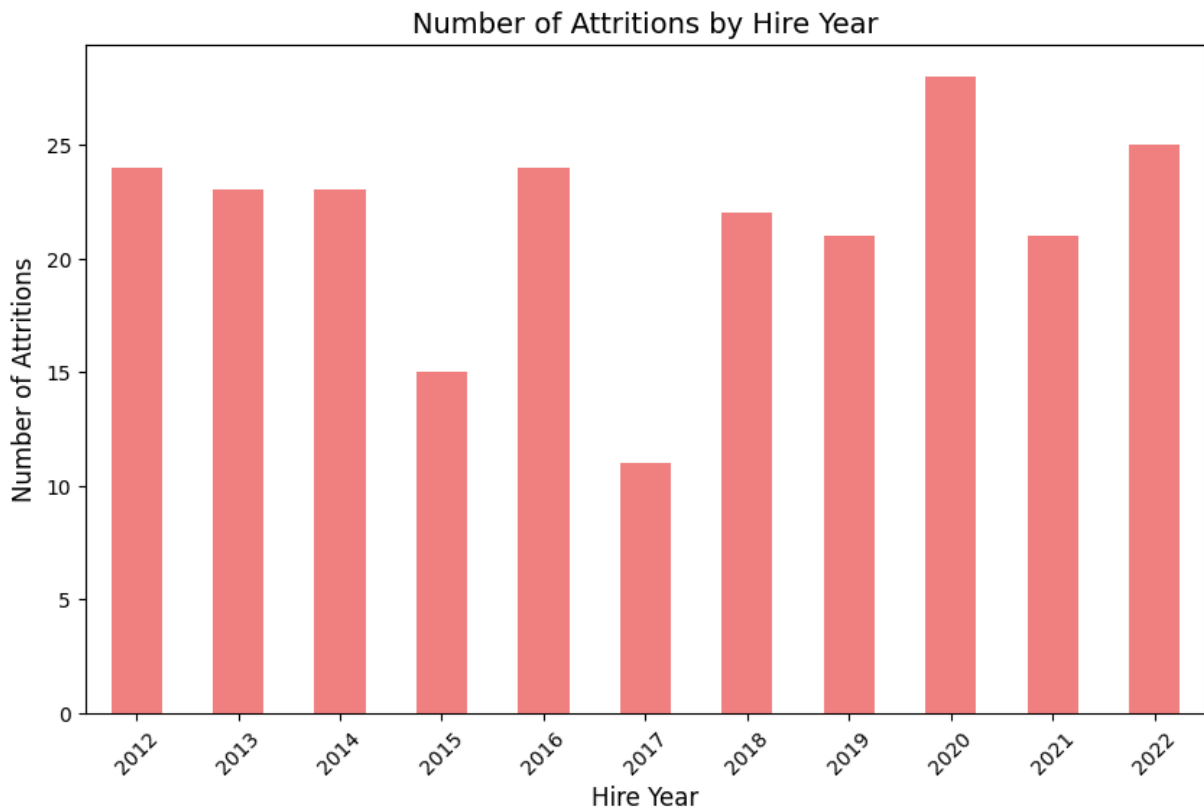
```
In [69]: plt.figure(figsize=(12, 6))
attrition_by_jobrole.plot(kind='bar', color='orange')
plt.title('Number of Attritions by Job Role', fontsize=14)
plt.xlabel('Job Role', fontsize=12)
plt.ylabel('Number of Attritions', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



```
In [73]: #Attrition Per HireDate
df_employee['HireYear'] = df_employee['HireDate'].dt.year
attrition_by_hireyear = df_employee[df_employee['Attrition'] == 'Yes'].groupby('Hir
print(attrition_by_hireyear)
```

```
HireYear
2012    24
2013    23
2014    23
2015    15
2016    24
2017    11
2018    22
2019    21
2020    28
2021    21
2022    25
dtype: int64
```

```
In [74]: plt.figure(figsize=(10, 6))
attrition_by_hireyear.plot(kind='bar', color='lightcoral')
plt.title('Number of Attritions by Hire Year', fontsize=14)
plt.xlabel('Hire Year', fontsize=12)
plt.ylabel('Number of Attritions', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```

