Project Employee Attrition Project

In [ ]:

Data Cleaning

In [2]:
```
import pandas as pd
import numpy as np

data = pd.read_csv(r"E:\Unified Mentor\Projects\Employee Attrition Analysis\Attriti
```

In [3]:
```
data.head()
```

Out[3]:

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Educati
0	1	51	No	Travel_Rarely	Sales	6	
1	2	31	Yes	Travel_Frequently	Research & Development	10	
2	3	32	No	Travel_Frequently	Research & Development	17	
3	4	38	No	Non-Travel	Research & Development	2	
4	5	32	No	Travel_Rarely	Research & Development	10	

5 rows × 29 columns

In [4]:
```
# Drop Irrelevant Columns

data.drop(['EmployeeID', 'EmployeeCount', 'Over18', 'StandardHours'], axis=1, inpla
```

In [ ]:

In [5]:
```
data.shape
```

Out[5]: (4410, 25)

In [6]:
```
data.describe()
```

Out[6]:

	Age	DistanceFromHome	Education	JobLevel	MonthlyIncome	NumCo
count	4410.000000	4410.000000	4410.000000	4410.000000	4410.000000	
mean	36.923810	9.192517	2.912925	2.063946	65029.312925	
std	9.133301	8.105026	1.023933	1.106689	47068.888559	
min	18.000000	1.000000	1.000000	1.000000	10090.000000	
25%	30.000000	2.000000	2.000000	1.000000	29110.000000	
50%	36.000000	7.000000	3.000000	2.000000	49190.000000	
75%	43.000000	14.000000	4.000000	3.000000	83800.000000	
max	60.000000	29.000000	5.000000	5.000000	199990.000000	

In [ ]:

In [7]:

```
# Checking Null Values

data.isnull().sum()
```

Out[7]:

```
Age                        0
Attrition                  0
BusinessTravel             0
Department                 0
DistanceFromHome           0
Education                  0
EducationField             0
Gender                     0
JobLevel                   0
JobRole                    0
MaritalStatus              0
MonthlyIncome              0
NumCompaniesWorked        19
PercentSalaryHike          0
StockOptionLevel           0
TotalWorkingYears          9
TrainingTimesLastYear      0
YearsAtCompany             0
YearsSinceLastPromotion    0
YearsWithCurrManager       0
EnvironmentSatisfaction   25
JobSatisfaction           20
WorkLifeBalance           38
JobInvolvement             0
PerformanceRating          0
dtype: int64
```

In [ ]:

In [8]:

```
# Calculating Total Zeros
```

```python
(data == 0).sum()
```

Out[8]:

```
Age                          0
Attrition                    0
BusinessTravel               0
Department                   0
DistanceFromHome             0
Education                    0
EducationField               0
Gender                       0
JobLevel                     0
JobRole                      0
MaritalStatus                0
MonthlyIncome                0
NumCompaniesWorked         586
PercentSalaryHike            0
StockOptionLevel          1893
TotalWorkingYears           33
TrainingTimesLastYear      162
YearsAtCompany             132
YearsSinceLastPromotion   1743
YearsWithCurrManager       789
EnvironmentSatisfaction      0
JobSatisfaction              0
WorkLifeBalance              0
JobInvolvement               0
PerformanceRating            0
dtype: int64
```

In [ ]:

In [9]:

```python
# Replace 0 to nan

data = data.replace(0, np.nan)
```

In [10]:

```python
# Again checking Null values

data.isnull().sum()
```

Out[10]:
```
Age                          0
Attrition                    0
BusinessTravel               0
Department                   0
DistanceFromHome             0
Education                    0
EducationField               0
Gender                       0
JobLevel                     0
JobRole                      0
MaritalStatus                0
MonthlyIncome                0
NumCompaniesWorked         605
PercentSalaryHike            0
StockOptionLevel          1893
TotalWorkingYears           42
TrainingTimesLastYear      162
YearsAtCompany             132
YearsSinceLastPromotion   1743
YearsWithCurrManager       789
EnvironmentSatisfaction     25
JobSatisfaction             20
WorkLifeBalance             38
JobInvolvement               0
PerformanceRating            0
dtype: int64
```

In [ ]:

In [11]:
```
data.shape
```

Out[11]:
```
(4410, 25)
```

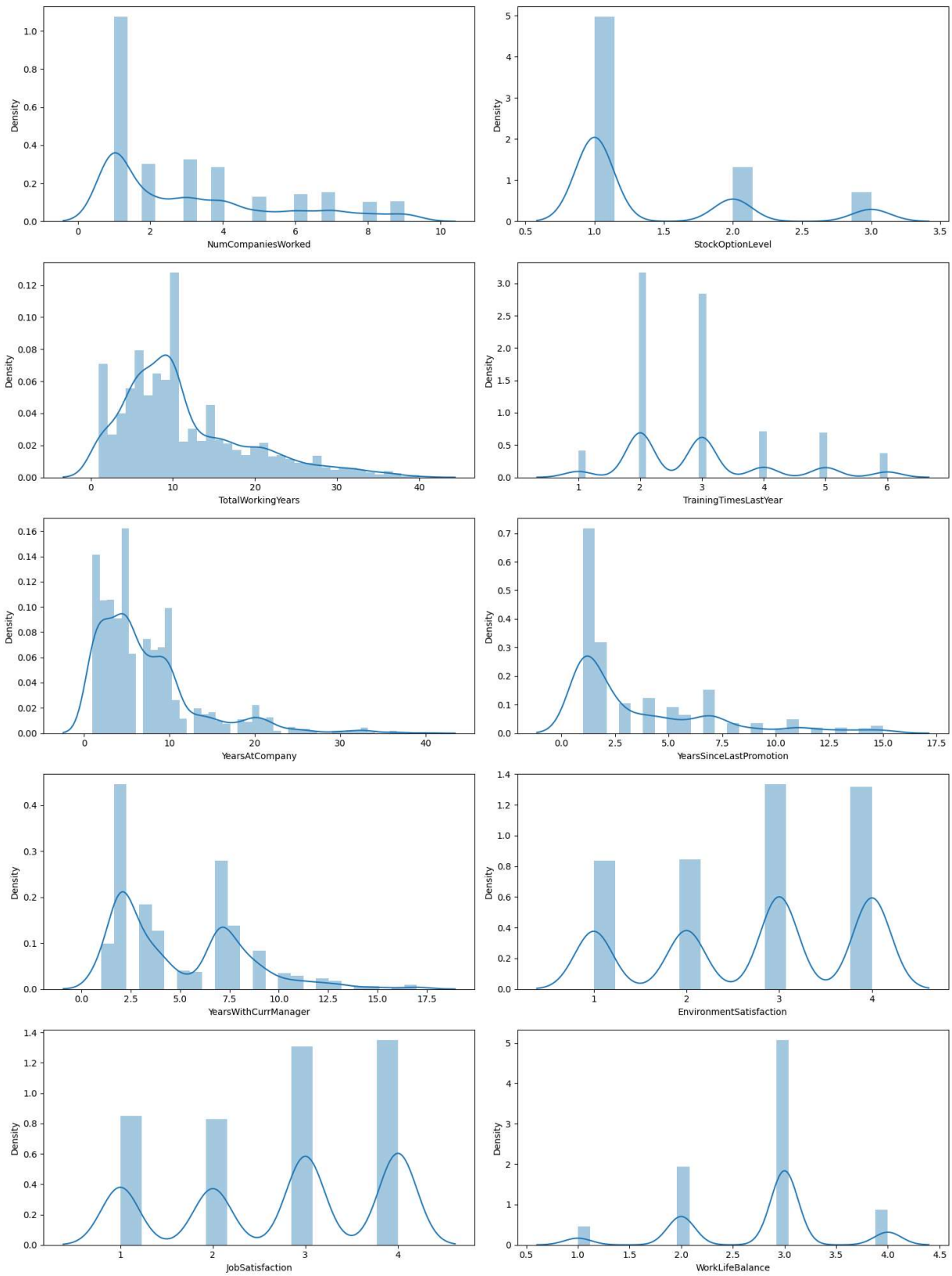
In [ ]:

Dealing with Missing Values

In [12]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [13]:
```
fig, axes = plt.subplots(5, 2, figsize=(15, 20))

sns.distplot(data["NumCompaniesWorked"], ax=axes[0, 0])
sns.distplot(data["StockOptionLevel"], ax=axes[0, 1])
sns.distplot(data["TotalWorkingYears"], ax=axes[1, 0])
sns.distplot(data["TrainingTimesLastYear"], ax=axes[1, 1])
sns.distplot(data["YearsAtCompany"], ax=axes[2, 0])
sns.distplot(data["YearsSinceLastPromotion"], ax=axes[2, 1])
sns.distplot(data["YearsWithCurrManager"], ax=axes[3, 0])
sns.distplot(data["EnvironmentSatisfaction"], ax=axes[3, 1])
sns.distplot(data["JobSatisfaction"], ax=axes[4, 0])
sns.distplot(data["WorkLifeBalance"], ax=axes[4, 1])
```

