Employee Salary Analysis

- **Problem Statement:** The goal of this analysis is to explore employee salary data, identify patterns, and generate insights based on job titles, departments, and other factors. This report includes data visualization and statistical summaries to better understand salary distribution. The analysis aims to provide actionable insights for HR managers and business leaders to ensure fair and competitive compensation structures.
- **Name:** [Aditya Kanaujiya]
- **Roll No.:** [202401100300015]
- **Course:** [Artificial Intelligence]
- **Institution:** [Kiet group of Institution]

Introduction:
Salary analysis is a crucial aspect of human resource management. It helps in understanding salary trends across various job roles and departments. In this study, we aim to analyze employee salaries, compare salary distributions, and visualize the data to derive meaningful conclusions.

The importance of salary analysis includes:

1. **Fair Compensation:** Ensuring that employees are paid fairly based on industry standards and job responsibilities.
2. **Retention & Motivation:** Competitive salaries help retain skilled employees and keep them motivated.
3. **Budget Planning:** Helps organizations allocate budgets effectively across different departments.

4. **Gender & Diversity Pay Gaps:** Identifies potential pay disparities and ensures inclusivity.

This analysis involves data preprocessing, statistical computations, and visual representations using Python. The study provides data-driven insights to help organizations optimize their compensation strategies.

Methodology:
The analysis follows a structured approach to examine the employee salary dataset and generate insights. The key steps involved are:

1. **Data Collection:** The dataset includes details such as employee job titles, department information, and salary figures. It is assumed to be extracted from an organization's HR database or external sources.
2. **Data Preprocessing:** Cleaning the dataset to remove missing values, handling incorrect or inconsistent entries, and ensuring uniform formatting for better analysis.
3. **Exploratory Data Analysis (EDA):** Performing descriptive statistical analysis to examine salary distributions and summarize key trends.
4. **Visualization Techniques:** Utilizing various graphical representations, such as:
   - **Box plots:** To compare salaries across job titles and departments.
   - **Bar charts:** To highlight average salary differences.
   - **Histograms:** To understand the overall distribution of salaries.
5. **Insights & Conclusion:** Analyzing observed trends to draw meaningful conclusions, such as:
   - Which department has the highest average salary?
   - Are there any job roles that show significant salary variations?
   - Identifying any anomalies or outliers in salary data.

Code:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns
```

```python
# Load dataset
file_path = "/content/employee_data.csv"  # Update the path if needed
df = pd.read_csv("/content/employee_data.csv")

# Display basic statistics
print("Dataset Overview:\n", df.head())
print("\nBasic Statistics:\n", df.describe())

# Check for missing values
print("\nMissing Values:\n", df.isnull().sum())

# Salary distribution plot
plt.figure(figsize=(10, 5))
sns.histplot(df["Salary"], bins=20, kde=True, color='blue')
plt.title("Salary Distribution")
plt.xlabel("Salary")
plt.ylabel("Frequency")
plt.show()
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
file_path = "/content/employee_data.csv "
df = pd.read_csv("/content/employee_data.csv")