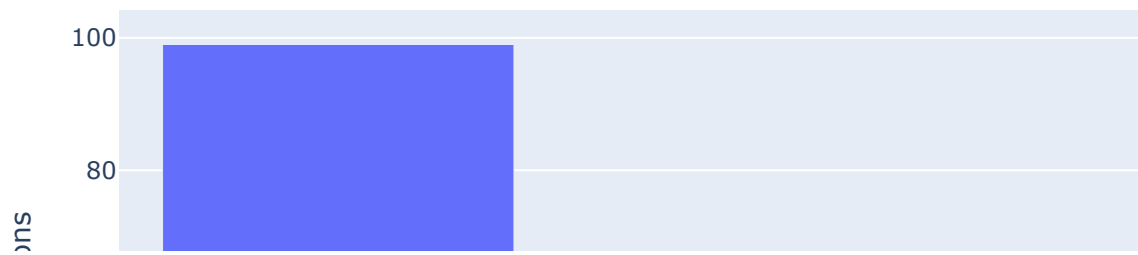


```
In [77]: #Attrition per Education Level
attrition_by_education_level = df_employee[df_employee['Attrition'] == 'Yes'].group
print(attrition_by_education_level)
```

```
EducationLevel
Bachelors          99
Doctorate           5
High School        44
Masters            58
No Formal Qualifications  31
dtype: int64
```

```
In [82]: import plotly.express as px
attrition_by_education_level = df_employee[df_employee['Attrition'] == 'Yes'].group
fig = px.bar(attrition_by_education_level, x='EducationLevel', y='Attrition Count',
             title='Number of Attritions by Education Level',
             labels={'EducationLevel': 'Education Level', 'Attrition Count': 'Number of Attritions'},
             color='EducationLevel')
fig.show()
```

