HR Analytics

Introduction:

- In this project, we delve into the realm of HR Analytics, utilizing Python and leveraging the extensive capabilities of libraries available. Our dataset, sourced from Kaggle, serves as the foundation for our exploration into understanding various aspects of human resource management through data-driven insights.

Aim:

- The primary aim of this project is to harness the power of Python libraries to analyze HR data comprehensively.
- By employing statistical techniques, data visualization, and machine learning algorithms, we aim to uncover patterns, trends, and correlations within the dataset.
- Our focus lies in gaining actionable insights that can inform decision-making processes in the realm of human resource management.

- This project focuses on HR analytics conducted in Python, utilizing a specific library. The dataset utilized in this project was collected from Kaggle, a renowned platform for data science enthusiasts and professionals.

- These are some of the essential libraries utilized in Python for HR analytics projects, providing functionalities for data manipulation (Pandas), numerical computing (NumPy), data visualization (Matplotlib), and enhanced visualizations (Seaborn).

- These libraries offer a wide range of functionalities for data manipulation, visualization, statistical analysis, machine learning, and deep learning, making them essential tools for HR analytics projects conducted in Python.

IMPORTING PACKAGES | LIBRARIES

In [54]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import missingno as msno
```

READING DATA FROM CSV FILE

In [15]:
```python
df = pd.read_csv("HR-Employee.csv")
df
```

Out[15]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumbe
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	
...	...	...	...	...	...	...	...	...	...	.
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	206
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	206
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	206
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	206
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	206

1470 rows × 35 columns

EXPLORATORY DATA ANALYSIS

In [16]:
```
df.head() # top 5 record
```

Out[16]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	

5 rows × 35 columns

In [17]:
```
df.tail() # last 5 record
```

Out[17]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumbe
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	206
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	206
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	206
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	206
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	206

5 rows × 35 columns

In [11]:
```
# Total no of columns in Dataset
df.columns
```

Out[11]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

In [12]:
```
# Information About Dataset
df.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 [13]:
```
# More About Dataset
df.describe()
```

Out[13]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	Job
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156	
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428	
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000	
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000	
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000	

8 rows × 26 columns

In [18]:

```
# More About Dataset with Transpose ('T')
df.describe().T
```

Out[18]:

	count	mean	std	min	25%	50%	75%	max
Age	1470.0	36.923810	9.135373	18.0	30.00	36.0	43.00	60.0
DailyRate	1470.0	802.485714	403.509100	102.0	465.00	802.0	1157.00	1499.0
DistanceFromHome	1470.0	9.192517	8.106864	1.0	2.00	7.0	14.00	29.0
Education	1470.0	2.912925	1.024165	1.0	2.00	3.0	4.00	5.0
EmployeeCount	1470.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
EmployeeNumber	1470.0	1024.865306	602.024335	1.0	491.25	1020.5	1555.75	2068.0
EnvironmentSatisfaction	1470.0	2.721769	1.093082	1.0	2.00	3.0	4.00	4.0
HourlyRate	1470.0	65.891156	20.329428	30.0	48.00	66.0	83.75	100.0
JobInvolvement	1470.0	2.729932	0.711561	1.0	2.00	3.0	3.00	4.0
JobLevel	1470.0	2.063946	1.106940	1.0	1.00	2.0	3.00	5.0
JobSatisfaction	1470.0	2.728571	1.102846	1.0	2.00	3.0	4.00	4.0
MonthlyIncome	1470.0	6502.931293	4707.956783	1009.0	2911.00	4919.0	8379.00	19999.0
MonthlyRate	1470.0	14313.103401	7117.786044	2094.0	8047.00	14235.5	20461.50	26999.0
NumCompaniesWorked	1470.0	2.693197	2.498009	0.0	1.00	2.0	4.00	9.0
PercentSalaryHike	1470.0	15.209524	3.659938	11.0	12.00	14.0	18.00	25.0
PerformanceRating	1470.0	3.153741	0.360824	3.0	3.00	3.0	3.00	4.0
RelationshipSatisfaction	1470.0	2.712245	1.081209	1.0	2.00	3.0	4.00	4.0
StandardHours	1470.0	80.000000	0.000000	80.0	80.00	80.0	80.00	80.0
StockOptionLevel	1470.0	0.793878	0.852077	0.0	0.00	1.0	1.00	3.0
TotalWorkingYears	1470.0	11.279592	7.780782	0.0	6.00	10.0	15.00	40.0
TrainingTimesLastYear	1470.0	2.799320	1.289271	0.0	2.00	3.0	3.00	6.0
WorkLifeBalance	1470.0	2.761224	0.706476	1.0	2.00	3.0	3.00	4.0
YearsAtCompany	1470.0	7.008163	6.126525	0.0	3.00	5.0	9.00	40.0
YearsInCurrentRole	1470.0	4.229252	3.623137	0.0	2.00	3.0	7.00	18.0
YearsSinceLastPromotion	1470.0	2.187755	3.222430	0.0	0.00	1.0	3.00	15.0

	count	mean	std	min	25%	50%	75%	max
YearsWithCurrManager	1470.0	4.123129	3.568136	0.0	2.00	3.0	7.00	17.0

In [19]:

```
# Checking for null values in Dataset
df.isnull()
```

Out[19]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
1465	False	False	False	False	False	False	False	False	False	False
1466	False	False	False	False	False	False	False	False	False	False
1467	False	False	False	False	False	False	False	False	False	False
1468	False	False	False	False	False	False	False	False	False	False
1469	False	False	False	False	False	False	False	False	False	False

1470 rows × 35 columns

In [ ]:

```
# Dropping duplicates
df = df.drop_ duplicate()
```

In [ ]:

```
#removing NaN Values
df= df.dropna()
```

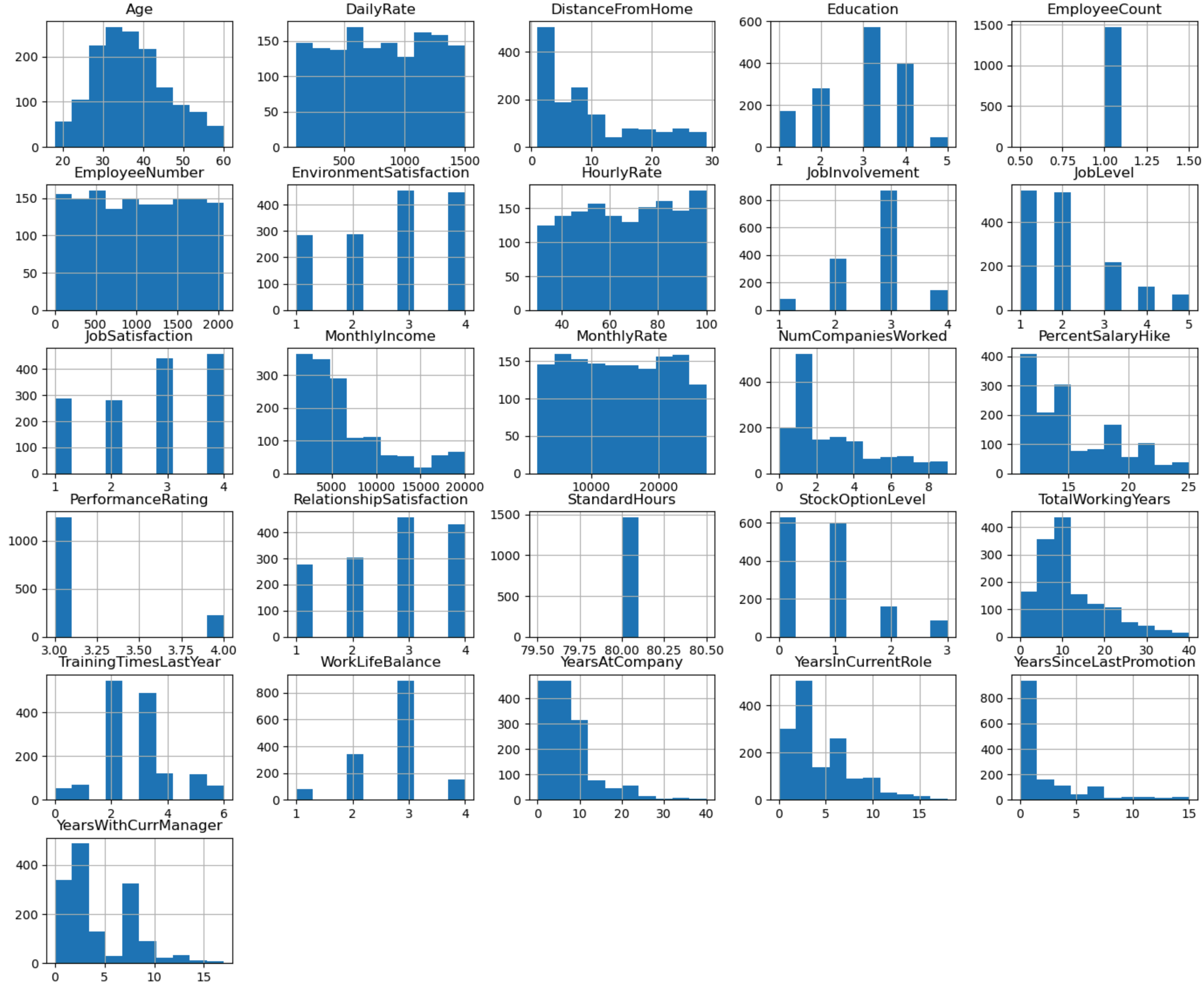
In [20]:

```
# Checking Total null values in Dataset
df.isnull().sum()
```

