Project Overview

This project analyzes an HR dataset using Python to understand employee demographics, work patterns, salary distribution, and attrition trends.

The goal of this analysis is to help HR teams and management make informed decisions related to workforce planning, employee retention, compensation, and organizational growth.

Key Objectives:

- Analyze employee status and attrition patterns
- Understand department-wise employee distribution
- Compare salaries across departments and job roles
- Analyze remote vs onsite work patterns
- Study hiring trends over time

In [1]:
```
# importing important libraries
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
# importing  the dataset
df = pd.read_csv(r"C:\Users\adity\OneDrive\Desktop\hr data\HR_Analytics_Project_
```

In [3]:
```
df
```

Out[3]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Perf
0	1	Employee_1	R&D	Research Analyst	2012-05-10	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	
2	3	Employee_3	Sales	Sales Executive	2013-03-30	City, Germany	
3	4	Employee_4	HR	Recruiter	2010-07-09	City, UK	
4	5	Employee_5	Finance	Analyst	2017-11-28	City, USA	
...	...	...	...	...	...	...	
29995	29996	Employee_29996	Sales	Sales Executive	2018-09-22	City, Germany	
29996	29997	Employee_29997	IT	IT Manager	2015-04-09	City, India	
29997	29998	Employee_29998	Finance	Accountant	2016-08-18	City, Germany	
29998	29999	Employee_29999	Sales	Sales Manager	2010-05-12	City, UK	
29999	30000	Employee_30000	HR	HR Executive	2024-08-05	City, UK	

30000 rows × 11 columns

In [4]:
```
# getting basic information about the dataset

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   EmployeeID         30000 non-null  int64 
 1   EmployeeName       30000 non-null  object
 2   Department         30000 non-null  object
 3   JobTitle           30000 non-null  object
 4   HireDate           30000 non-null  object
 5   Location           30000 non-null  object
 6   PerformanceRating  30000 non-null  int64 
 7   ExperienceYears    30000 non-null  int64 
 8   Status             30000 non-null  object
 9   WorkMode           30000 non-null  object
 10  Salary             30000 non-null  int64 
dtypes: int64(4), object(7)
memory usage: 2.5+ MB
```

In [5]:
```
# change the data type of HireDate column

df['HireDate']= pd.to_datetime(df['HireDate'])
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   EmployeeID         30000 non-null  int64
 1   EmployeeName       30000 non-null  object
 2   Department         30000 non-null  object
 3   JobTitle           30000 non-null  object
 4   HireDate           30000 non-null  datetime64[ns]
 5   Location           30000 non-null  object
 6   PerformanceRating  30000 non-null  int64
 7   ExperienceYears    30000 non-null  int64
 8   Status             30000 non-null  object
 9   WorkMode           30000 non-null  object
 10  Salary             30000 non-null  int64
dtypes: datetime64[ns](1), int64(4), object(6)
memory usage: 2.5+ MB
```

In [7]:
```
df.head()
```

Out[7]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Performanc
0	1	Employee_1	R&D	Research Analyst	2012-05-10	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	
2	3	Employee_3	Sales	Sales Executive	2013-03-30	City, Germany	
3	4	Employee_4	HR	Recruiter	2010-07-09	City, UK	
4	5	Employee_5	Finance	Analyst	2017-11-28	City, USA	

In [8]:
```
df['ExperienceYears'].nunique()
```

Out[8]: 16

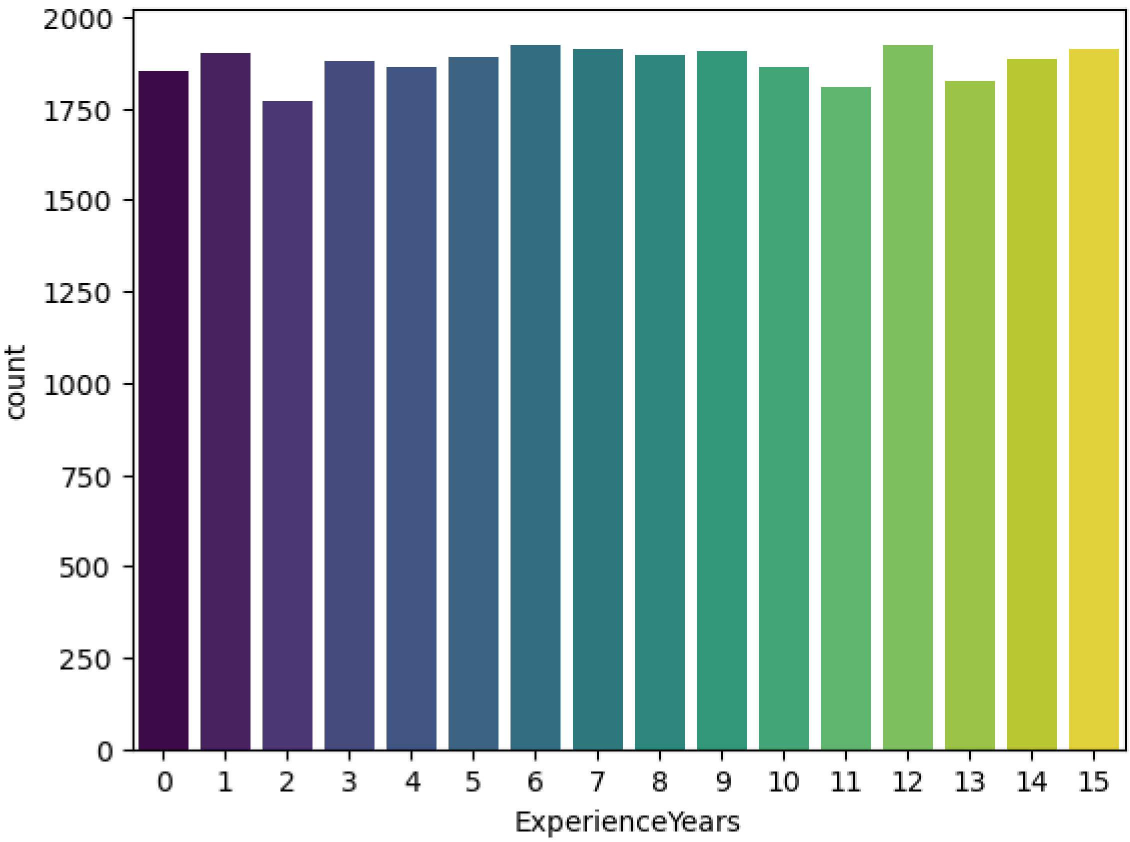
In [9]:
```
df['ExperienceYears'].unique()
```

Out[9]: array([4, 1, 5, 6, 8, 7, 13, 11, 9, 15, 3, 2, 0, 10, 14, 12])

In [10]:
```
df['ExperienceYears'].mean()
```

Out[10]: np.float64(7.518966666666667)

In [11]:
```
sns.countplot( x = 'ExperienceYears', data = df,
hue ='ExperienceYears', palette='viridis', legend = False)
plt.show()
```



In [12]:
```
df['ExperienceYears'].value_counts()
```

Out[12]:
```
ExperienceYears
6     1924
12    1920
15    1914
7     1909
9     1907
1     1900
8     1896
5     1888
14    1886
3     1879
10    1864
4     1863
0     1853
13    1822
11    1808
2     1767
Name: count, dtype: int64
```

In [13]:
```
df.select_dtypes(include= 'object')
```

Out[13]:

	EmployeeName	Department	JobTitle	Location	Status	WorkMode
0	Employee_1	R&D	Research Analyst	City, UK	Active	Remote
1	Employee_2	Marketing	SEO Specialist	City, UK	Resigned	Onsite
2	Employee_3	Sales	Sales Executive	City, Germany	Active	Remote
3	Employee_4	HR	Recruiter	City, UK	Terminated	Onsite
4	Employee_5	Finance	Analyst	City, USA	Terminated	Onsite
...	...	...	...	...	...	...
29995	Employee_29996	Sales	Sales Executive	City, Germany	Active	Onsite
29996	Employee_29997	IT	IT Manager	City, India	Resigned	Remote
29997	Employee_29998	Finance	Accountant	City, Germany	Active	Remote
29998	Employee_29999	Sales	Sales Manager	City, UK	Active	Onsite
29999	Employee_30000	HR	HR Executive	City, UK	Active	Onsite

30000 rows × 6 columns

In [14]:
```
df.select_dtypes(include= 'number')
```

Out[14]:

	EmployeeID	PerformanceRating	ExperienceYears	Salary
0	1	5	4	117498
1	2	5	1	35311
2	3	2	5	114478
3	4	4	6	99092
4	5	4	8	98148
...	...	...	...	...
29995	29996	2	2	45840
29996	29997	3	5	132637
29997	29998	2	5	60773
29998	29999	4	7	110392
29999	30000	4	7	68221

30000 rows × 4 columns

Q.1) What is the distribution of Employee status(Active, resigned, retired, terminated) ?

