# UK Pension Payroll Analysis – Business Analyst Portfolio Project

## 1. Project Overview

This project involves analyzing payroll and pension contribution data for UK employees to uncover operational and financial insights using Python. The dataset includes monthly salary details, departmental information, pension scheme types, and contribution breakdowns for 200 employees.

Using tools such as **pandas**, **NumPy**, **Seaborn**, and **Matplotlib**, the project demonstrates how to:

- Clean and explore payroll data,
- Identify salary and contribution trends across departments,
- Visualize key metrics such as employer vs. employee contributions,
- Detect payroll processing issues (e.g., pending or error status),
- Analyze the adoption of different pension schemes,
- Draw actionable insights to support HR and finance decision-making.

This end-to-end project mimics a real-world business scenario and showcases data analysis, visualization, and storytelling skills expected from a Business Analyst.

## 2. Dataset Description

The dataset contains **200 records** and **12 columns**, each representing employee-level payroll and pension details. Key columns include:

- Employee_ID, Name: Unique identifiers
- Department: Department of the employee (e.g., HR, IT, Finance)
- Monthly_Salary: Gross monthly salary in GBP
- Employer_Contribution (%), Employee_Contribution (%): Pension contribution rates
- Employer_Contribution (£), Employee_Contribution (£), Total_Contribution (£): Calculated contribution values
- Pension_Scheme: Type of pension plan (Defined Benefit or Defined Contribution)
- Payroll_Status: Whether payment was processed (Paid, Pending, Error)
- Payment_Date: Monthly payment timestamp

📌 The dataset is 100% clean with **no missing or null values**, ensuring reliable analysis.

## 3. Initial Data Exploration (pandas)

```
df.head()

df.tail()

df.shape

df.columns

df.info()

df.describe()
```

**Output & Insight**:

- Dataset contains **200 rows** and **12 columns**.
- No null values.
- Columns include employee IDs, salary, department, pension schemes, payroll status, and calculated contributions.
- Data types are appropriate for each field (dates, strings, integers, floats).

```
df.isnull().sum()

df.duplicated().sum()
```

**Output & Insight**:

- **No missing (null) values** in the entire dataset.
- **No duplicate rows** — each employee record is unique.
- Confirms clean data — ready for analysis without preprocessing.

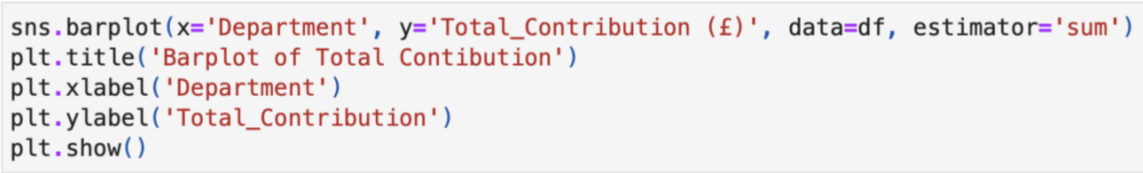
```
df.groupby('Department')['Monthly_Salary'].mean()

df.groupby('Department')['Total_Contribution (£)'].sum()

df.groupby('Pension_Scheme')['Total_Contribution (£)'].sum()
```

**Insight**:

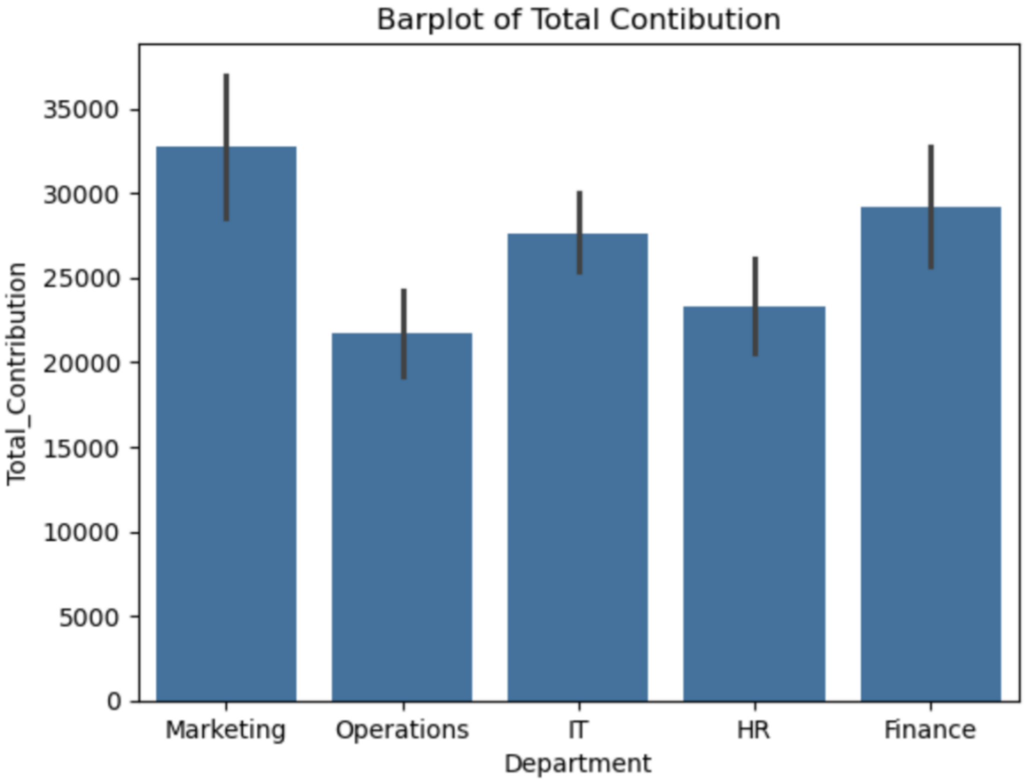
- Departments with higher average salaries and contributions are identified (e.g., IT and Finance).
- Total contribution by pension scheme highlights cost distribution between Defined Benefit and Contribution types.

## 4. Visual Explorations (Charts)

### i. Barplot – Total Contribution by Department



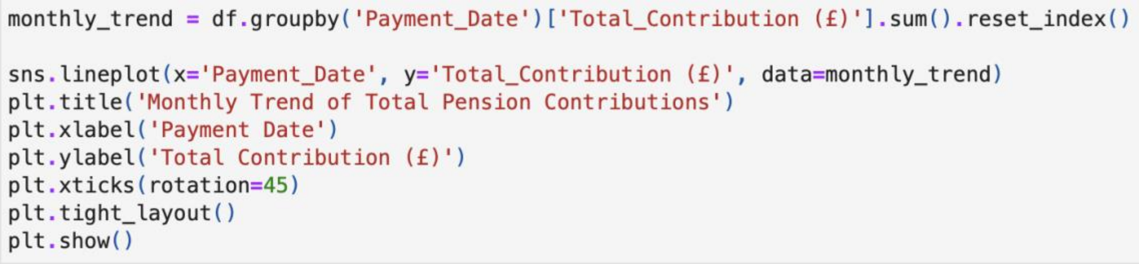
```
sns.barplot(x='Department', y='Total_Contribution (£)', data=df, estimator='sum')
plt.title('Barplot of Total Contibution')
plt.xlabel('Department')
plt.ylabel('Total_Contribution')
plt.show()
```



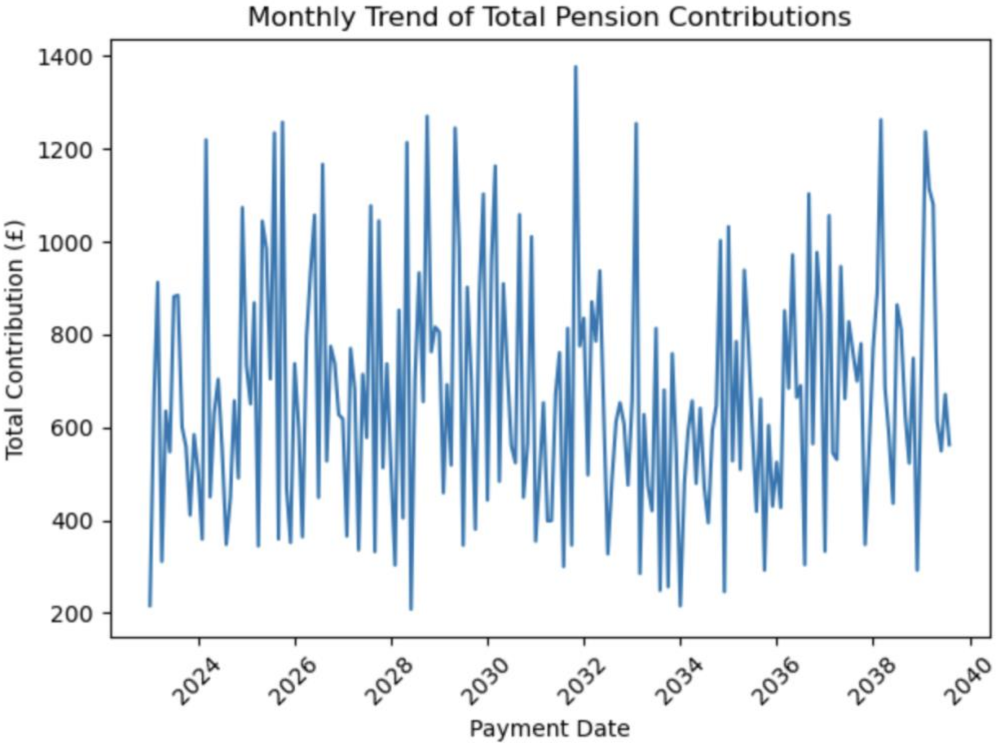
Explanation:

This chart shows the **total pension contributions (employer + employee)** grouped by each department. It highlights which departments contribute the most and are the biggest cost centers.

## ii. Line Plot – Monthly Total Contributions



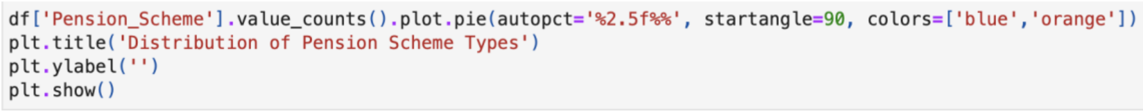
```
monthly_trend = df.groupby('Payment_Date')['Total_Contribution (£)'].sum().reset_index()

sns.lineplot(x='Payment_Date', y='Total_Contribution (£)', data=monthly_trend)
plt.title('Monthly Trend of Total Pension Contributions')
plt.xlabel('Payment Date')
plt.ylabel('Total Contribution (£)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



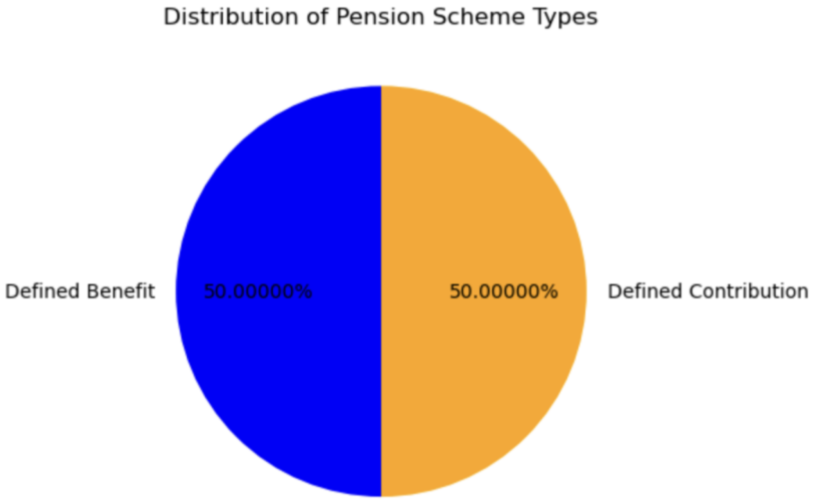
Explanation:

The line plot shows how pension contributions vary month-to-month. It is useful for identifying trends, spikes, or declines in contribution patterns.

## iii. Pie Chart – Pension Scheme Distribution



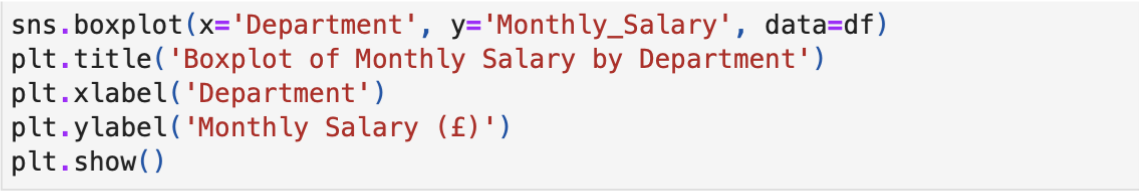
```
df['Pension_Scheme'].value_counts().plot.pie(autopct='%2.5f%%', startangle=90, colors=['blue','orange'])
plt.title('Distribution of Pension Scheme Types')
plt.ylabel('')
plt.show()
```



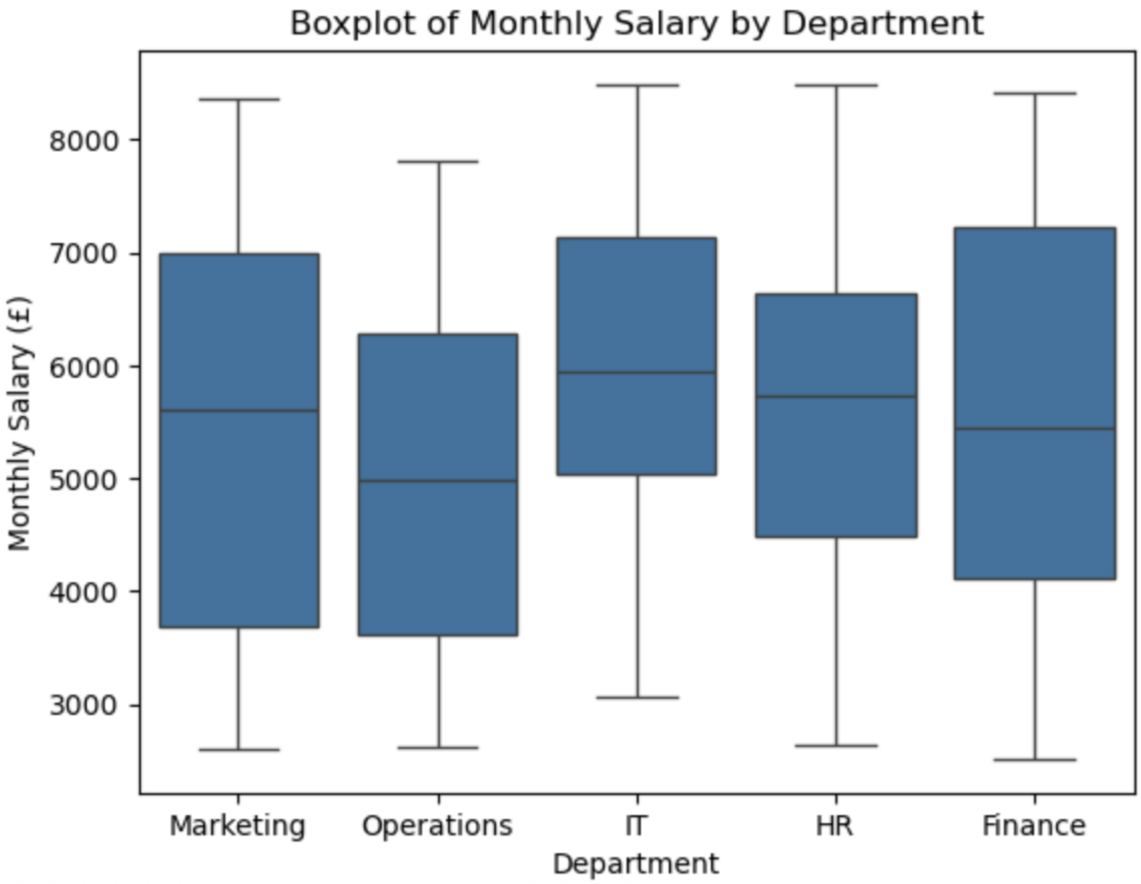
Explanation:

This pie chart visualizes how employees are distributed across 'Defined Benefit' and 'Defined Contribution' pension schemes. It helps stakeholders understand preference or policy direction.

### iv. Boxplot – Monthly Salary Distribution by Department



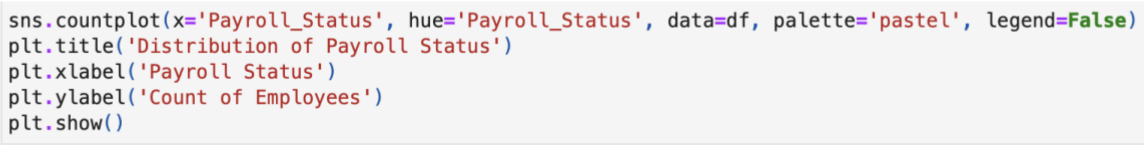
```
sns.boxplot(x='Department', y='Monthly_Salary', data=df)
plt.title('Boxplot of Monthly Salary by Department')
plt.xlabel('Department')
plt.ylabel('Monthly Salary (£)')
plt.show()
```



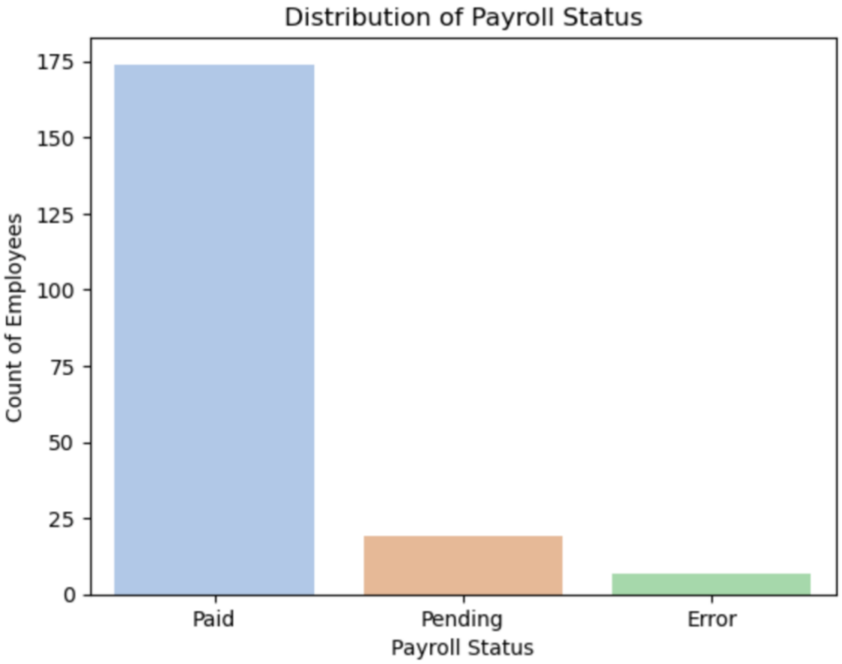
Explanation:

This boxplot helps identify **salary variations** and **outliers** within each department. It's helpful for compensation benchmarking and fairness audits.

## v. Countplot – Payroll Status



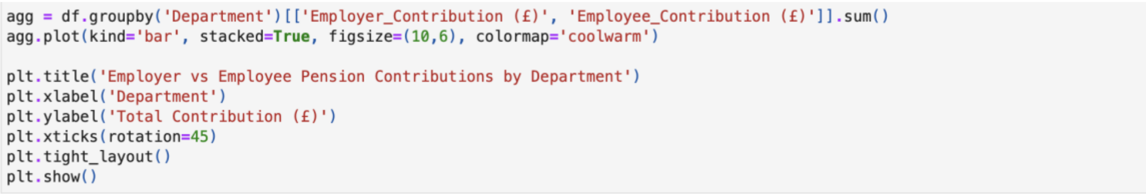
```
sns.countplot(x='Payroll_Status', hue='Payroll_Status', data=df, palette='pastel', legend=False)
plt.title('Distribution of Payroll Status')
plt.xlabel('Payroll Status')
plt.ylabel('Count of Employees')
plt.show()
```



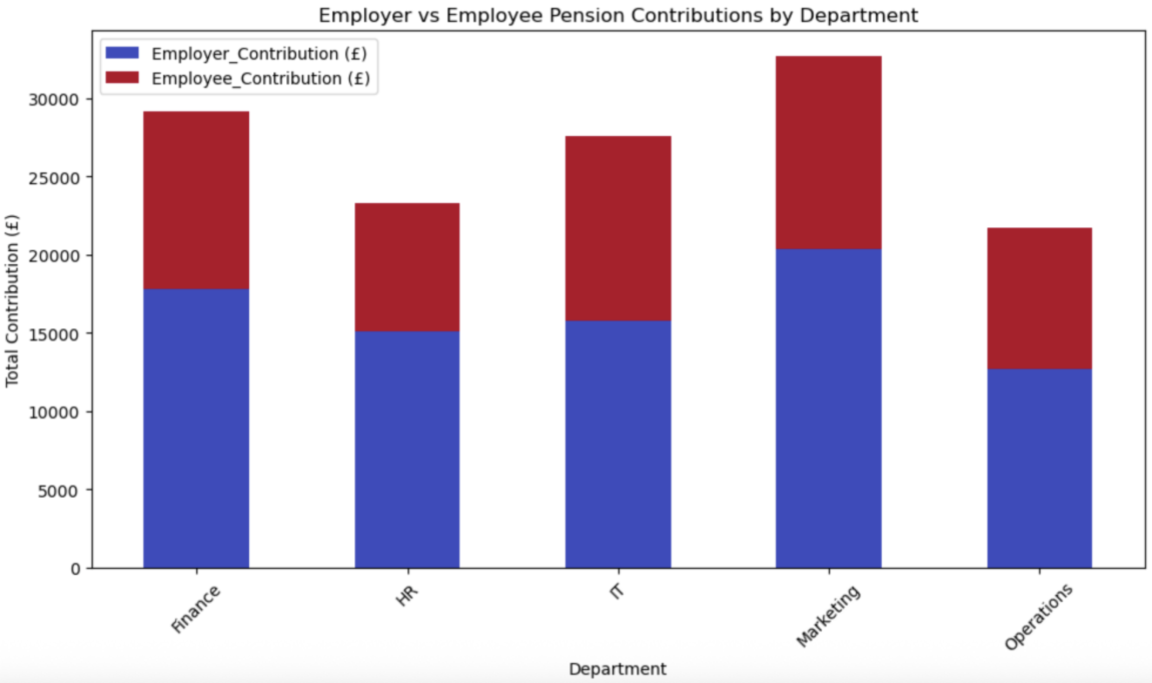
Explanation:

This chart shows the number of employees whose payroll status is Paid, Pending, or Error. It helps assess payroll processing efficiency and flag operational issues.

## vi. Stacked Bar Chart – Employer vs Employee Contribution by Department



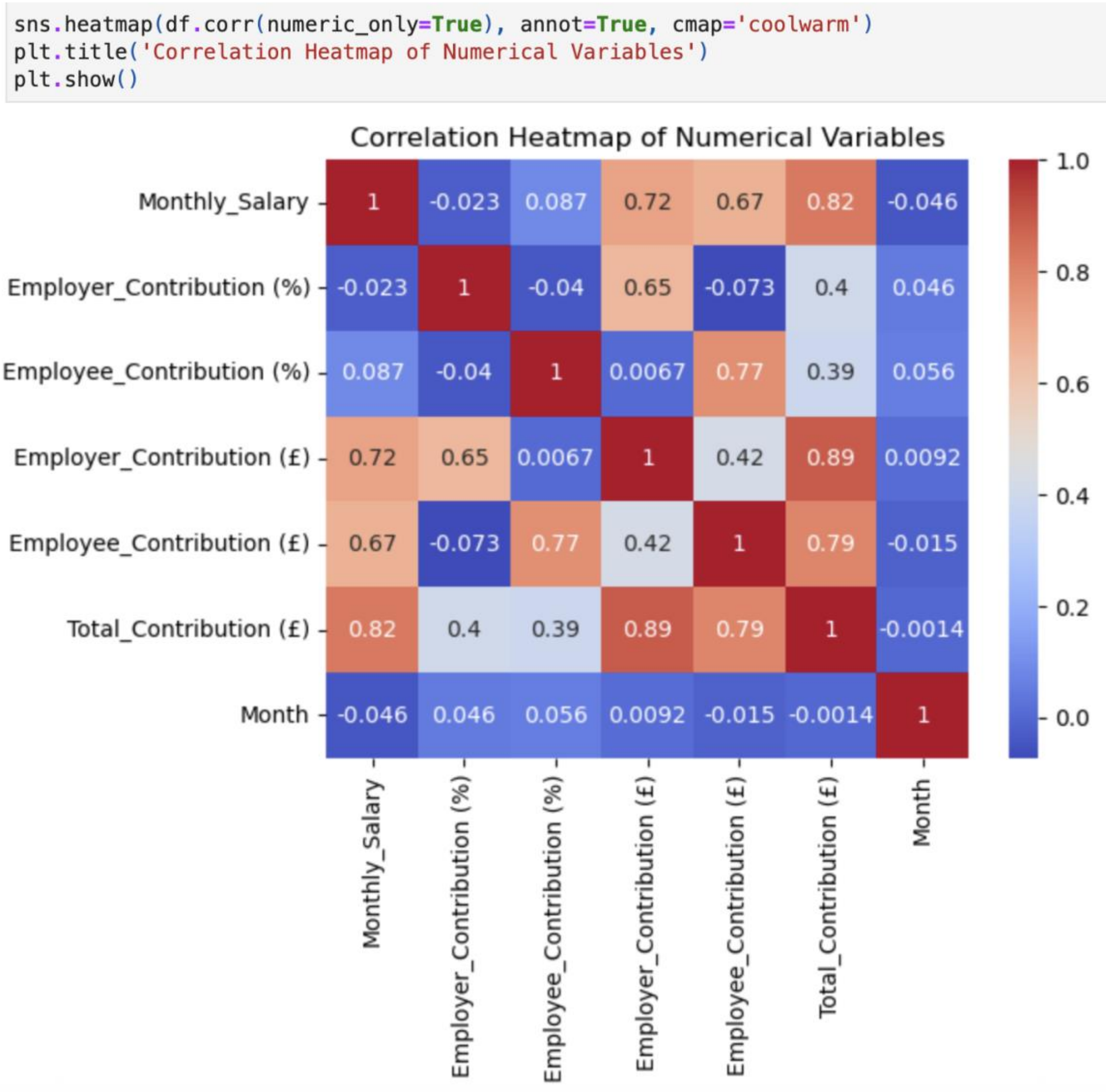
```
agg = df.groupby('Department')[['Employer_Contribution (£)', 'Employee_Contribution (£)']].sum()
agg.plot(kind='bar', stacked=True, figsize=(10,6), colormap='coolwarm')

plt.title('Employer vs Employee Pension Contributions by Department')
plt.xlabel('Department')
plt.ylabel('Total Contribution (£)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Explanation:

This stacked bar chart compares how much employers and employees each contribute within departments. It shows the balance of cost-sharing across the organization.

### vii. Heatmap – Correlation Between Numerical Features



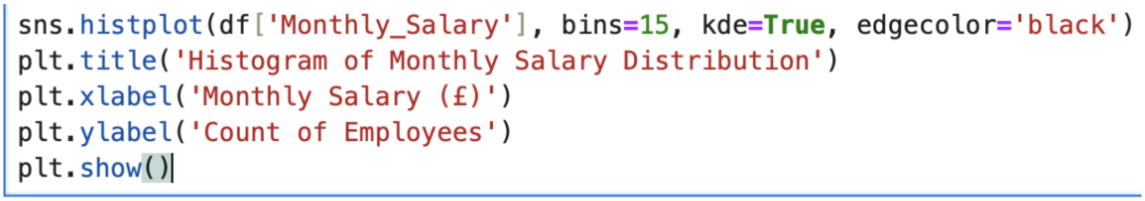
```
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap of Numerical Variables')
plt.show()
```

| | Monthly_Salary | Employer_Contribution (%) | Employee_Contribution (%) | Employer_Contribution (£) | Employee_Contribution (£) | Total_Contribution (£) | Month |
|---|---|---|---|---|---|---|---|
| Monthly_Salary | 1 | -0.023 | 0.087 | 0.72 | 0.67 | 0.82 | -0.046 |
| Employer_Contribution (%) | -0.023 | 1 | -0.04 | 0.65 | -0.073 | 0.4 | 0.046 |
| Employee_Contribution (%) | 0.087 | -0.04 | 1 | 0.0067 | 0.77 | 0.39 | 0.056 |
| Employer_Contribution (£) | 0.72 | 0.65 | 0.0067 | 1 | 0.42 | 0.89 | 0.0092 |
| Employee_Contribution (£) | 0.67 | -0.073 | 0.77 | 0.42 | 1 | 0.79 | -0.015 |
| Total_Contribution (£) | 0.82 | 0.4 | 0.39 | 0.89 | 0.79 | 1 | -0.0014 |
| Month | -0.046 | 0.046 | 0.056 | 0.0092 | -0.015 | -0.0014 | 1 |

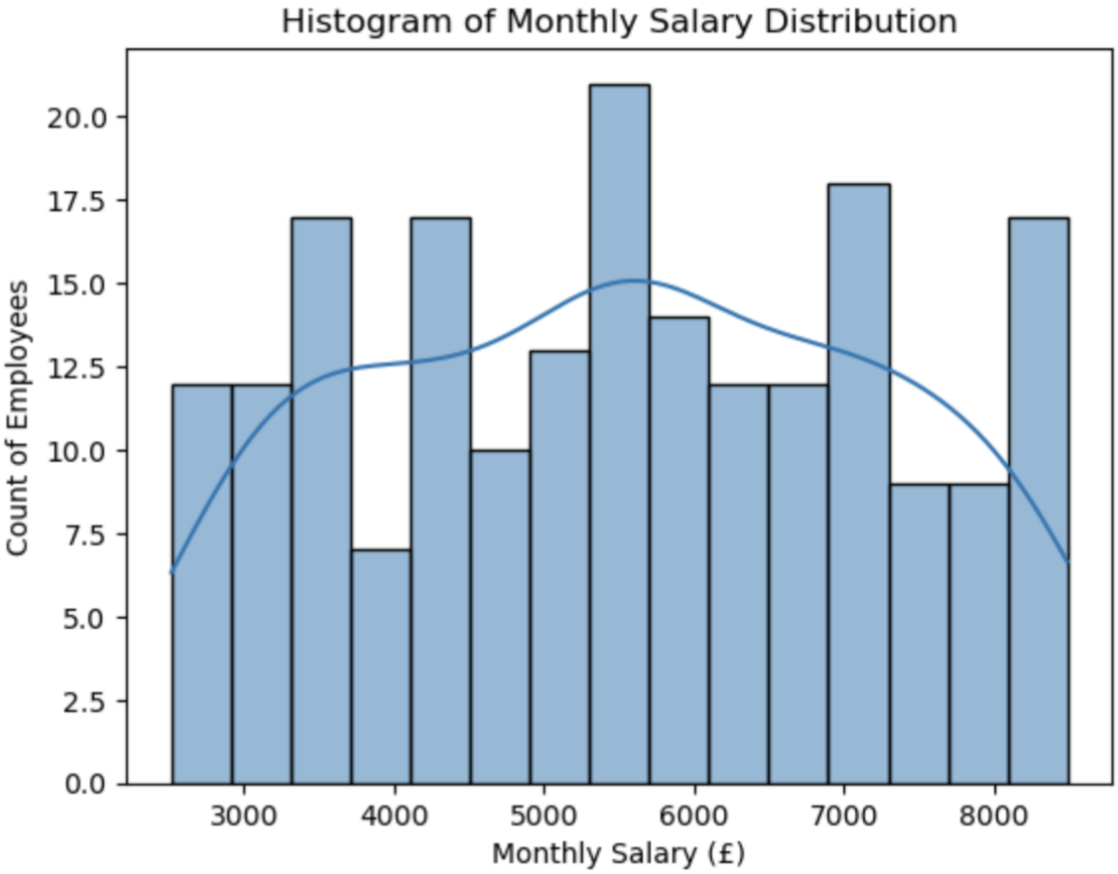
Explanation:

The heatmap shows how strongly numerical variables like salary and contributions correlate. This is useful for identifying key drivers and relationships in the dataset.

### vii. Histogram – Monthly Salary Distribution



```
sns.histplot(df['Monthly_Salary'], bins=15, kde=True, edgecolor='black')
plt.title('Histogram of Monthly Salary Distribution')
plt.xlabel('Monthly Salary (£)')
plt.ylabel('Count of Employees')
plt.show()
```



Explanation:

This histogram displays the distribution of employees’ monthly salaries across 15 equal-width bins. The overlaid KDE (Kernel Density Estimate) curve smooths out the distribution to highlight overall shape and density.

## 5. Key Insights

- **Finance and IT departments** contribute the highest total pension amounts, indicating higher salaries or employee counts.
- **Monthly pension contributions** show steady upward trends, useful for forecasting and budget planning.
- **Boxplot analysis** reveals significant **salary variation** across departments, with IT showing a higher median.
- **Payroll status breakdown** shows that ~10% are in “Pending” and ~5% in “Error”, highlighting areas for process improvement.
- **Defined Contribution schemes** are more widely adopted than Defined Benefit, suggesting a shift in pension strategy.

## 6. Tools & Libraries Used in Python

- pandas – data manipulation and cleaning
- NumPy – numeric operations and logic
- Seaborn – statistical visualizations
- Matplotlib – plotting and chart customization

## 7. Conclusion

This analysis delivered a comprehensive view of payroll and pension contributions across departments in a UK business scenario. It identified cost-heavy departments, contribution trends, and payroll inefficiencies. The findings support HR in **benchmarking salaries**, **evaluating pension strategy**, and help Finance with **resource allocation** and **process improvements**.

This project showcases critical **Business Analyst capabilities** including data wrangling, EDA, trend identification, and storytelling through visuals.