## Number of Attritions by Education Level

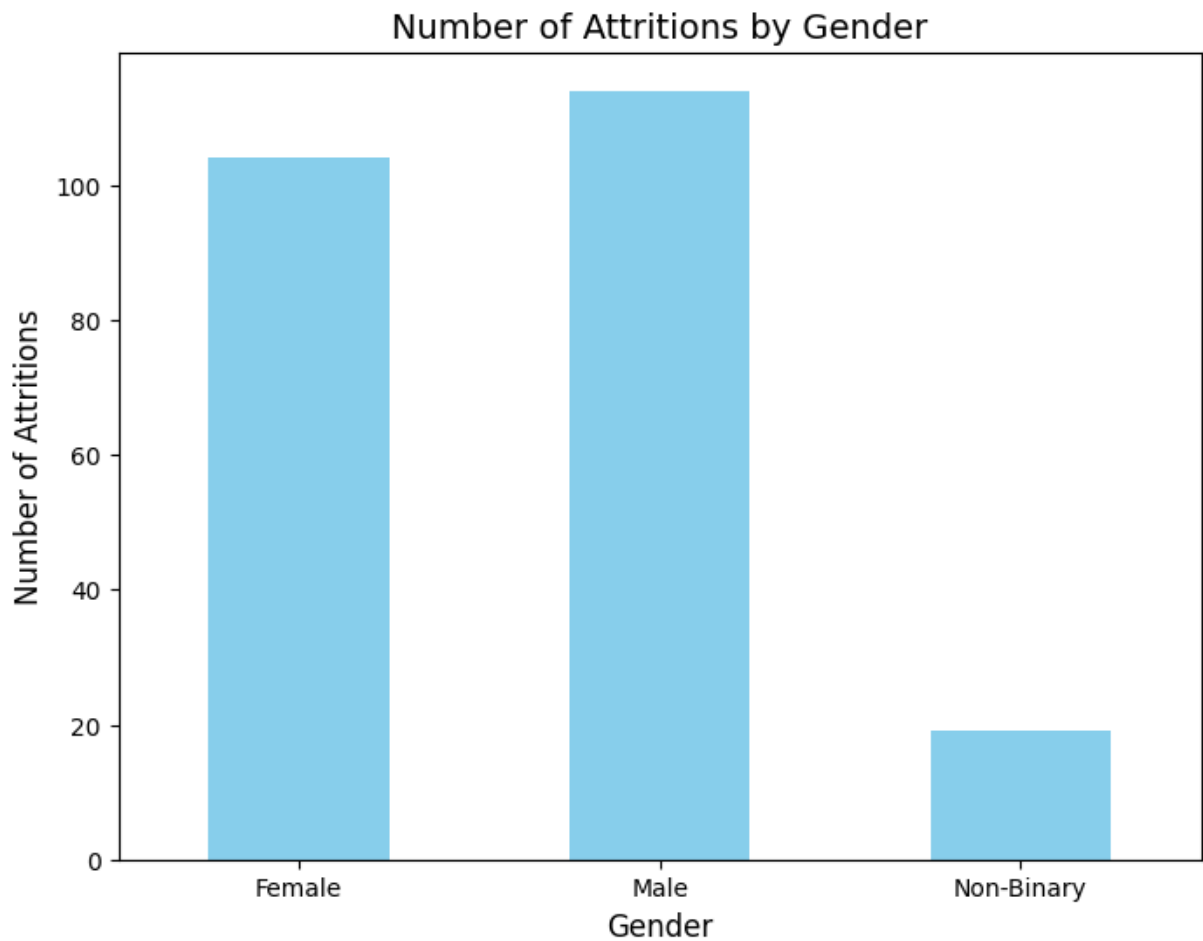


```
In [83]: attrition_by_gender = df_employee[df_employee['Attrition'] == 'Yes'].groupby('Gender')
print(attrition_by_gender)
```

```
Gender
Female      104
Male       114
Non-Binary   19
dtype: int64
```

```
In [84]: plt.figure(figsize=(8, 6))
attrition_by_gender.plot(kind='bar', color='skyblue')
plt.title('Number of Attritions by Gender', fontsize=14)
plt.xlabel('Gender', fontsize=12)
plt.ylabel('Number of Attritions', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```

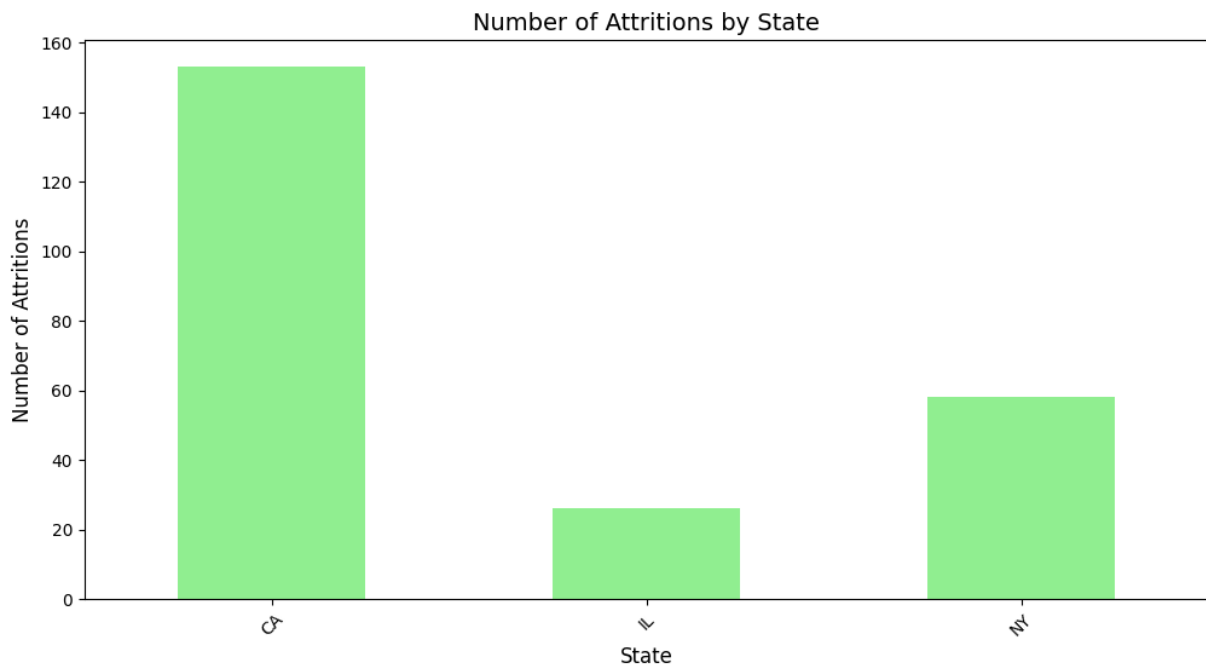