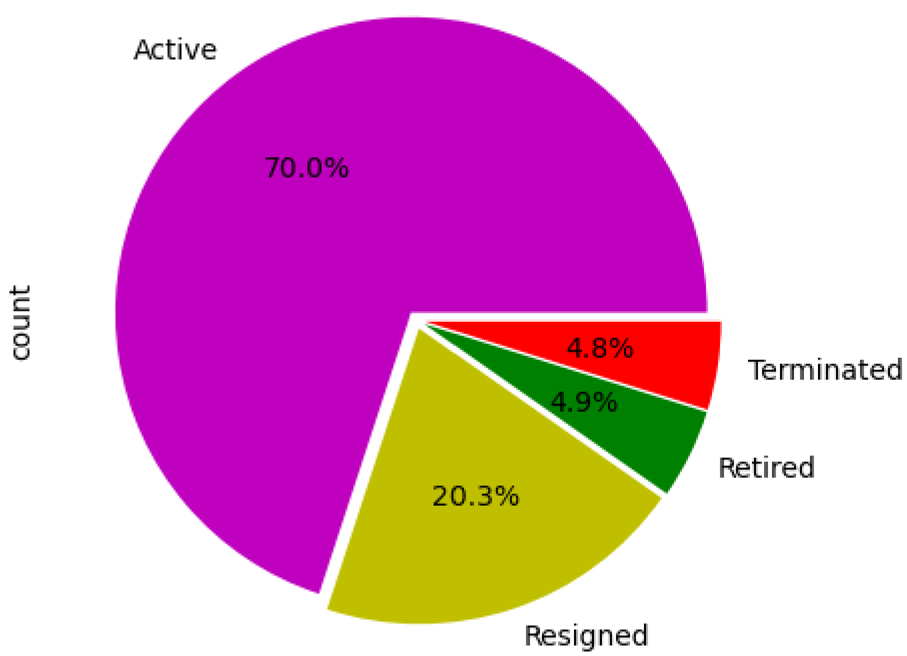
In [15]:
```
status = df['Status'].value_counts()
status
```

Out[15]:
```
Status
Active        20990
Resigned       6092
Retired        1471
Terminated     1447
Name: count, dtype: int64
```

In [16]:
```
type(status)
```

Out[16]:
```
pandas.core.series.Series
```

In [17]:
```
status.plot(kind = 'pie', colors = 'mygr', autopct= '%1.1f%%', explode= (0.03,0.
plt.show()
```



Q.2) What is the distribution of Wprk Modes(On-Site , Remote) ?

In [18]:
```
df.head(2)
```

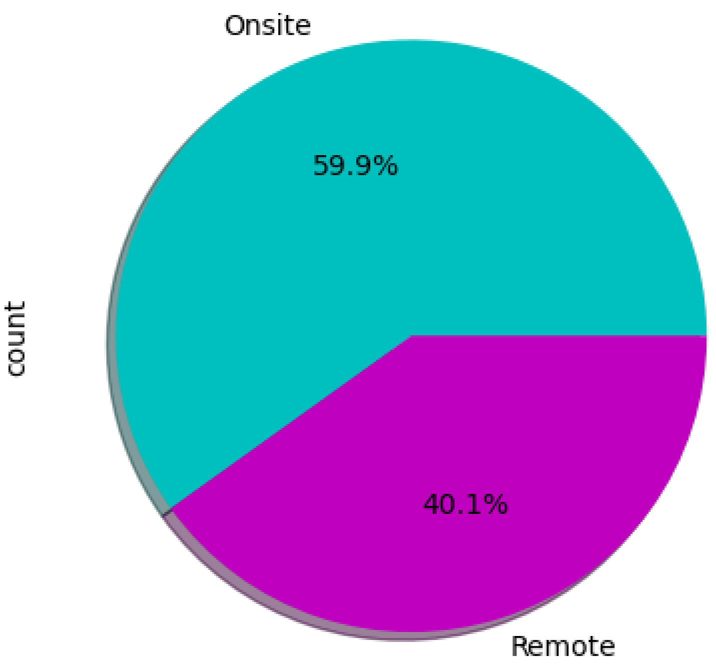
Out[18]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Performan
0	1	Employee_1	R&D	Research Analyst	2012-05-10	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	

In [19]:
```
work = df['WorkMode'].value_counts()
work
```

Out[19]:
```
WorkMode
Onsite     17965
Remote     12035
Name: count, dtype: int64
```

In [20]:
```
work.plot(kind = 'pie', colors= 'cm', autopct= '%1.1f%%', shadow = True)
plt.show()
```



Q.3) How many employees are there in each department ?

In [21]:
```
df.head(3)
```

Out[21]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Performanc
0	1	Employee_1	R&D	Research Analyst	2012-05-10	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	
2	3	Employee_3	Sales	Sales Executive	2013-03-30	City, Germany	

In [22]:
```
emp = df['Department'].value_counts()
emp
```

Out[22]:
```
Department
HR            4349
R&D           4319
Operations    4305
Sales         4291
Marketing     4279
Finance       4242
IT            4215
Name: count, dtype: int64
```

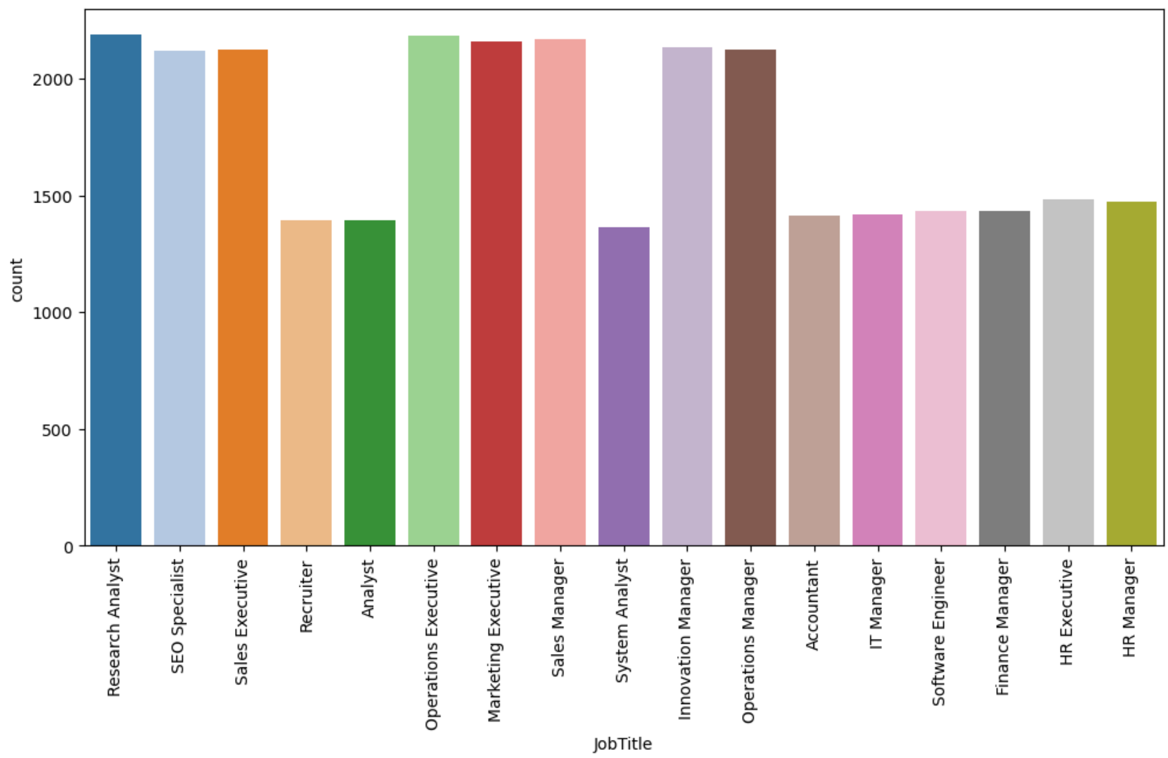
In [23]:
```
sns.countplot(x ='Department', data = df,
hue= 'Department', palette=  'viridis', legend = False)
plt.show()
```