# Print column names to check
```

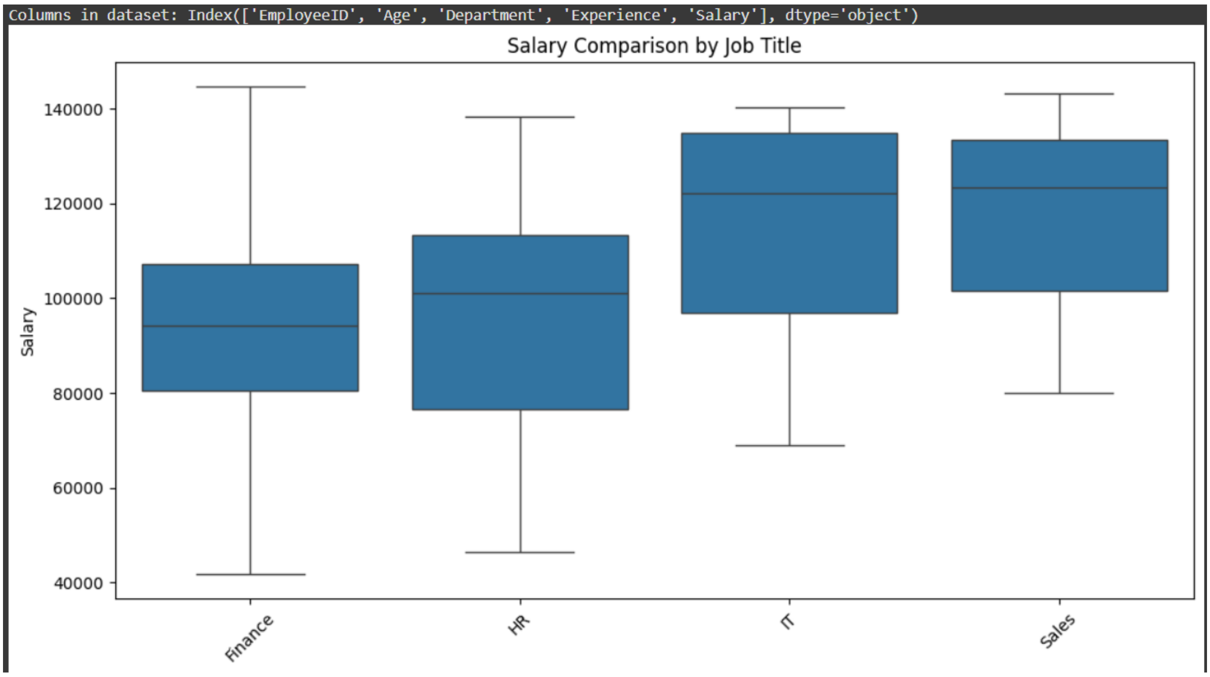
```python
print("Columns in dataset:", df.columns)

# Standardizing column names (remove spaces)
df.columns = df.columns.str.strip()

# Convert Salary to numeric (handling errors)
df['Salary'] = pd.to_numeric(df['Salary'], errors='coerce')

# Drop NaN values in Salary
df.dropna(subset=['Salary'], inplace=True)

# Box plot for salary comparison by job title
plt.figure(figsize=(12, 6))
sns.boxplot(x="Department", y="Salary", data=df)  # Ensure column name matches
plt.xticks(rotation=45)
plt.title("Salary Comparison by Job Title")
plt.show()

# Average salary by department
avg_salary_by_dept = df.groupby("Department")["Salary"].mean().sort_values()
print("\nAverage Salary by Department:\n", avg_salary_by_dept)

# Plot average salary by department
plt.figure(figsize=(12, 6))
avg_salary_by_dept.plot(kind='bar', color='green')
plt.title("Average Salary by Department")
plt.xlabel("Department")
plt.ylabel("Average Salary")
```

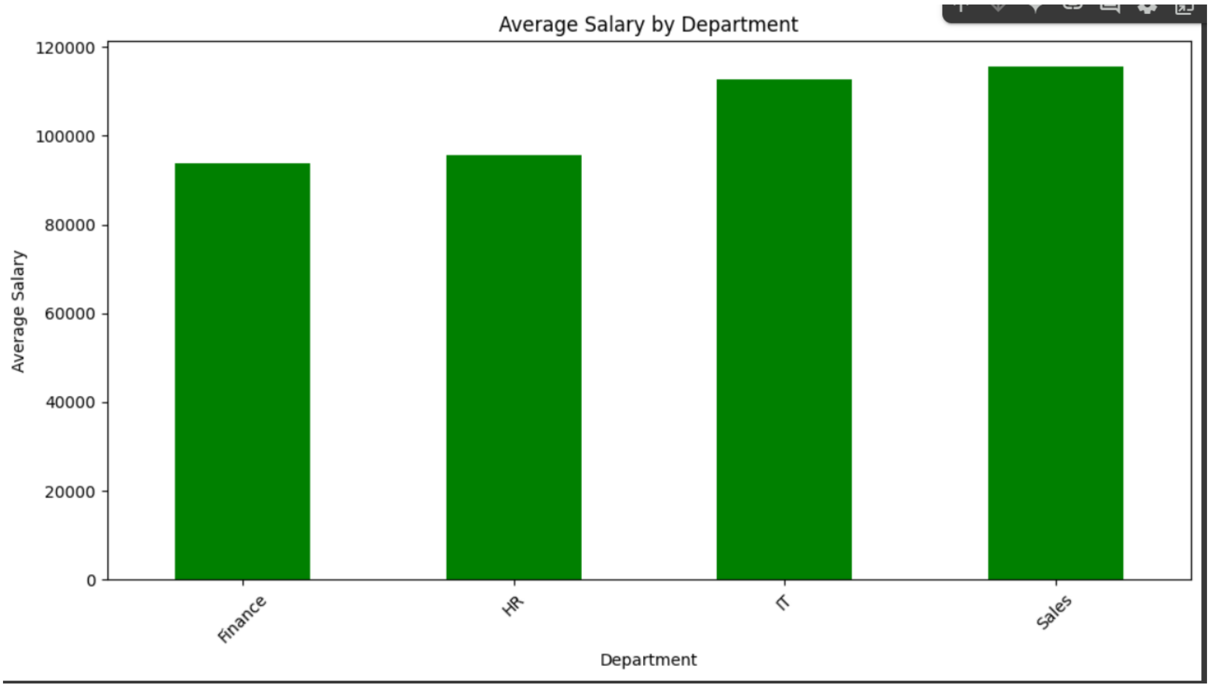
```python
plt.xticks(rotation=45)