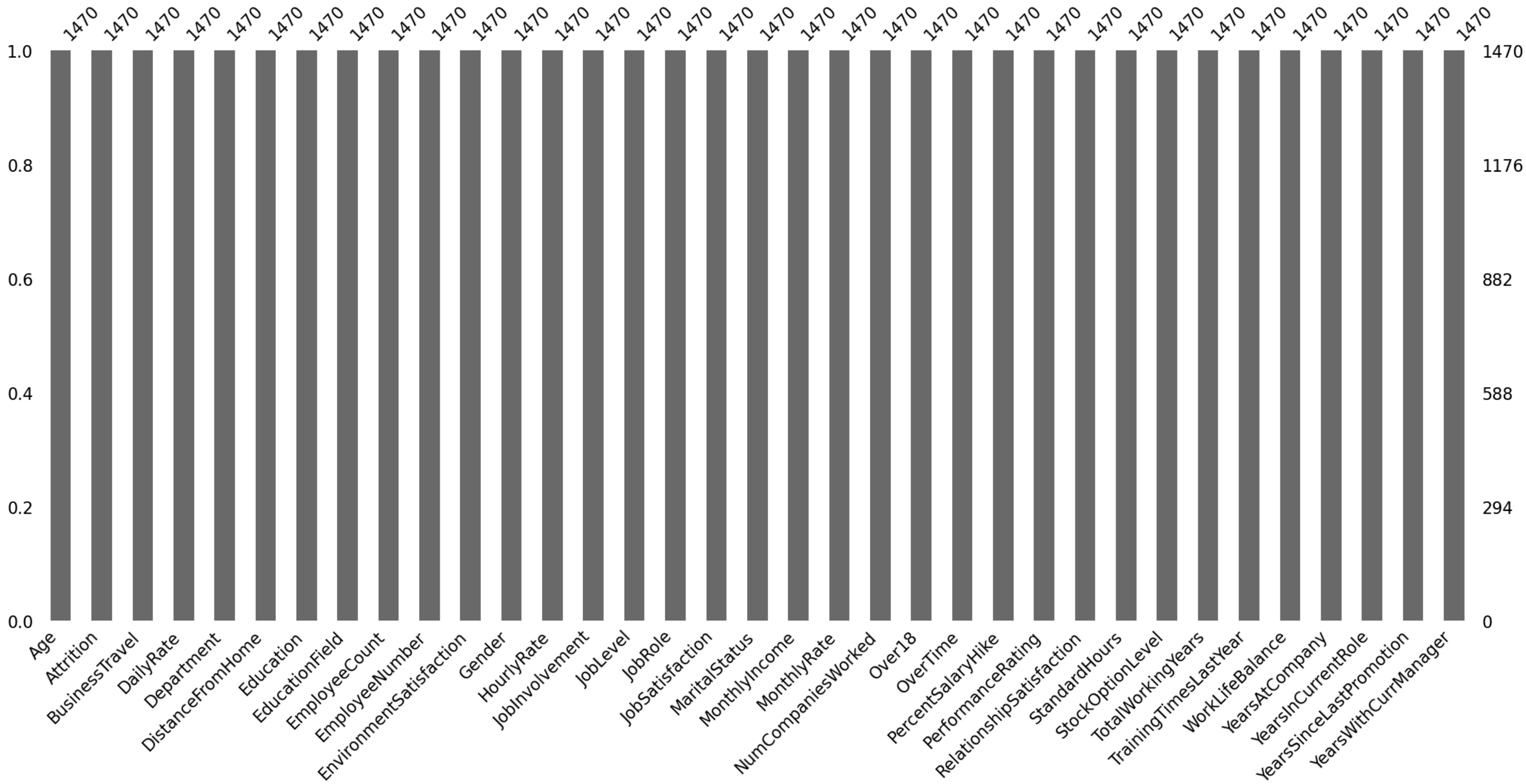
Out[20]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              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
```

DATA VISUALIZATION

In [26]:
```
# Plotting The Data Distribution Plots
df.hist(figsize = (17,14))
plt.show()
```

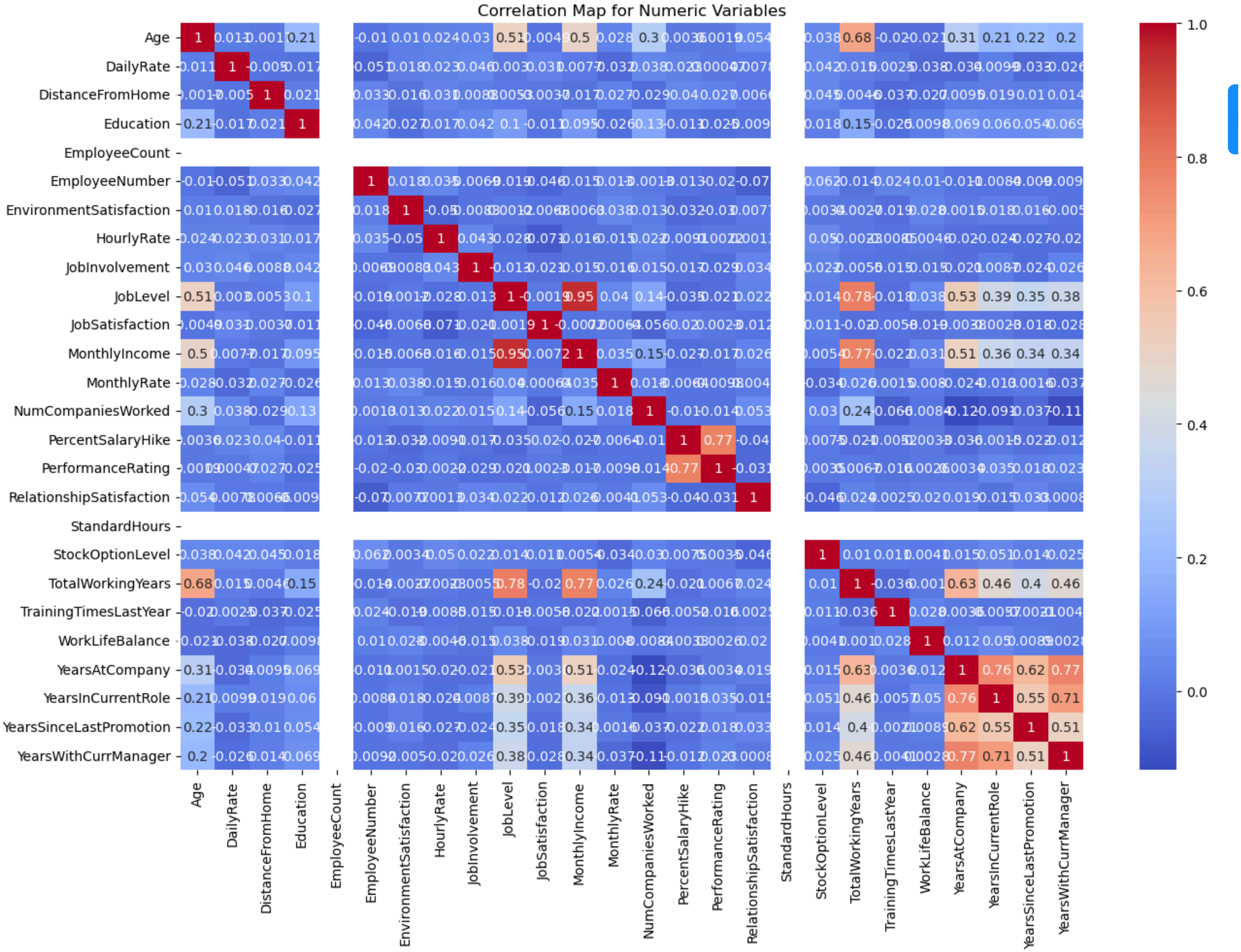

Age
DailyRate
DistanceFromHome
Education
EmployeeCount
EmployeeNumber
EnvironmentSatisfaction
HourlyRate
JobInvolvement
JobLevel
JobSatisfaction
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StandardHours
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager

In [56]:
```
P = msno.bar(df)
```

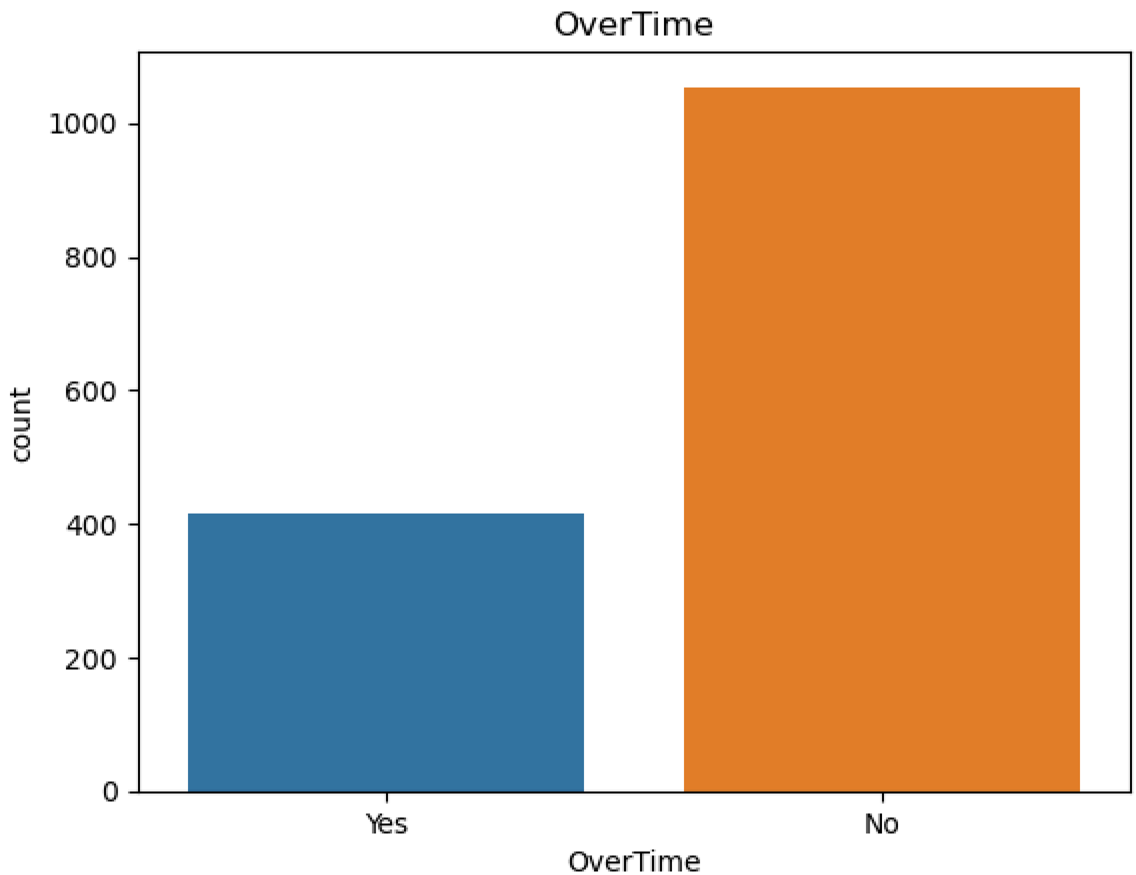


In [27]:
```
# Showing a correlation map for all numeric values
corr_matrix = df.corr()