In [24]:
```
title = df['JobTitle'].value_counts()

title
```

Out[24]:
```
JobTitle
Research Analyst        2187
Operations Executive    2183
Sales Manager           2168
Marketing Executive     2160
Innovation Manager      2132
Sales Executive         2123
Operations Manager      2122
SEO Specialist          2119
HR Executive            1484
HR Manager              1472
Finance Manager         1435
Software Engineer       1434
IT Manager              1419
Accountant              1412
Analyst                 1395
Recruiter               1393
System Analyst          1362
Name: count, dtype: int64
```

In [25]:
```
plt.figure(figsize=(12,6))
sns.countplot(x = 'JobTitle', data = df, hue= 'JobTitle', palette= 'tab20', lege
plt.xticks(rotation =  'vertical')
plt.show()
```



Q.4) What is the average salary by department?

In [26]:
```
df.head(2)
```

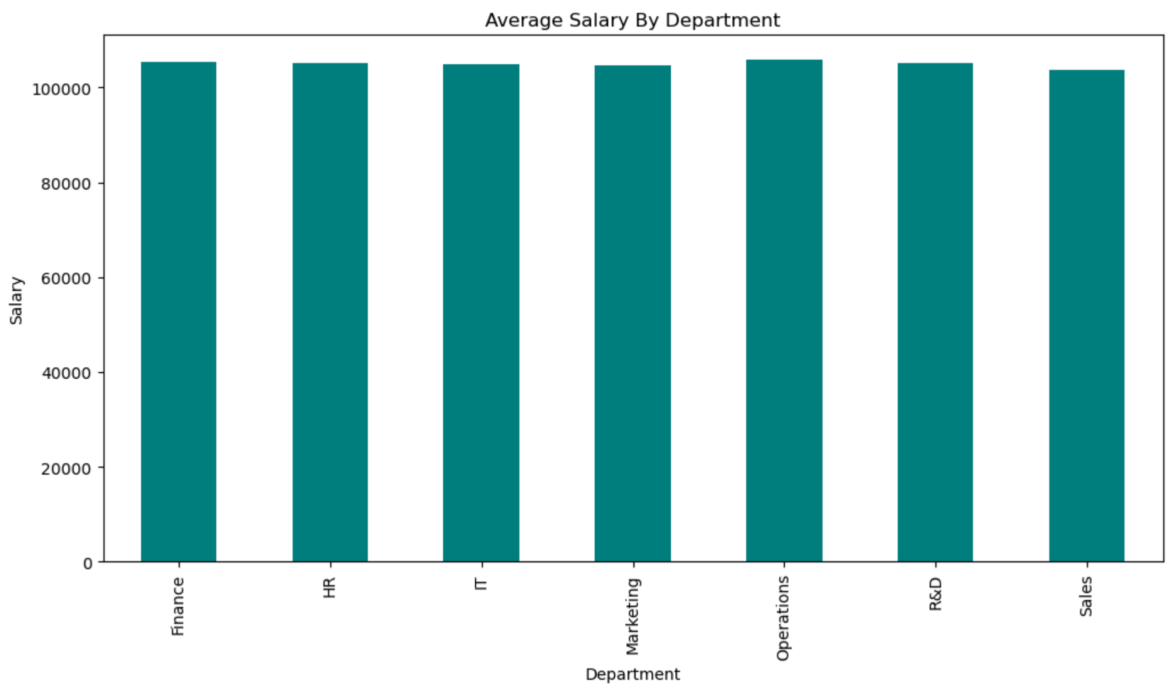
Out[26]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Performan
0	1	Employee_1	R&D	Research Analyst	2012-05-10	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	

In [27]:
```
avg_salary = df.groupby('Department')['Salary'].mean()
avg_salary.apply(lambda x: f"{x:,.2f}")
```

Out[27]:
```
Department
Finance       105,483.03
HR            105,239.91
IT            104,875.64
Marketing     104,726.72
Operations    105,777.59
R&D           105,046.74
Sales         103,803.98
Name: Salary, dtype: object
```

In [28]:
```
plt.figure(figsize=(12,6))
avg_salary.plot(x = avg_salary.index, y = avg_salary.values, kind = 'bar', color
plt.title("Average Salary By Department")
plt.ylabel("Salary")
plt.show()
```

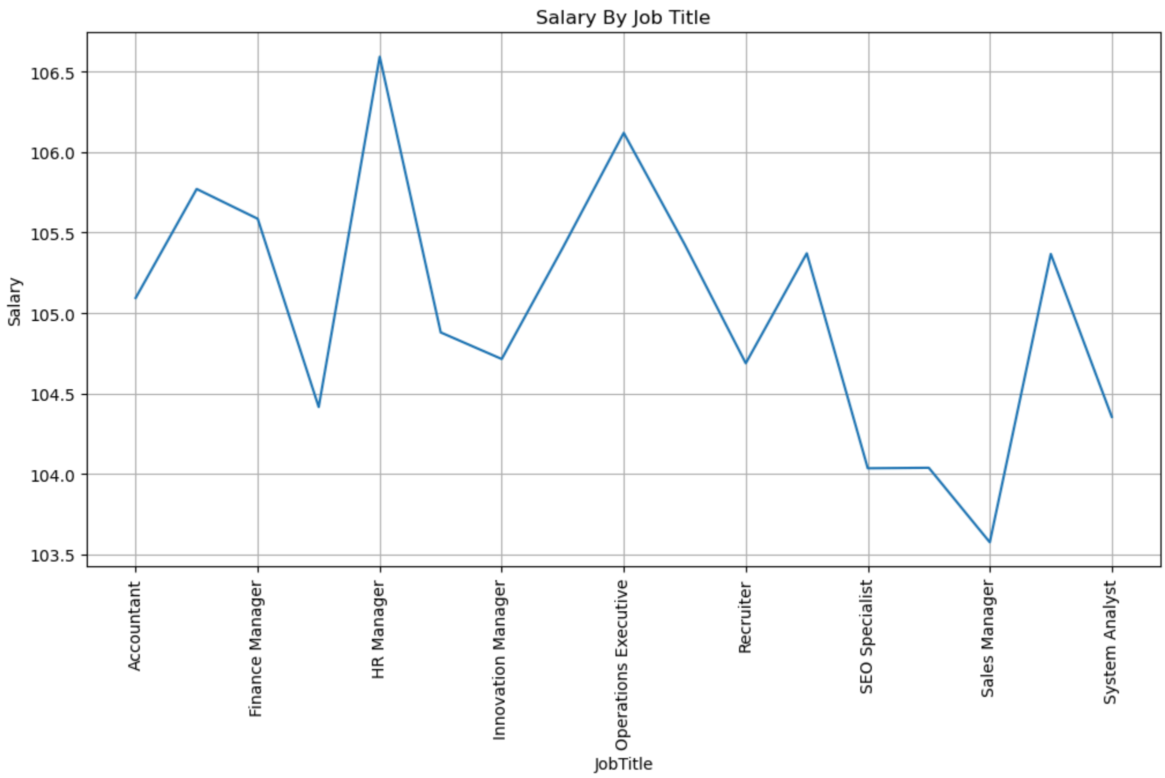


Q.5) Which job title has the highest salary?