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



Need to use "Mean" -->> Normal Distribution (datas in charts evenly distributed means)

Need to use "Median" -->> left or right sided skewed (Curve will be in single side(not normally distributed means)

In [14]:
```
# Most of the Plots are Left Side Skewed and few Plots are Right Side Skewed

# So we are Going to Fillna with Median Accordingly
```

In [ ]:

In [15]:
```
# Fill na Using Median

data["NumCompaniesWorked"].fillna(data["NumCompaniesWorked"].median(), inplace=True
data["StockOptionLevel"].fillna(data["StockOptionLevel"].median(), inplace=True)
data["TotalWorkingYears"].fillna(data["TotalWorkingYears"].median(), inplace=True)
data["TrainingTimesLastYear"].fillna(data["TrainingTimesLastYear"].median(), inplac
data["YearsAtCompany"].fillna(data["YearsAtCompany"].median(), inplace=True)
data["YearsSinceLastPromotion"].fillna(data["YearsSinceLastPromotion"].median(), in
data["YearsWithCurrManager"].fillna(data["YearsWithCurrManager"].median(), inplace=
data["EnvironmentSatisfaction"].fillna(data["EnvironmentSatisfaction"].median(), in
data["JobSatisfaction"].fillna(data["JobSatisfaction"].median(), inplace=True)
data["WorkLifeBalance"].fillna(data["WorkLifeBalance"].median(), inplace=True)
```

In [16]:
```
data.isnull().sum()
```

Out[16]:
```
Age                        0
Attrition                  0
BusinessTravel             0
Department                 0
DistanceFromHome           0
Education                  0
EducationField             0
Gender                     0
JobLevel                   0
JobRole                    0
MaritalStatus              0
MonthlyIncome              0
NumCompaniesWorked         0
PercentSalaryHike          0
StockOptionLevel           0
TotalWorkingYears          0
TrainingTimesLastYear      0
YearsAtCompany             0
YearsSinceLastPromotion    0
YearsWithCurrManager       0
EnvironmentSatisfaction    0
JobSatisfaction            0
WorkLifeBalance            0
JobInvolvement             0
PerformanceRating          0
dtype: int64
```

In [ ]:

In [17]:
```
# Save the Cleaned data

data.to_csv('E:/Unified Mentor/Projects/Employee_Attrition_Analysis_DataCleaned.csv
```

In [ ]:

EDA

Step 1 - Descriptive Statistics

Get an overview of the dataset by summarizing numerical features and counting the values in categorical features.

Summary Statistics for Numerical Features:

This includes mean, median, standard deviation, minimum, and maximum values.

In [18]:
```
data.describe()
```

Out[18]:

	Age	DistanceFromHome	Education	JobLevel	MonthlyIncome	NumCo
count	4410.000000	4410.000000	4410.000000	4410.000000	4410.000000	
mean	36.923810	9.192517	2.912925	2.063946	65029.312925	
std	9.133301	8.105026	1.023933	1.106689	47068.888559	
min	18.000000	1.000000	1.000000	1.000000	10090.000000	
25%	30.000000	2.000000	2.000000	1.000000	29110.000000	
50%	36.000000	7.000000	3.000000	2.000000	49190.000000	
75%	43.000000	14.000000	4.000000	3.000000	83800.000000	
max	60.000000	29.000000	5.000000	5.000000	199990.000000	

In [ ]:

Count of Categorical Features:

For categorical features, we look at the count of each category.

In [19]:
```
# Count of unique values in categorical features

categorical_features = data.select_dtypes(include=['object']).columns

for col in categorical_features:
    print(f"{col}:\n{data[col].value_counts()}\0")
```

```
Attrition:
Attrition
No     3699
Yes     711
Name: count, dtype: int64
BusinessTravel:
BusinessTravel
Travel_Rarely        3129
Travel_Frequently     831
Non-Travel            450
Name: count, dtype: int64
Department:
Department
Research & Development    2883
Sales                     1338
Human Resources            189
Name: count, dtype: int64
EducationField:
EducationField
Life Sciences       1818
Medical             1392
Marketing            477
Technical Degree     396
Other                246
Human Resources       81
Name: count, dtype: int64
Gender:
Gender
Male      2646
Female    1764
Name: count, dtype: int64
JobRole:
JobRole
Sales Executive              978
Research Scientist           876
Laboratory Technician        777
Manufacturing Director       435
Healthcare Representative    393
Manager                      306
Sales Representative         249
Research Director            240
Human Resources              156
Name: count, dtype: int64
MaritalStatus:
MaritalStatus
Married     2019
Single      1410
Divorced     981
Name: count, dtype: int64
```

Observation from Step 1:

1. The dataset includes a variety of features like age, income, and job satisfaction. The numerical features show varying ranges, with some like "Age" and "MonthlyIncome" having wide distributions.

2. Categorical features like "Department" and "JobRole" have distinct categories, which will help in understanding the distribution of employees across different areas.

In [ ]:

Step 2 - Attrition Analysis

Understand the distribution of the target variable "Attrition"

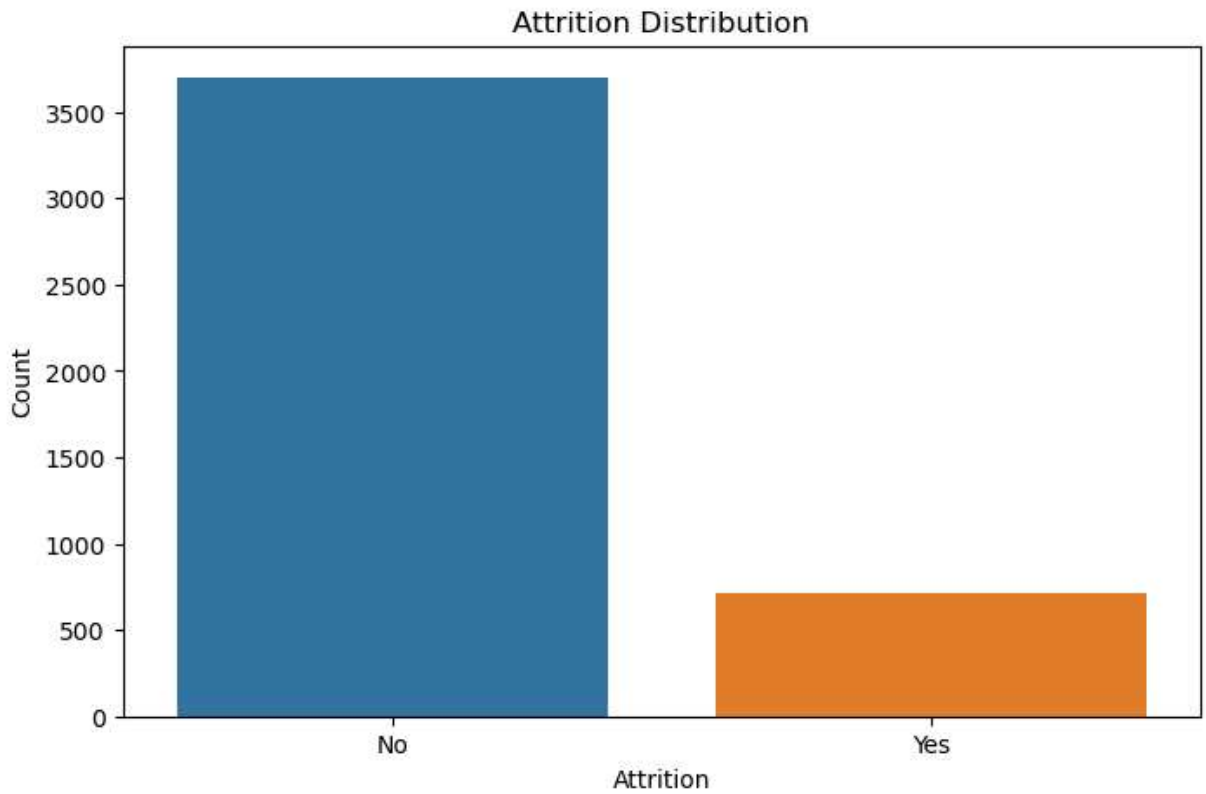
Count and Percentage of Attrition: Calculate the number and percentage of employees who have left (Yes) versus those who stayed (No).