```
In [85]: #Attrition Per State
attrition_by_state = df_employee[df_employee['Attrition'] == 'Yes'].groupby('State')
print(attrition_by_state)
```

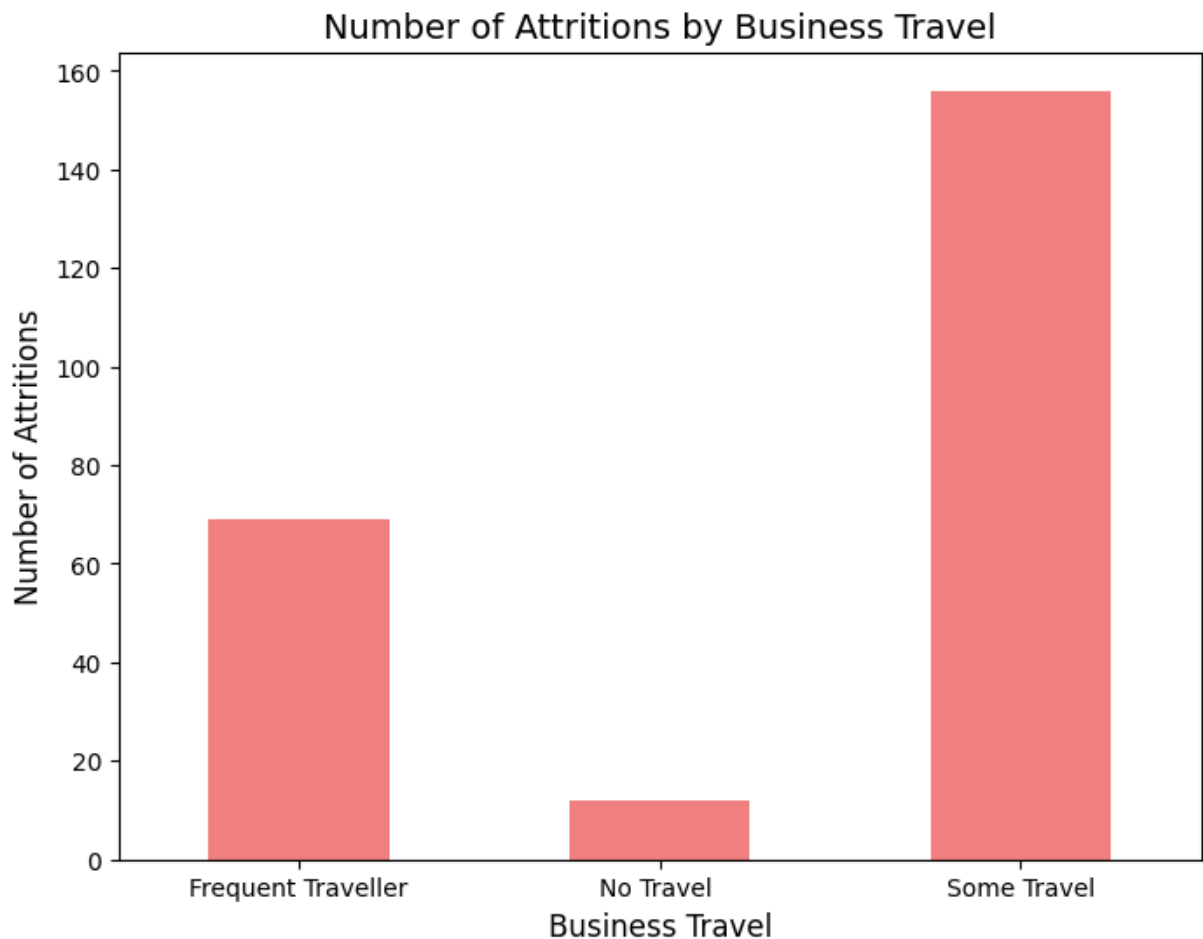
```
State
CA    153
IL     26
NY     58
dtype: int64
```