In [29]:
```
salary = df.groupby('JobTitle')['Salary'].mean()/1000
salary.apply(lambda x: f"{x:,.2f}")
```

Out[29]:
```
JobTitle
Accountant              105.09
Analyst                 105.77
Finance Manager         105.59
HR Executive            104.42
HR Manager              106.59
IT Manager              104.88
Innovation Manager      104.71
Marketing Executive     105.41
Operations Executive    106.12
Operations Manager      105.43
Recruiter               104.69
Research Analyst        105.37
SEO Specialist          104.03
Sales Executive         104.04
Sales Manager           103.57
Software Engineer       105.37
System Analyst          104.35
Name: Salary, dtype: object
```

In [30]:
```
plt.figure(figsize=(12,6))
salary.plot(x = salary.index, y = salary.values)
plt.grid(), plt.title("Salary By Job Title"), plt.ylabel("Salary")
plt.xticks(rotation = 90)
plt.show()
```



Q6) What is the average salary in different department based on job title?

In [31]:
```
df.head()
```

Out[31]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Performanc
0	1	Employee_1	R&D	Research Analyst	2012-05-10	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	
2	3	Employee_3	Sales	Sales Executive	2013-03-30	City, Germany	
3	4	Employee_4	HR	Recruiter	2010-07-09	City, UK	
4	5	Employee_5	Finance	Analyst	2017-11-28	City, USA	

In [32]:

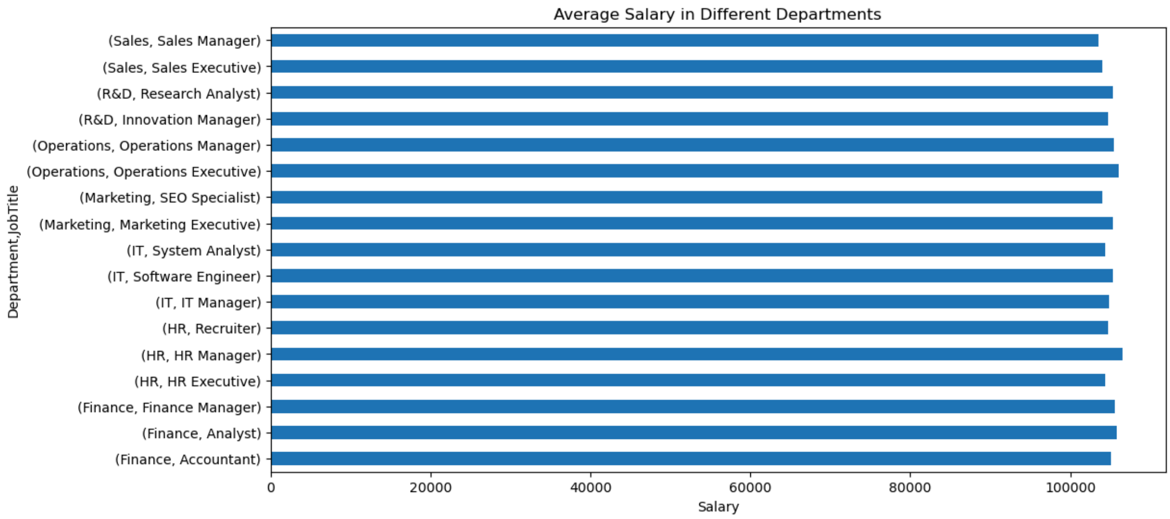
```
dep= df.groupby(['Department','JobTitle'])['Salary'].mean()
dep.apply(lambda x: f"{x: ,.2f}")
```

Out[32]:

```
Department  JobTitle
Finance     Accountant                 105,093.94
            Analyst                    105,771.09
            Finance Manager            105,585.84
HR          HR Executive               104,415.75
            HR Manager                 106,593.45
            Recruiter                  104,687.60
IT          IT Manager                 104,879.44
            Software Engineer          105,366.90
            System Analyst             104,354.43
Marketing   Marketing Executive        105,405.42
            SEO Specialist             104,034.88
Operations  Operations Executive       106,119.88
            Operations Manager         105,425.46
R&D         Innovation Manager         104,714.04
            Research Analyst           105,371.07
Sales       Sales Executive            104,037.97
            Sales Manager              103,574.85
Name: Salary, dtype: object
```

In [33]:

```
dep.plot(kind= 'barh', figsize = (12,6))
plt.title("Average Salary in Different Departments")
plt.xlabel("Salary")
plt.show()
```



Q.7) How many employees resigned & terminated in each department?

In [34]:
```
df.head(2)
```

Out[34]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Performan
0	1	Employee_1	R&D	Research Analyst	2012-05-10	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	

In [35]:
```
df.Status.unique()
```

Out[35]:
```
array(['Active', 'Resigned', 'Terminated', 'Retired'], dtype=object)
```

In [36]:
```
df_resigned = df[df['Status'] == 'Resigned']
df_resigned
```

Out[36]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Perfc
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	
15	16	Employee_16	R&D	Innovation Manager	2019-01-28	City, Germany	
16	17	Employee_17	Sales	Sales Manager	2019-12-07	City, Germany	
22	23	Employee_23	R&D	Innovation Manager	2018-11-30	City, UK	
25	26	Employee_26	R&D	Research Analyst	2017-12-29	City, India	
...	...	...	...	...	...	...	
29985	29986	Employee_29986	Sales	Sales Executive	2011-02-05	City, Germany	
29987	29988	Employee_29988	Marketing	SEO Specialist	2024-04-23	City, Germany	
29990	29991	Employee_29991	HR	Recruiter	2010-01-07	City, India	
29993	29994	Employee_29994	Sales	Sales Manager	2019-08-21	City, India	
29996	29997	Employee_29997	IT	IT Manager	2015-04-09	City, India	

6092 rows × 11 columns

In [37]:

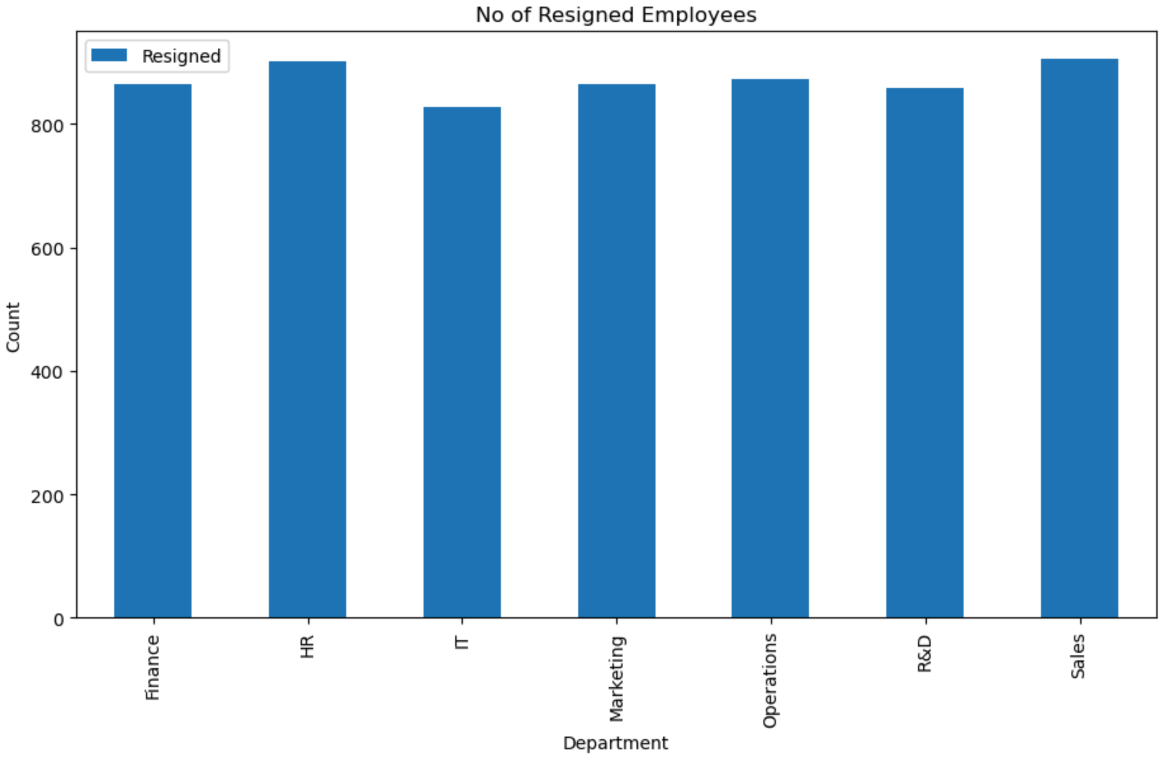
```
r_emp = df_resigned.groupby('Department')['Status'].count()
r_emp
```