plt.figure(figsize=(15,10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Map for Numeric Variables')
plt.show()
```

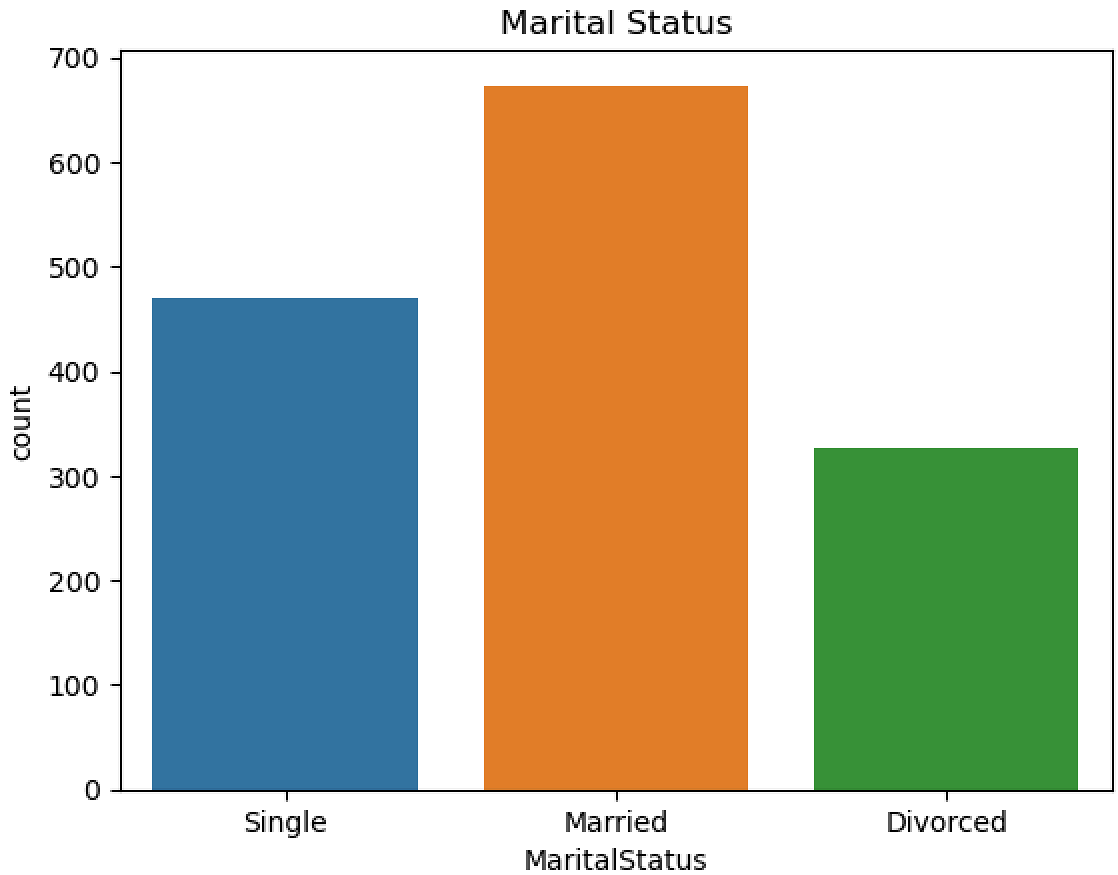
```
C:\Users\prata\AppData\Local\Temp\ipykernel_9804\3910158270.py:2: FutureWarning: The default value of numeric_only in Dat
aFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  corr_matrix = df.corr()
```


Correlation Map for Numeric Variables
Age
DailyRate
DistanceFromHome
Education
EmployeeCount
EmployeeNumber
EnvironmentSatisfaction
HourlyRate
JobInvolvement
JobLevel
JobSatisfaction
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StandardHours
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
1.0
0.8
0.6
0.4
0.2
0.0

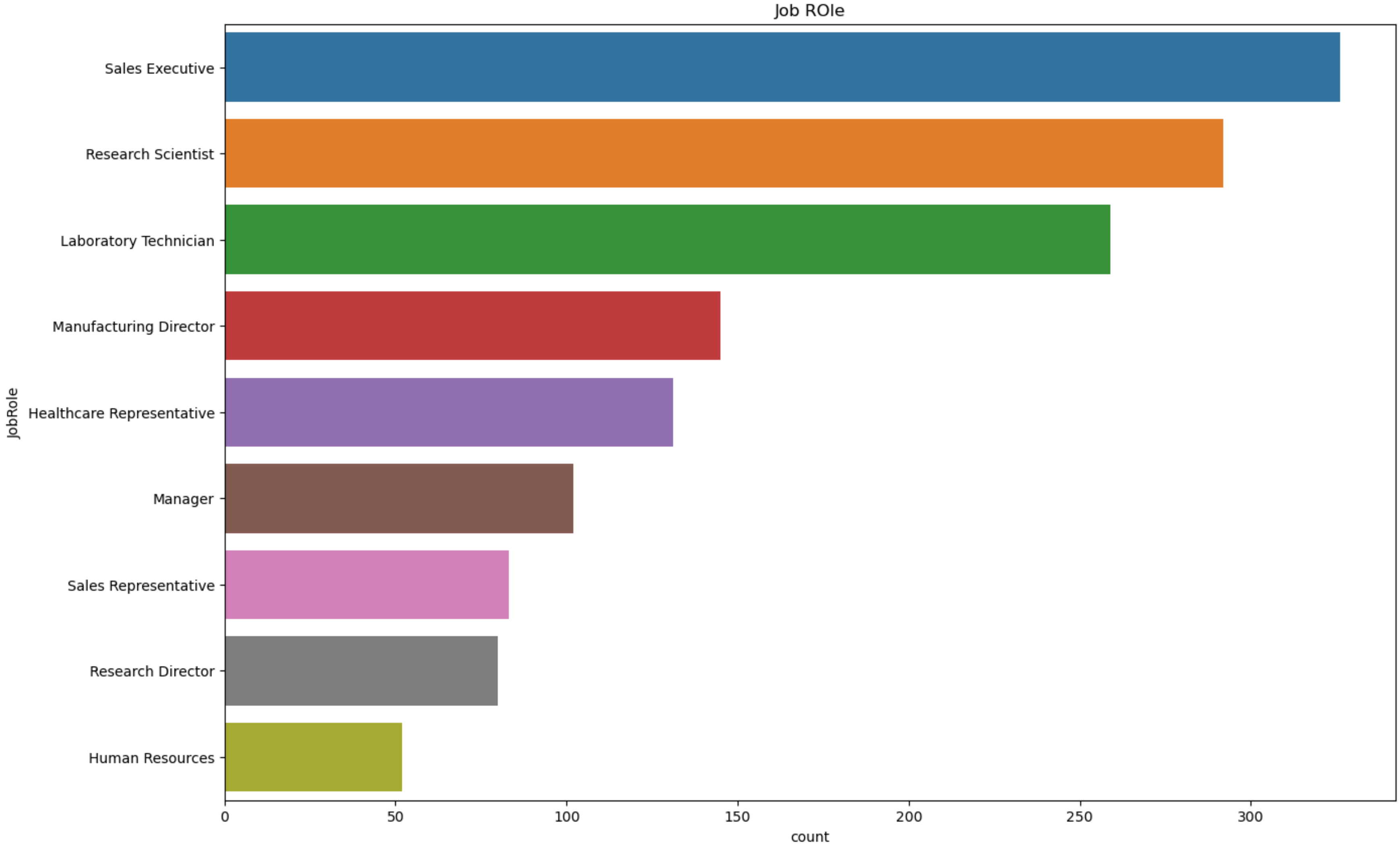
In [28]:
```
