Project Title	Green Destinations
Tools	Python, Jupyter Notebook, Machine Learning
Domain	Domain Data Analyst and Data Scientist
Project Difficulties Levels	Intermediate

Dataset : Dataset is available in the given link. You can download it at your convenience.

[Click here to download the data set]

About Dataset Overview:
The dataset from Green Destinations contains detailed information about employees, with the primary goal of analysing attrition trends whether employees choose to leave the company. Spanning from March 2024 to the present, the dataset is updated daily and reflects a live view of the organization's workforce. Each record represents an individual employee and includes a wide range of attributes such as demographic details (age, gender, marital status), job-related data (department, job role, years at the company, salary), and workplace factors (job satisfaction, overtime status, work-life balance, and performance ratings). The key column, Attrition, indicates whether an employee has left the organization, making this dataset highly valuable for identifying the factors that contribute to employee turnover and supporting data driven HR decisions.

Dataset Overview:
Overview of the dataset structure.
Key columns:
Attrition, Age, Monthly Income, Years At Company, etc.
Data quality checks (missing values, data types, etc.)

Attrition Rate Calculation:
Formula used:
Attrition Rate =Employees Who Left/Total Employees ×100
Actual attrition rate derived from the data.

Factor Analysis:
1.Age vs Attrition:
Distribution of age groups and their attrition rates.
Any noticeable patterns or age brackets with high attrition.

2. Years at Company vs Attrition:
Are newer employees leaving more?
Average tenure of employees who left vs who stayed.
3. Monthly Income vs Attrition:
Comparing income levels of employees who left vs stayed.
Are lower-income employees more likely to leave?

Visualize the Insights:
Use plots to support your findings:
Bar/line charts for attrition by age group.
Box plots for income vs attrition.
Histograms plots for years at the company.

Optional: Predictive Modelling:
Build a simple classification model (like logistic regression or decision tree) to predict attrition using the identified features.

Project Outline: Green Destinations

Dataset Columns:
Here's a detailed explanation of each column in the Green Destinations dataset to help you understand what the data represents.

Column Name	Description
Age	Age of the employee
Attrition	Whether the employee left the company (Yes/No)
BusinessTravel	Frequency of business travel
DailyRate	Daily wage of the employee
Department	Department where the employee works (e.g., Sales, R&D)
DistanceFromHome	Distance from employee's home to workplace (in km or miles)
Education	Education level (1: Below College, 2: College, ..., 5: Doctor)
EducationField	Field of education (e.g., Life Sciences, Medical, Marketing)
EmployeeCount	Usually always 1 (legacy field)
EmployeeNumber	Unique identifier for each employee
EnvironmentSatisfaction	Satisfaction with work environment (1–4 scale)
Gender	Gender of the employee
HourlyRate	Hourly wage rate
JobInvolvement	Level of job involvement (1–4 scale)
JobLevel	Level of the job in the organization (1 = entry-level)

Column Name	Description
JobRole	Job title/role (e.g., Sales Executive, Research Scientist)
JobSatisfaction	Employee's job satisfaction (1–4 scale)
MaritalStatus	Marital status (e.g., Single, Married)
MonthlyIncome	Monthly salary
MonthlyRate	Monthly pay rate
NumCompaniesWorked	Number of companies the employee has worked for
Over18	Whether the employee is over 18 (typically always 'Y')
OverTime	Whether the employee works overtime (Yes/No)
PercentSalaryHike	Percentage increase in salary
PerformanceRating	Performance rating (typically 1–4)
RelationshipSatisfaction	Satisfaction with relationships at work (1–4 scale)
StandardHours	Standard number of work hours (typically 80, legacy field)
StockOptionLevel	Stock option level (0–3)
TotalWorkingYears	Total years of work experience
TrainingTimesLastYear	Number of training sessions attended last year
WorkLifeBalance	Work-life balance rating (1–4 scale)
YearsAtCompany	Years spent at the current company
YearsInCurrentRole	Years in the current job role
YearsSinceLastPromotion	Years since the last promotion
YearsWithCurrManager	Years working with the current manager

<u>Example: You can get the basic idea how you can create a project from here. Machine Learning Project:</u> Green Destinations. Green Destinations is a well-known travel agency. The HR Director has recently noticed an increase in employees leaving (attrition). She would like to figure out any trends or patterns. She has surveyed the staff of Green Destinations and provided you with the data. She would like to know what the attrition rate is (% of people who have left). She would also like to know if factors like age, years at the company and income play a part in determining if people will leave or not.

Sample code

Import Necessary Libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import the Data Set:

```
data = pd.read_csv("greendestination (1).csv")
print(data.head())
print(data.info())
```

Data Cleaning and Preprocessing:

```
print(data.isnull().sum())
# Drop any rows with missing values
data.dropna(inplace=True)
# Check for duplicates
data.drop_duplicates(inplace=True)
```

Calculate the Attrition Rate

```
# Total employees
total_employees = len(data)
# Employees who left
employees_left = data[data["Attrition"] == "Yes"]
print(f"Total Employees: {total_employees}")
print(f"Employees Who Left: {len(employees_left)}")
# Attrition rate
attrition_rate = (len(employees_left) / total_employees) * 100
print(f"Attrition Rate: {attrition_rate:.2f}%")
```

Analyze the Attrition by Monthly Income:

```
# Compare income
income_data = data.groupby("Attrition")["MonthlyIncome"].mean()
print("Average Income by Attrition Status:\n", income_data)
```

Analyze the Attrition by Years at Company:

```
#Analyze the Attrition by Years at Company
#Compare years at company
years_stats = data.groupby("Attrition")["YearsAtCompany"].mean()
print("Average Years at Company by Attrition Status:\n", years_stats)
```

Analyze the Attrition by Age

```
#Average age of employees who left vs stayed
age_stats = data.groupby("Attrition")["Age"].mean()
print("Average Age by Attrition Status:\n", age_stats)
```

Create a Crosstab (for visual comparison)

```
# Example: Attrition rate by department (if the column exists)
department_attrition = pd.crosstab(data["Department"], data["Attrition"],
normalize='index') * 100
print("Attrition Rate by Department:\n", department_attrition)

import matplotlib.pyplot as plt
import seaborn as sns

# Set Seaborn style for better visuals
sns.set(style="whitegrid")

# Create a figure with two subplots side by side
fig, axes = plt.subplots(1, 2, figsize=(16, 6))
```

```
# Histogram of Age by Attrition with enhancements
sns.histplot(
    data=data,
    x="Age",
    hue="Attrition",
    multiple="stack",
    palette="Set2",
    edgecolor="black",
    alpha=0.8,
    ax=axes[0]
)
axes[0].set_title("Age Distribution by Attrition Status", fontsize=14)
axes[0].set_xlabel("Age", fontsize=12)
axes[0].set_ylabel("Count", fontsize=12)
axes[0].legend(title="Attrition")

# Boxplot of MonthlyIncome by Attrition with enhancements
sns.boxplot(
    x="Attrition",
    y="MonthlyIncome",
    data=data,
    palette="Set3",
    width=0.5,
    ax=axes[1]
)
axes[1].set_title("Monthly Income by Attrition Status", fontsize=14)
axes[1].set_xlabel("Attrition", fontsize=12)
axes[1].set_ylabel("Monthly Income", fontsize=12)

# Adjust layout and display the plots
plt.tight_layout()
plt.show()
```

Main visualization

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
# Set Seaborn style
sns.set(style="whitegrid")
# Create side-by-side box plots for key features by Attrition
fig, axes = plt.subplots(1, 3, figsize=(18, 5))
# Boxplot for Age
sns.boxplot(data=data, x='Attrition', y='Age', ax=axes[0])
axes[0].set_title('Age by Attrition')
# Boxplot for Monthly Income
sns.boxplot(data=data, x='Attrition', y='MonthlyIncome', ax=axes[1])
axes[1].set_title('MonthlyIncome by Attrition')
```

```
# Boxplot for Years at Company
sns.boxplot(data=data, x='Attrition', y='YearsAtCompany', ax=axes[2])
axes[2].set_title('Years at Company by Attrition')
# Adjust layout and show plot
plt.tight_layout()
plt.show()
```

Exploratory Data Analysis (EDA):

```
import matplotlib.pyplot as plt
import seaborn as sns
# Set plot style
sns.set(style="whitegrid")
# Plot: Age distribution by Attrition
plt.figure(figsize=(10, 6))
sns.histplot(data=data, x='Age', hue='Attrition', kde=True, element='step', bins=30)
plt.title('Age Distribution by Attrition')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```

Attrition by Monthly Income:

```
# Plot: Monthly Income vs Attrition
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='Attrition', y='MonthlyIncome')
plt.title('Monthly Income by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Monthly Income')
plt.show()
```

Attrition by Years At Company:

```
# Plot: Years at Company by Attrition
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='Attrition', y='YearsAtCompany')
plt.title('Years at Company by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Years at Company')
plt.show()
```

Statistical And Visual Insights:

```
import seaborn as sns
import matplotlib.pyplot as plt
# Convert 'Attrition' to numeric (Yes:1, No:0) for correlation
data_corr = data.copy()
data_corr['Attrition'] = data_corr['Attrition'].map({'Yes': 1, 'No': 0})
# Compute correlation matrix
correlation = data_corr.corr(numeric_only=True)
# Plot heatmap
plt.figure(figsize=(14, 10))
```

```
sns.heatmap(correlation[['Attrition']].sort_values(by='Attrition', ascending=False),
annot=True, cmap='coolwarm')
plt.title('Correlation of Numeric Features with Attrition')
plt.show()
```

Visual Patterns:

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Attrition', y='MonthlyIncome', data=data)
plt.title('Monthly Income by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Monthly Income')
plt.show()
```

Boxplot: Age vs. Attrition:

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Attrition', y='Age', data=data)
plt.title('Age by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Age')
plt.show()
```

Boxplot: Years at Company vs. Attrition:

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Attrition', y='YearsAtCompany', data=data)
plt.title('Years at Company by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Years at Company')
plt.show()
```

Countplot: Overtime vs. Attrition:

```
plt.figure(figsize=(8, 5))
sns.countplot(x='OverTime', hue='Attrition', data=data)
plt.title('Attrition Count by Overtime Status')
plt.xlabel('OverTime')
plt.ylabel('Number of Employees')
plt.show()
```

Conclusion:

The employee attrition rate at Green Destinations stands at 16.12%, indicating a moderate turnover within the organization. Through data-driven analysis, several clear patterns have emerged:

Younger employees are significantly more likely to leave, suggesting a need for stronger engagement or growth opportunities for early-career staff.

Lower-income employees show a higher tendency to leave, indicating that compensation may be a key factor influencing retention.

A majority of employees who left had spent less than three years at the company, which highlights the critical importance of the early employment experience.

Overtime work showed the strongest correlation with attrition — those working extra hours were far more likely to leave, hinting at possible burnout or work-life balance issues

Final Insight:
To reduce attrition, Green Destinations should focus on improving work-life balance, re-evaluating early-career compensation, and supporting new hires more effectively through structured onboarding and mentorship programs.

Sample Code and Output

Import libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load data set

```
data = pd.read_csv("greendestination (1).csv")
# Check the first few rows
print(data.head())
print(data.info())
```

```
  Age Attrition    BusinessTravel  DailyRate            Department  \
0  41       Yes     Travel_Rarely       1102                 Sales
1  49        No Travel_Frequently        279  Research & Development
2  37       Yes     Travel_Rarely       1373  Research & Development
3  33        No Travel_Frequently       1392  Research & Development
4  27        No     Travel_Rarely        591  Research & Development

  DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber  \
0            1          2  Life Sciences            1              1
1            8          1  Life Sciences            1              2
2            2          2          Other            1              4
3            3          4  Life Sciences            1              5
4            2          1        Medical            1              7

  ...  RelationshipSatisfaction StandardHours  StockOptionLevel  \
0 ...                  1          80              0
1 ...                  4          80              1
2 ...                  2          80              0
3 ...                  3          80              0
4 ...                  4          80              1

  TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0            8              0          1            6
1           10              3          3           10
2            7              3          3            0
3            8              3          3            8
4            6              3          3            2

 YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0            4              0              5
1            7              1              7
2            0              0              0
3            7              3              0
4            2              2              2

[5 rows x 35 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
```

```
 0  Age                        1470 non-null  int64
 1  Attrition                  1470 non-null  object
 2  BusinessTravel             1470 non-null  object
 3  DailyRate                  1470 non-null  int64
 4  Department                 1470 non-null  object
 5  DistanceFromHome           1470 non-null  int64
 6  Education                  1470 non-null  int64
 7  EducationField             1470 non-null  object
 8  EmployeeCount              1470 non-null  int64
 9  EmployeeNumber             1470 non-null  int64
 10 EnvironmentSatisfaction    1470 non-null  int64
 11 Gender                     1470 non-null  object
 12 HourlyRate                 1470 non-null  int64
 13 JobInvolvement             1470 non-null  int64
 14 JobLevel                   1470 non-null  int64
 15 JobRole                    1470 non-null  object
 16 JobSatisfaction            1470 non-null  int64
 17 MaritalStatus              1470 non-null  object
 18 MonthlyIncome              1470 non-null  int64
 19 MonthlyRate                1470 non-null  int64
 20 NumCompaniesWorked         1470 non-null  int64
 21 Over18                     1470 non-null  object
 22 OverTime                   1470 non-null  object
 23 PercentSalaryHike          1470 non-null  int64
 24 PerformanceRating          1470 non-null  int64
 25 RelationshipSatisfaction   1470 non-null  int64
 26 StandardHours              1470 non-null  int64
 27 StockOptionLevel           1470 non-null  int64
 28 TotalWorkingYears          1470 non-null  int64
 29 TrainingTimesLastYear      1470 non-null  int64
 30 WorkLifeBalance            1470 non-null  int64
 31 YearsAtCompany             1470 non-null  int64
 32 YearsInCurrentRole         1470 non-null  int64
 33 YearsSinceLastPromotion    1470 non-null  int64
 34 YearsWithCurrManager       1470 non-null  int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
None
```

```python
print(data.isnull().sum())
# Drop any rows with missing values
data.dropna(inplace=True)
# Check for duplicates
data.drop_duplicates(inplace=True)
```

output

```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
```

TotalWorkingYears	0
TrainingTimesLastYear	0
WorkLifeBalance	0
YearsAtCompany	0
YearsInCurrentRole	0
YearsSinceLastPromotion	0
YearsWithCurrManager	0

```
#Calculate the Attrition Rate
# Total employees
total_employees = len(data)
# Employees who left
employees_left = data[data["Attrition"] == "Yes"]
print(f"Total Employees: {total_employees}")
print(f"Employees Who Left: {len(employees_left)}")
# Attrition rate
attrition_rate = (len(employees_left) / total_employees) * 100
print(f"Attrition Rate: {attrition_rate:.2f}%")
```

Total Employees: 1470
Employees Who Left: 237
Attrition Rate: 16.12%

```
# Analyze Attrition by MonthlyIncome
# Compare income
income_data = data.groupby("Attrition")["MonthlyIncome"].mean()
print("Average Income by Attrition Status:\n", income_data)
```

Average Income by Attrition Status:
Attrition
No 6832.739659
Yes 4787.092827
Name: MonthlyIncome, dtype: float64

```
#Analyze the Attrition by Years at Company
#Compare years at company
years_stats = data.groupby("Attrition")["YearsAtCompany"].mean()
print("Average Years at Company by Attrition Status:\n", years_stats)
```

Average Years at Company by Attrition Status:
Attrition
No 7.369019
Yes 5.130802
Name: YearsAtCompany, dtype: float64

```
#Analyze the Attrition by Age
#Average age of employees who left vs stayed
age_stats = data.groupby("Attrition")["Age"].mean()
print("Average Age by Attrition Status:\n", age_stats)
```

Average Age by Attrition Status:
Attrition
No 37.561233
Yes 33.607595
Name: Age, dtype: float64

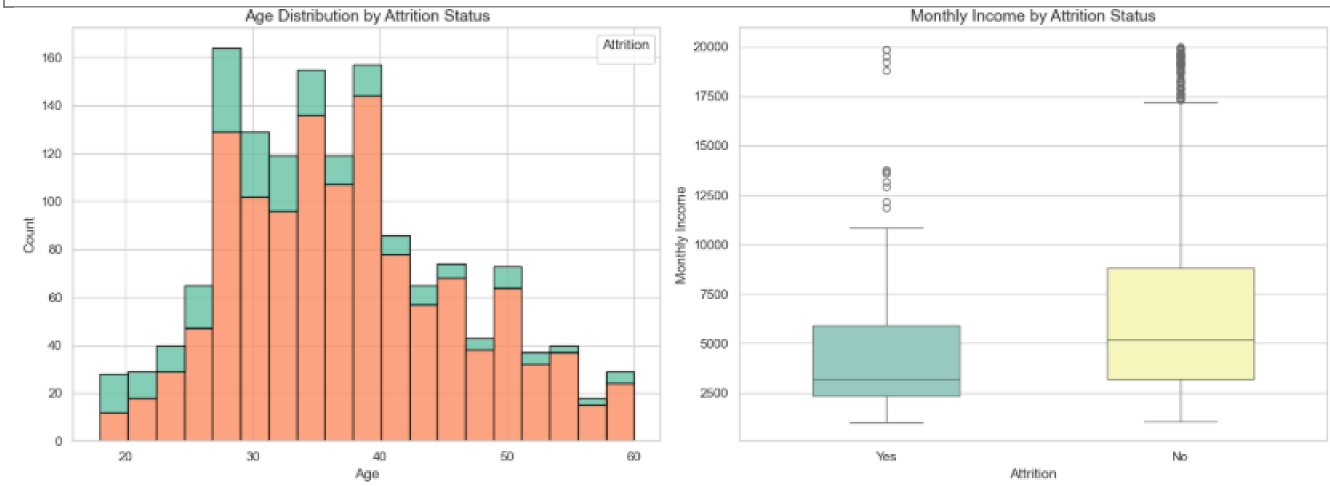
```
#Create a Crosstab (for visual comparison)
# Example: Attrition rate by department (if the column exists)
department_attrition = pd.crosstab(data["Department"], data["Attrition"],
normalize='index') * 100
print("Attrition Rate by Department:\n", department_attrition)
```

Attrition Rate by Department:

Attrition	No	Yes
Department		
Human Resources	80.952381	19.047619
Research & Development	86.160250	13.839750
Sales	79.372197	20.627803

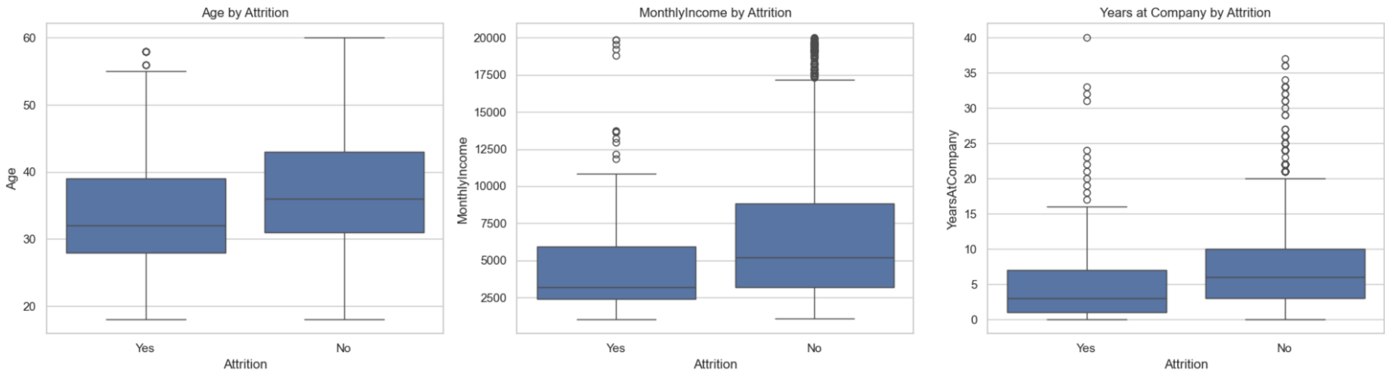
```
import matplotlib.pyplot as plt
import seaborn as sns

# Set Seaborn style for better visuals
sns.set(style="whitegrid")

# Create a figure with two subplots side by side
fig, axes = plt.subplots(1, 2, figsize=(16, 6))
```

```python
# Histogram of Age by Attrition with enhancements
sns.histplot(
    data=data,
    x="Age",
    hue="Attrition",
    multiple="stack",
    palette="Set2",
    edgecolor="black",
    alpha=0.8,
    ax=axes[0]
)
axes[0].set_title("Age Distribution by Attrition Status", fontsize=14)
axes[0].set_xlabel("Age", fontsize=12)
axes[0].set_ylabel("Count", fontsize=12)
axes[0].legend(title="Attrition")

# Boxplot of MonthlyIncome by Attrition with enhancements
sns.boxplot(
    x="Attrition",
    y="MonthlyIncome",
    data=data,
    palette="Set3",
    width=0.5,
    ax=axes[1]
)
axes[1].set_title("Monthly Income by Attrition Status", fontsize=14)
axes[1].set_xlabel("Attrition", fontsize=12)
axes[1].set_ylabel("Monthly Income", fontsize=12)

# Adjust layout and display the plots
plt.tight_layout()
plt.show()
```

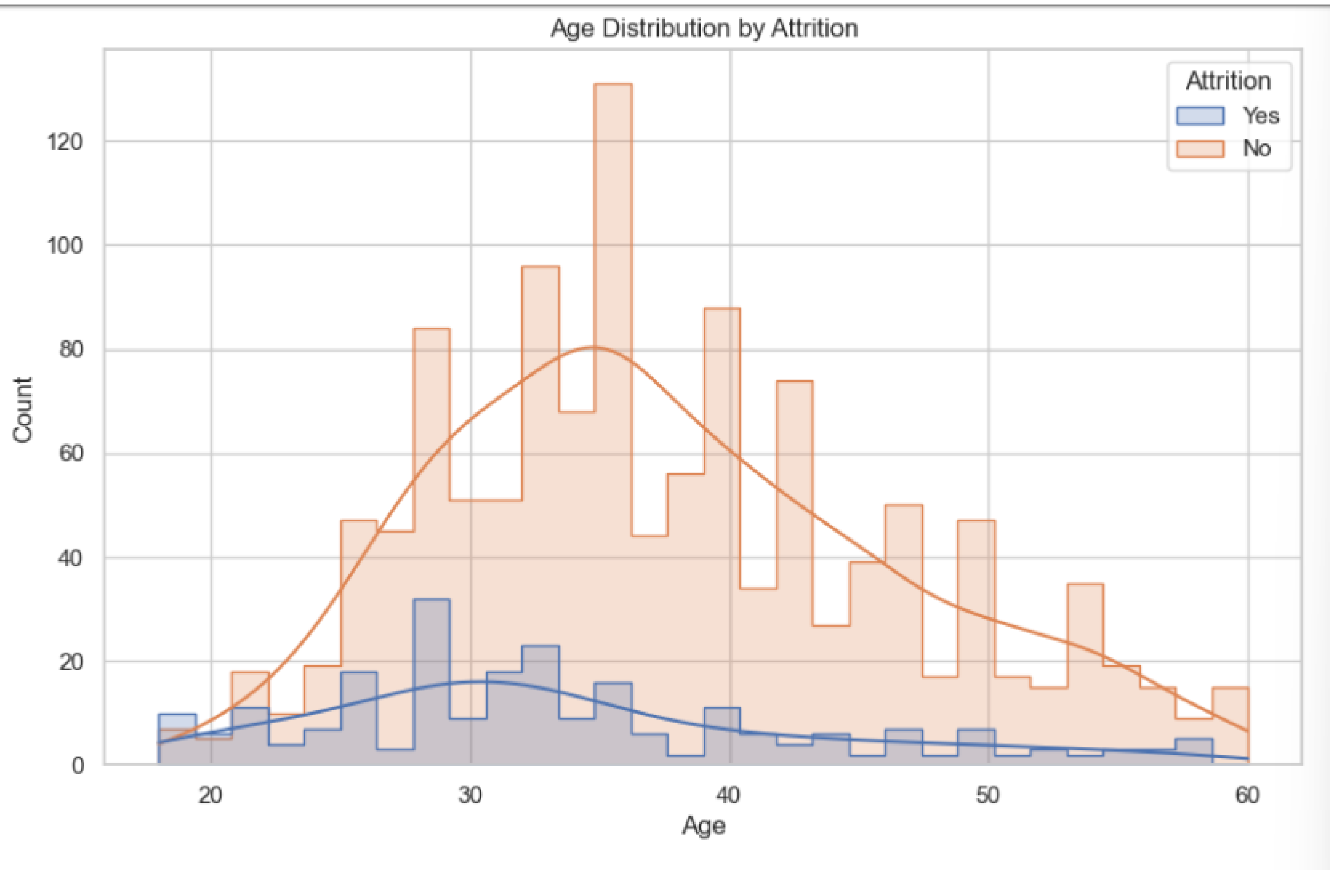



```python
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
# Set Seaborn style
sns.set(style="whitegrid")
# Create side-by-side box plots for key features by Attrition
fig, axes = plt.subplots(1, 3, figsize=(18, 5))
# Boxplot for Age
sns.boxplot(data=data, x='Attrition', y='Age', ax=axes[0])
axes[0].set_title('Age by Attrition')
# Boxplot for Monthly Income
sns.boxplot(data=data, x='Attrition', y='MonthlyIncome', ax=axes[1])
axes[1].set_title('MonthlyIncome by Attrition')
# Boxplot for Years at Company
sns.boxplot(data=data, x='Attrition', y='YearsAtCompany', ax=axes[2])
axes[2].set_title('Years at Company by Attrition')
# Adjust layout and show plot
plt.tight_layout()
```

```
plt.show()
```

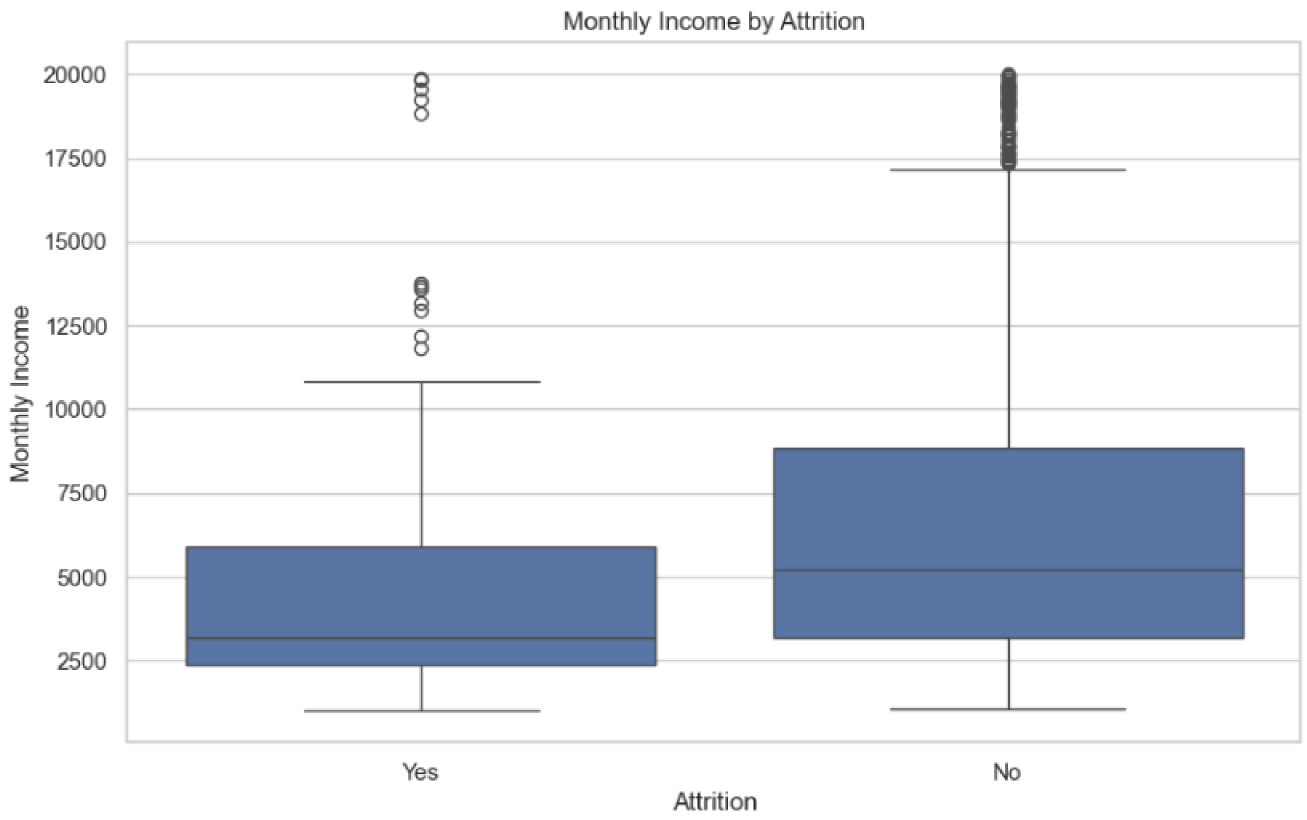



```
import matplotlib.pyplot as plt
import seaborn as sns
# Set plot style
sns.set(style="whitegrid")
# Plot: Age distribution by Attrition
plt.figure(figsize=(10, 6))
sns.histplot(data=data, x='Age', hue='Attrition', kde=True, element='step', bins=30)
plt.title('Age Distribution by Attrition')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```

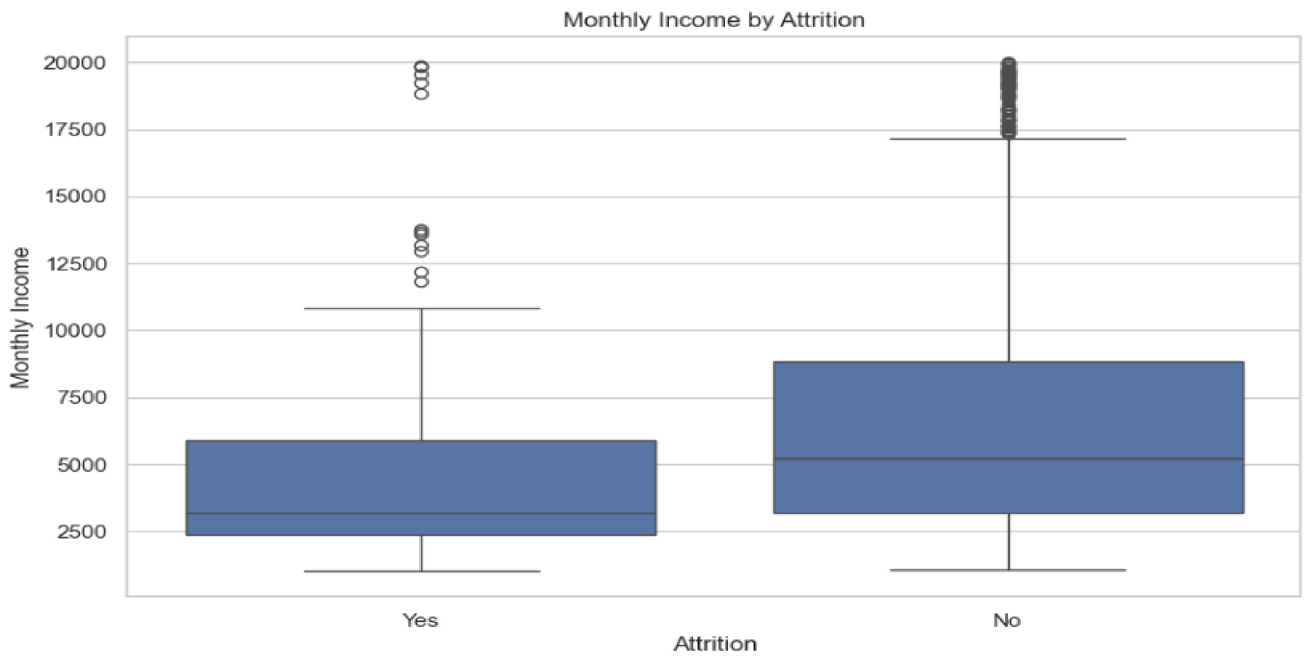



```
# Plot: Monthly Income vs Attrition
```

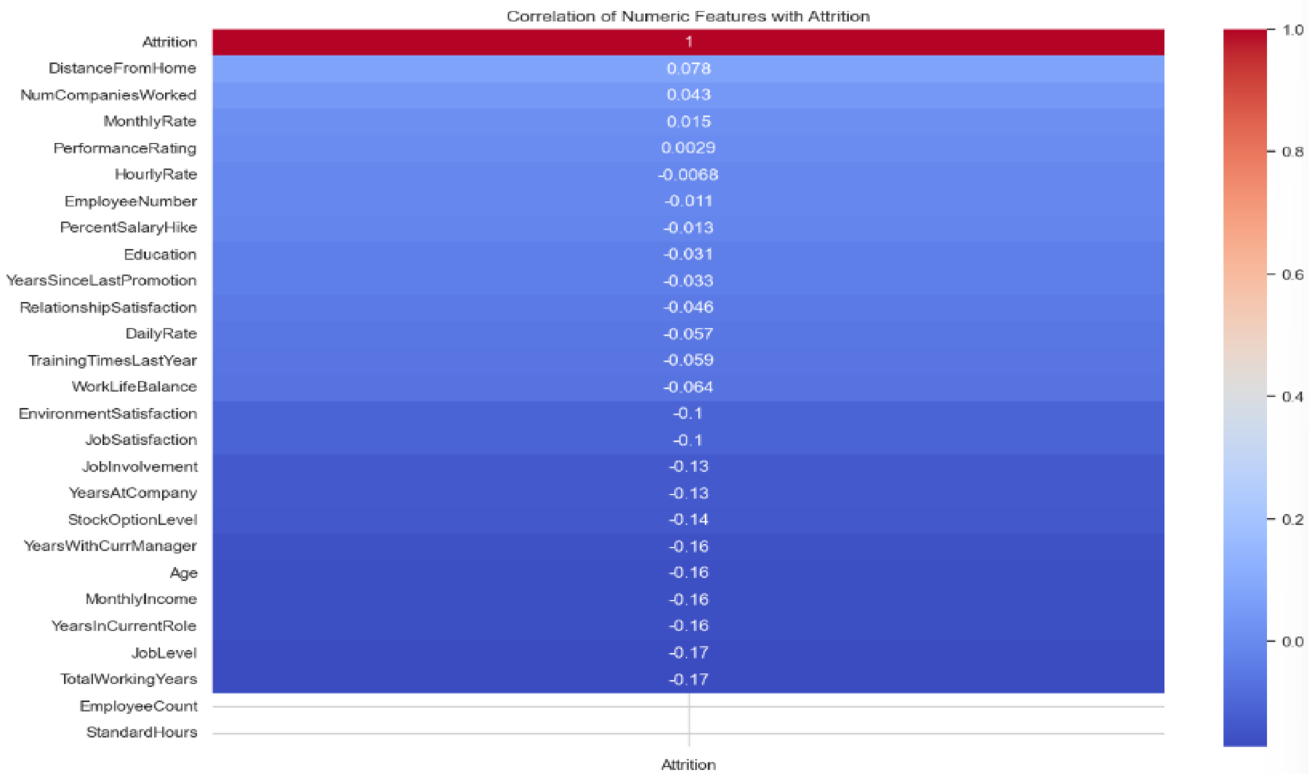
```python
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='Attrition', y='MonthlyIncome')
plt.title('Monthly Income by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Monthly Income')
plt.show()
```



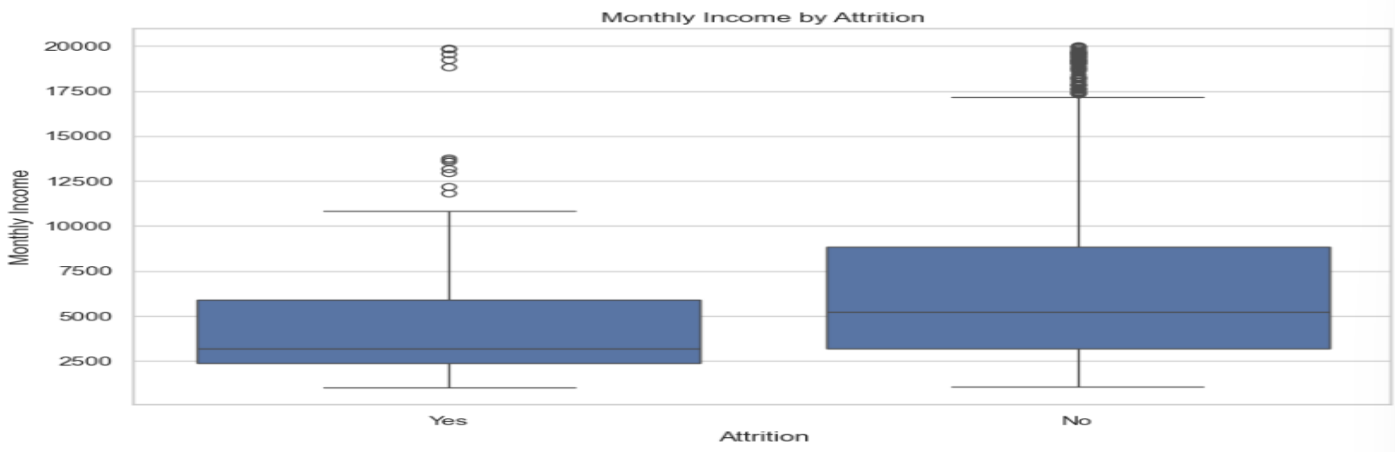

```python
# Plot: Years at Company by Attrition
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='Attrition', y='YearsAtCompany')
plt.title('Years at Company by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Years at Company')
plt.show()
```



```python
import seaborn as sns
import matplotlib.pyplot as plt
# Convert 'Attrition' to numeric (Yes:1, No:0) for correlation
data_corr = data.copy()
data_corr['Attrition'] = data_corr['Attrition'].map({'Yes': 1, 'No': 0})
# Compute correlation matrix
correlation = data_corr.corr(numeric_only=True)
# Plot heatmap
plt.figure(figsize=(14, 10))
sns.heatmap(correlation[['Attrition']].sort_values(by='Attrition', ascending=False),
annot=True, cmap='coolwarm')
plt.title('Correlation of Numeric Features with Attrition')
plt.show()
```

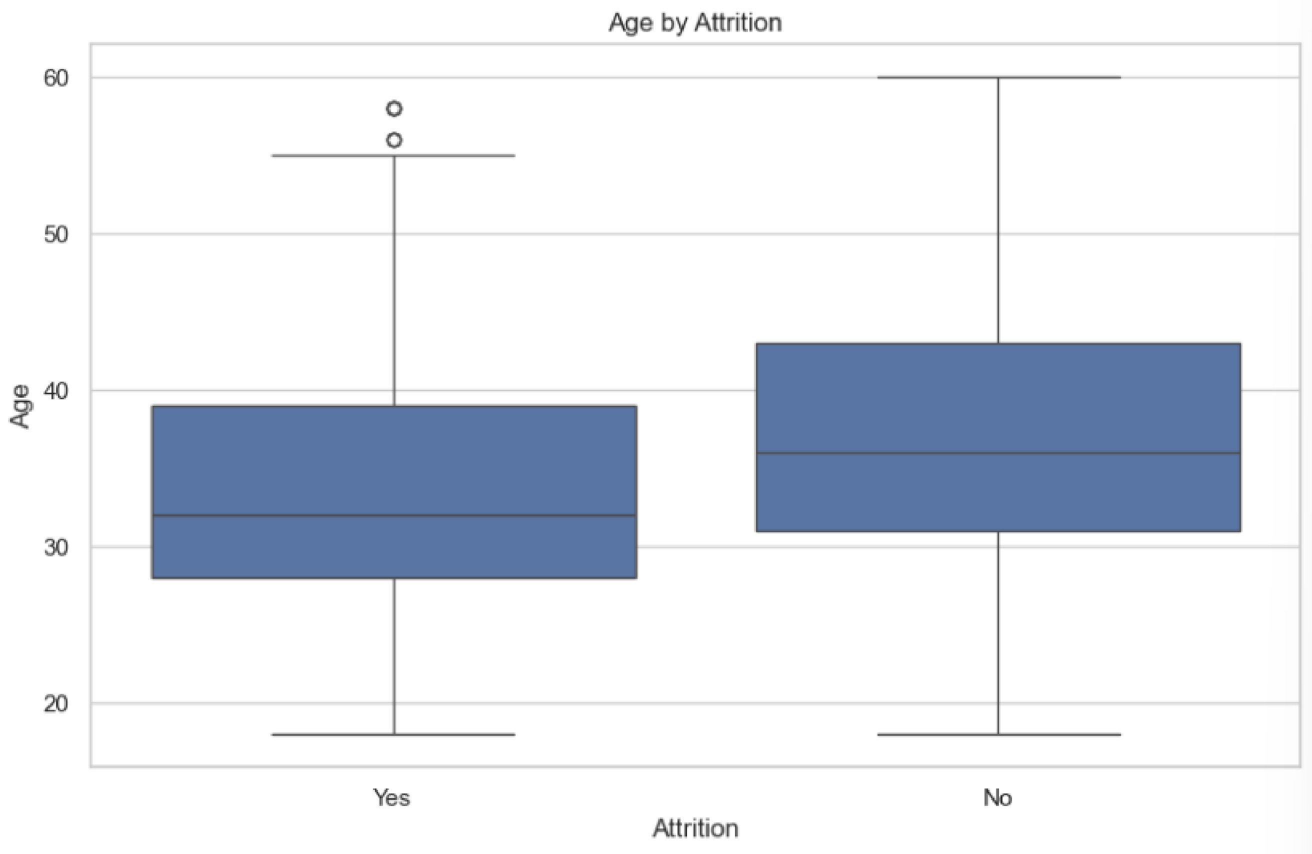



```python
plt.figure(figsize=(10, 6))
sns.boxplot(x='Attrition', y='MonthlyIncome', data=data)
plt.title('Monthly Income by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Monthly Income')
plt.show()
```

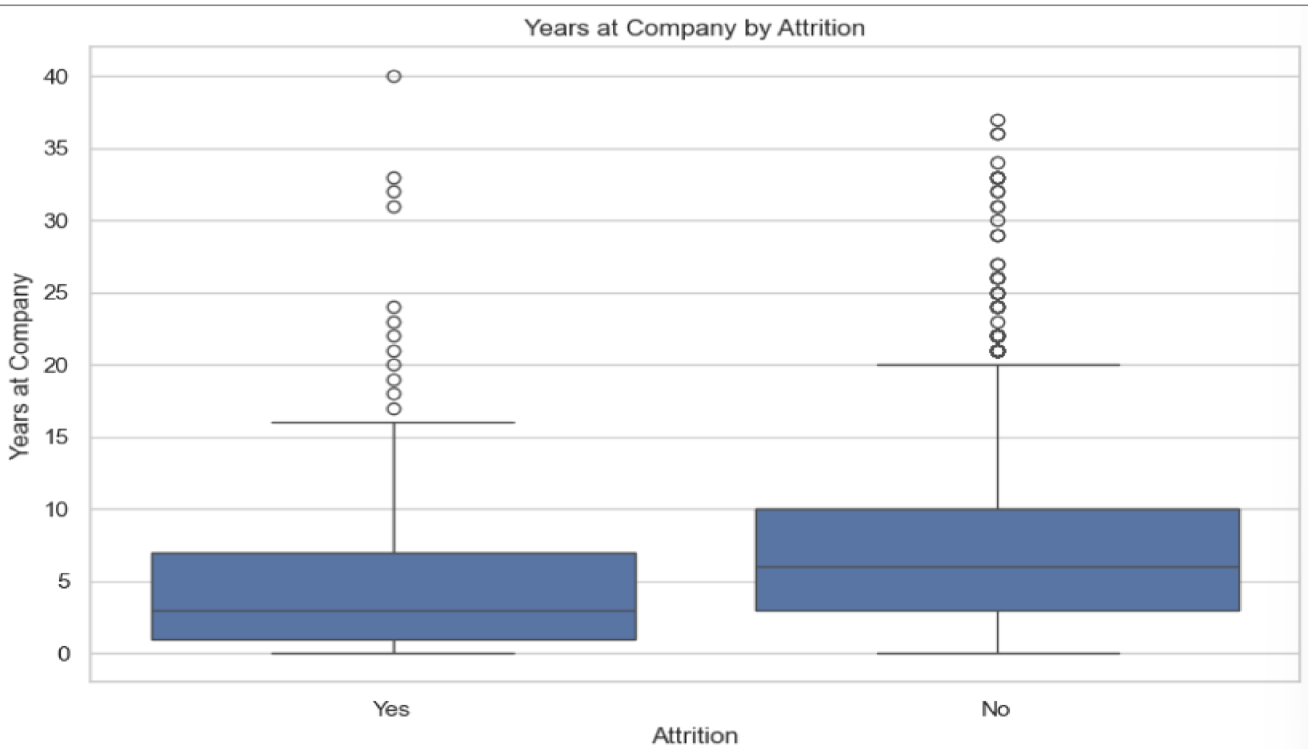



```python
plt.figure(figsize=(10, 6))
sns.boxplot(x='Attrition', y='Age', data=data)
plt.title('Age by Attrition')
plt.xlabel('Attrition')
```

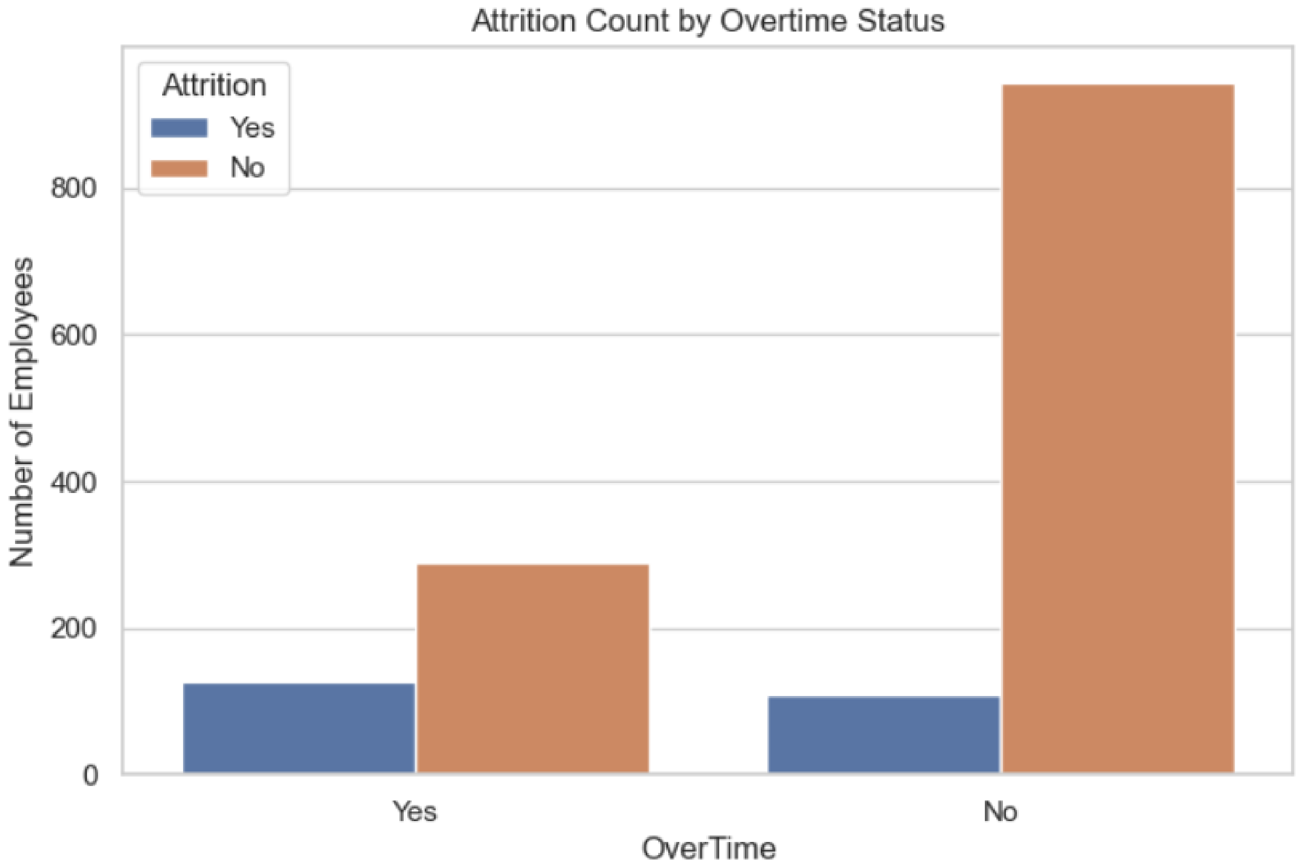
```
plt.ylabel('Age')
plt.show()
```




```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Attrition', y='YearsAtCompany', data=data)
plt.title('Years at Company by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Years at Company')
plt.show()
```



```
plt.figure(figsize=(8, 5))
sns.countplot(x='OverTime', hue='Attrition', data=data)
plt.title('Attrition Count by Overtime Status')
plt.xlabel('OverTime')
plt.ylabel('Number of Employees')
plt.show()
```



Inference:

Based on the analysis of employee data at Green Destinations, we can infer the following:

1. Attrition is influenced by multiple interrelated factors — particularly age, income, tenure, and overtime work.
2. Younger and less experienced employees are more likely to leave the organization. This suggests potential gaps in early-career engagement, support, or satisfaction .
3. Lower monthly income is associated with higher attrition. Employees in lower salary brackets may feel undervalued or see better opportunities elsewhere.
4. Employees working overtime are significantly more prone to leave, indicating a strong link between workload, burnout, and turnover.
5. Short tenure (<3 years) is common among those who exited, which highlights that the early phase of employment is the most vulnerable to attrition.

Key Inference:

Improving early employee experience, offering competitive compensation, and managing overtime effectively could significantly reduce attrition and improve long-term retention.

Referance link