Out[37]:

```
Department
Finance       865
HR            901
IT            827
Marketing     864
Operations    872
R&D           858
Sales         905
Name: Status, dtype: int64
```

In [38]:

```
plt.figure(figsize=(11,6))
r_emp.plot(x= r_emp.index, y = r_emp.values, kind = 'bar', legend = True , label
plt.title("No of Resigned Employees"), plt.ylabel("Count")
plt.show()
```



In [39]:
```
df_terminated = df[df['Status'] == 'Terminated']
df_terminated
```

Out[39]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Perf
3	4	Employee_4	HR	Recruiter	2010-07-09	City, UK	
4	5	Employee_5	Finance	Analyst	2017-11-28	City, USA	
40	41	Employee_41	R&D	Research Analyst	2021-05-09	City, India	
55	56	Employee_56	Marketing	Marketing Executive	2011-02-17	City, Germany	
67	68	Employee_68	IT	Software Engineer	2023-10-09	City, Canada	
...	...	...	...	...	...	...	
29845	29846	Employee_29846	Operations	Operations Manager	2015-06-18	City, India	
29855	29856	Employee_29856	Sales	Sales Manager	2014-01-27	City, India	
29940	29941	Employee_29941	R&D	Innovation Manager	2017-07-07	City, Germany	
29973	29974	Employee_29974	Sales	Sales Executive	2022-01-22	City, India	
29994	29995	Employee_29995	IT	Software Engineer	2012-06-17	City, UK	

1447 rows × 11 columns

In [40]:

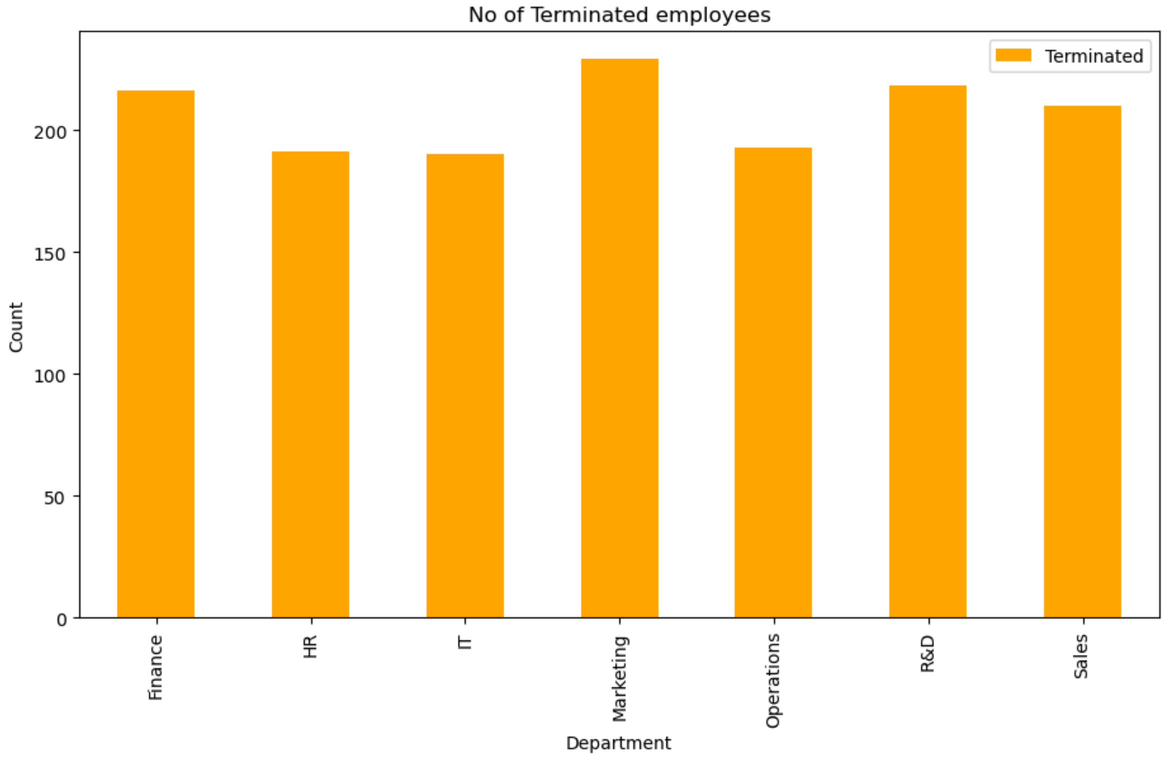
```
t_emp = df_terminated.groupby('Department')['Status'].count()
t_emp
```

Out[40]:

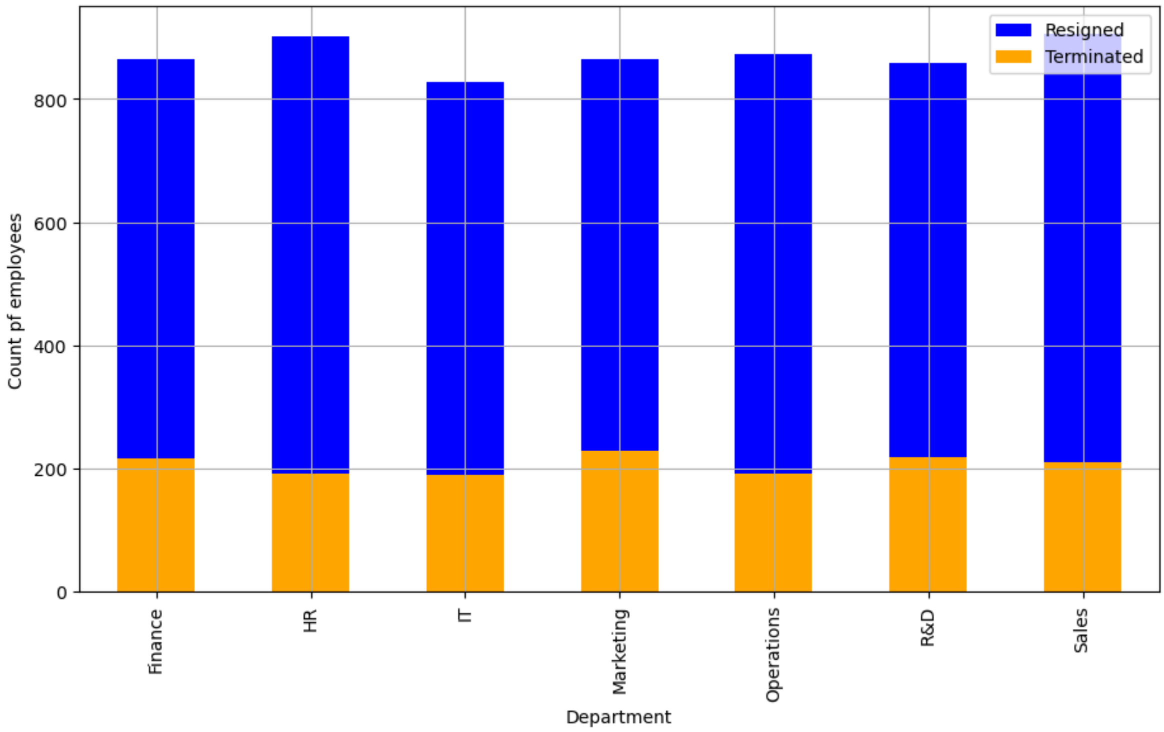
```
Department
Finance       216
HR            191
IT            190
Marketing     229
Operations    193
R&D           218
Sales         210
Name: Status, dtype: int64
```

In [41]:

```
plt.figure(figsize=(11,6))
t_emp.plot(x = t_emp.index, y = t_emp.values, kind = 'bar', color= 'orange', leg
plt.title("No of Terminated employees"), plt.ylabel("Count")
plt.show()
```



In [42]:
```
plt.figure(figsize=(11,6))
r_emp.plot(x= r_emp.index, y = r_emp.values, kind = 'bar',color = 'blue', legend
t_emp.plot(x = t_emp.index, y = t_emp.values, kind = 'bar', color= 'orange', leg
plt.ylabel('Count pf employees')
plt.grid()
```



Q.8) How does the salay vary with years of experience?

