eda project 2

June 22, 2024

[26]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

[27]:
```
df= pd.read_csv("Salary_dataset.csv")
df.head()
```

[27]:
```
   Unnamed: 0  ID  Age  Gender   Department  Years_of_Experience  \
0           0   1   60  Female    Marketing                  4.0
1           1   2   50  Female  Engineering                 25.0
2           2   3   36  Female      Finance                 14.0
3           3   4   64  Female      Finance                  9.0
4           4   5   29    Male  Engineering                 26.0

  Education_Level    Salary  Bonus  Performance_Score Region
0       Bachelors  114065.0   6514                  5   East
1             PhD   49268.0   8432                  3   East
2       Bachelors   52185.0   6474                  8  North
3       Bachelors  103704.0   7892                  6   East
4       Bachelors   79099.0   5561                  3   East
```

Data Cleaning tasks:

[28]:
```
df.dtypes
```

[28]:
```
Unnamed: 0               int64
ID                       int64
Age                      int64
Gender                  object
Department              object
Years_of_Experience    float64
Education_Level         object
Salary                 float64
Bonus                    int64
Performance_Score        int64
Region                  object
dtype: object
```

[29]:
```
df.drop(columns = "Unnamed: 0", inplace= True)
```

[30]:
```
df.isnull().sum()
```

[30]:
```
ID                     0
Age                    0
Gender                 0
Department             0
Years_of_Experience    1
Education_Level        0
Salary                 1
Bonus                  0
Performance_Score      0
Region                 0
dtype: int64
```

[31]:
```
df[df.duplicated()]
```

[31]:
```
Empty DataFrame
Columns: [ID, Age, Gender, Department, Years_of_Experience, Education_Level,
Salary, Bonus, Performance_Score, Region]
Index: []
```

[32]:
```
df["Years_of_Experience"] = df["Years_of_Experience"].
 ↪fillna(df["Years_of_Experience"].mode()[0])
```

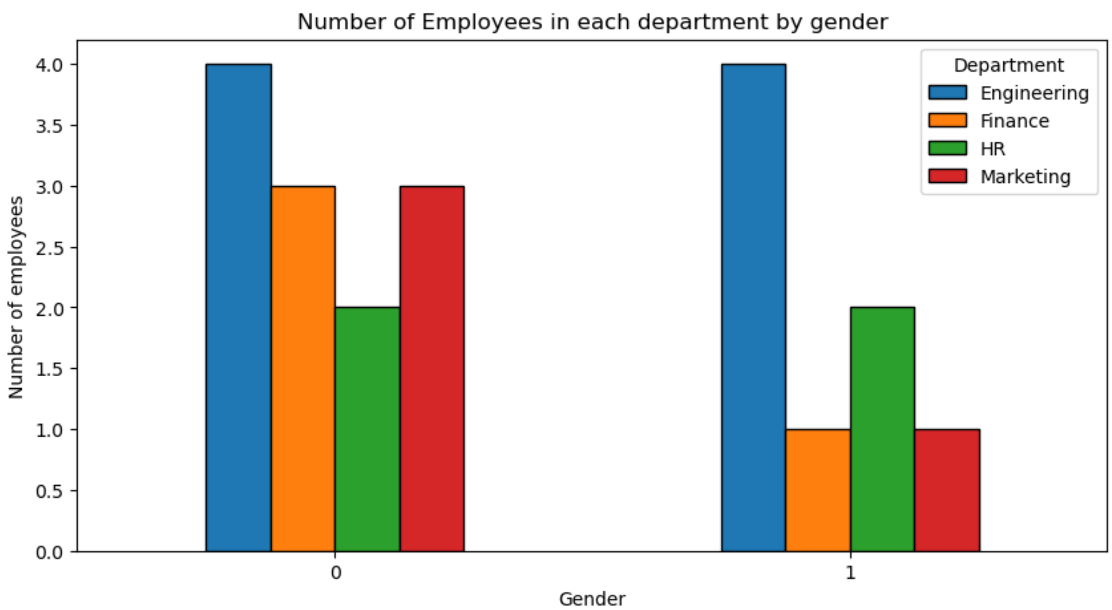
[33]:
```
df["Salary"] = df["Salary"].fillna(df["Salary"].mode()[0])
```