```
In [87]: plt.figure(figsize=(12, 6))
attrition_by_state.plot(kind='bar', color='lightgreen')
plt.title('Number of Attritions by State', fontsize=14)
plt.xlabel('State', fontsize=12)
plt.ylabel('Number of Attritions', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



```
In [88]: #Attrition Per
attrition_by_business_travel = df_employee[df_employee['Attrition'] == 'Yes'].group
print(attrition_by_business_travel)
plt.figure(figsize=(8, 6))
attrition_by_business_travel.plot(kind='bar', color='lightcoral')
plt.title('Number of Attritions by Business Travel', fontsize=14)
plt.xlabel('Business Travel', fontsize=12)
plt.ylabel('Number of Attritions', fontsize=12)
plt.xticks(rotation=0)
plt.show()
```

```
BusinessTravel
Frequent Traveller    69
No Travel              12
Some Travel           156
dtype: int64
```



```
In [4]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the employee dataset
df = pd.read_csv(r"D:\Data Analysis\Ahmed Samier\PROJECT\Final project\Final\Employee Attrition.csv")

# Convert 'HireDate' to datetime
df['HireDate'] = pd.to_datetime(df['HireDate'])

# Filter employees who resigned
df_resigned = df[df['Attrition'] == 'Yes'].copy()

# Extract year from the HireDate (proxy for resignation year)
df_resigned['ResignYear'] = df_resigned['HireDate'].dt.year

# Group by year to count resignations
annual_attrition = df_resigned.groupby('ResignYear').size().reset_index(name='Attritions')

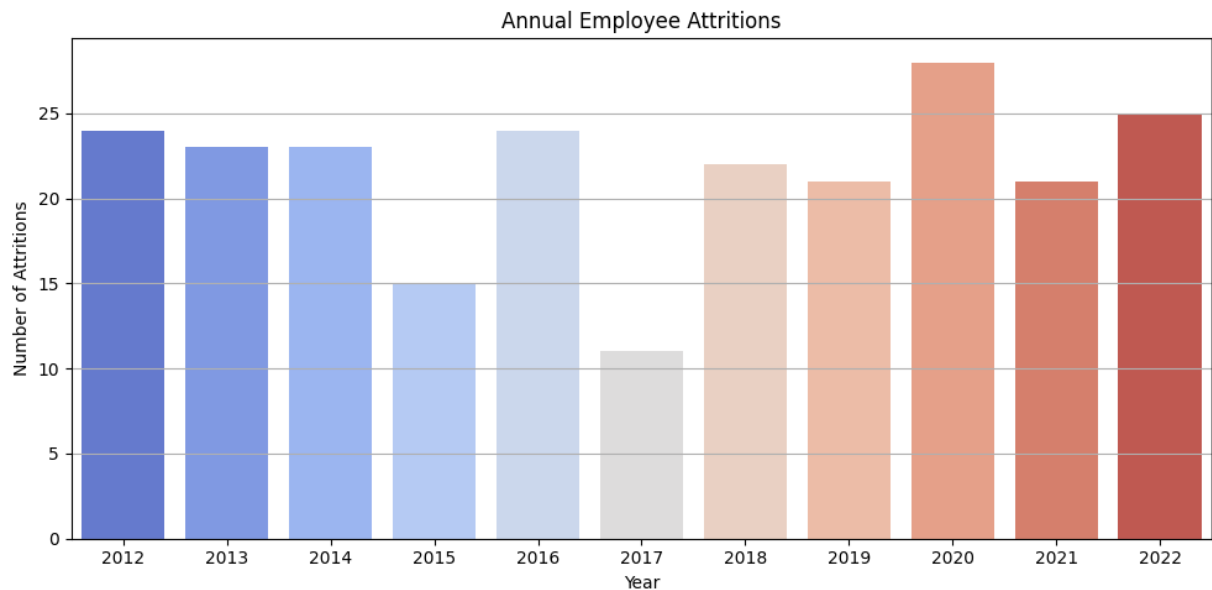
# Plotting
plt.figure(figsize=(10, 5))
sns.barplot(x='ResignYear', y='Attritions', data=annual_attrition, palette='coolwarm')
plt.title('Annual Employee Attritions')
plt.xlabel('Year')
plt.ylabel('Number of Attritions')
plt.grid(True, axis='y')
```

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

C:\Users\adm.e\AppData\Local\Temp\ipykernel\_22828\2015713429.py:23: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x='ResignYear', y='Attritions', data=annual_attrition, palette='coolwa  
rm')
```



In [ ]:

In [ ]: