```
pip install pandas seaborn matplotlib scikit-learn
```

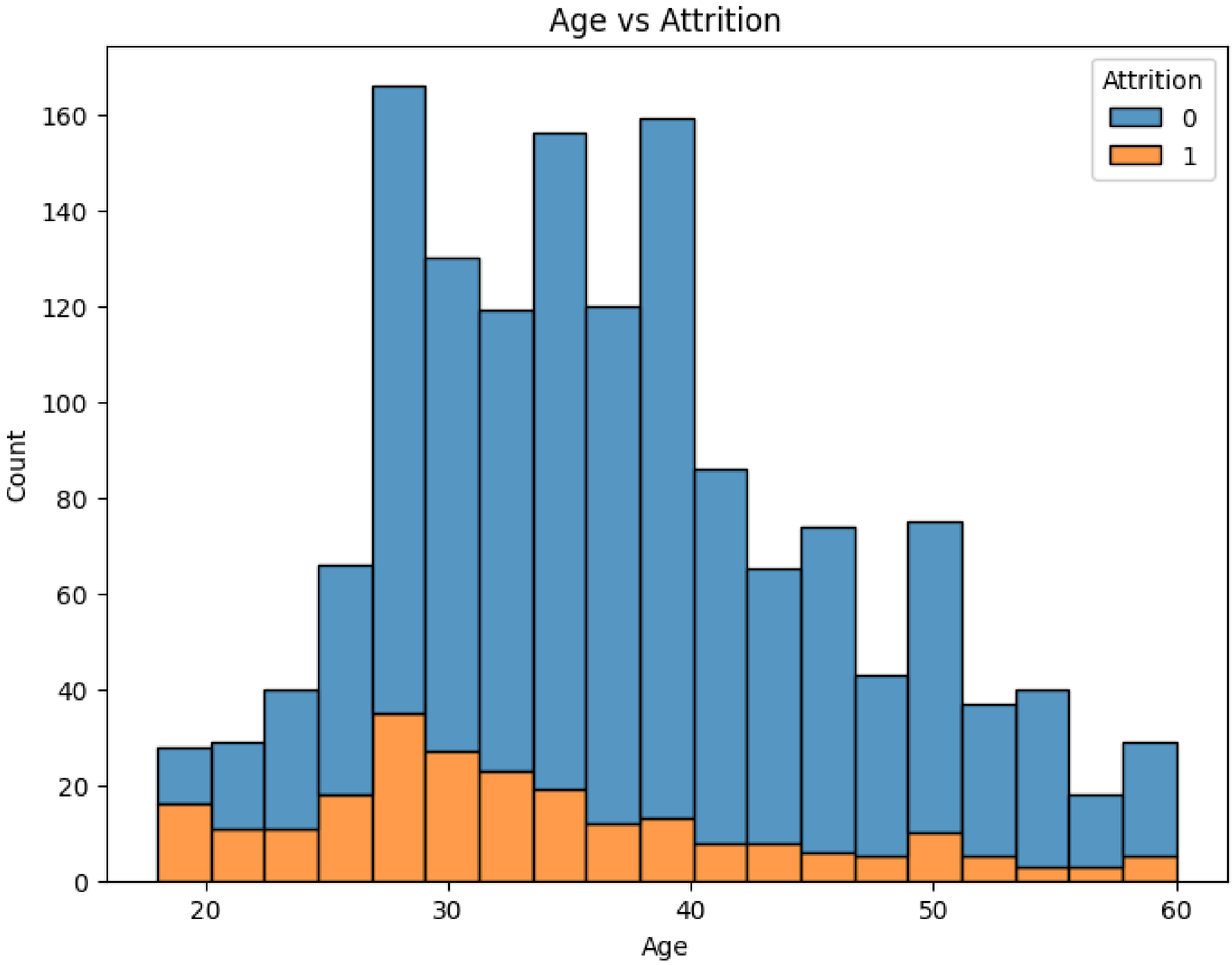
```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.1
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dis
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-package
```

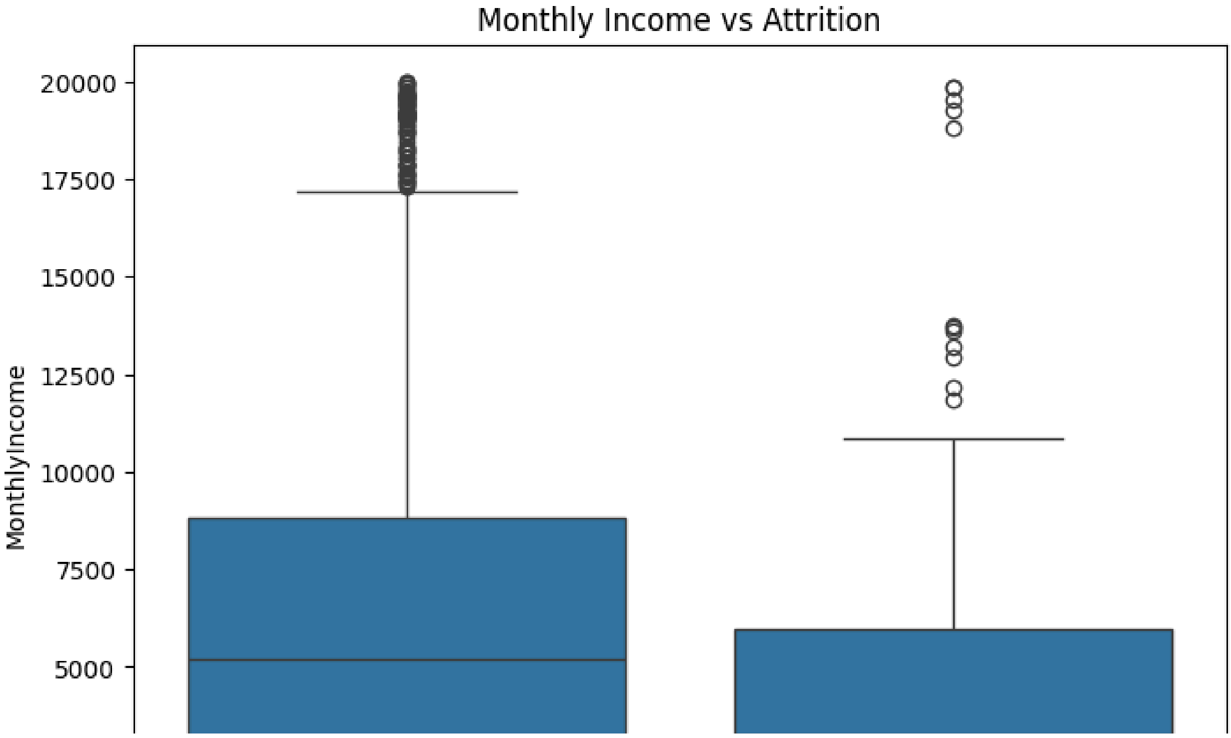
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.preprocessing import LabelEncoder

# --- 1. Load Data Safely ---
# This handles messy CSVs with inconsistent commas
import csv

file_path = '/content/drive/MyDrive/HR_Analytics.csv'

with open(file_path, encoding='utf-8') as f:
    reader = csv.reader(f)
    data = list(reader)

# Extract header and rows
header = data[0]
rows = data[1:]

# Create DataFrame
df = pd.DataFrame(rows, columns=header)

# --- 2. Clean and Convert Columns ---
# Convert numeric columns
numeric_cols = ['Age', 'DailyRate', 'MonthlyIncome', 'HourlyRate', 'YearsAtCompany']
for col in numeric_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')
```

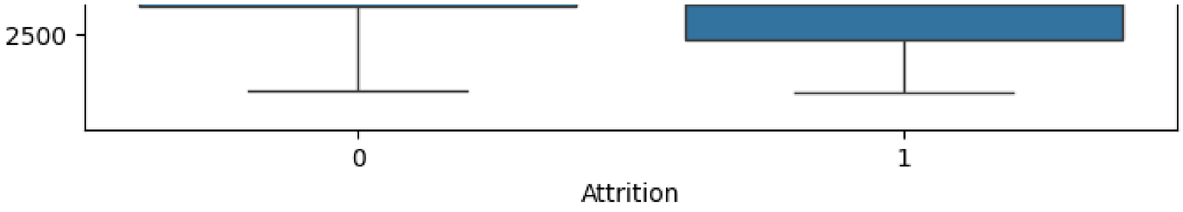
```
# Fill missing values
df['Attrition'] = df['Attrition'].map({'Yes': 1, 'No': 0})
df = df.dropna(subset=['Attrition'])  # Remove rows without Attrition info

# Encode categorical variables
le = LabelEncoder()
cat_cols = ['Department', 'Gender', 'JobRole', 'MaritalStatus']
for col in cat_cols:
    df[col] = le.fit_transform(df[col].astype(str))

