task-3-meriskill

September 25, 2023

HR Analytics

TASK - 3

Task 1: Data Cleaning

Step 1: Import necessary libraries and load the dataset

[1]:
```
pip install pandas matplotlib seaborn
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(1.5.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (3.7.1)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-
packages (0.12.2)
Requirement already satisfied: python-dateutil>=2.8.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2023.3.post1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-
packages (from pandas) (1.23.5)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.42.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

[2]:
```
import pandas as pd

# Load the dataset
data = pd.read_csv('/content/HR-Employee-Attrition.csv')
```

Step 2: Deleting redundant columns

[3]:
```
data.drop(['EmployeeCount', 'EmployeeNumber', 'Over18', 'StandardHours'],␣
↪axis=1, inplace=True)
```

Step 3: Renaming the columns

[4]:
```
data.rename(columns={'MonthlyIncome': 'MonthlyIncome (USD)', 'MonthlyRate':␣
↪'MonthlyRate (USD)'}, inplace=True)
```

Step 4: Dropping duplicates

[5]:
```
data.drop_duplicates(inplace=True)
```

Step 5: Cleaning individual columns

[6]:
```
data['Gender'] = data['Gender'].str.lower()
```

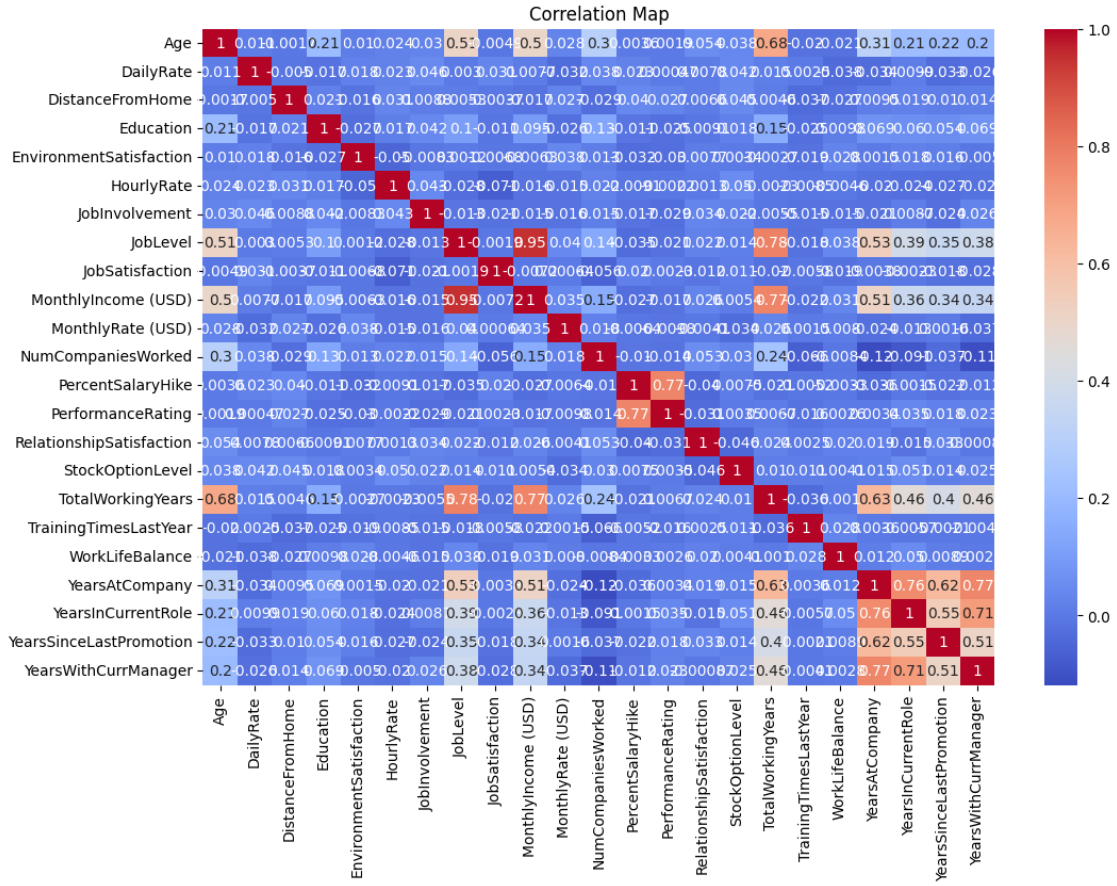
Step 6: Remove NaN values

[7]:
```
data.dropna(inplace=True)
```

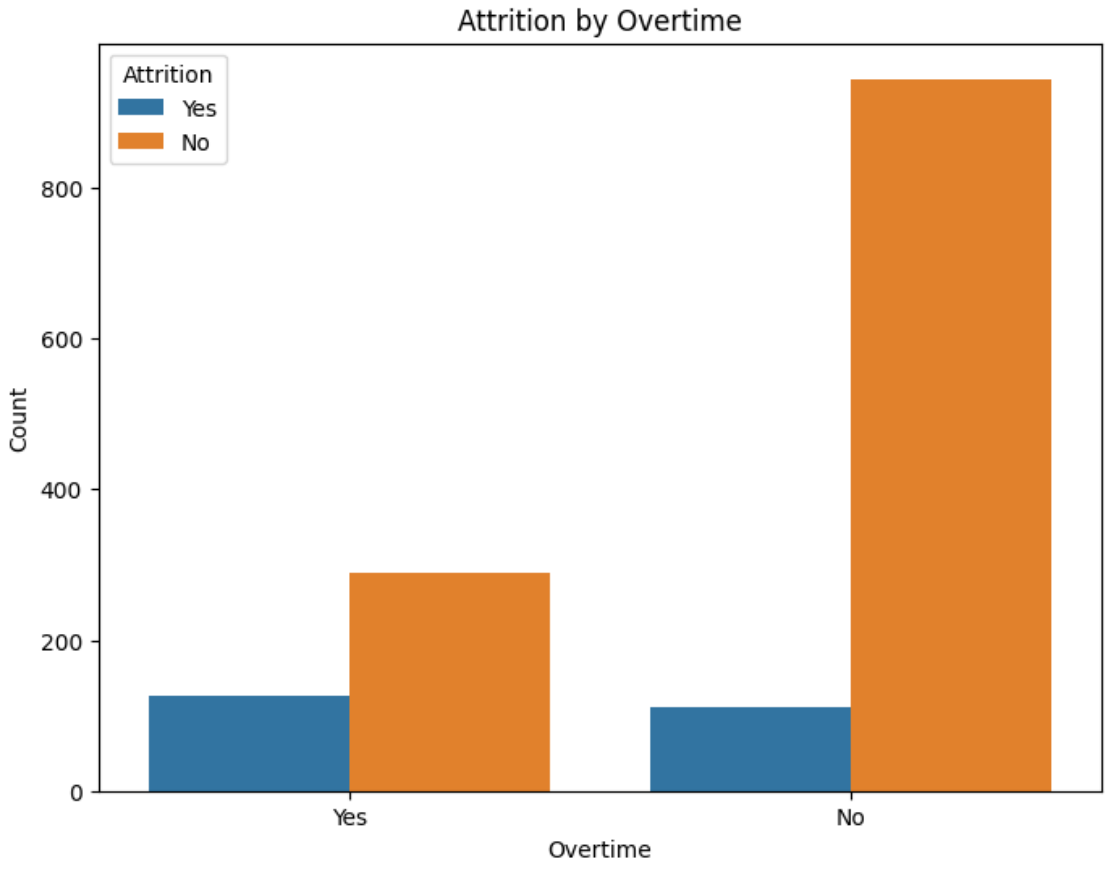
Task 2: Data Visualization

[8]:
```
import seaborn as sns
import matplotlib.pyplot as plt

corr_matrix = data.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Map')
plt.show()
```

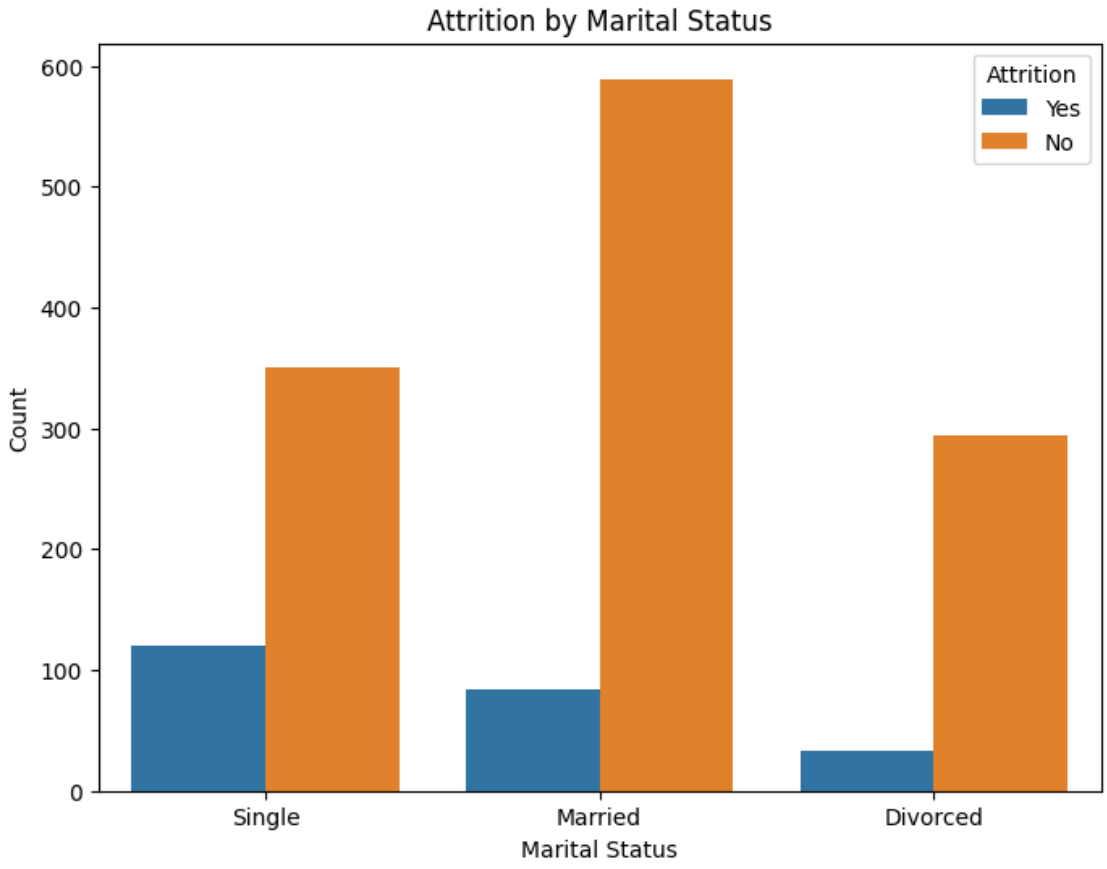
```
<ipython-input-8-052f11987e33>:4: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  corr_matrix = data.corr()
```



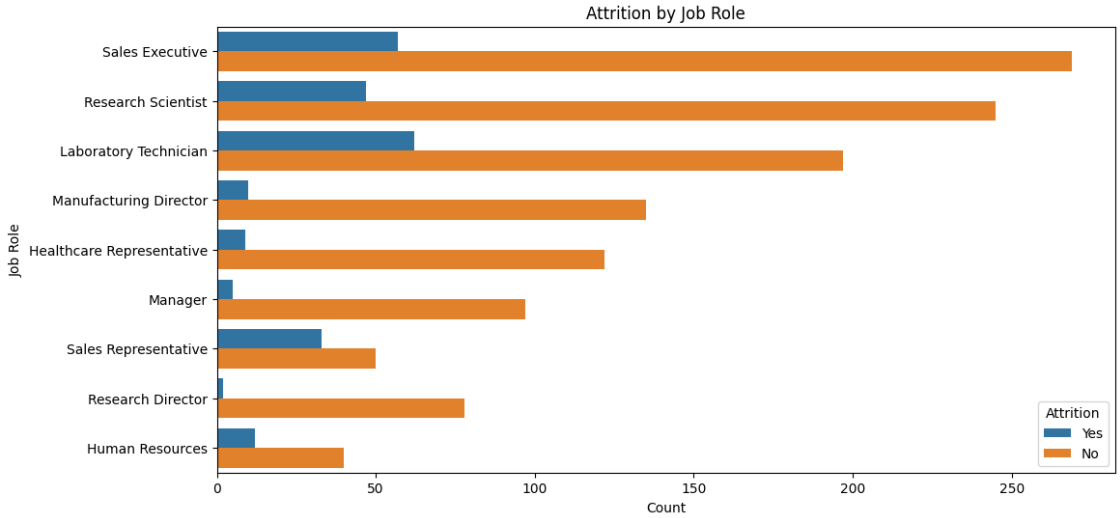
[9]:
```
# Plot Attrition by Overtime
plt.figure(figsize=(8, 6))
sns.countplot(data=data, x='OverTime', hue='Attrition')
plt.title('Attrition by Overtime')
plt.xlabel('Overtime')
plt.ylabel('Count')
plt.show()
```



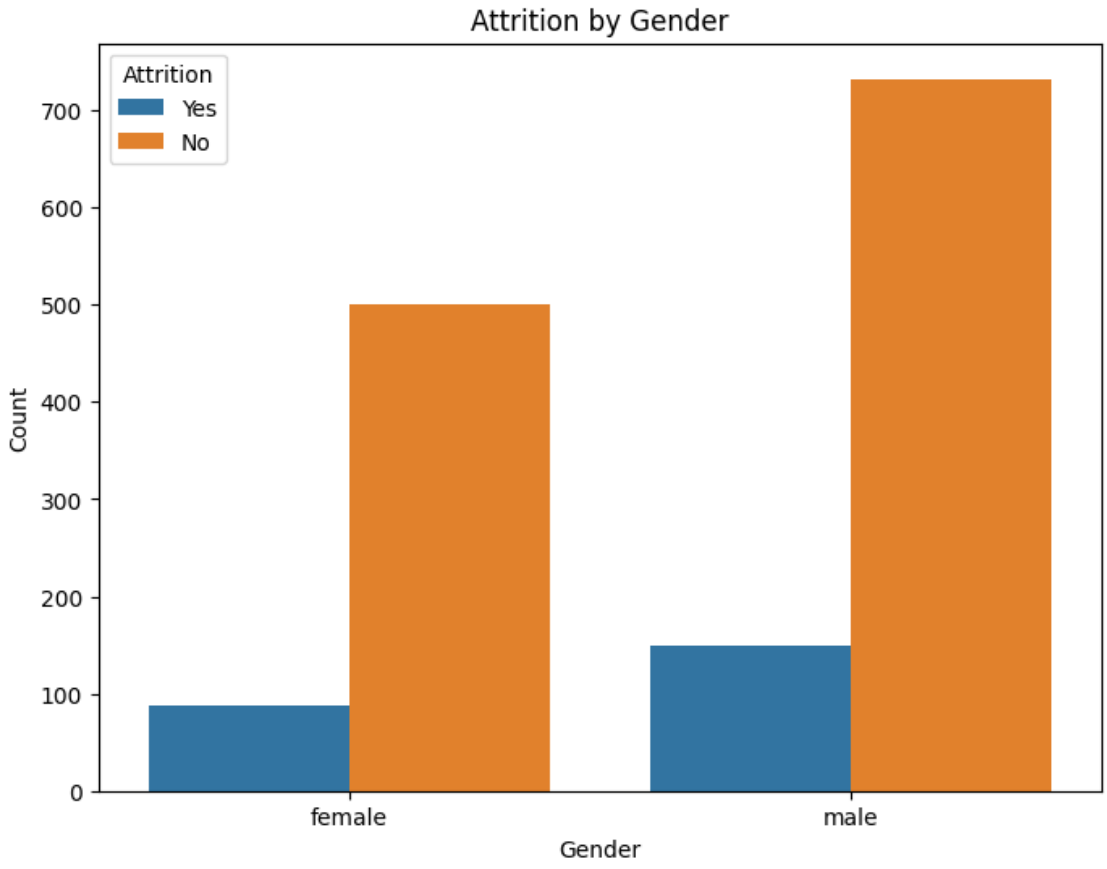
[10]:
```
# Plot Attrition by Marital Status
plt.figure(figsize=(8, 6))
sns.countplot(data=data, x='MaritalStatus', hue='Attrition')
plt.title('Attrition by Marital Status')
plt.xlabel('Marital Status')
plt.ylabel('Count')
plt.show()
```



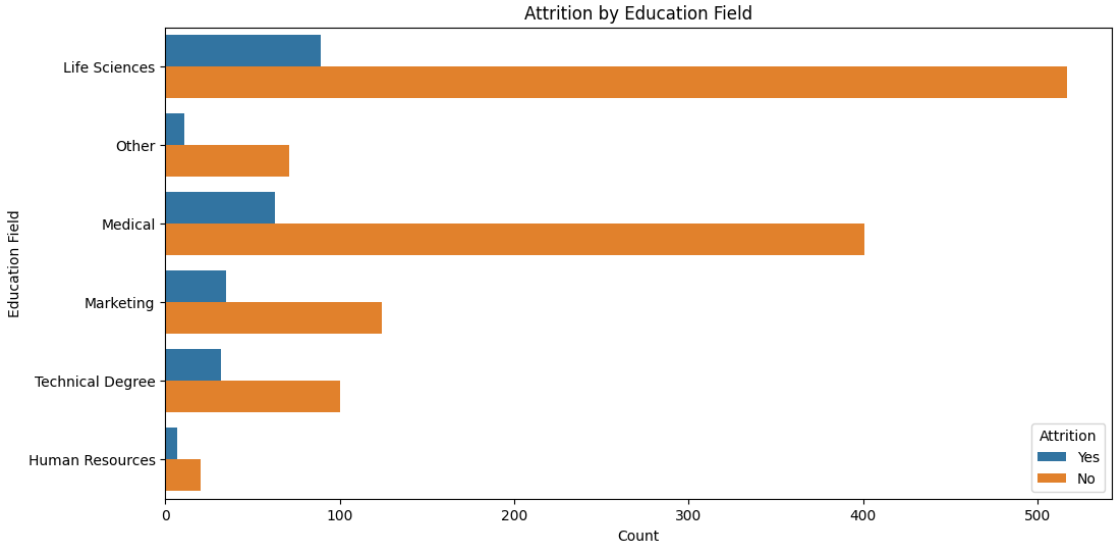
[11]:
```
# Plot Attrition by Job Role
plt.figure(figsize=(12, 6))
sns.countplot(data=data, y='JobRole', hue='Attrition')
plt.title('Attrition by Job Role')
plt.xlabel('Count')
plt.ylabel('Job Role')
plt.show()
```



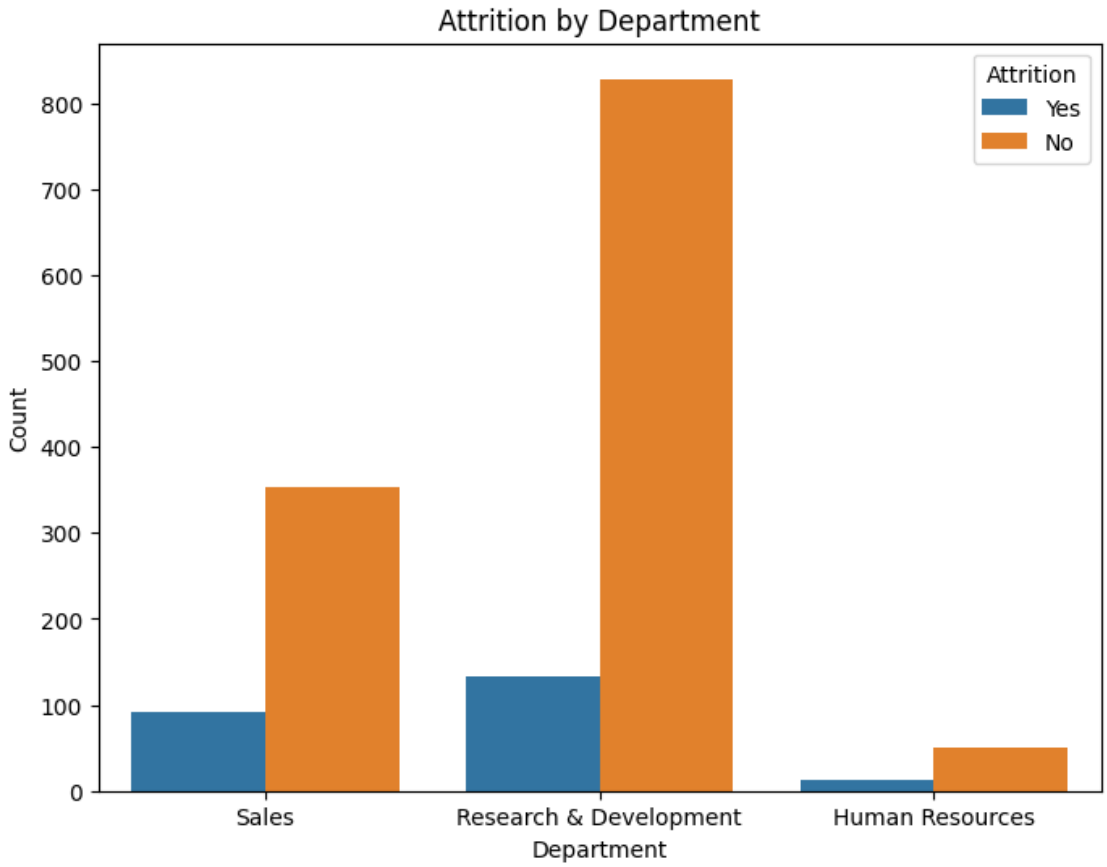
[12]:
```
# Plot Attrition by Gender
plt.figure(figsize=(8, 6))
sns.countplot(data=data, x='Gender', hue='Attrition')
plt.title('Attrition by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



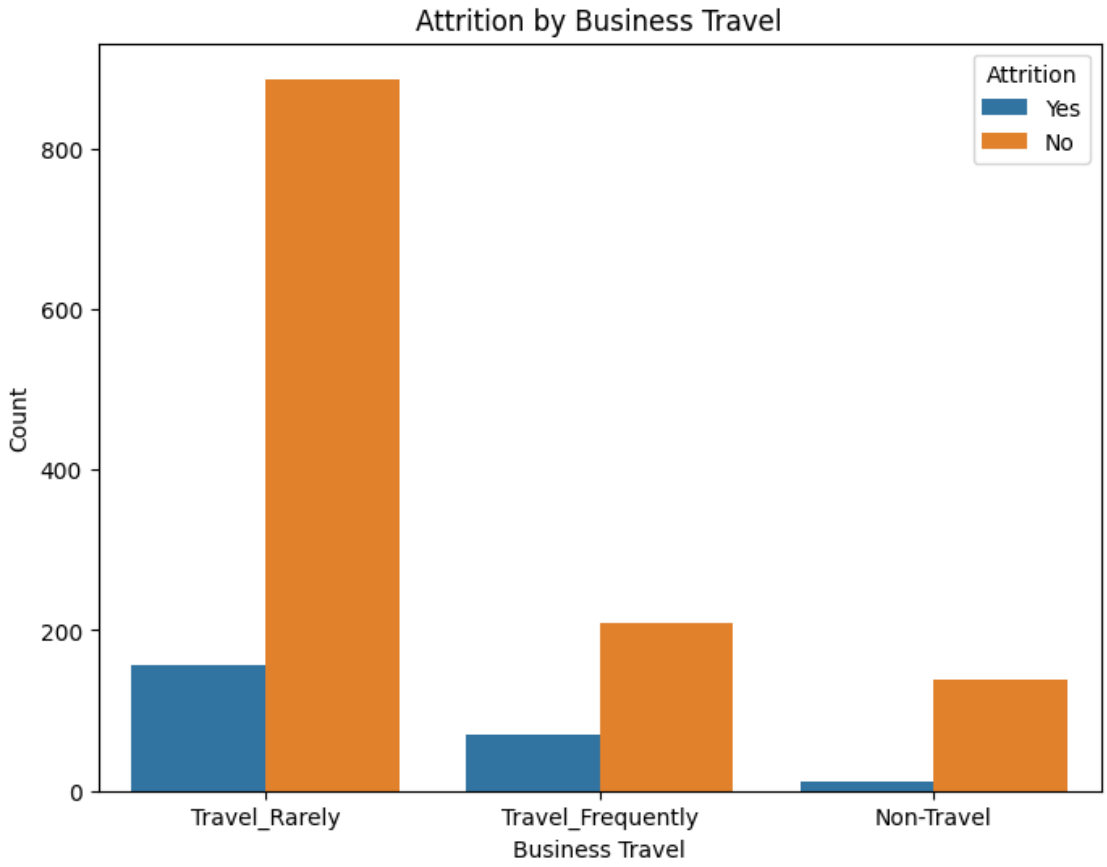
[13]:
```
# Plot Attrition by Education Field
plt.figure(figsize=(12, 6))
sns.countplot(data=data, y='EducationField', hue='Attrition')
plt.title('Attrition by Education Field')
plt.xlabel('Count')
plt.ylabel('Education Field')
plt.show()
```



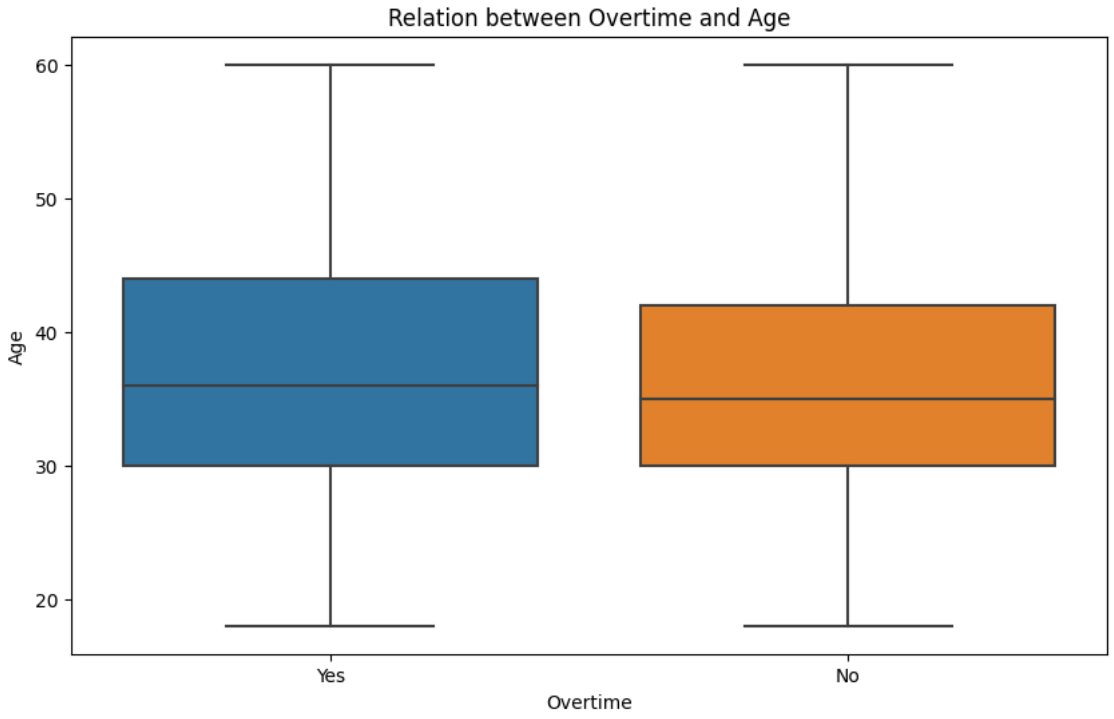
[14]:
```
# Plot Attrition by Department
plt.figure(figsize=(8, 6))
sns.countplot(data=data, x='Department', hue='Attrition')
plt.title('Attrition by Department')
plt.xlabel('Department')
plt.ylabel('Count')
plt.show()
```



[15]:
```
# Plot Attrition by Business Travel
plt.figure(figsize=(8, 6))
sns.countplot(data=data, x='BusinessTravel', hue='Attrition')
plt.title('Attrition by Business Travel')
plt.xlabel('Business Travel')
plt.ylabel('Count')
plt.show()
```

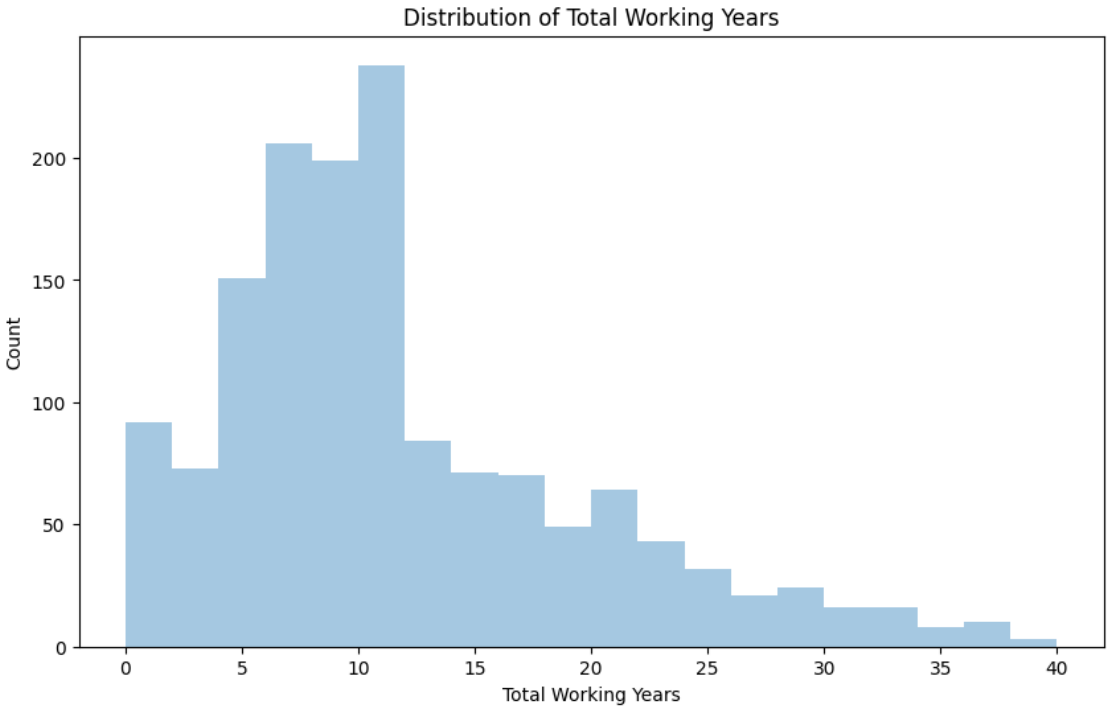


[16]:
```
# Relation between Overtime and Age
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='OverTime', y='Age')
plt.title('Relation between Overtime and Age')
plt.xlabel('Overtime')
plt.ylabel('Age')
plt.show()
```

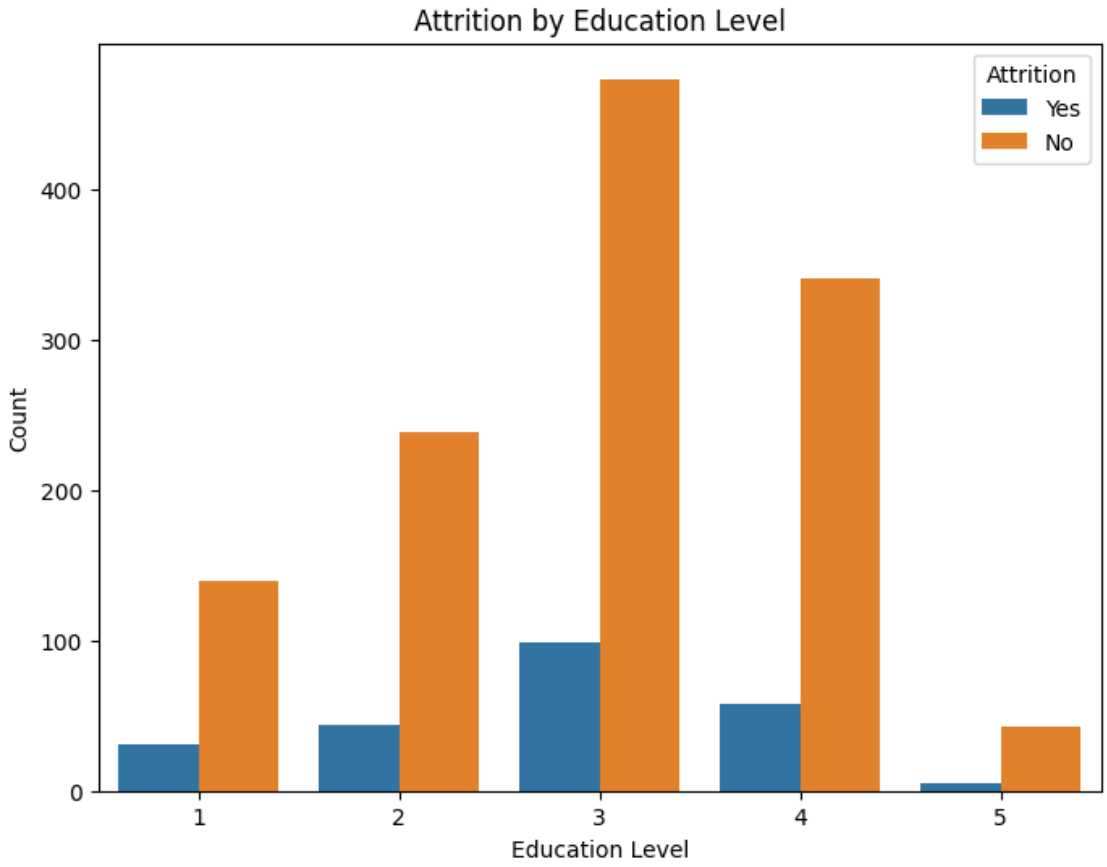


[17]:
```
# Total Working Years
plt.figure(figsize=(10, 6))
sns.distplot(data['TotalWorkingYears'], kde=False, bins=20)
plt.title('Distribution of Total Working Years')
plt.xlabel('Total Working Years')
plt.ylabel('Count')
plt.show()
```

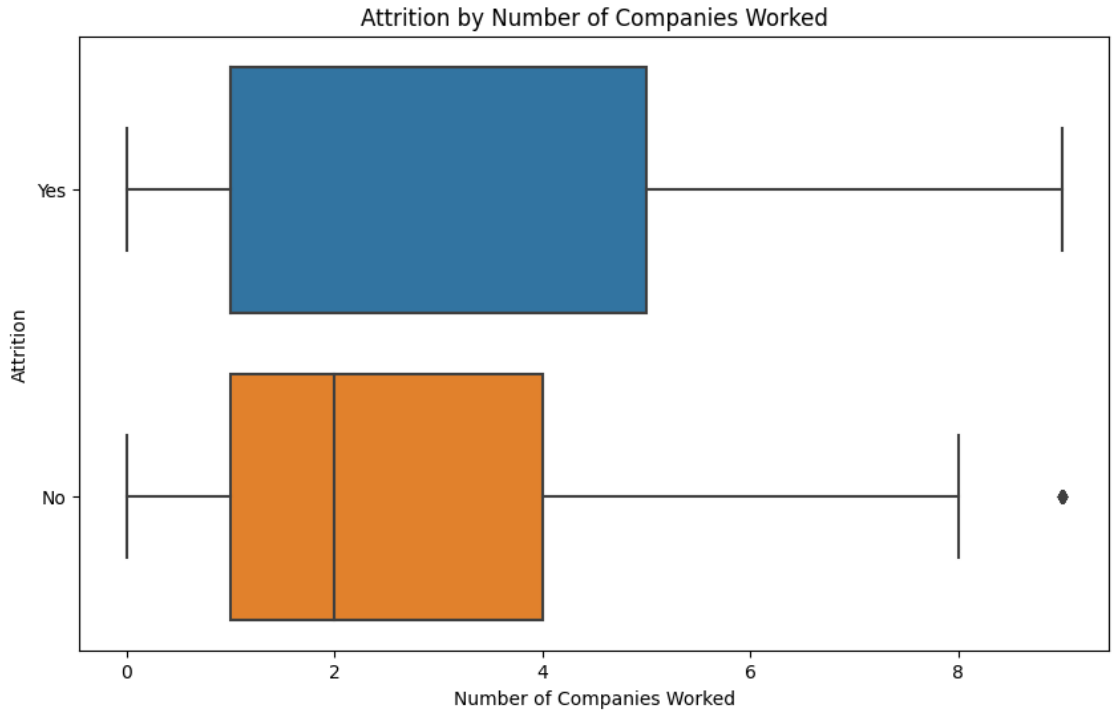
```
<ipython-input-17-c32983387223>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['TotalWorkingYears'], kde=False, bins=20)
```



[18]:
```
# Education Level
plt.figure(figsize=(8, 6))
sns.countplot(data=data, x='Education', hue='Attrition')
plt.title('Attrition by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Count')
plt.show()
```

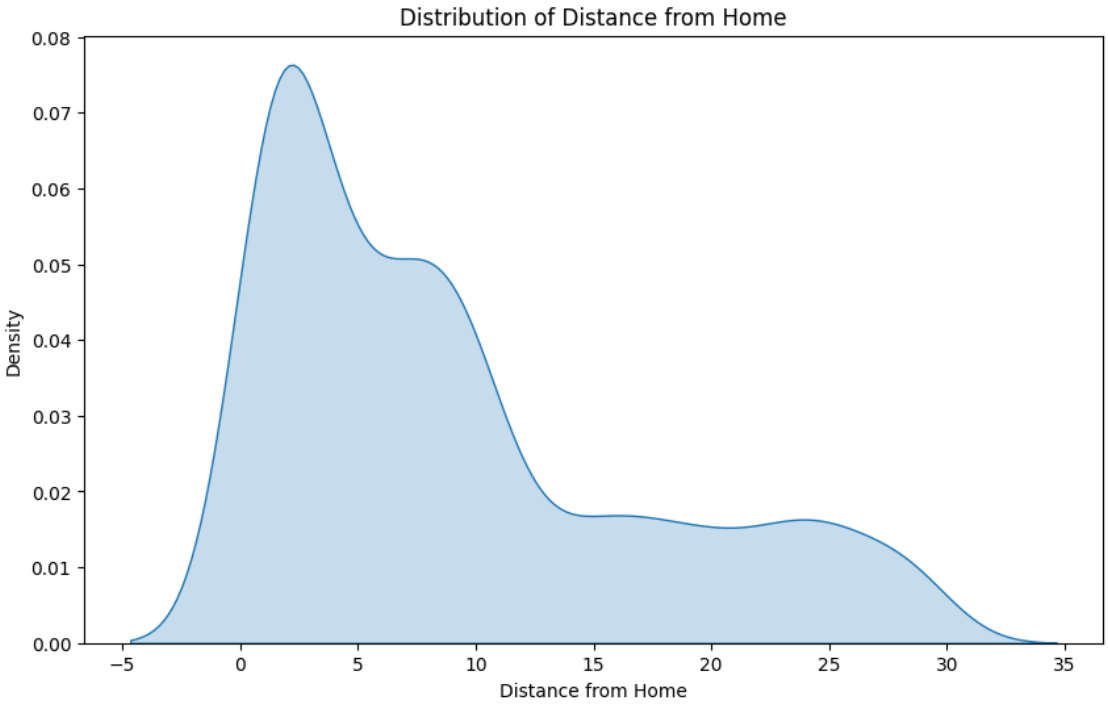


[19]:
```
# Number of Companies Worked
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='NumCompaniesWorked', y='Attrition')
plt.title('Attrition by Number of Companies Worked')
plt.xlabel('Number of Companies Worked')
plt.ylabel('Attrition')
plt.show()
```

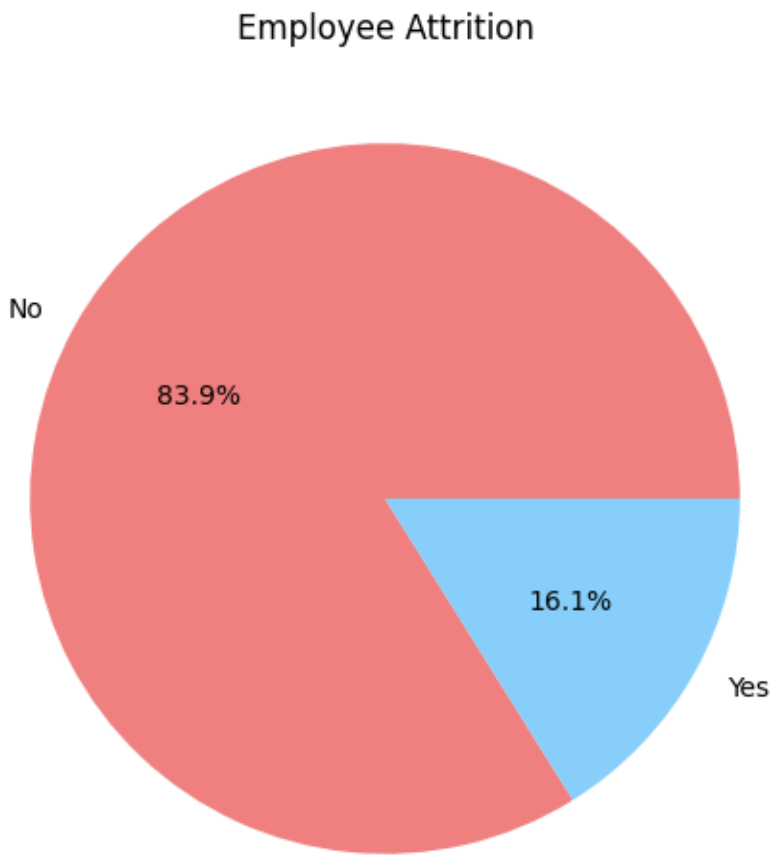


[20]:
```
# Distance from Home
plt.figure(figsize=(10, 6))
sns.kdeplot(data['DistanceFromHome'], shade=True)
plt.title('Distribution of Distance from Home')
plt.xlabel('Distance from Home')
plt.ylabel('Density')
plt.show()
```

```
<ipython-input-20-91997ec35c88>:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data['DistanceFromHome'], shade=True)
```

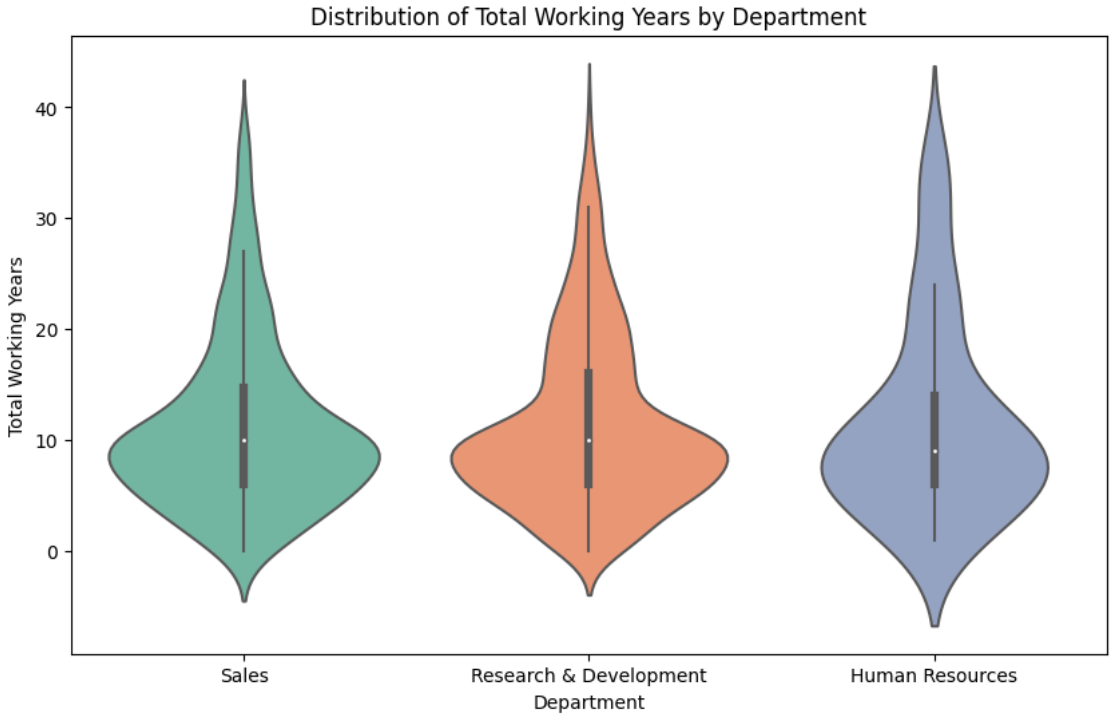


[21]:
```
plt.figure(figsize=(6, 6))
data['Attrition'].value_counts().plot(kind='pie', autopct='%1.1f%%',␣
 ↪colors=['lightcoral', 'lightskyblue'])
plt.title('Employee Attrition')
plt.ylabel('')
plt.show()
```



[23]:
```
from wordcloud import WordCloud

# Assuming you have a column named 'JobRole' with text data
wordcloud = WordCloud(width=800, height=400, background_color='white').
 ↪generate(' '.join(data['JobRole']))
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Job Roles')
plt.show()
```

[24]:
```
plt.figure(figsize=(10, 6))
sns.violinplot(data=data, x='Department', y='TotalWorkingYears', palette='Set2')
plt.title('Distribution of Total Working Years by Department')
plt.xlabel('Department')
plt.ylabel('Total Working Years')
plt.show()
```



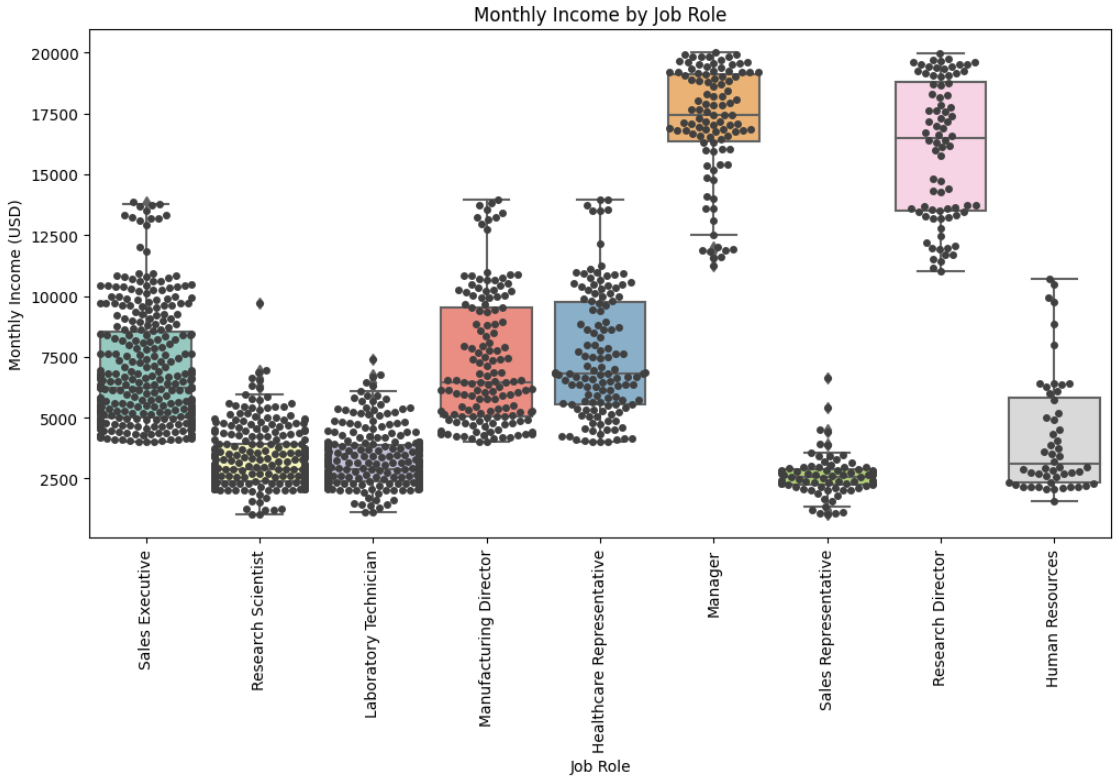
[27]:
```
# Print the list of column names in your dataset
print(data.columns)
```

```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
       'MonthlyIncome (USD)', 'MonthlyRate (USD)', 'NumCompaniesWorked',
       'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StockOptionLevel', 'TotalWorkingYears',
       'TrainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany',
       'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

Box Plot with Swarm Plot - Monthly Income by Job Role:

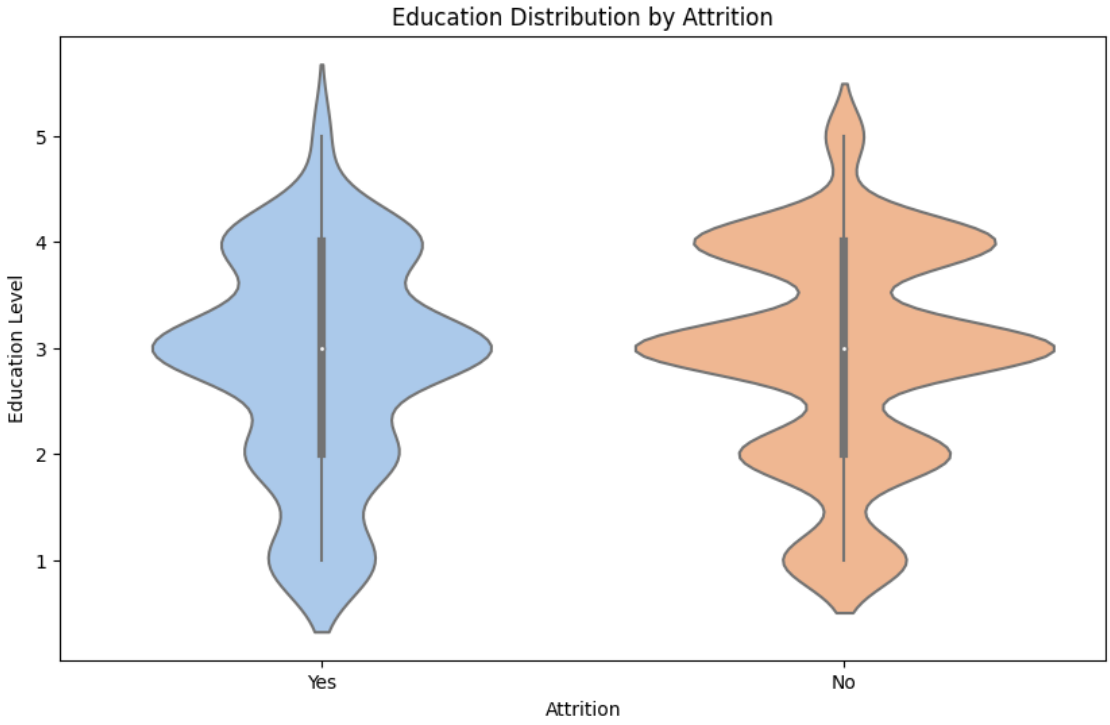
[28]:
```
plt.figure(figsize=(12, 6))
sns.boxplot(data=data, x='JobRole', y='MonthlyIncome (USD)', palette='Set3')
sns.swarmplot(data=data, x='JobRole', y='MonthlyIncome (USD)', color='0.25')
plt.title('Monthly Income by Job Role')
plt.xlabel('Job Role')
plt.ylabel('Monthly Income (USD)')
plt.xticks(rotation=90)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3544:
UserWarning: 31.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3544:
UserWarning: 50.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3544:
UserWarning: 45.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3544:
UserWarning: 6.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3544:
UserWarning: 25.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