[34]:
```
# Convert the 'Gender' column to a numerical format (e.g., Male=1, Female=0).
df["Gender"].replace({"Male":1,"Female":0},inplace=True)
df["Gender"]=df["Gender"].astype(int)
df.head()
```

[34]:
```
   ID  Age  Gender   Department  Years_of_Experience Education_Level  \
0   1   60       0    Marketing                  4.0       Bachelors
1   2   50       0  Engineering                 25.0             PhD
2   3   36       0      Finance                 14.0       Bachelors
3   4   64       0      Finance                  9.0       Bachelors
4   5   29       1  Engineering                 26.0       Bachelors

     Salary  Bonus  Performance_Score Region
0  114065.0   6514                  5   East
1   49268.0   8432                  3   East
2   52185.0   6474                  8  North
3  103704.0   7892                  6   East
4   79099.0   5561                  3   East
```

1 Matplotlib

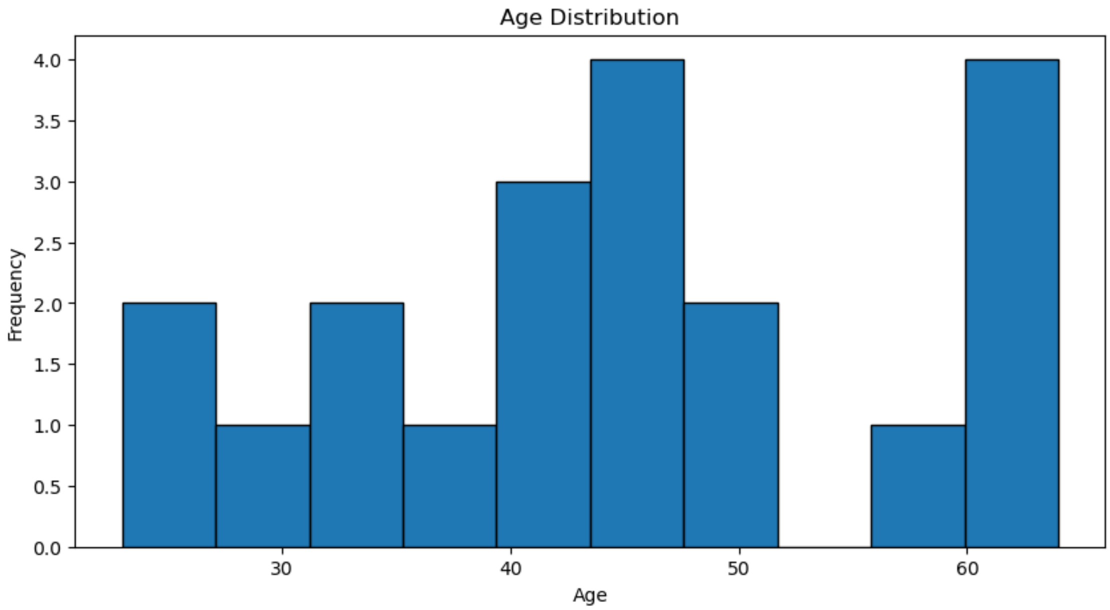
[35]:
```
#Create a bar chart to show the number of employees in each department by␣
↪gender.

emp = df.groupby(['Gender', 'Department']).size().unstack()
emp.plot(kind="bar",figsize=(10, 5),edgecolor="black")
plt.title("Number of Employees in each department by gender")
plt.xlabel("Gender")
plt.ylabel("Number of employees")
plt.xticks(rotation=0)
plt.legend(title="Department")
plt.show()
```



Insight:

Some departments have a more balanced gender distribution, while others might be predominantly male or female. This insight can be useful for diversity and inclusion initiatives and to identify areas where gender balance can be improved.