plt.show()
```

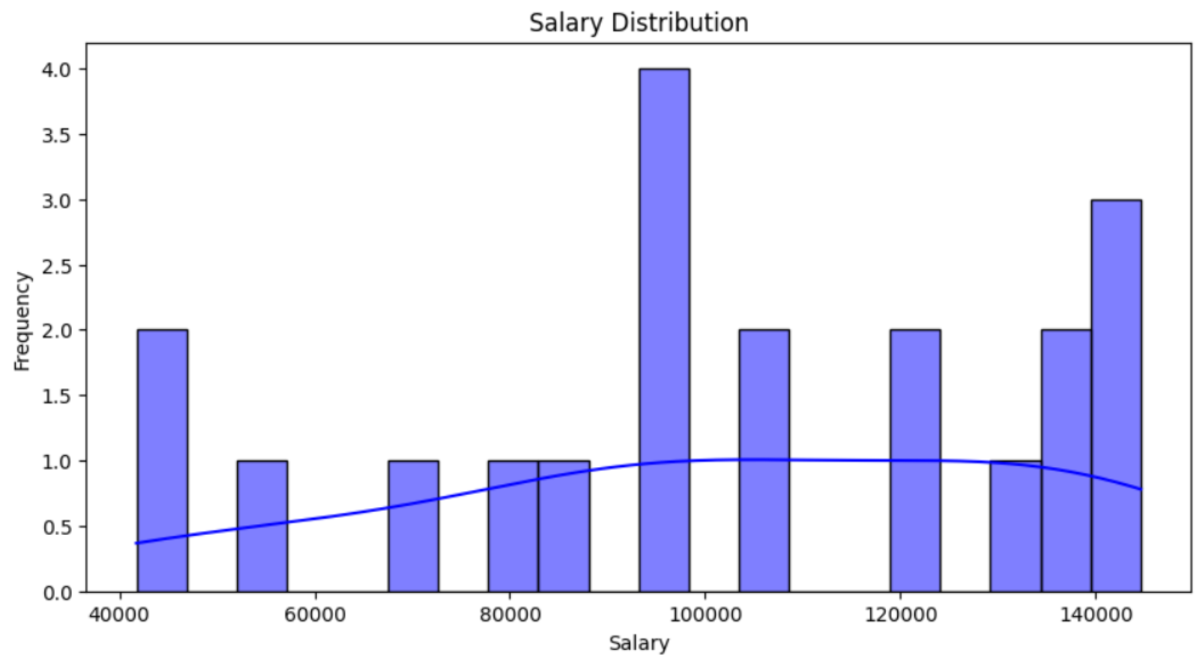
Output/Result:
The analysis provided the following key findings:

- **Salary Distribution:** The box plots revealed that salaries vary significantly across different job roles. Some job titles exhibit a wider range of salaries, indicating hierarchical levels within roles.
- **Departmental Differences:** The bar chart analysis showed that certain departments offer higher average salaries than others. For example, departments related to technology and finance had the highest salary levels, while administrative roles had relatively lower pay.
- **Outliers in Salary Data:** The analysis identified a few extreme values in the dataset, suggesting the presence of high-paying executive roles or potential data inconsistencies.

(Screenshots of graphs and data insights should be attached here.)







References/Credits:

- **Dataset Source:** [Kaggle]
- **Python Libraries Used:** Pandas, Matplotlib, Seaborn
- **Additional References:** External articles or documentation that helped in data preprocessing or visualization techniques

---