# --- 3. Define Features and Target ---
X = df[['Age', 'MonthlyIncome', 'Department', 'YearsAtCompany', 'JobRole', 'Gender']]
y = df['Attrition']

# --- 4. Train Model ---
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

preds = model.predict(X_test)
print("Model Evaluation:")
print(classification_report(y_test, preds))

# --- 5. Visualize Key Factors ---

# Age vs Attrition
plt.figure(figsize=(8,6))
sns.histplot(data=df, x='Age', hue='Attrition', multiple='stack')
plt.title('Age vs Attrition')
plt.show()

# Monthly Income vs Attrition
plt.figure(figsize=(8,6))
sns.boxplot(x='Attrition', y='MonthlyIncome', data=df)
plt.title('Monthly Income vs Attrition')
plt.show()

# Department vs Attrition
plt.figure(figsize=(10,6))
sns.barplot(x='Department', y='Attrition', data=df, estimator=np.mean)
plt.title('Attrition Rate by Department')
plt.xticks(rotation=45)
plt.show()

# Feature Importance
importances = model.feature_importances_
features = pd.DataFrame({'Feature': X.columns, 'Importance': importances})
features = features.sort_values(by='Importance', ascending=False)

plt.figure(figsize=(8,6))
sns.barplot(x='Importance', y='Feature', data=features)
plt.title('Key Factors Influencing Attrition')
```

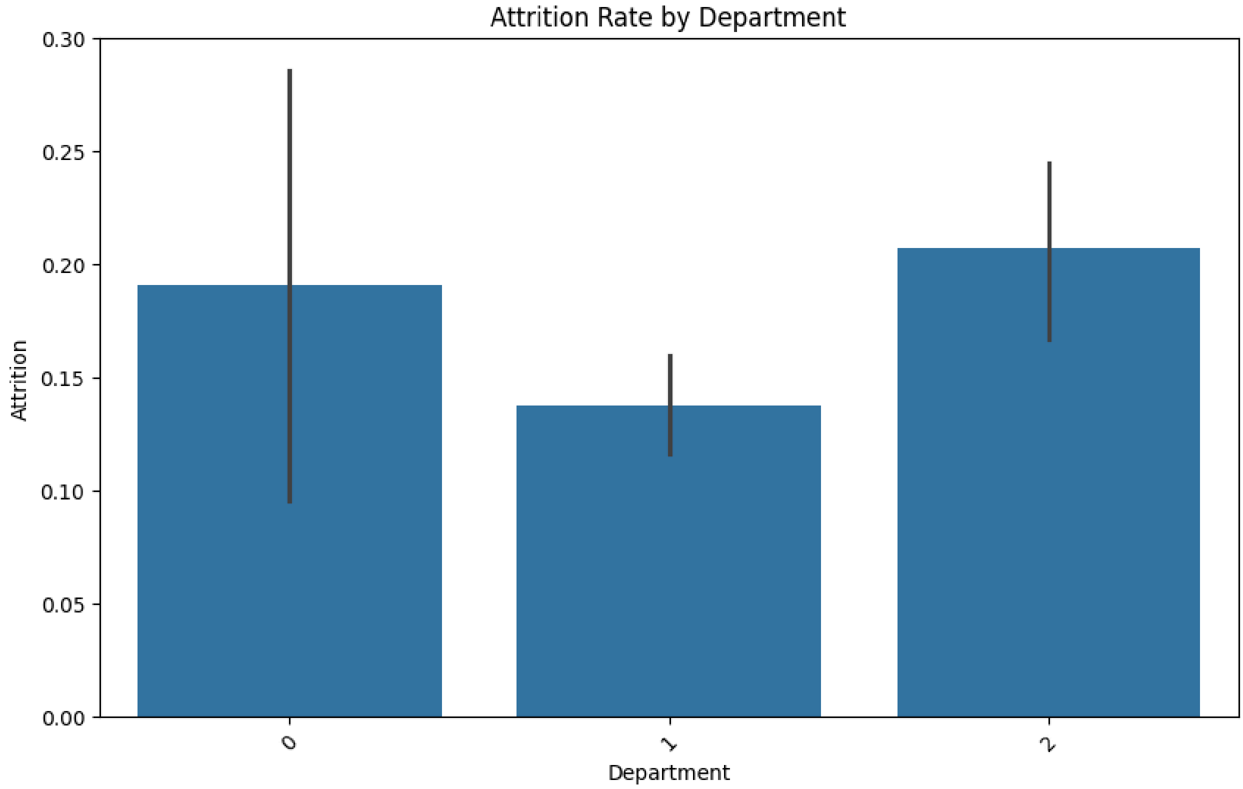
```
    plt.tight_layout()
    plt.show()
```

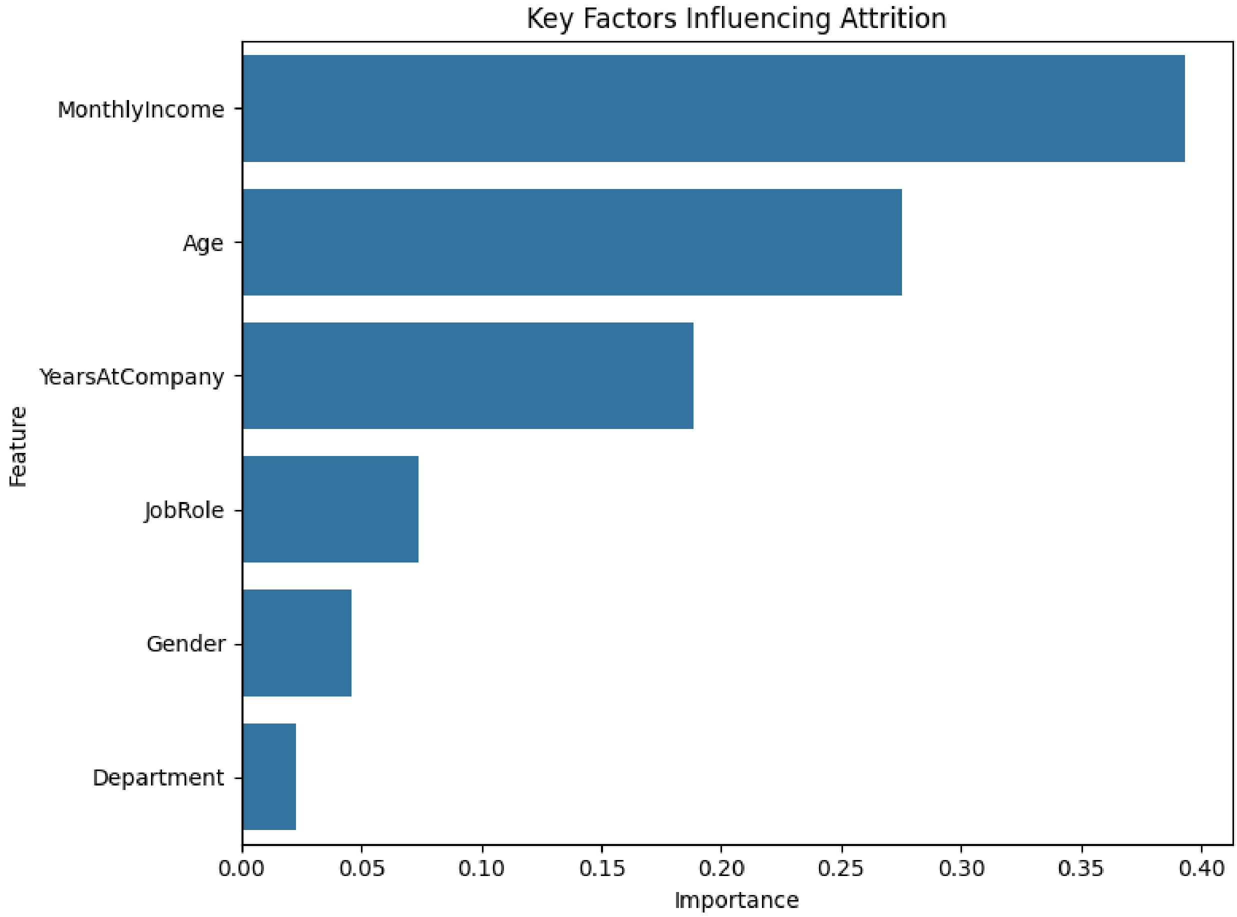
```
Model Evaluation:
              precision    recall  f1-score   support

           0       0.85      0.95      0.90       249
           1       0.29      0.11      0.16        47

    accuracy                           0.82       296
   macro avg       0.57      0.53      0.53       296
weighted avg       0.76      0.82      0.78       296
```






2500
0
1
Attrition


Attrition Rate by Department
Attrition
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
Department


Key Factors Influencing Attrition
Feature
MonthlyIncome
Age
YearsAtCompany
JobRole
Gender
Department
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Importance

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

```
df = pd.read_csv('/content/drive/MyDrive/HR_Analytics.csv')
df.head()
df.info()
df['Attrition'].value_counts()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1480 entries, 0 to 1479
Data columns (total 38 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   EmpID                     1480 non-null   object 
 1   Age                       1480 non-null   int64  
 2   AgeGroup                  1480 non-null   object 
 3   Attrition                 1480 non-null   object 
 4   BusinessTravel            1480 non-null   object 
 5   DailyRate                 1480 non-null   int64  
 6   Department                1480 non-null   object 
 7   DistanceFromHome          1480 non-null   int64  
 8   Education                 1480 non-null   int64  
 9   EducationField            1480 non-null   object 
 10  EmployeeCount             1480 non-null   int64  
 11  EmployeeNumber            1480 non-null   int64  
 12  EnvironmentSatisfaction   1480 non-null   int64  
 13  Gender                    1480 non-null   object 
 14  HourlyRate                1480 non-null   int64  
 15  JobInvolvement            1480 non-null   int64  
 16  JobLevel                  1480 non-null   int64  
 17  JobRole                   1480 non-null   object 
 18  JobSatisfaction           1480 non-null   int64  
 19  MaritalStatus             1480 non-null   object 
 20  MonthlyIncome             1480 non-null   int64  
 21  SalarySlab                1480 non-null   object 
 22  MonthlyRate               1480 non-null   int64  
 23  NumCompaniesWorked        1480 non-null   int64  
 24  Over18                    1480 non-null   object 
 25  OverTime                  1480 non-null   object 
 26  PercentSalaryHike         1480 non-null   int64  
 27  PerformanceRating         1480 non-null   int64  
 28  RelationshipSatisfaction  1480 non-null   int64  
 29  StandardHours             1480 non-null   int64  
 30  StockOptionLevel          1480 non-null   int64  
 31  TotalWorkingYears         1480 non-null   int64  
 32  TrainingTimesLastYear     1480 non-null   int64  
 33  WorkLifeBalance           1480 non-null   int64  
 34  YearsAtCompany            1480 non-null   int64  
 35  YearsInCurrentRole        1480 non-null   int64  
 36  YearsSinceLastPromotion   1480 non-null   int64  
 37  YearsWithCurrManager      1423 non-null   float64
dtypes: float64(1), int64(25), object(12)
memory usage: 439.5+ KB
```

	count
Attrition	
No	1242
Yes	238

dtype: int64

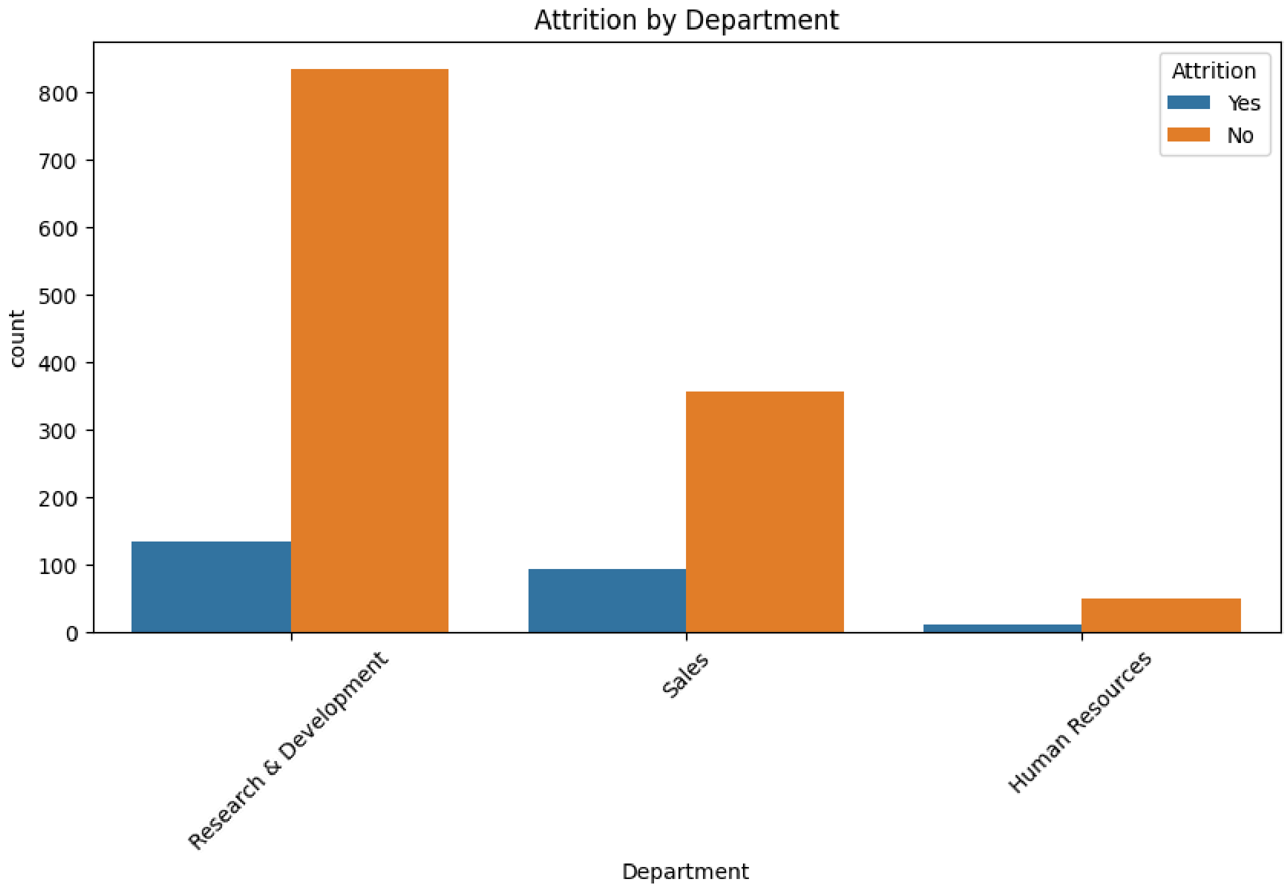
```
attrition_rate = df['Attrition'].value_counts(normalize=True)['Yes'] * 100
print(f"Attrition Rate: {attrition_rate:.2f}%")
```

```
Attrition Rate: 16.08%
```

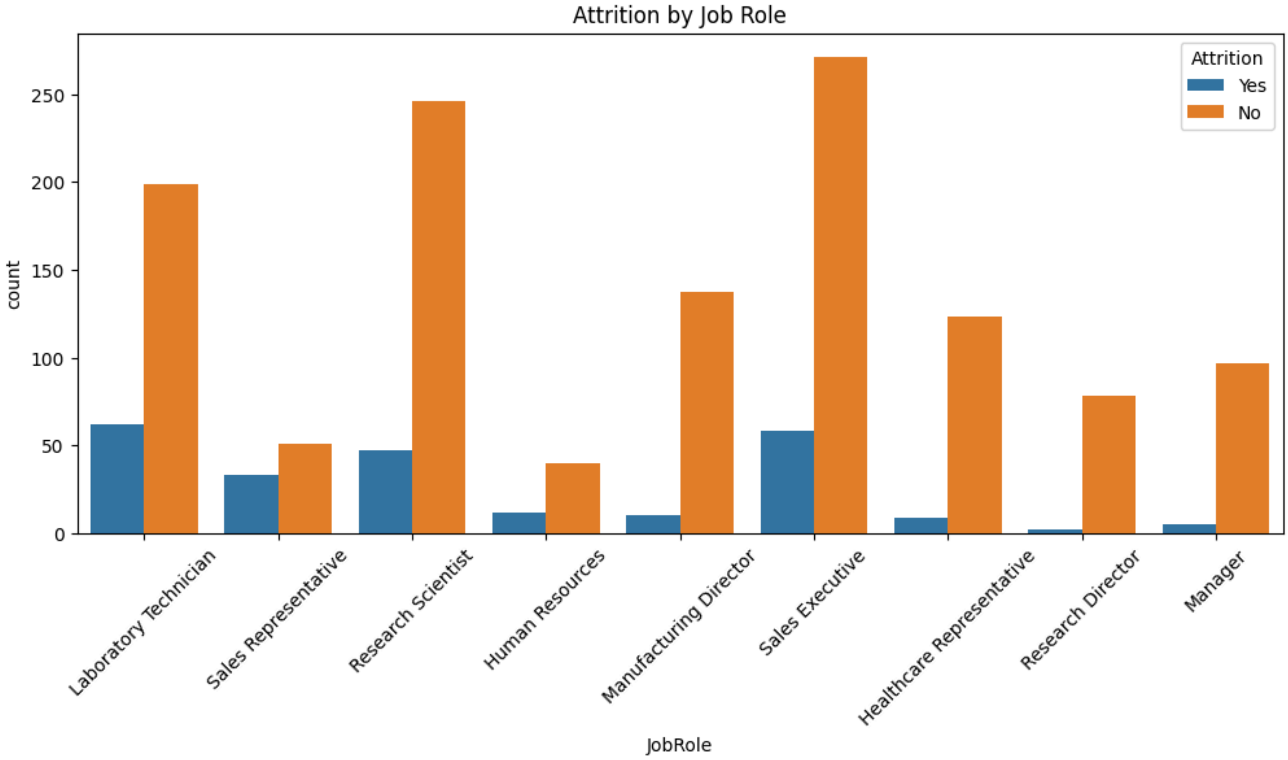
```
avg_tenure_leavers = df[df['Attrition'] == 'Yes']['YearsAtCompany'].mean()
```

```
dept_attrition = df[df['Attrition'] == 'Yes']['Department'].value_counts()
```

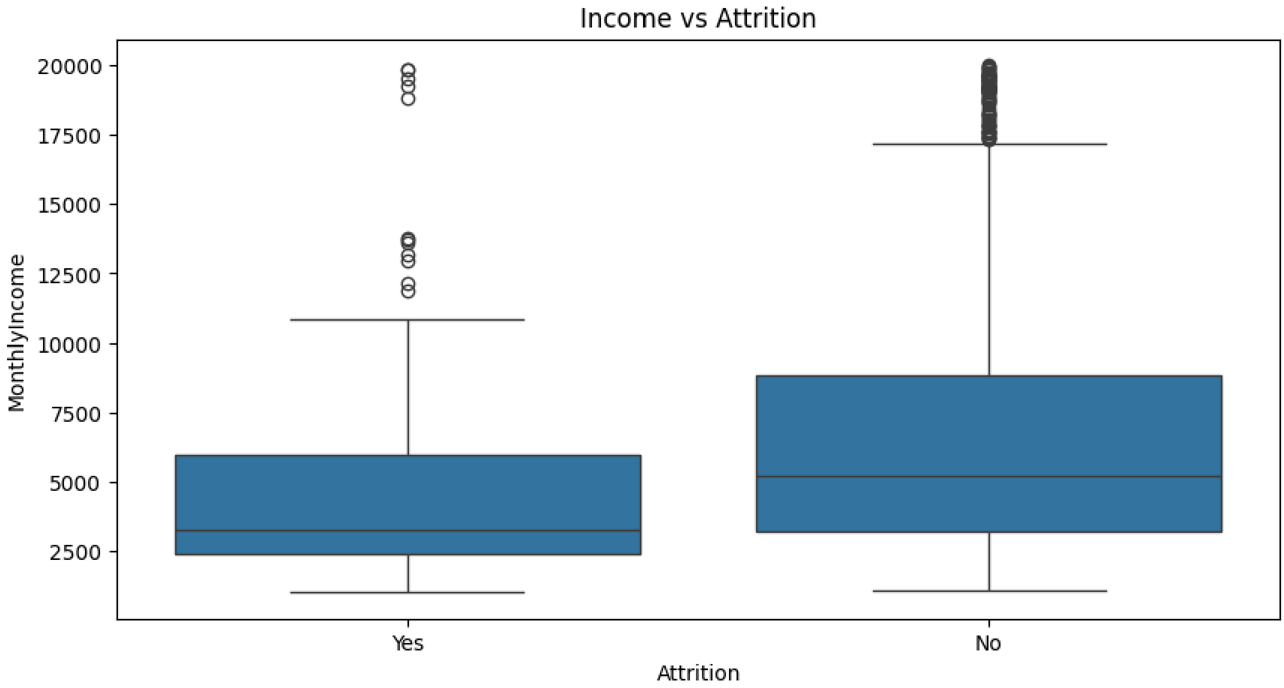
```
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='Department', hue='Attrition')
plt.title("Attrition by Department")
plt.xticks(rotation=45)
plt.show()
```



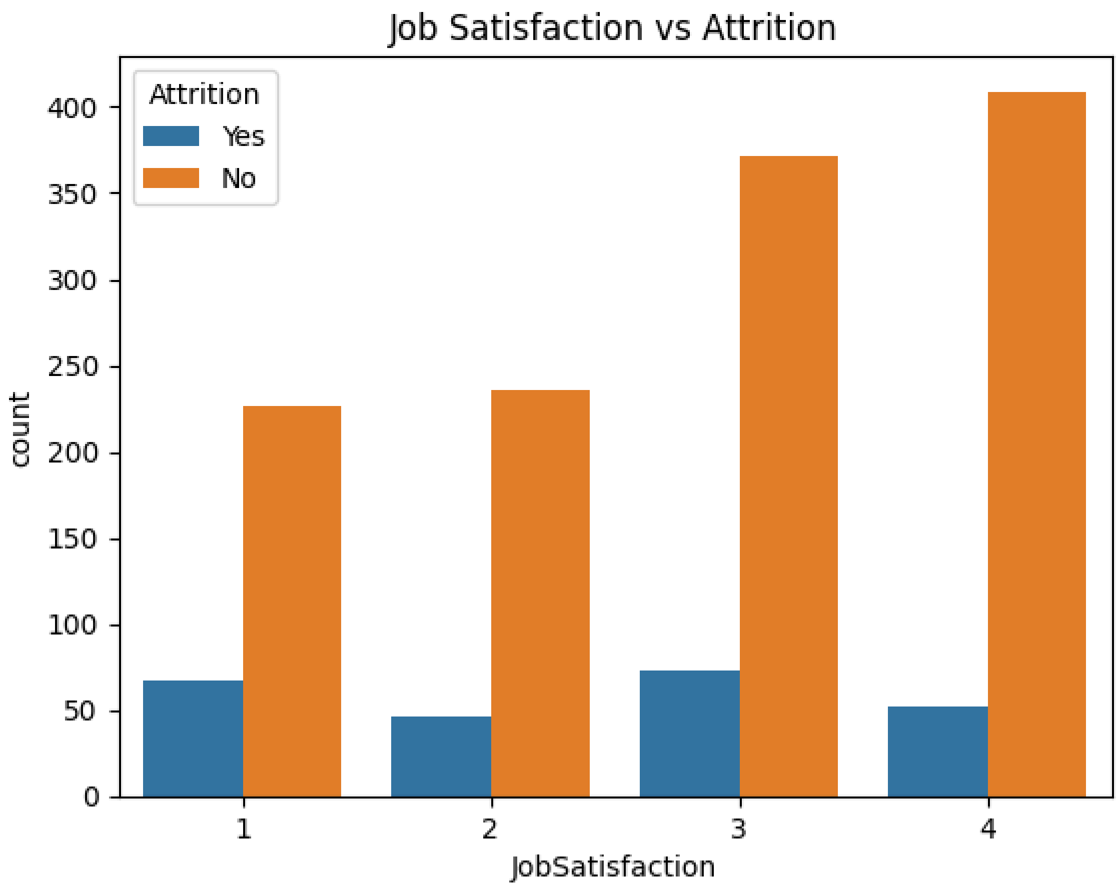
```
plt.figure(figsize=(12, 5))
sns.countplot(data=df, x='JobRole', hue='Attrition')
plt.title("Attrition by Job Role")
plt.xticks(rotation=45)
plt.show()
```



```
plt.figure(figsize=(10, 5))
sns.boxplot(data=df, x='Attrition', y='MonthlyIncome')
plt.title("Income vs Attrition")
plt.show()
```



```
sns.countplot(data=df, x='JobSatisfaction', hue='Attrition')
plt.title("Job Satisfaction vs Attrition")
plt.show()
```



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

# Load the dataset
df = pd.read_csv('/content/drive/MyDrive/HR_Analytics.csv')  # Update path if needed
df.head()
```

	EmpID	Age	AgeGroup	Attrition	BusinessTravel	DailyRate	Department	Distan
0	RM297	18	18-25	Yes	Travel_Rarely	230	Research & Development	
1	RM302	18	18-25	No	Travel_Rarely	812	Sales	
2	RM458	18	18-25	Yes	Travel_Frequently	1306	Sales	
3	RM728	18	18-25	No	Non-Travel	287	Research & Development	
4	RM829	18	18-25	Yes	Non-Travel	247	Research & Development	

5 rows × 38 columns

```
# 1. Attrition Rate
attrition_rate = df['Attrition'].value_counts(normalize=True)['Yes'] * 100
print(f"✅ Attrition Rate: {attrition_rate:.2f}%")
```