[36]:
```
#Ploting a histogram to show the distribution of ages in the dataset
plt.figure(figsize=(10, 5))
plt.hist(df["Age"],bins=10,edgecolor="black")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.title("Age Distribution")
plt.show()
```



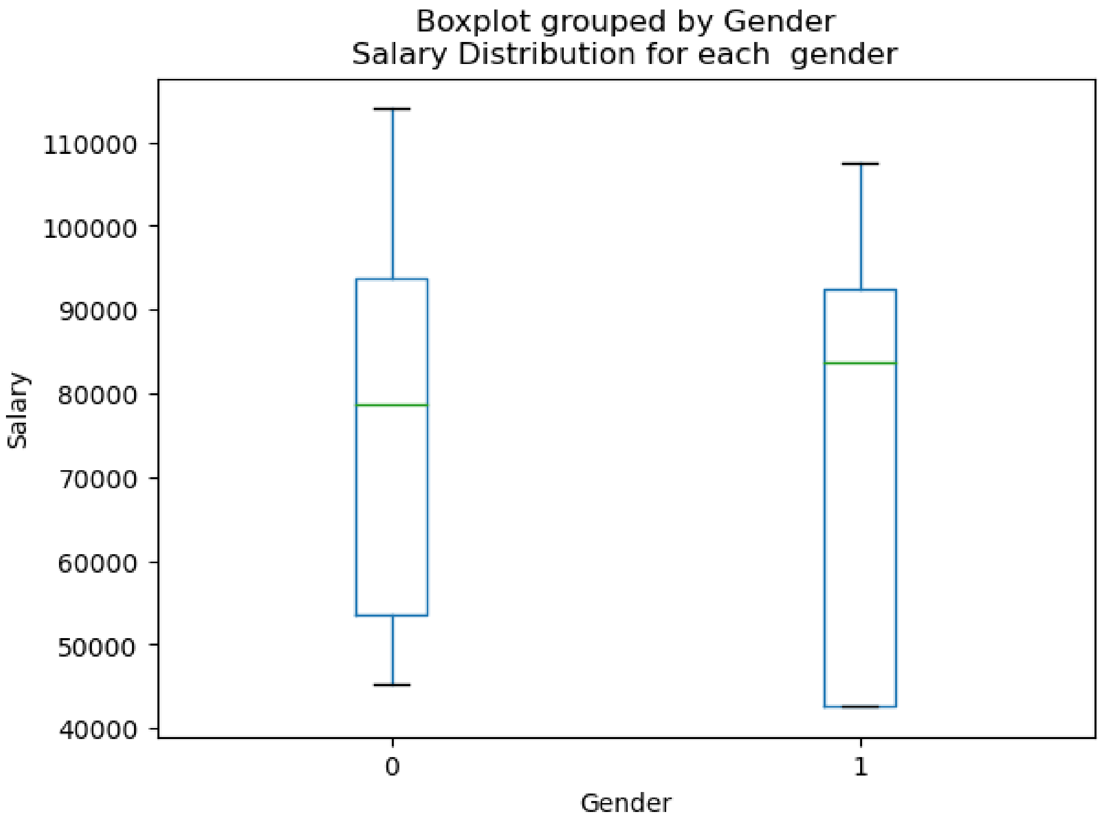
Insight:

The age distribution shows that the most frequent age group is around 40-50 years, indicating that a significant portion of employees are likely in their mid-career stage. There is a notable spread in ages, with fewer employees in the younger (20-30 years) and older (60+ years) age brackets, suggesting a workforce that is predominantly composed of middle-aged employees.

[37]:

```
#Creating a box plot to visualize the salary distribution for each gender.
plt.figure(figsize=(10,5))
df.boxplot(column='Salary', by='Gender', grid=False)
plt.xlabel("Gender")
plt.ylabel("Salary")
plt.title("Salary Distribution for each  gender")
plt.show()
```

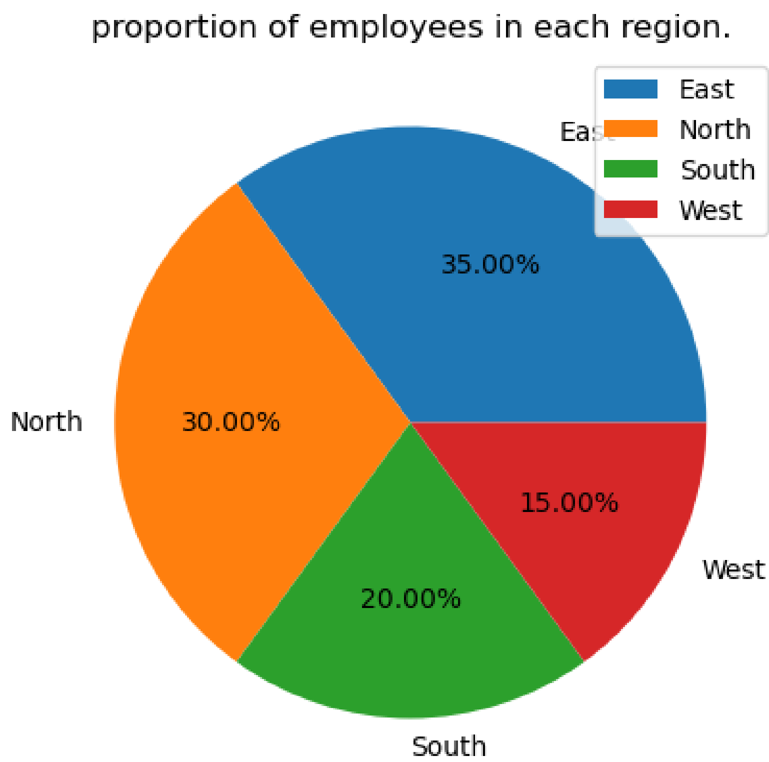
```
<Figure size 1000x500 with 0 Axes>
```



Insight:

The median salary for both genders appears similar, suggesting that there might not be a significant gender pay gap in terms of median salary. The range of salaries for males seems slightly wider compared to females, indicating more variability in male salaries. There may be outliers present, particularly in the higher salary range for both genders.

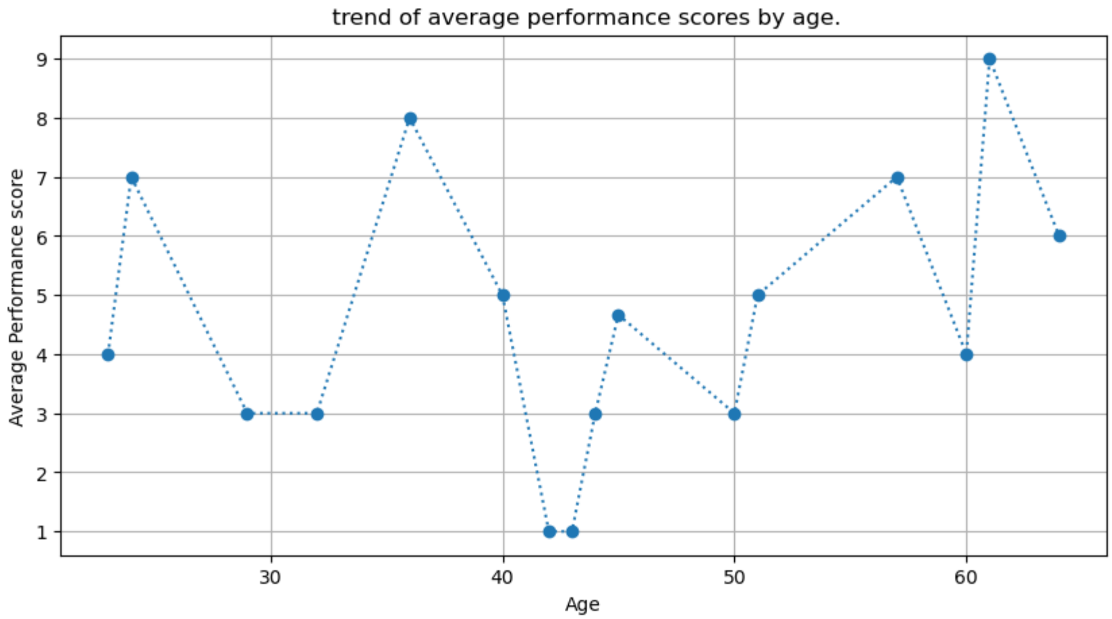
[38]:
```
#Ploting a pie chart to show the proportion of employees in each region.

region= df["Region"].value_counts()
plt.figure(figsize=(12,5))
plt.pie(region,labels=region.index,autopct='%1.2f%%')
plt.title("proportion of employees in each region.")
plt.legend()
plt.show()
```



Insight:

The distribution of employees is fairly balanced among different regions, though one region might have a slightly higher proportion. This insight can help in regional resource allocation and workforce planning.

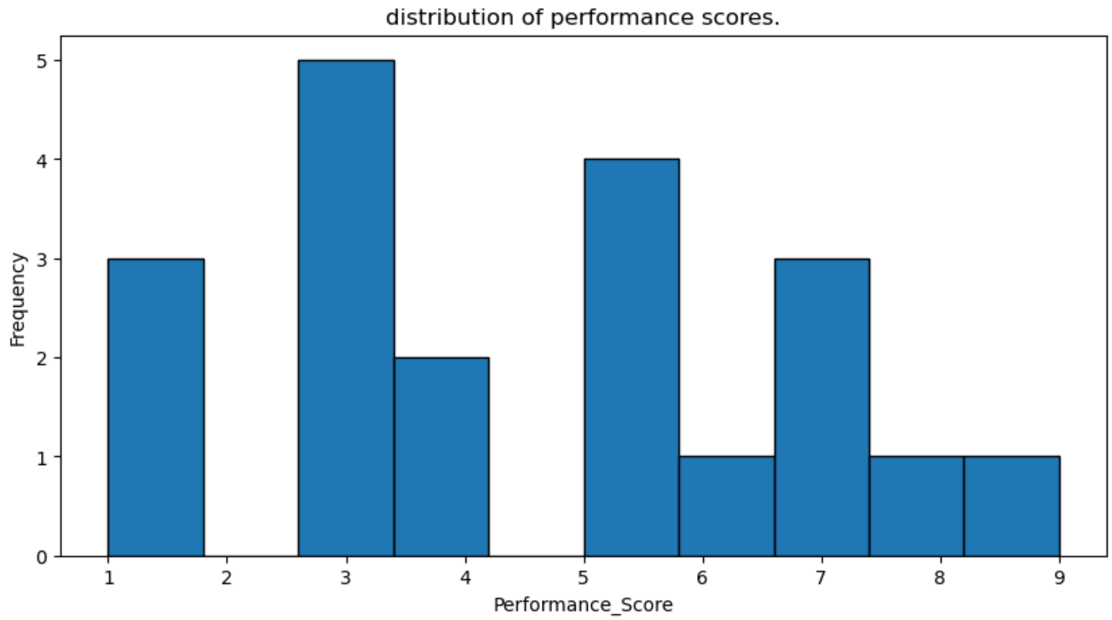
[39]:
```
#Creating a line plot to show the trend of average performance scores by age.
avg_score = df.groupby("Age")["Performance_Score"].mean()
plt.figure(figsize=(10,5))
plt.plot(avg_score.index,avg_score.values,marker="o" ,linestyle = 'dotted')
plt.title(" trend of average performance scores by age.")
plt.xlabel("Age")
plt.ylabel("Average Performance score")
plt.grid(True)
plt.show()
```



Insight:

The line plot reveals that the average performance score tends to increase with age up to a certain point, indicating that experience may contribute positively to performance. After reaching a peak, the average performance score stabilizes or slightly declines, suggesting that factors other than age, such as job role or tenure, might start to influence performance