# Overtime
sns.countplot(df, x='OverTime')
plt.title('OverTime')
plt.show()
```



In [37]:
```
# Marital status
sns.countplot(df, x='MaritalStatus')
plt.title('Marital Status')
plt.show()
```

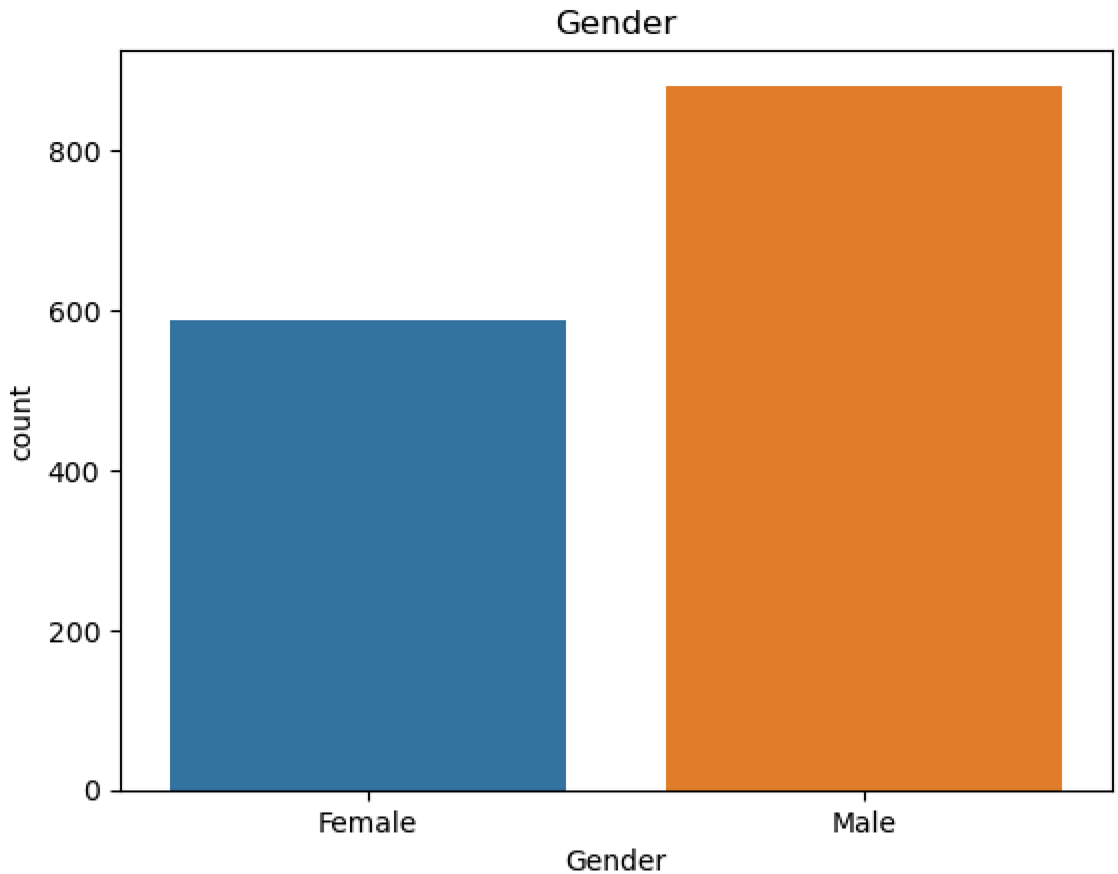


In [38]:
```
# Job Role
plt.figure(figsize = (15,10))
sns.countplot(df, y ='JobRole')
plt.title('Job ROle')
plt.show()
```

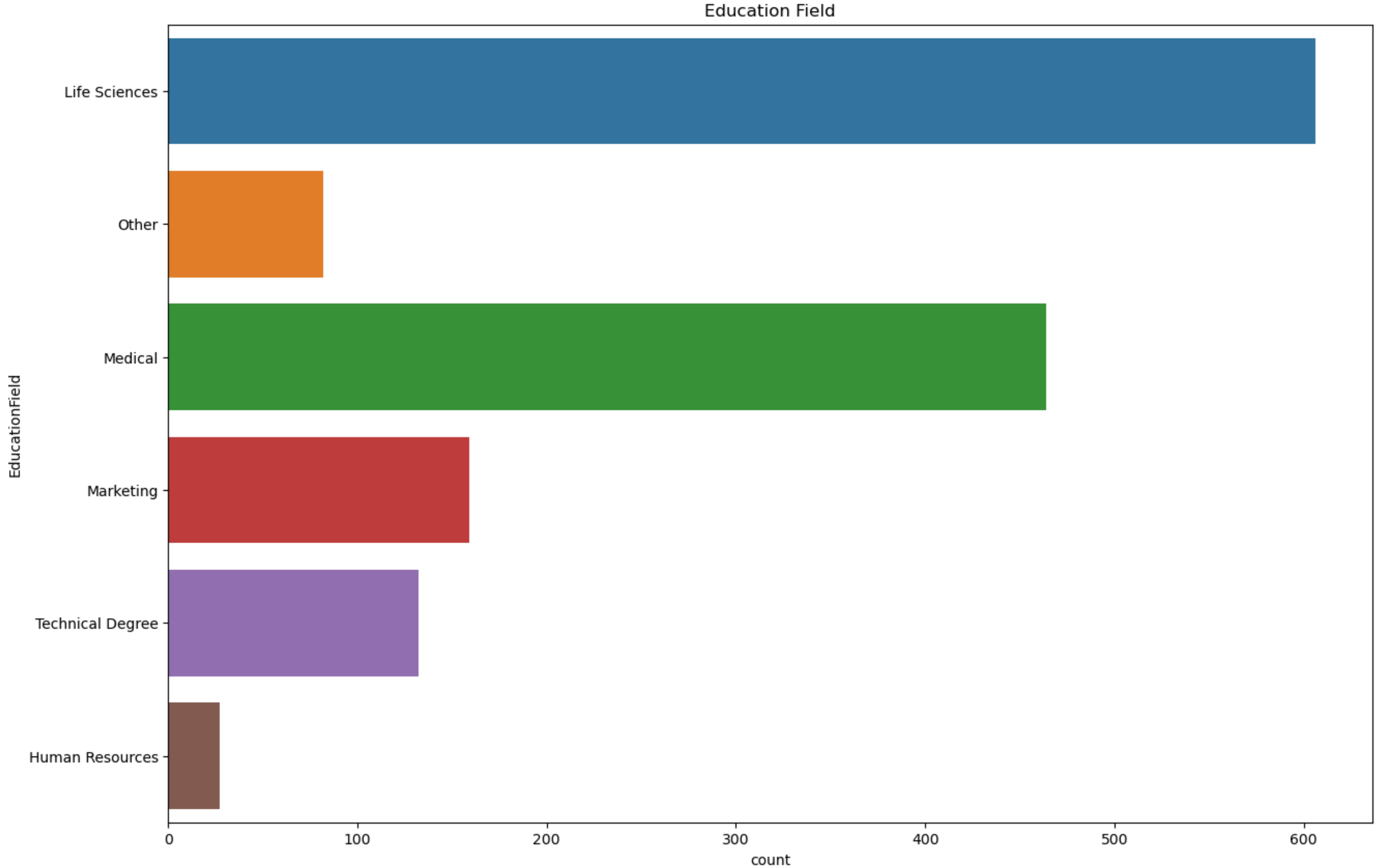


In [40]:

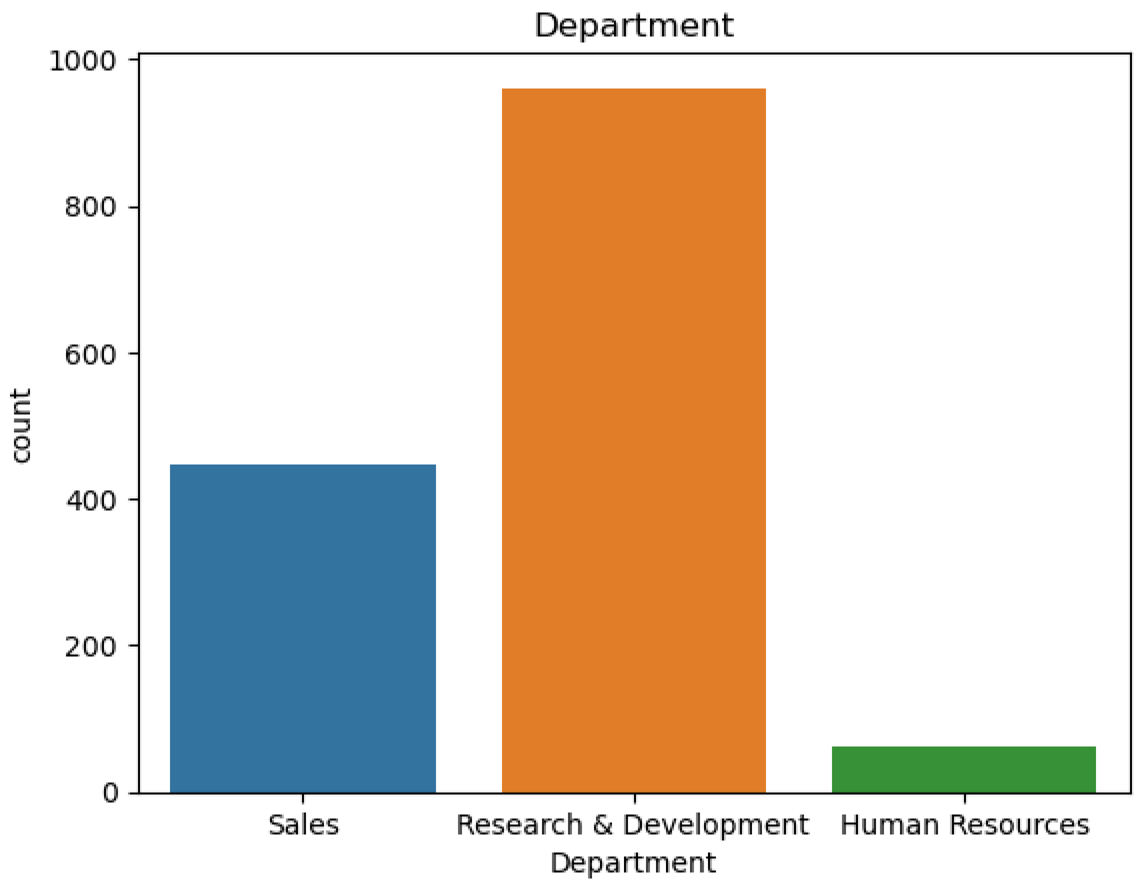
```
# Gender
sns.countplot(df, x ='Gender')
plt.title('Gender')
plt.show()
```



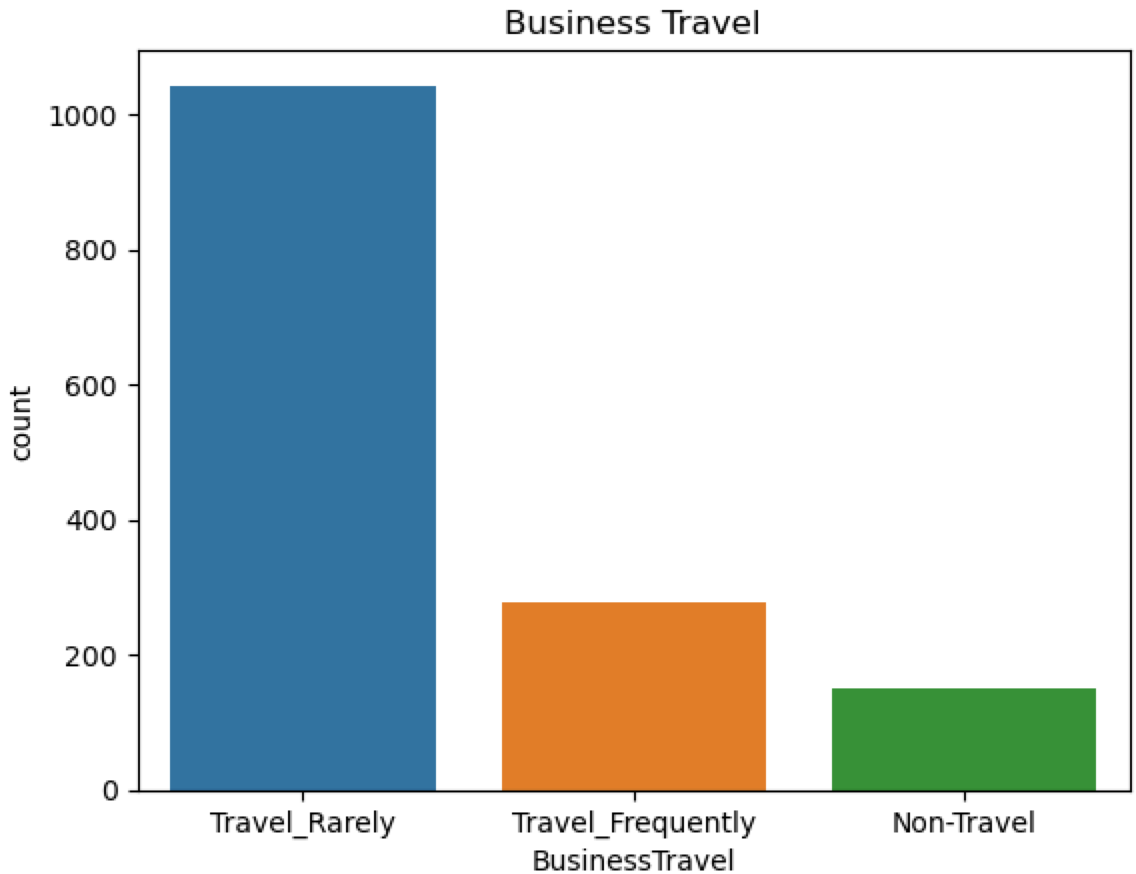
In [41]:
```
# Education Field
plt.figure(figsize = (15,10))
sns.countplot(df, y ='EducationField')
plt.title('Education Field')
plt.show()
```



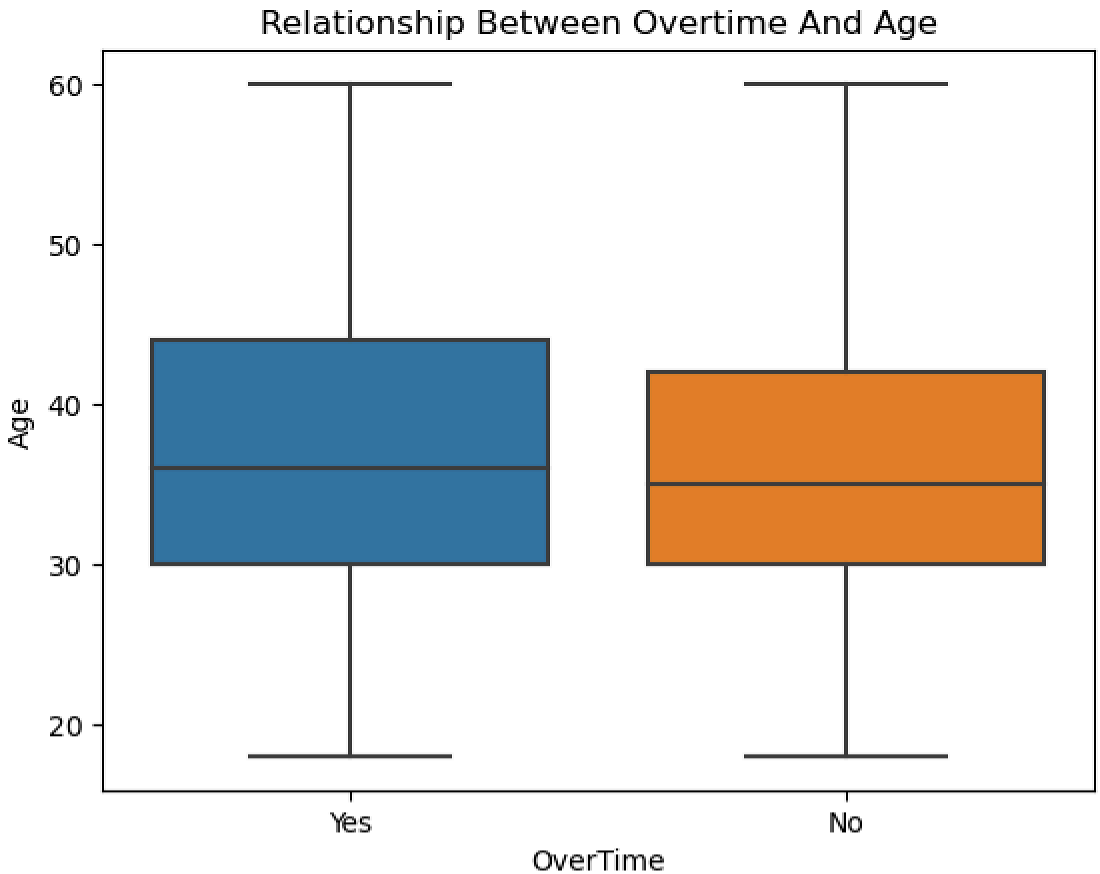
In [44]:
```
# Department
sns.countplot(df, x ='Department')
plt.title('Department')
plt.show()
```



In [45]:
```
# Business Travel
sns.countplot(df, x ='BusinessTravel')
plt.title('Business Travel')
plt.show()
```

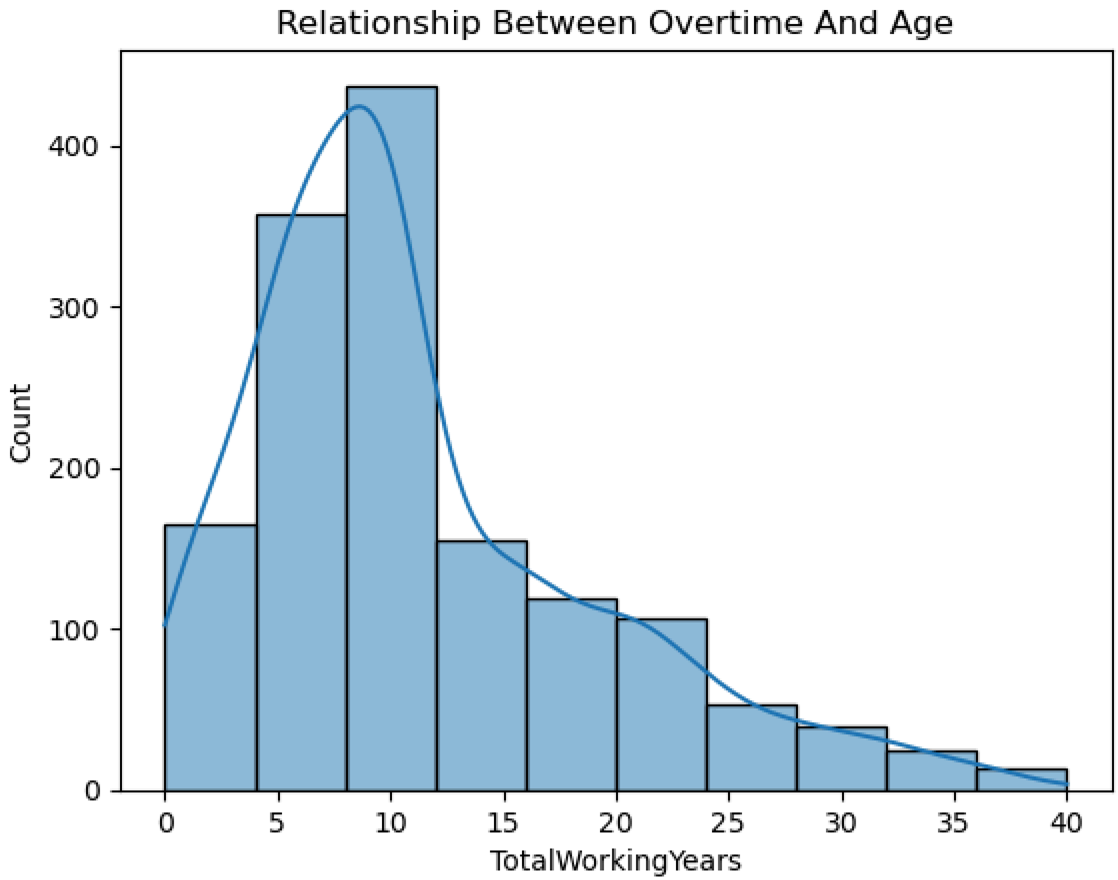


In [47]:
```
# Relationship Between Overtime And Age
sns.boxplot(df, x ='OverTime', y = 'Age')
plt.title('Relationship Between Overtime And Age')
plt.show()
```

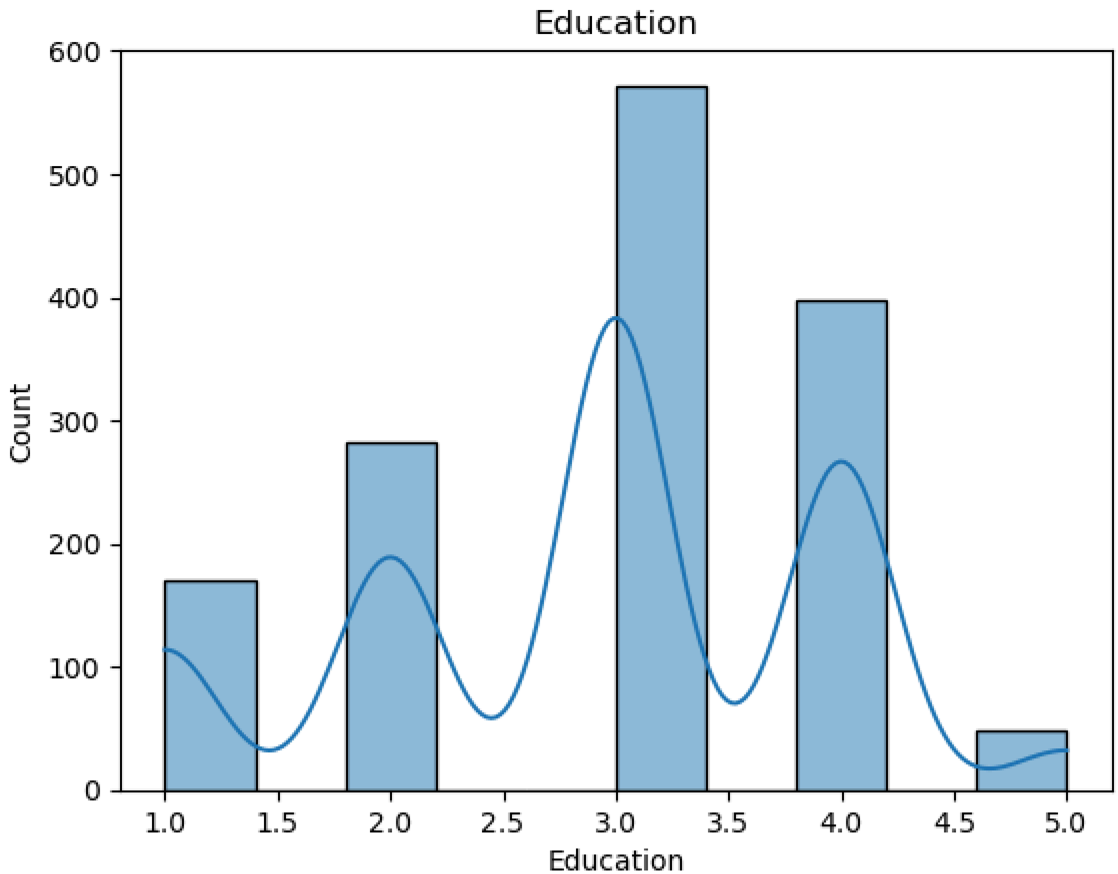


Plotting Numerical Values

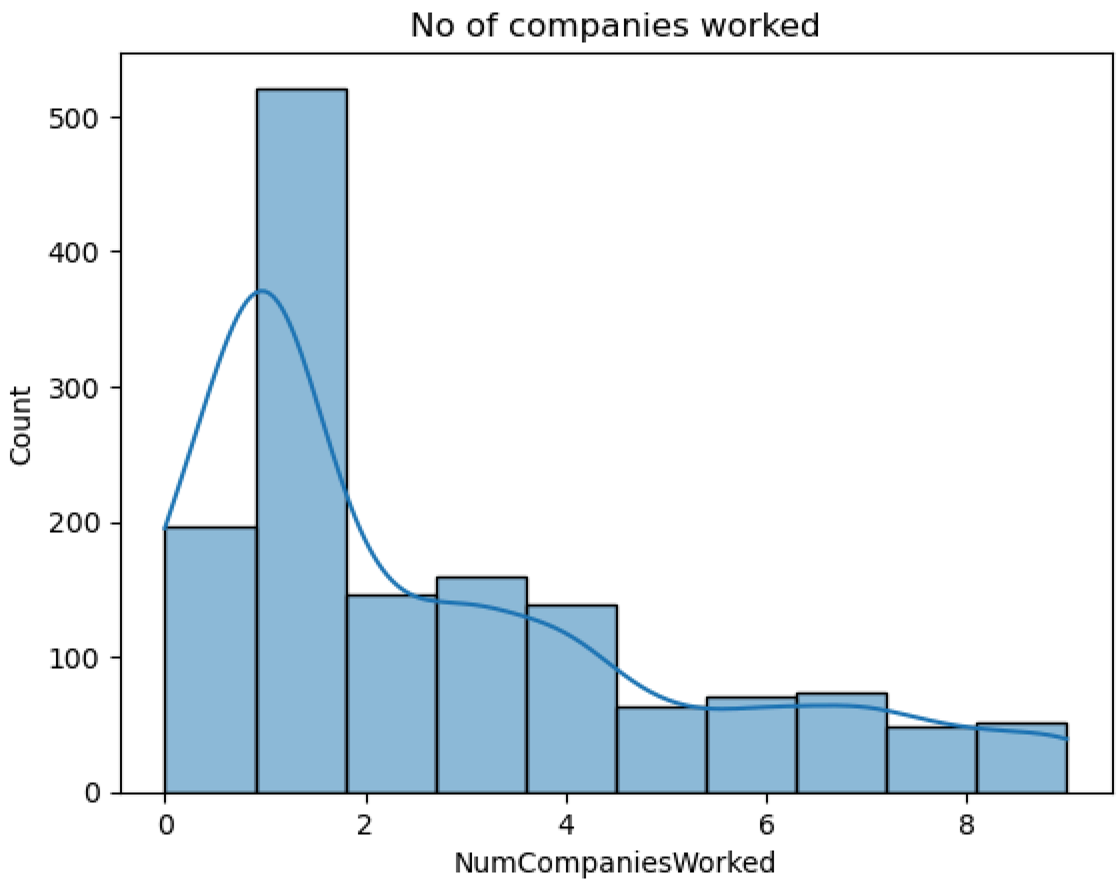
In [48]:
```
# Total working years
sns.histplot(df, x ='TotalWorkingYears',bins = 10,kde = True)
plt.title('Relationship Between Overtime And Age')
plt.show()
```



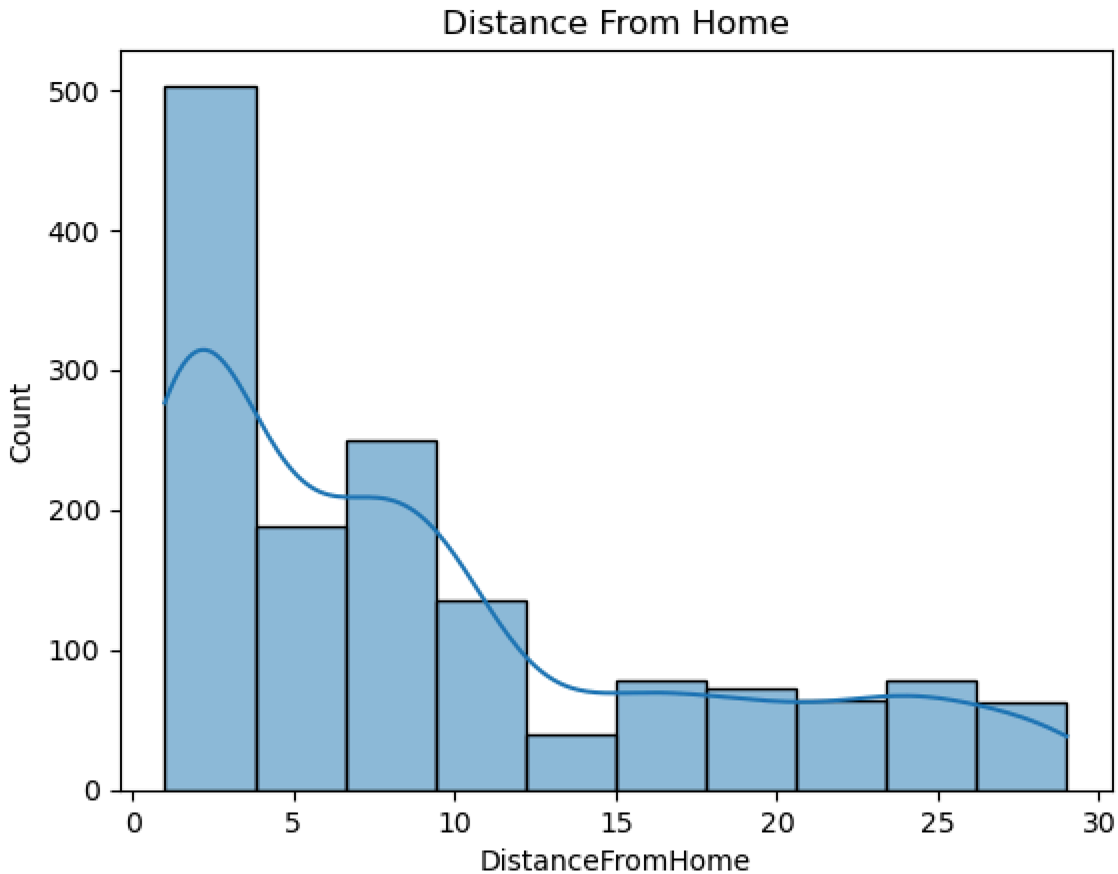
In [49]:
```
# Education
sns.histplot(df, x ='Education',bins = 10,kde = True)
plt.title('Education')
plt.show()
```



In [50]:
```
# No Of Companies Worked
sns.histplot(df, x ='NumCompaniesWorked',bins = 10,kde = True)
plt.title('No of companies worked')
plt.show()
```

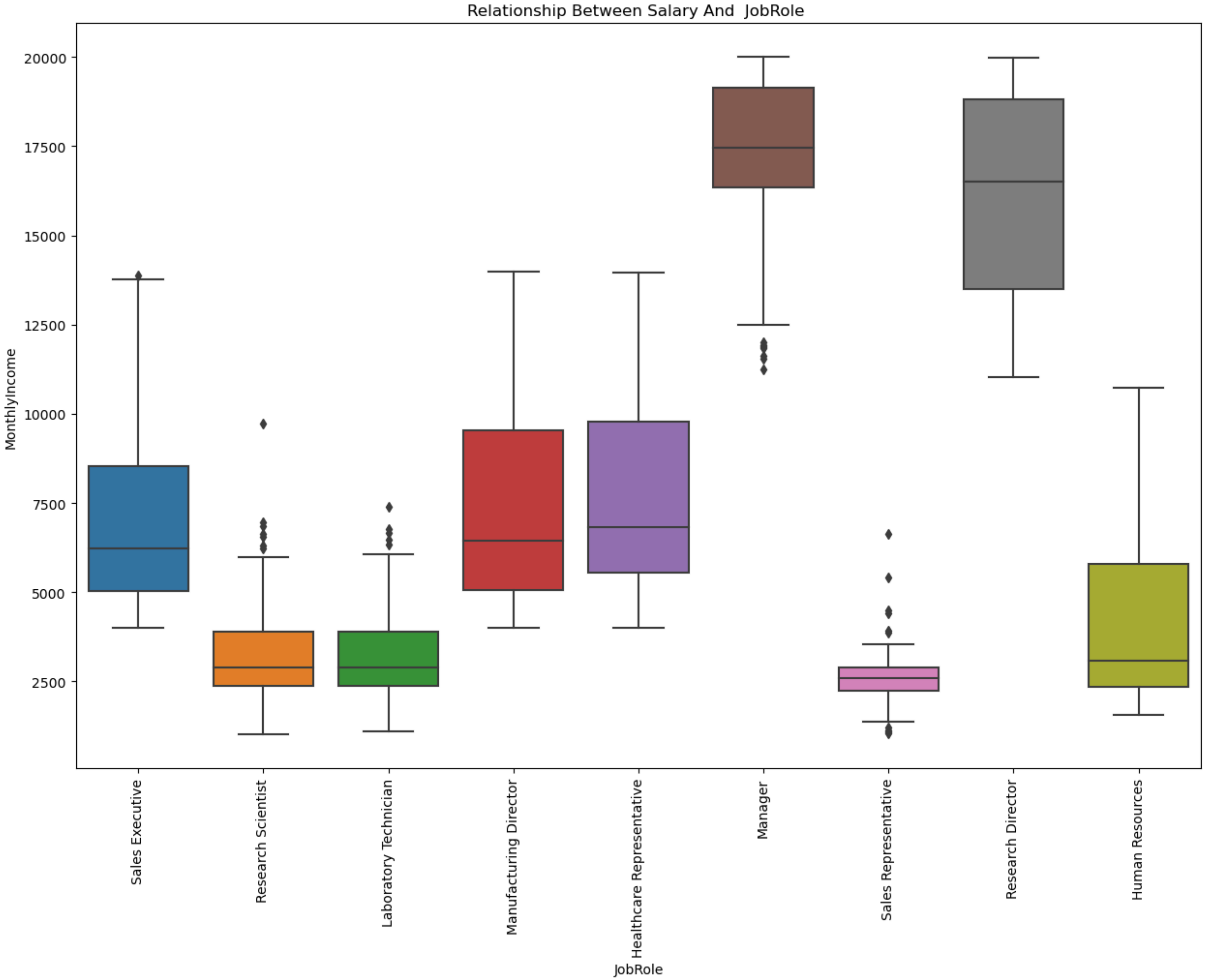


In [51]:
```
# Distance From Home
sns.histplot(df, x ='DistanceFromHome',bins = 10,kde = True)
plt.title('Distance From Home')
plt.show()
```

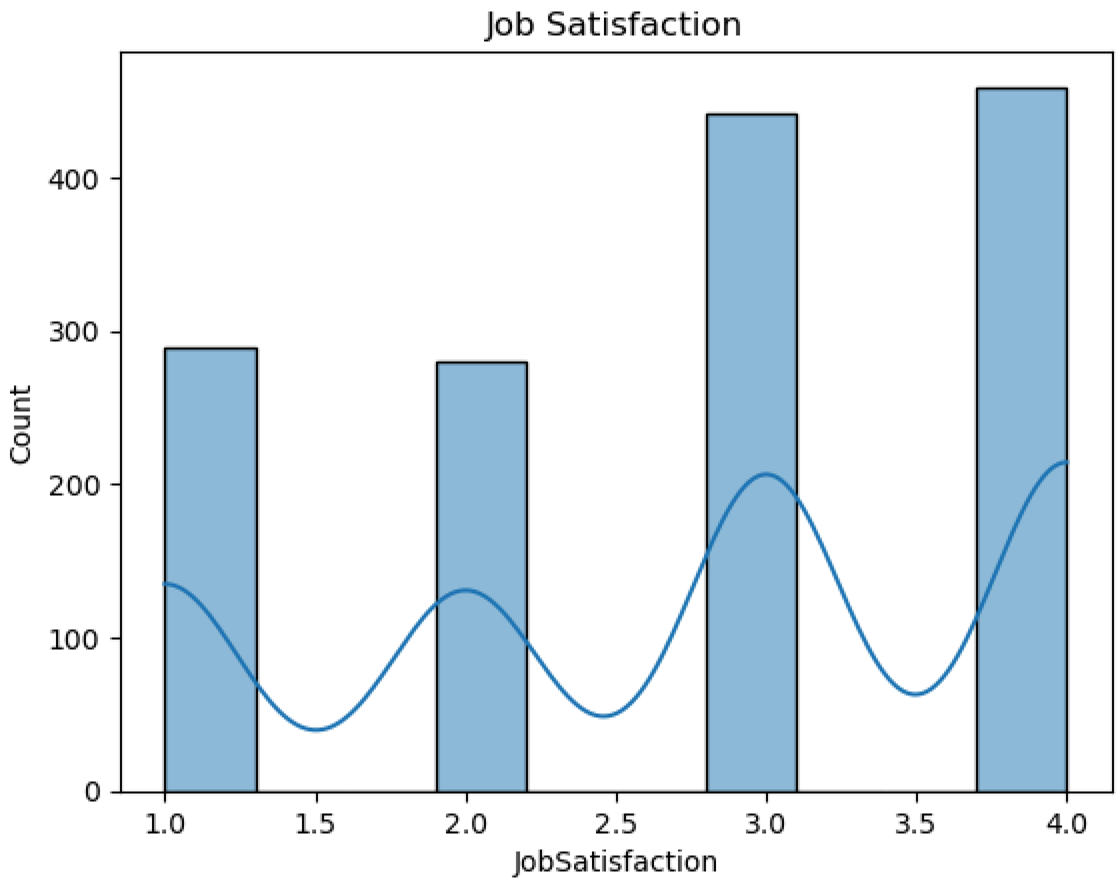


In [52]:
```
# Monthly Income
sns.histplot(df, x ='MonthlyIncome',bins = 10,kde = True)
plt.title('Monthly Income')
plt.show()
```

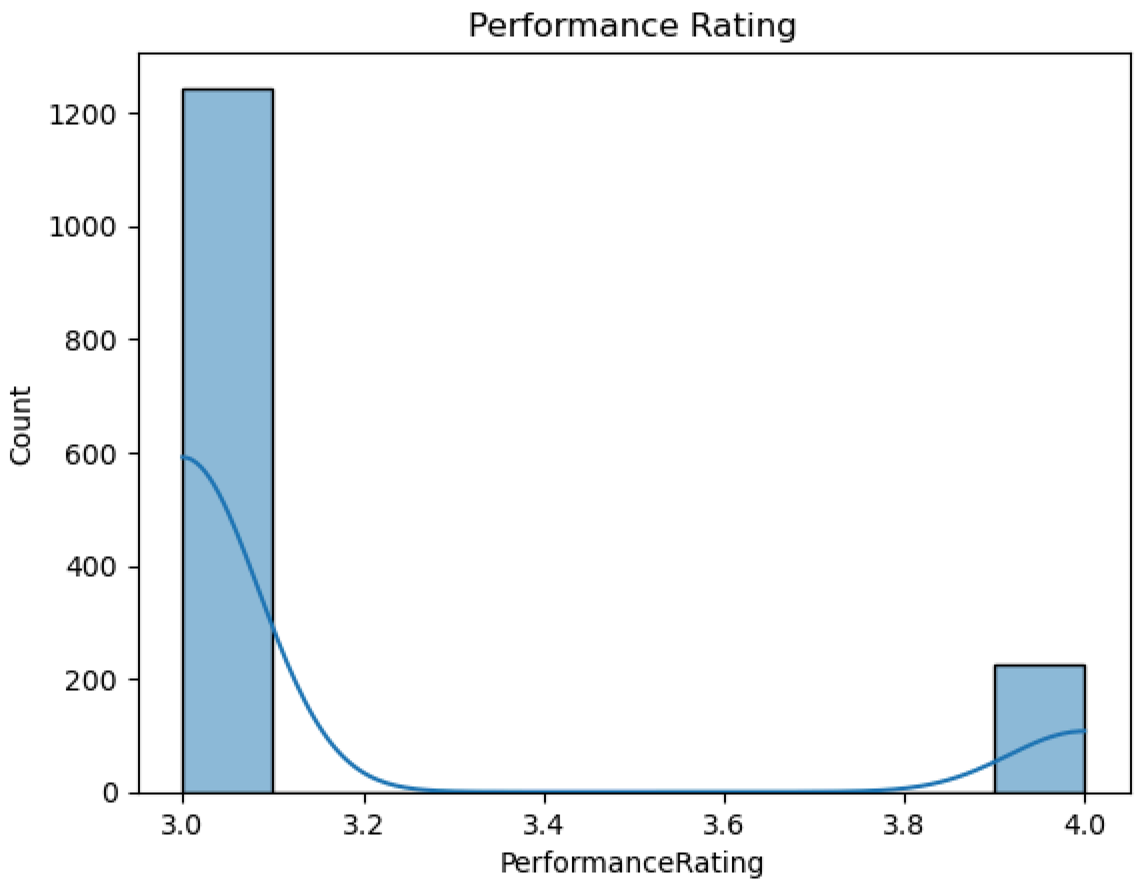
In [59]:
```
# Relationship Between Salary And  JobRole
plt.figure(figsize = (15,10))
sns.boxplot(df, x ='JobRole', y = 'MonthlyIncome')