In [43]:
```
df.head()
```

Out[43]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Performanc
0	1	Employee_1	R&D	Research Analyst	2012-05-10	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	
2	3	Employee_3	Sales	Sales Executive	2013-03-30	City, Germany	
3	4	Employee_4	HR	Recruiter	2010-07-09	City, UK	
4	5	Employee_5	Finance	Analyst	2017-11-28	City, USA	

In [44]:

```
df['ExperienceYears'].nunique()
```

Out[44]:

```
16
```

In [45]:

```
df.groupby('ExperienceYears')['Salary'].mean()/1000
```

Out[45]:

```
ExperienceYears
0     105.619209
1     104.792950
2     103.889826
3     105.517338
4     104.422228
5     104.914294
6     105.600741
7     103.841741
8     105.496129
9     104.827534
10    106.178953
11    104.066749
12    104.602680
13    105.407892
14    105.116777
15    105.534246
Name: Salary, dtype: float64
```

Q.9) What is the average performance ratiing by department?

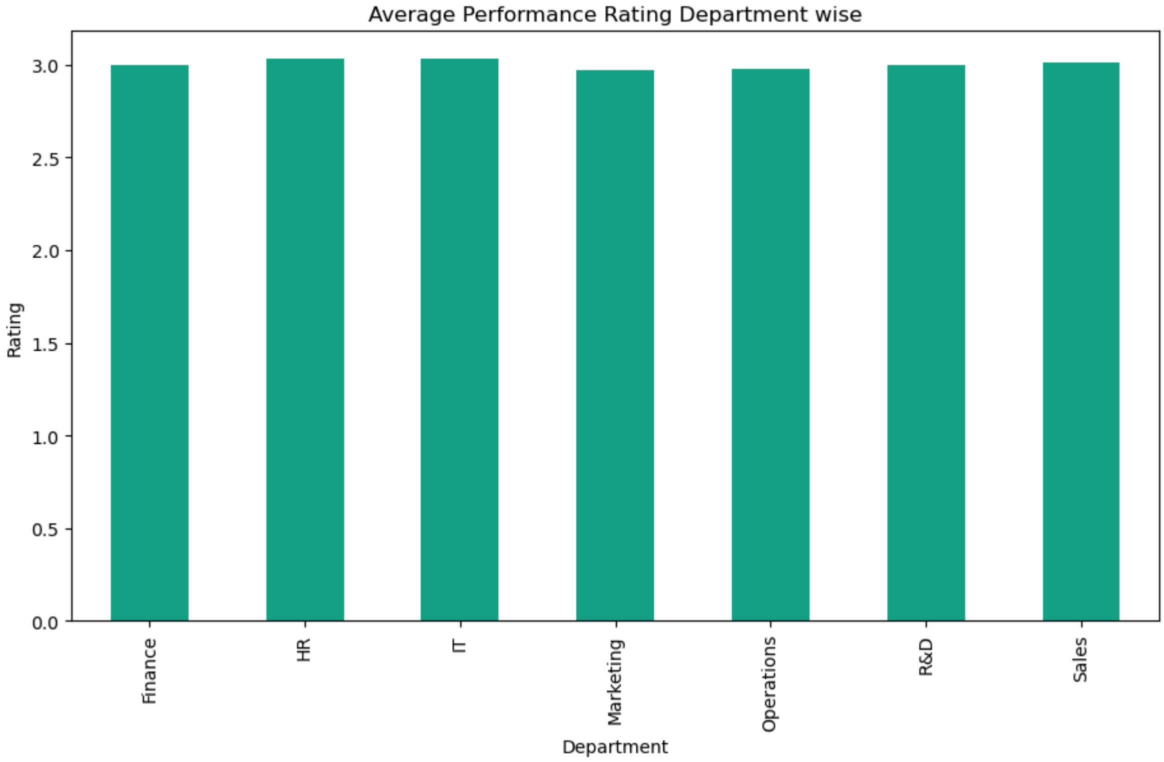
In [46]:

```
pr = df.groupby('Department')['PerformanceRating'].mean().round(2)
pr
```

Out[46]:

```
Department
Finance       3.00
HR            3.03
IT            3.03
Marketing     2.97
Operations    2.98
R&D           3.00
Sales         3.01
Name: PerformanceRating, dtype: float64
```

In [47]:
```
plt.figure(figsize=(11,6))
pr.plot(x= pr.index, y = pr.values, kind = 'bar', color= '#16A085')
plt.title("Average Performance Rating Department wise"), plt.ylabel("Rating")
plt.show()
```



Q.10) Which Country has the highest concentration of employees?

In [48]:
```
df['Country']= df['Location'].apply(lambda x: str(x.split(',')[1]))
```

In [49]:
```
df.head()
```

Out[49]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Location	Performanc
0	1	Employee_1	R&D	Research Analyst	2012-05-10	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	City, UK	
2	3	Employee_3	Sales	Sales Executive	2013-03-30	City, Germany	
3	4	Employee_4	HR	Recruiter	2010-07-09	City, UK	
4	5	Employee_5	Finance	Analyst	2017-11-28	City, USA	

In [50]:
```
df.Country.nunique()
```

Out[50]: 5

In [51]:
```
df.Country.value_counts()
```

Out[51]:
```
Country
Canada     6091
India      6071
UK         6002
Germany    5994
USA        5842
Name: count, dtype: int64
```

Q.11) Is there any correlation beetween performance Rating and Salary?

In [52]:
```
corr_df = df[['PerformanceRating', 'ExperienceYears', 'Salary']].corr()
corr_df
```

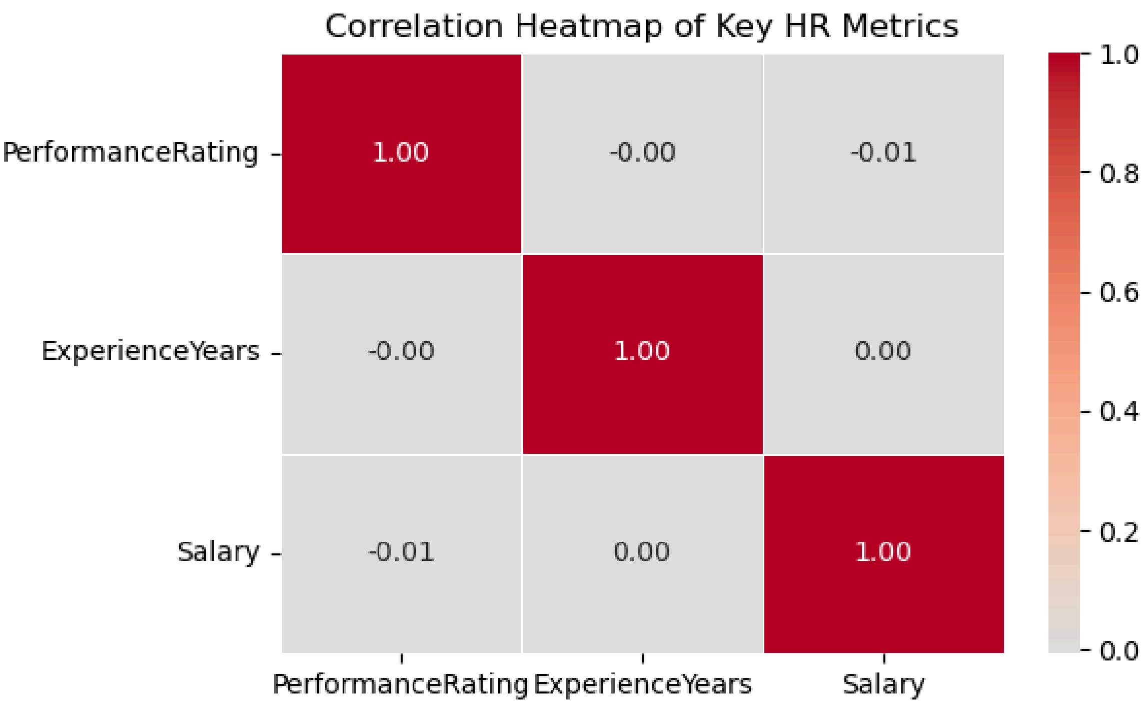
Out[52]:

	PerformanceRating	ExperienceYears	Salary
PerformanceRating	1.000000	-0.001251	-0.006667
ExperienceYears	-0.001251	1.000000	0.002221
Salary	-0.006667	0.002221	1.000000

In [53]:
```
#showing Coorelation with heatmap
plt.figure(figsize=(6, 4))

sns.heatmap(
    corr_df,
    annot=True,          # show numbers
    fmt=".2f",           # 2 decimals
    cmap="coolwarm",     # diverging palette
    center=0,            # correct correlation center
    linewidths=0.5
)

plt.title("Correlation Heatmap of Key HR Metrics")
plt.show()
```



Q.12) How has the number of hires changed over time(per year)?