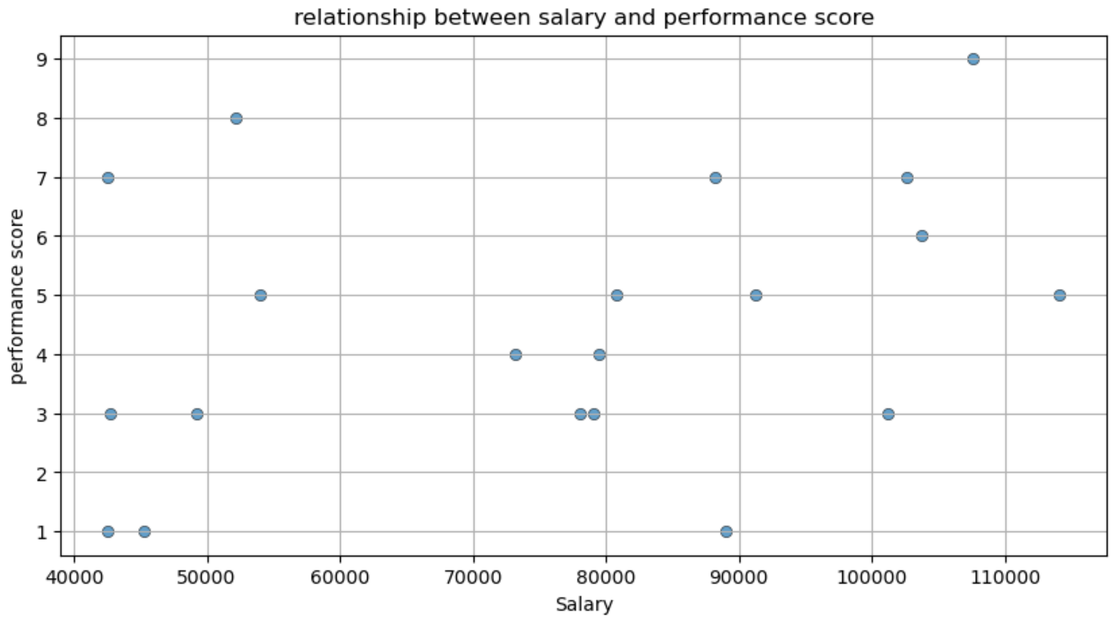
[40]:
```
#Create a histogram to show the distribution of performance scores.
plt.figure(figsize=(10,5))
plt.hist(df["Performance_Score"],bins=10,edgecolor="black")
plt.xlabel("Performance_Score")
plt.ylabel("Frequency")
plt.title("distribution of performance scores.")
plt.show()
```



Insight:

The histogram indicates that performance scores are widely distributed, with certain score ranges being more common, suggesting variability in employee performance. The presence of multiple peaks suggests that there might be distinct groups or clusters of performance levels within the organization.

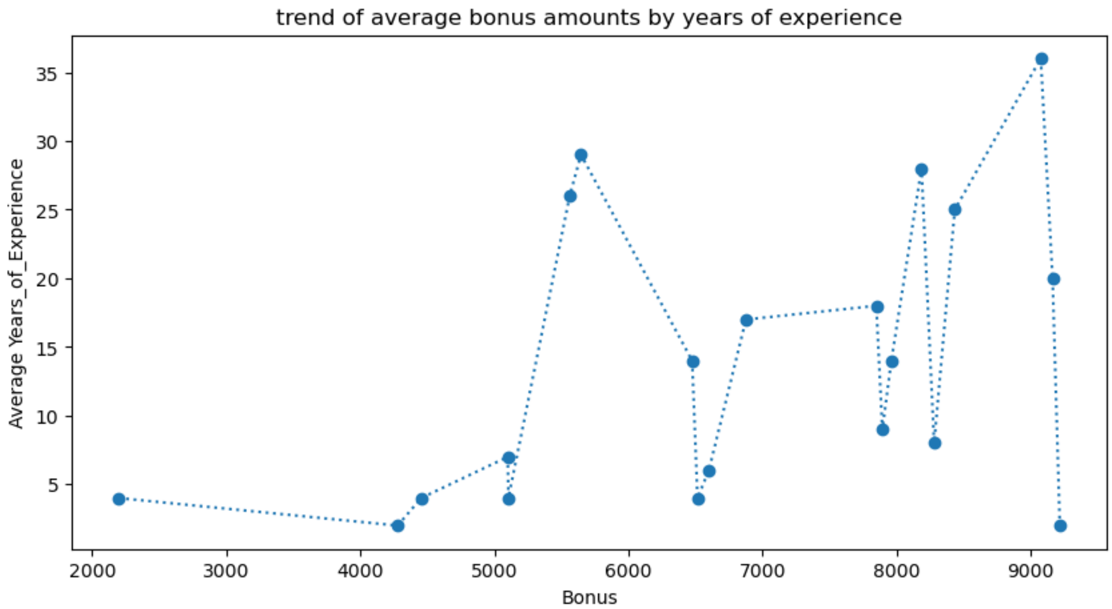
[47]:
```
# Plot a scatter plot to show the relationship between salary and performance␣
↪score.
plt.figure(figsize=(10,5))
plt.scatter(df["Salary"],df["Performance_Score"],alpha=0.
↪7,edgecolors="black",linewidths=0.5)
plt.title("relationship between salary and performance score")
plt.xlabel("Salary")
plt.ylabel("performance score")
plt.grid(True)
plt.show()
```