plt.title('Relationship Between Salary And  JobRole ')
plt.xticks(rotation = 90)
plt.show()
```


Relationship Between Salary And JobRole
MonthlyIncome
20000
17500
15000
12500
10000
7500
5000
2500
Sales Executive
Research Scientist
Laboratory Technician
Manufacturing Director
Healthcare Representative
Manager
Sales Representative
Research Director
Human Resources
JobRole

In [63]:
```
# Job Satisfaction
sns.histplot(df, x ='JobSatisfaction',bins = 10,kde = True)
plt.title('Job Satisfaction')
plt.show()
```

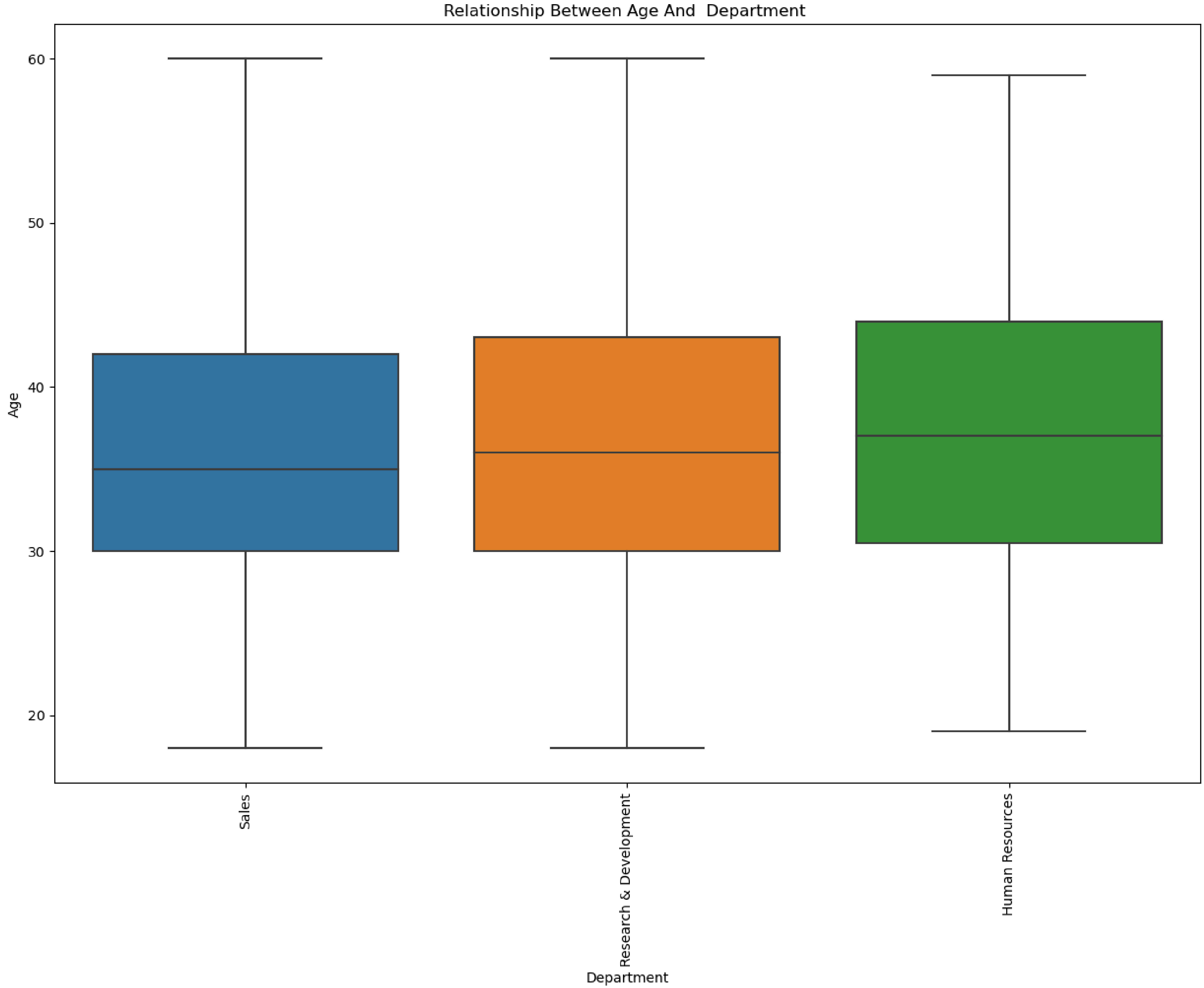


In [64]:
```
# Performance Rating
sns.histplot(df, x ='PerformanceRating',bins = 10,kde = True)
plt.title('Performance Rating')
plt.show()
```

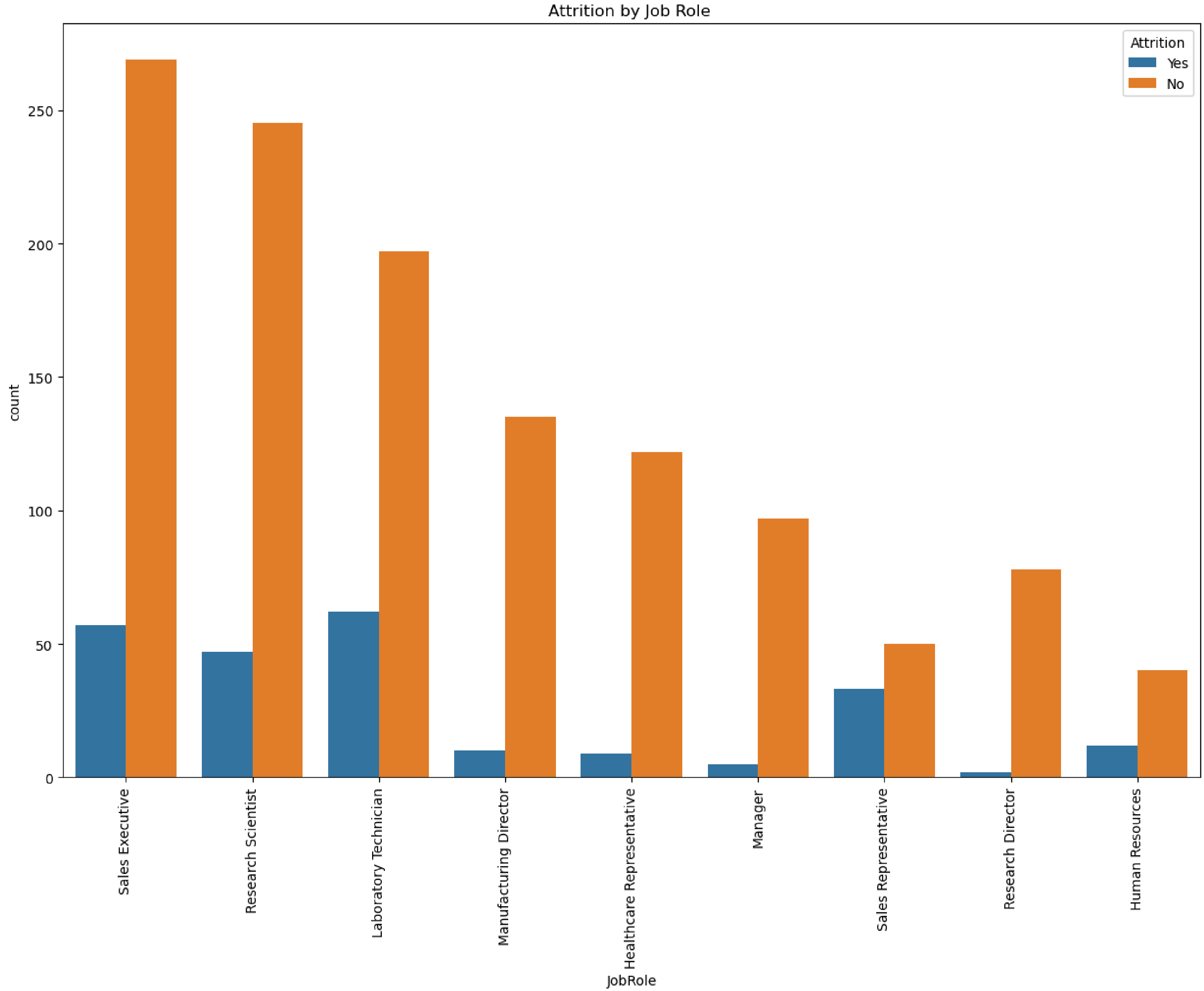


In [69]:
```
#  Employes Tenure in Companies