In [54]:
```
df.HireDate.dtype
```

Out[54]:
```
dtype('<M8[ns]')
```

In [55]:
```
df.insert(5, 'Year', df['HireDate'].dt.year)
```

In [56]:
```
df
```

Out[56]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Year	Locatior
0	1	Employee_1	R&D	Research Analyst	2012-05-10	2012	City, Uk
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	2017	City, Uk
2	3	Employee_3	Sales	Sales Executive	2013-03-30	2013	City Germany
3	4	Employee_4	HR	Recruiter	2010-07-09	2010	City, Uk
4	5	Employee_5	Finance	Analyst	2017-11-28	2017	City USA
...	...	...	...	...	...	...	..
29995	29996	Employee_29996	Sales	Sales Executive	2018-09-22	2018	City Germany
29996	29997	Employee_29997	IT	IT Manager	2015-04-09	2015	City India
29997	29998	Employee_29998	Finance	Accountant	2016-08-18	2016	City Germany
29998	29999	Employee_29999	Sales	Sales Manager	2010-05-12	2010	City, Uk
29999	30000	Employee_30000	HR	HR Executive	2024-08-05	2024	City, Uk

30000 rows × 13 columns

In [57]:
```
df.Year.unique()
```

Out[57]:
```
array([2012, 2017, 2013, 2010, 2014, 2023, 2020, 2022, 2018, 2024, 2021,
       2019, 2025, 2015, 2011, 2016], dtype=int32)
```

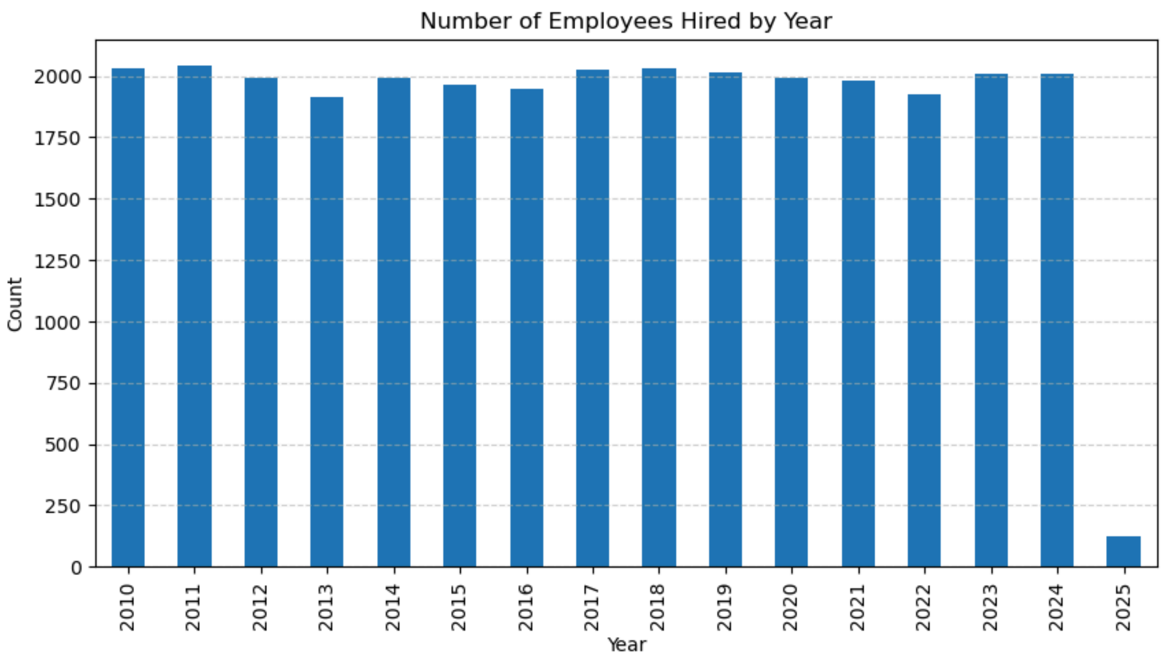
In [58]:
```
df.Year.nunique()
```

Out[58]:
```
16
```

In [59]:
```
hire = df.groupby('Year')['HireDate'].count().sort_index()
hire
```

Out[59]:
```
Year
2010    2031
2011    2045
2012    1991
2013    1917
2014    1991
2015    1967
2016    1947
2017    2028
2018    2033
2019    2013
2020    1991
2021    1979
2022    1926
2023    2009
2024    2008
2025     124
Name: HireDate, dtype: int64
```

In [60]:
```
plt.figure(figsize=(10, 5))
hire.plot(x=  hire.index, y = hire.values, kind = 'bar', color = '#1F77B4')
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.title("Number of Employees Hired by Year")
plt.ylabel("Count")
plt.show()
```



Q.13) Compare salaries of Remte vs on-site empolyees---Is there any significant difference?

In [61]:
```
df.head()
```

Out[61]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Year	Location	Perf
0	1	Employee_1	R&D	Research Analyst	2012-05-10	2012	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	2017	City, UK	
2	3	Employee_3	Sales	Sales Executive	2013-03-30	2013	City, Germany	
3	4	Employee_4	HR	Recruiter	2010-07-09	2010	City, UK	
4	5	Employee_5	Finance	Analyst	2017-11-28	2017	City, USA	

In [62]:
```
df.groupby('WorkMode')['Salary'].mean().round(2)
```

Out[62]:
```
WorkMode
Onsite     104811.32
Remote     105266.36
Name: Salary, dtype: float64
```

Q.14) Find the top 10 employees with highest salaries from each department.

In [63]:
```
top_10= df.groupby('Department')\
  .apply(lambda x: x.nlargest(10, "Salary"), include_groups=False)
```

In [64]:
```
top_10.head(40)
```

Out[64]:

Department		EmployeeID	EmployeeName	JobTitle	HireDate	Year	Locatior
Finance	**15033**	15034	Employee_15034	Accountant	2010-12-26	2010	City Canada
	3813	3814	Employee_3814	Finance Manager	2023-07-27	2023	City Canada
	16565	16566	Employee_16566	Finance Manager	2022-08-16	2022	City USA
	20504	20505	Employee_20505	Accountant	2017-01-10	2017	City India
	19149	19150	Employee_19150	Accountant	2011-04-21	2011	City Canada
	16053	16054	Employee_16054	Analyst	2010-10-11	2010	City USA
	20228	20229	Employee_20229	Accountant	2024-08-07	2024	City India
	299	300	Employee_300	Analyst	2012-08-01	2012	City Canada
	1594	1595	Employee_1595	Analyst	2024-11-15	2024	City, UK
	1447	1448	Employee_1448	Accountant	2024-02-07	2024	City Germany
HR	**20410**	20411	Employee_20411	HR Manager	2023-06-28	2023	City Germany
	17484	17485	Employee_17485	Recruiter	2018-09-13	2018	City USA
	12758	12759	Employee_12759	HR Executive	2016-01-25	2016	City Canada
	25869	25870	Employee_25870	HR Manager	2019-02-18	2019	City India
	6796	6797	Employee_6797	HR Manager	2010-05-24	2010	City, UK
	14647	14648	Employee_14648	Recruiter	2020-03-27	2020	City India
	26948	26949	Employee_26949	HR Executive	2022-10-08	2022	City Germany
	20116	20117	Employee_20117	Recruiter	2020-10-10	2020	City India
	28122	28123	Employee_28123	Recruiter	2022-05-24	2022	City USA
	4641	4642	Employee_4642	Recruiter	2017-03-17	2017	City Canada