Insight:

The scatter plot can reveal whether there is a positive correlation between salary and performance score. If higher performance scores are generally associated with higher salaries, it indicates that the company rewards high performers with better pay. If there is no clear pattern, it might suggest that salary is not strongly linked to performance scores, which could be an area for HR to investigate further.v

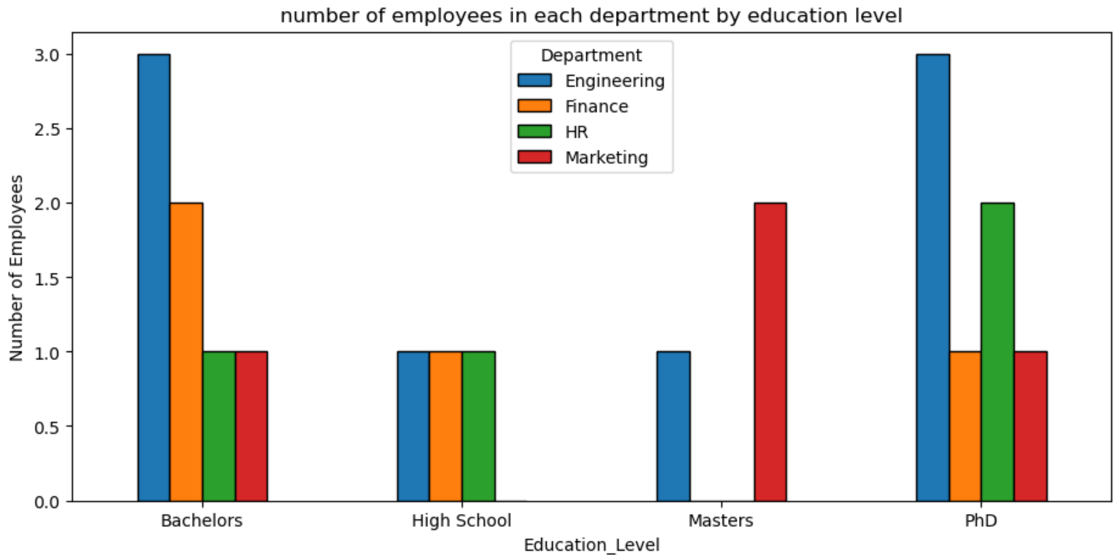
[43]:
```
#Create a line plot to show the trend of average bonus amounts by years of␣
↪experience.
avg_bonus = df.groupby("Bonus")["Years_of_Experience"].mean()
plt.figure(figsize=(10,5))
plt.plot(avg_bonus.index,avg_bonus.values,marker="o",linestyle=":")
plt.title("trend of average bonus amounts by years of experience")
plt.xlabel("Bonus")
plt.ylabel("Average Years_of_Experience")
plt.show()
```



Insight:

The trend line show an increase in performance scores with age up to a certain point, followed by a plateau or decline. This can indicate that experience contributes to better performance up to a certain age, after which other factors might influence performance.

[53]:
```
#Plot a bar chart to show the number of employees in each department by␣
↪education level.
emp=df.groupby(["Education_Level","Department"]).size().unstack()
emp.plot(kind="bar",figsize=(11,5),edgecolor="black")
plt.title("number of employees in each department by education level")
plt.xlabel("Education_Level")
plt.ylabel("Number of Employees")
plt.xticks(rotation=0)
plt.legend(title="Department")
plt.show()
```



Insight:

The bar chart reveals that certain departments have a higher concentration of employees with specific education levels, indicating potential educational requirements or preferences for those departments. Comparing the bars across different education levels shows that higher education levels e.g., Masters or Ph.D. are more prevalent in some departments, which could reflect the specialized skills needed for those roles.

[ ]: