Load and study the data

Import the libraries that will be used in this notebook

In [1]:

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

Load the csv file as pandas dataframe

In [2]:

```
# Read in the "Dataset" file as a Pandas Data Frame
data=pd.read_csv('greendestination (1).csv')
data.head()
```

Out[2]:

	Age	Attrition	BusinessTravel	DailyRate	D
0	41	Yes	Travel_Rarely	1102	
1	49	No	Travel_Frequently	279	R De
2	37	Yes	Travel_Rarely	1373	R De
3	33	No	Travel_Frequently	1392	R De
4	27	No	Travel_Rarely	591	R De

5 rows × 35 columns

In [3]:

```
# Take a brief look at the data
data.shape
```

Out[3]:

```
(1470, 35)
```

In [4]:

```
# Look at basic information about the dataframe
data.info()
```

```
<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
```

In [5]:

```
#Checking for null values
data.isnull().sum()
```

Out[5]:

```
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
dtype: int64
```

Calculate the attrition rate

In [6]:

```
# Attrition rate
total_employees = data.shape[0]
attrition_count = data[data['Attrition'] == 'Yes'].sh
attrition_rate = (attrition_count / total_employees)

attrition_rate
```

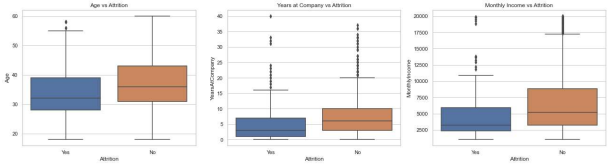
Out[6]:

```
16.122448979591837
```

Data Visualization

In [7]:

```
# Set up the visual style
sns.set(style="whitegrid")

# Plot distributions for Age, YearsAtCompany, and Mon
fig, axes = plt.subplots(1, 3, figsize=(18, 5))

# Age vs Attrition
sns.boxplot(x='Attrition', y='Age', data=data, ax=axe
axes[0].set_title('Age vs Attrition')

# Years at Company vs Attrition
sns.boxplot(x='Attrition', y='YearsAtCompany', data=d
axes[1].set_title('Years at Company vs Attrition')

# Monthly Income vs Attrition
sns.boxplot(x='Attrition', y='MonthlyIncome', data=da
axes[2].set_title('Monthly Income vs Attrition')

plt.tight_layout()
plt.show()
```



The boxplots provide the following insights:

Age vs Attrition: Employees who have left (Attrition = Yes) tend to be slightly younger on average than those who have stayed. There is some overlap, but age could be a factor influencing attrition.

Years at Company vs Attrition: Employees who have left tend to have spent fewer years at the company compared to those who stayed. This suggests that newer employees might be more likely to leave.

Monthly Income vs Attrition: There is no strong visual difference in monthly income between those who have stayed and those who have left. Income may not be a major determining factor.

Data Visualization

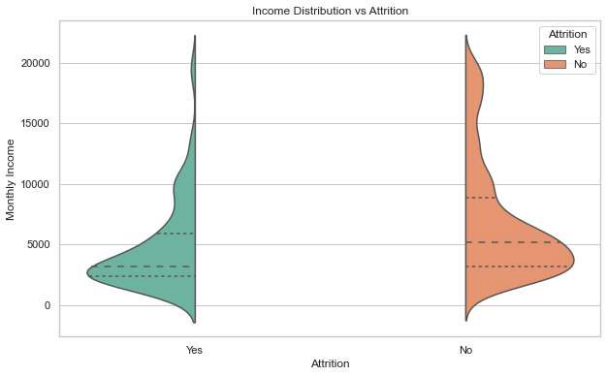
In [8]:

```
# Visualization 2: Income Distribution by Attrition (
plt.figure(figsize=(10, 6))
sns.kdeplot(data=data, x='MonthlyIncome', hue='Attrit
plt.title('Monthly Income Distribution by Attrition')
plt.xlabel('Monthly Income')
plt.ylabel('Density')
plt.show()
```

We use a KDE plot to compare Monthly Income between employees who stayed and those who left, you can see the income ranges where employees are more likely to leave the company versus where they are more likely to stay.

In [9]:

```
# Visualization 3: Age and Income Interaction with At
plt.figure(figsize=(10, 6))
sns.violinplot(x='Attrition', y='MonthlyIncome', hue=
plt.title('Income Distribution vs Attrition')
plt.ylabel('Monthly Income')
plt.xlabel('Attrition')
plt.show()
```



A violin plot of Monthly Income against attrition status (Yes/No) allows you to observe the income distribution for employees who left versus those who stayed, along with key statistics like median income.

In [10]:

```
# Visualization 4: Years at Company vs. Attrition (Sc
plt.figure(figsize=(10, 6))
sns.scatterplot(x='YearsAtCompany', y='Age', hue='Att
plt.title('Years at Company vs Age by Attrition')
plt.xlabel('Years at Company')
plt.ylabel('Age')
plt.show()
```

We have used scatter plot to distinguish between employees who have left (Attrition = Yes) and those who have stayed (Attrition = No).