		EmployeeID	EmployeeName	JobTitle	HireDate	Year	Locatior
Department							
IT	**24009**	24010	Employee_24010	System Analyst	2019-09-25	2019	City Germany
	317	318	Employee_318	IT Manager	2011-10-08	2011	City, Uk
	2320	2321	Employee_2321	Software Engineer	2019-12-28	2019	City USA
	3012	3013	Employee_3013	IT Manager	2024-11-22	2024	City Canada
	28987	28988	Employee_28988	System Analyst	2013-09-05	2013	City India
	27709	27710	Employee_27710	System Analyst	2013-05-31	2013	City, Uk
	20636	20637	Employee_20637	IT Manager	2021-03-29	2021	City, Uk
	7362	7363	Employee_7363	IT Manager	2022-10-23	2022	City, Uk
	991	992	Employee_992	Software Engineer	2017-07-10	2017	City Canada
	3002	3003	Employee_3003	Software Engineer	2017-01-24	2017	City USA
Marketing	**25440**	25441	Employee_25441	SEO Specialist	2013-03-30	2013	City Germany
	3873	3874	Employee_3874	SEO Specialist	2023-08-15	2023	City USA
	6702	6703	Employee_6703	SEO Specialist	2011-11-03	2011	City, Uk
	26009	26010	Employee_26010	Marketing Executive	2016-02-13	2016	City Germany
	23704	23705	Employee_23705	Marketing Executive	2013-06-06	2013	City India
	25748	25749	Employee_25749	SEO Specialist	2012-06-30	2012	City USA
	5294	5295	Employee_5295	SEO Specialist	2011-01-12	2011	City India
	11758	11759	Employee_11759	Marketing Executive	2013-06-26	2013	City USA
	645	646	Employee_646	Marketing Executive	2015-02-13	2015	City Germany
	2976	2977	Employee_2977	SEO Specialist	2016-11-08	2016	City, Uk

In [65]:
```
top_10.tail(30)
```

Out[65]:

		EmployeeID	EmployeeName	JobTitle	HireDate	Year	Location
Department							
Operations	**6028**	6029	Employee_6029	Operations Manager	2010-09-20	2010	City, Canada
	26660	26661	Employee_26661	Operations Executive	2011-06-17	2011	City, UK
	27139	27140	Employee_27140	Operations Executive	2014-03-12	2014	City, India
	20141	20142	Employee_20142	Operations Manager	2016-12-14	2016	City, India
	17784	17785	Employee_17785	Operations Executive	2011-08-23	2011	City, Germany
	6739	6740	Employee_6740	Operations Executive	2017-12-25	2017	City, UK
	9878	9879	Employee_9879	Operations Manager	2022-05-27	2022	City, India
	10439	10440	Employee_10440	Operations Manager	2019-01-01	2019	City, India
	13968	13969	Employee_13969	Operations Executive	2022-08-09	2022	City, USA
	13221	13222	Employee_13222	Operations Manager	2016-10-01	2016	City, Germany
R&D	**28033**	28034	Employee_28034	Innovation Manager	2021-03-21	2021	City, UK
	26920	26921	Employee_26921	Research Analyst	2022-07-23	2022	City, USA
	4642	4643	Employee_4643	Innovation Manager	2019-08-22	2019	City, India
	1617	1618	Employee_1618	Research Analyst	2015-02-09	2015	City, Germany
	5262	5263	Employee_5263	Innovation Manager	2012-03-21	2012	City, Canada
	24494	24495	Employee_24495	Research Analyst	2020-04-19	2020	City, Germany
	11123	11124	Employee_11124	Research Analyst	2020-08-22	2020	City, Germany
	27565	27566	Employee_27566	Research Analyst	2011-10-19	2011	City, India
	28695	28696	Employee_28696	Innovation Manager	2018-09-06	2018	City, USA
	17917	17918	Employee_17918	Innovation Manager	2010-02-04	2010	City, Canada

		EmployeeID	EmployeeName	JobTitle	HireDate	Year	Location
Department							
Sales	**22759**	22760	Employee_22760	Sales Manager	2012-11-06	2012	City, Germany
	28513	28514	Employee_28514	Sales Manager	2015-02-15	2015	City, UK
	18384	18385	Employee_18385	Sales Manager	2022-08-14	2022	City, USA
	16471	16472	Employee_16472	Sales Manager	2016-04-30	2016	City, India
	1813	1814	Employee_1814	Sales Manager	2020-06-12	2020	City, USA
	11526	11527	Employee_11527	Sales Executive	2023-03-22	2023	City, Germany
	2833	2834	Employee_2834	Sales Manager	2024-06-30	2024	City, Germany
	5937	5938	Employee_5938	Sales Executive	2020-01-24	2020	City, USA
	17445	17446	Employee_17446	Sales Manager	2013-09-02	2013	City, India
	15383	15384	Employee_15384	Sales Executive	2020-05-02	2020	City, UK

Q.15) Identify the Department with the highest attrition rate(Resigned %).

In [66]:
```
df.head(2)
```

Out[66]:

	EmployeeID	EmployeeName	Department	JobTitle	HireDate	Year	Location	Perf
0	1	Employee_1	R&D	Research Analyst	2012-05-10	2012	City, UK	
1	2	Employee_2	Marketing	SEO Specialist	2017-12-29	2017	City, UK	

In [67]:
```
dept_counts= df.groupby('Department')['Status'].agg(total_emp= 'count' , resigne
dept_counts
```

Out[67]:

Department	total_emp	resigned
Finance	4242	865
HR	4349	901
IT	4215	827
Marketing	4279	864
Operations	4305	872
R&D	4319	858
Sales	4291	905

In [68]:
```
type(dept_counts)
```

Out[68]: pandas.core.frame.DataFrame

In [69]:
```
# Calculate attrition rate
dept_counts['attrition_rate_%']= (dept_counts['resigned'] / dept_counts['total_e
```

In [70]:
```
dept_counts
```

Out[70]:

Department	total_emp	resigned	attrition_rate_%
Finance	4242	865	20.391325
HR	4349	901	20.717406
IT	4215	827	19.620403
Marketing	4279	864	20.191634
Operations	4305	872	20.255517
R&D	4319	858	19.865710
Sales	4291	905	21.090655

In [71]:
```
dept_counts.sort_values('attrition_rate_%',ascending = False)
```

Out[71]:

Department	total_emp	resigned	attrition_rate_%
Sales	4291	905	21.090655
HR	4349	901	20.717406
Finance	4242	865	20.391325
Operations	4305	872	20.255517
Marketing	4279	864	20.191634
R&D	4319	858	19.865710
IT	4215	827	19.620403

Key Insights

- Most employees in the organization are currently active, indicating workforce stability.
- Certain departments have significantly higher employee counts, showing their importance in business operations.
- Salary distribution varies across departments, reflecting differences in role complexity and seniority.
- Remote employees show comparable salary trends to onsite employees, indicating effective remote work adoption.
- Some departments experience higher attrition, which may point to workload or satisfaction challenges.
- Hiring trends reveal growth in specific years, suggesting expansion phases within the organization.

Business Recommendations

- HR should focus retention efforts on departments with higher attrition by improving employee engagement and career growth opportunities.
- Compensation structures should be reviewed periodically to ensure salaries remain competitive across departments.
- The organization can continue or expand remote work policies, as remote employees show stable salary and employment patterns.
- Workforce planning should account for historical hiring trends to better manage future growth phases.
- Additional employee satisfaction and performance data should be collected to gain deeper insights into resignation patterns.

Conclusion & Future Scope

This HR data analysis project successfully explored key workforce metrics using Python, pandas, and data visualization libraries. The insights generated from this analysis can support HR decision-making related to employee retention, compensation planning, and workforce optimization.

Future Enhancements:

- Build predictive models to forecast employee attrition
- Integrate employee performance and satisfaction data
- Create interactive dashboards using Power BI or Tableau
- Automate HR reporting workflows