In [20]:
```
# Count and percentage of Attrition

attrition_count = data["Attrition"].value_counts(normalize = True) * 100
print(round(attrition_count))
```

```
Attrition
No     84.0
Yes    16.0
Name: proportion, dtype: float64
```

In [ ]:

In [21]:
```
# Plotting the distribution plot for "Attrition"

plt.figure(figsize = (8,5))

sns.countplot( x = "Attrition", data = data)
plt.title("Attrition Distribution")
plt.xlabel("Attrition")
plt.ylabel("Count")

plt.show()
```



Observation from Step 2

1. The "Attrition" column shows that around 16% of employees have left the company. To understanding the reasons behind this attrition is key to addressing the problem.
2. Visualizing the distribution highlighted that attrition is not evenly spread, indicating potential patterns or factors at play.

In [ ]:

Step 3 - Feature Analysis

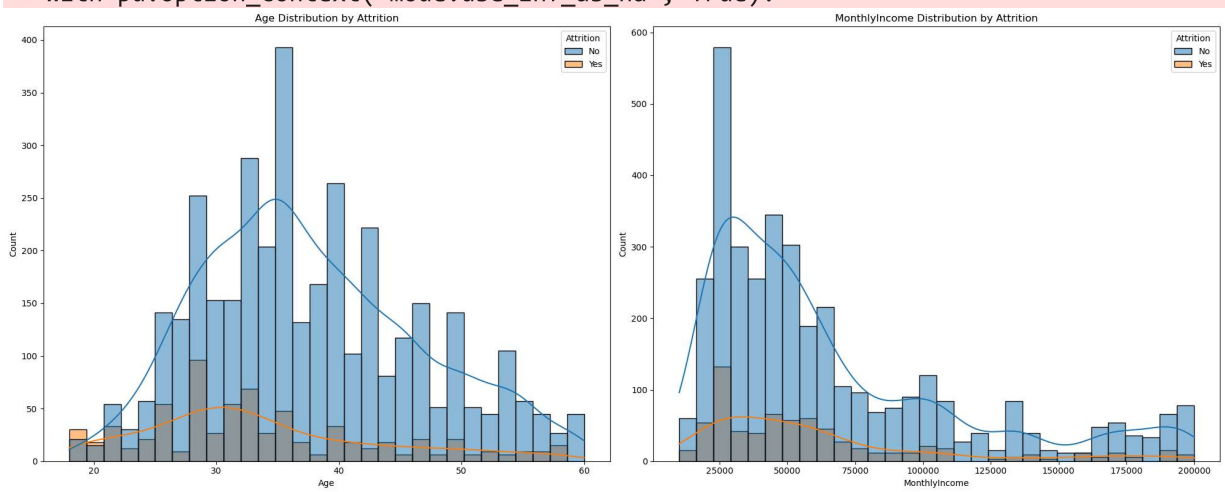
Explore the distributions of key features and compare them across Attrition categories.

1. Numerical Features - Age and Monthly Income

In [22]:
```
fig, axes = plt.subplots(1, 2, figsize=(20, 8))


# Plotting Age distribution by Attrition

sns.histplot(data, x ='Age', hue ='Attrition', kde = True, bins = 30, ax = axes[0])
```

```
axes[0].set_title('Age Distribution by Attrition')
axes[0].set_xlabel('Age')
axes[0].set_ylabel('Count')


# Plotting MonthlyIncome distribution by Attrition

sns.histplot(data, x ='MonthlyIncome', hue ='Attrition', kde = True, bins = 30, ax

axes[1].set_title('MonthlyIncome Distribution by Attrition')
axes[1].set_xlabel('MonthlyIncome')
axes[1].set_ylabel('Count')

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

```
C:\Users\Ravim\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Ravim\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

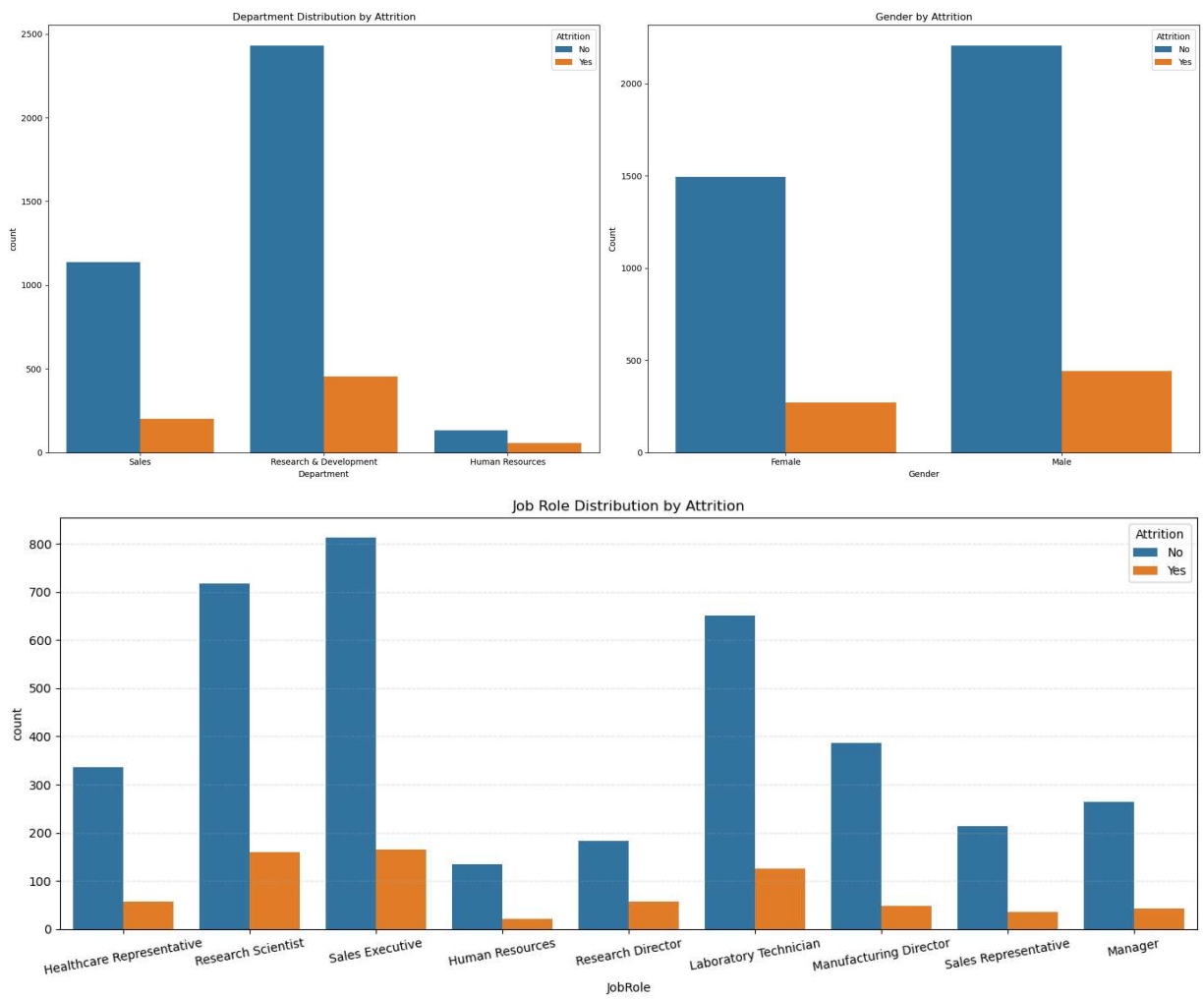


In [ ]:

2. Categorical Features - Department, Gender and JobRole

In [23]:
```
fig, axes = plt.subplots(1, 2, figsize = (20, 8))


# Plotting Department distribution by Attrition

sns.countplot(data = data, x ='Department', hue ='Attrition', ax = axes[0])

axes[0].set_title('Department Distribution by Attrition')
plt.ylabel('Count')
plt.xlabel('Department')
```

```python
# Plotting Gender distribution by Attrition

sns.countplot(data = data, x ='Gender', hue ='Attrition', ax = axes[1])

axes[1].set_title('Gender by Attrition')
plt.ylabel('Count')
plt.xlabel('Gender')

plt.tight_layout()
plt.show()


# Plotting JobRole distribution by Attrition

plt.figure(figsize=(14, 6))
sns.countplot(data=data, x='JobRole', hue='Attrition')
plt.title('Job Role Distribution by Attrition')
plt.xticks(rotation = 10)
plt.grid(axis='y', linestyle='--', alpha=0.3)

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



In [ ]:

Observation from Step 3

1. numerical Features:
   A. "Younger employees" have a higher attrition rate. (Below 35Age)
   B. "Low income levels" are associated with higher attrition. (Below 60000)

2. Categorical Features:
   A. Research and Development Department have higher attrition rates compared to other Departments.
   B. Male Employees have slightlt higher attrition rates compared to Female Employees.
   C. Sales Executives and Research Scientist have higher attrition rates compared to other Job Roles.

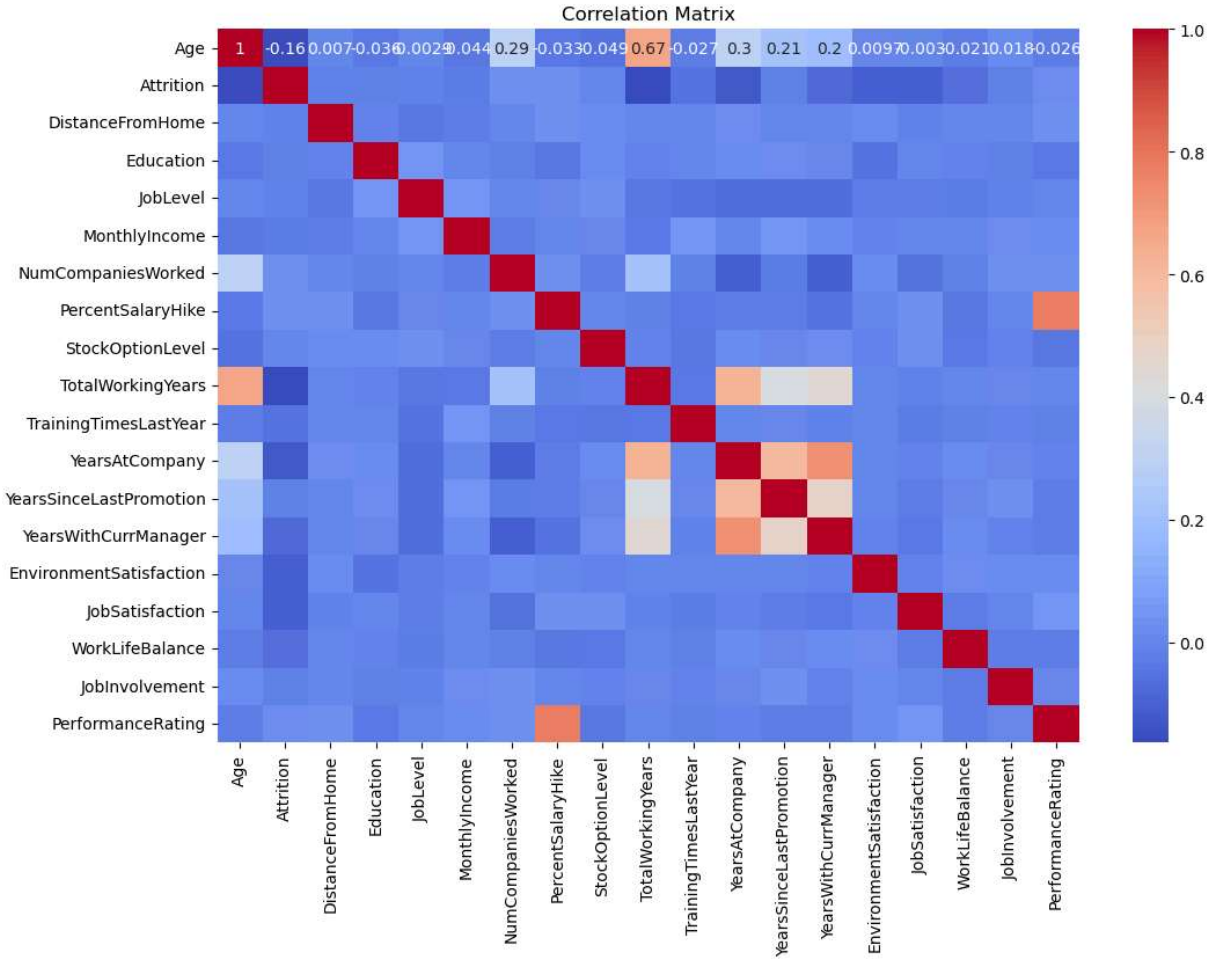
In [ ]:

Step 4 - Correlation Analysis

Identify relationships between features and Attrition

In [24]:
```
# Convert Attrition to numeric
data['Attrition'] = data['Attrition'].map({'Yes': 1, 'No': 0})
```

In [25]:
```
# Select only numeric columns for correlation analysis
numeric_data = data.select_dtypes(include=[float, int])