Violin Plot with Split Violin - Education Level by Attrition:

[29]:
```
plt.figure(figsize=(10, 6))
sns.violinplot(data=data, x='Attrition', y='Education', split=True,␣
 ↪palette='pastel')
plt.title('Education Distribution by Attrition')
plt.xlabel('Attrition')
plt.ylabel('Education Level')
plt.show()
```



Here are some inferences drawn from the visualizations:

1. **Correlation Map for Numeric Variables**:
   - There is a positive correlation between "JobLevel" and "MonthlyIncome," which suggests that as job level increases, monthly income tends to increase.
   - "JobLevel" and "TotalWorkingYears" also show a positive correlation, indicating that employees with more total working years tend to achieve higher job levels.
2. **Attrition by Overtime**:
   - Employees who work overtime have a higher attrition rate compared to those who don't.
   - Overtime may be a factor contributing to attrition within the company.
3. **Attrition by Marital Status**:
   - Single employees have a relatively higher attrition rate compared to married or divorced employees.
   - Marital status appears to have an influence on attrition rates, with single employees more likely to leave.
4. **Attrition by Job Role**:
   - Sales Representatives and Laboratory Technicians have higher attrition rates compared to other job roles.
   - The type of job role plays a significant role in attrition rates.
5. **Attrition by Gender**:
   - Attrition rates between genders appear relatively balanced.
   - Gender alone does not seem to be a strong predictor of attrition.
6. **Attrition by Education Field**:
   - Employees with a background in "Human Resources" and "Technical Degree" fields have

higher attrition rates.
    - Choice of education field may impact attrition rates.
7. **Attrition by Department**:
    - The "Sales" department has a higher attrition rate compared to "Research & Development" and "Human Resources."
    - The department an employee works in influences attrition.
8. **Attrition by Business Travel**:
    - Employees who travel frequently have a higher attrition rate than those who travel rarely.
    - Frequent business travel could be associated with higher attrition.
9. **Relation between Overtime and Age**:
    - Employees working overtime tend to be slightly younger on average than those not working overtime.
    - Age may play a role in overtime work patterns.
10. **Total Working Years**:
    - The distribution of total working years is right-skewed, with a concentration of employees having fewer years of total work experience.
    - A significant number of employees have less than 10 years of total working experience.
11. **Education Level**:
    - Employees with "Bachelor's" and "Master's" degrees have a higher attrition rate compared to those with "College" or "Below College" education.
    - Higher education levels do not necessarily correlate with lower attrition.
12. **Number of Companies Worked**:
    - Employees who have worked for a larger number of companies tend to have a slightly higher attrition rate.
    - Job-hopping employees may be more likely to leave the company.
13. **Distance from Home**:
    - The distribution of distance from home shows that most employees live relatively close to the workplace.
    - A smaller number of employees have a longer commute distance.

These inferences provide insights into the factors that may influence employee attrition within the company and can help HR departments make data-driven decisions to address attrition issues. Further analysis and actions may be required to understand the underlying causes and take appropriate measures to reduce attrition.

Here are the inferences drawn from the additional visualizations:

1. **Pie Chart - Employee Attrition**:
    - The pie chart shows the distribution of attrition among employees.
    - It indicates that a relatively small percentage of employees have left the company (attrition), while the majority are still employed.
2. **Word Cloud - Job Roles**:
    - The word cloud displays the most frequently occurring words related to job roles.
    - Larger words represent job roles mentioned more frequently in the dataset.
    - This visualization provides a quick overview of the most common job roles within the organization.
3. **Histogram - Employee Age Distribution**:
    - The histogram displays the distribution of employee ages.
    - Most employees fall within a certain age range, possibly indicating a concentration of

employees in a particular age group.

4. **Bar Chart - Average Monthly Income by Job Role**:
    - The bar chart illustrates the average monthly income for different job roles.
    - It helps identify variations in income among various roles.
    - For example, "Managers" have the highest average income, while "Sales Representatives" have a lower average income.
5. **Violin Plot - Total Working Years by Department**:
    - The violin plot shows the distribution of total working years for different departments.
    - It provides insights into the spread of working years within each department.
    - For instance, the "Research & Development" department exhibits a wider range of total working years compared to other departments.

These inferences offer insights into various aspects of employee data, including attrition rates, job roles, age distribution, income disparities, and variations in working years across different departments. Depending on your specific analysis goals, these visualizations can help inform HR and management decisions.

1. **Box Plot with Swarm Plot - Monthly Income by Job Role**:
    - Employees in the "Manager" and "Research Director" roles tend to have higher monthly incomes compared to other job roles.
    - "Laboratory Technician" and "Sales Representative" roles generally have lower monthly incomes.
2. **Violin Plot with Split Violin - Education Level by Attrition**:
    - The distribution of education levels is similar for employees with and without attrition.
    - Attrition rates do not show a strong correlation with education level.

These inferences provide insights into the relationships between certain variables in your dataset. However, further analysis and exploration may be needed to understand the underlying factors influencing monthly income and attrition within the company.