sns.histplot(df, x ='YearsAtCompany',bins = 10,kde = True)
plt.title('Employes Tenure in Companies')
plt.show()
```

In [70]:
```
# Relationship Between Age And  Department
plt.figure(figsize = (15,10))
sns.boxplot(df, x ='Department', y = 'Age')
plt.title('Relationship Between Age And  Department ')
plt.xticks(rotation = 90)
plt.show()
```


Relationship Between Age And Department
60
50
40
30
20
Age
Sales
Research & Development
Human Resources
Department

In [73]:
```
# Attrition by Job Role

plt.figure(figsize = (15,10))
sns.countplot(df, x ='JobRole',hue = 'Attrition')
plt.title('Attrition by Job Role ')
plt.xticks(rotation = 90)
plt.show()
```


Attrition by Job Role
Attrition
Yes
No
count
0
50
100
150
200
250
JobRole
Sales Executive
Research Scientist
Laboratory Technician
Manufacturing Director
Healthcare Representative
Manager
Sales Representative
Research Director
Human Resources

Conclusion:

- In conclusion, this project has provided valuable insights into HR Analytics using Python.
- Through exploratory data analysis, we identified key factors influencing employee attrition, satisfaction levels, and performance.
- Machine learning models enabled us to predict employee churn and classify potential candidates for promotion.
- Overall, this project highlights the significance of data-driven approaches in optimizing HR strategies and fostering a conducive work environment for organizational success.