# Calculate the correlation matrix for numeric columns only
corr_matrix = numeric_data.corr()

# Plotting the correlation matrix

plt.figure(figsize=(12, 8))

sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



Observation from Step 4

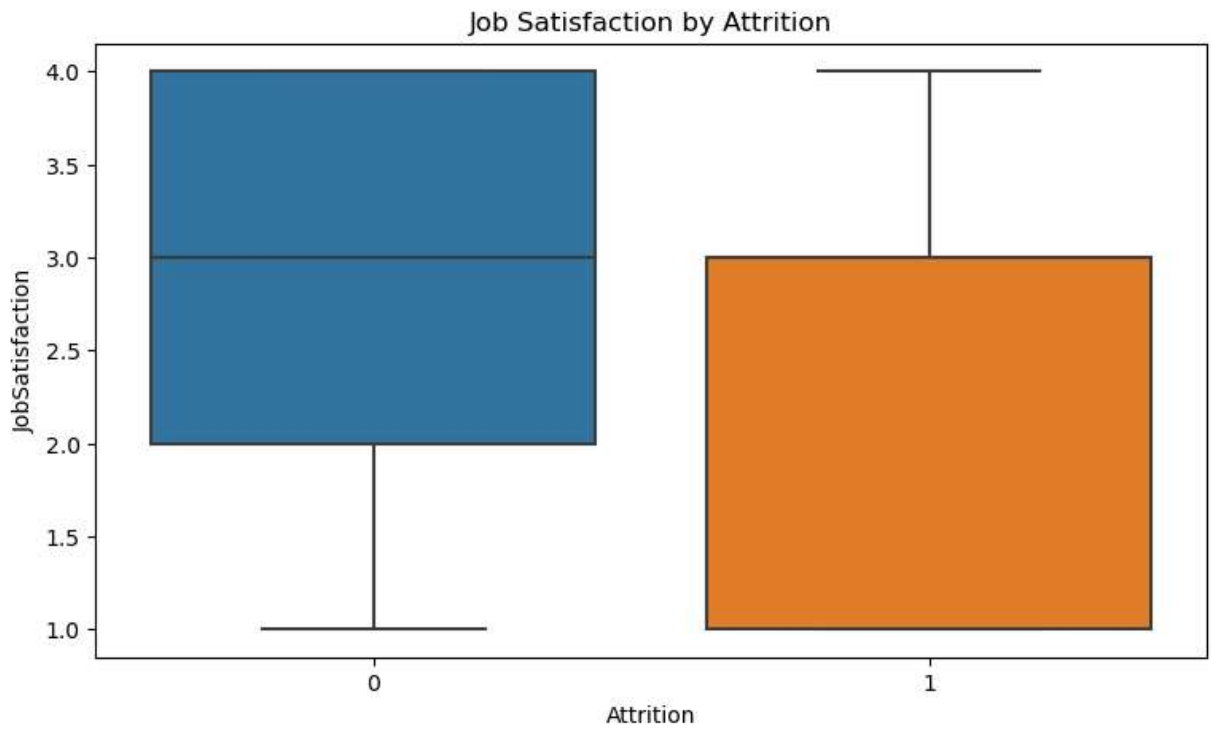
In [ ]:

1. Strong Positive Correlations: TotalWorkingYears and Age: Older employees tend to have more years of work experience, which is expected.
2. Weak or No Correlations: Many features show little to no correlation with others, indicating that they may be independent.
3. Factors to Focus On: Features like JobSatisfaction, EnvironmentSatisfaction, and WorkLifeBalance in relation to Attrition to understand if these factors significantly influence whether employees leave the company.

In [ ]:

- Identifing the Relationship between "Job Satiofation" and "Attrition"

In [26]:
```
# Plotting Job Satisfation  distribution by Attrition

plt.figure(figsize=(9, 5))

sns.boxplot(x='Attrition', y='JobSatisfaction', data=data)
plt.title('Job Satisfaction by Attrition')

plt.show()
```



- Observation - Higher dissatisfaction levels often correlate with higher attrition rates.

In [ ]:

Final Interpretation

- Key Factors Influencing Attrition:
  1. Younger employees (below 35 years old) have a higher attrition rate.
  2. Low income levels (below 60,000) are associated with higher attrition.
  3. The Research and Development department has a higher attrition rate compared to other departments.
  4. Male employees have slightly higher attrition rates compared to female employees.

5. Sales executives and research scientists have higher attrition rates compared to other job roles.

6. Higher Job dissatisfaction levels often correlate with higher attrition rates.