In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df=pd.read_csv('dataset.csv')
```

In [5]:
```
df
```

Out[5]:

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

1470 rows × 35 columns

In [ ]:
```
```

In [6]:
```
df.head()
```

Out[6]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education
0	41	Yes	Travel_Rarely	1102	Sales	1	2
1	49	No	Travel_Frequently	279	Research & Development	8	1
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2
3	33	No	Travel_Frequently	1392	Research & Development	3	4
4	27	No	Travel_Rarely	591	Research & Development	2	1

5 rows × 35 columns

In [7]:
```
df.tail()
```

Out[7]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Educat
1465	36	No	Travel_Frequently	884	Research & Development	23	
1466	39	No	Travel_Rarely	613	Research & Development	6	
1467	27	No	Travel_Rarely	155	Research & Development	4	
1468	49	No	Travel_Frequently	1023	Sales	2	
1469	34	No	Travel_Rarely	628	Research & Development	8	

5 rows × 35 columns

In [9]:
```
df.shape
```

Out[9]:
```
(1470, 35)
```

In [8]:
```
df.describe()
```

Out[8]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	Employ
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1
mean	36.923810	802.485714	9.192517	2.912925	1.0	1
std	9.135373	403.509100	8.106864	1.024165	0.0	
min	18.000000	102.000000	1.000000	1.000000	1.0	
25%	30.000000	465.000000	2.000000	2.000000	1.0	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1
max	60.000000	1499.000000	29.000000	5.000000	1.0	2

8 rows × 26 columns

In [11]:
```
df.nunique()
```

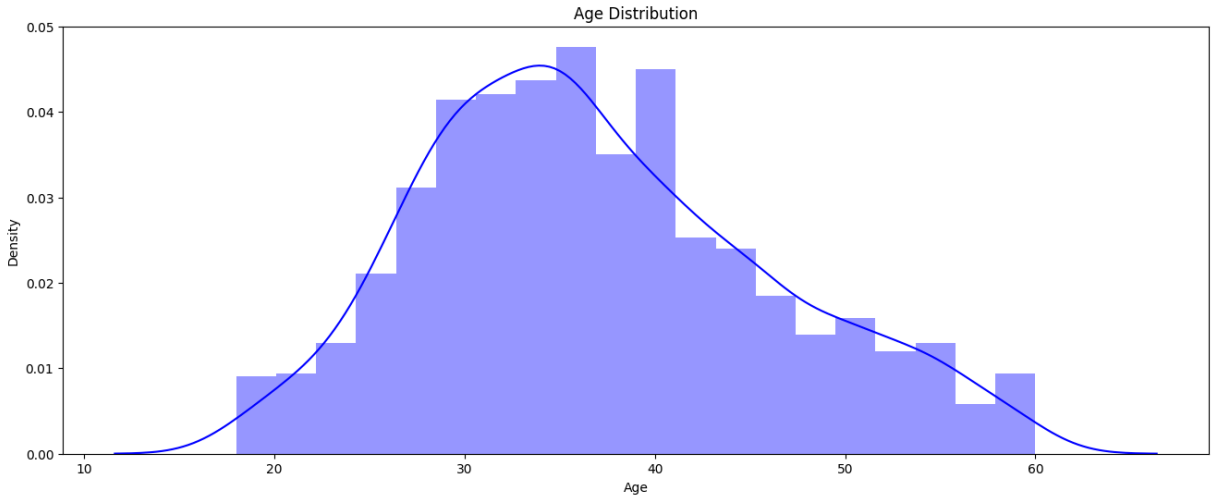
Out[11]:
```
Age                         43
Attrition                    2
BusinessTravel               3
DailyRate                  886
Department                   3
DistanceFromHome            29
Education                    5
EducationField               6
EmployeeCount                1
EmployeeNumber            1470
EnvironmentSatisfaction      4
Gender                       2
HourlyRate                  71
JobInvolvement               4
JobLevel                     5
JobRole                      9
JobSatisfaction              4
MaritalStatus                3
MonthlyIncome             1349
MonthlyRate               1427
NumCompaniesWorked          10
Over18                       1
OverTime                     2
PercentSalaryHike           15
PerformanceRating            2
RelationshipSatisfaction     4
StandardHours                1
StockOptionLevel             4
TotalWorkingYears           40
TrainingTimesLastYear        7
WorkLifeBalance              4
YearsAtCompany              37
YearsInCurrentRole          19
YearsSinceLastPromotion     16
YearsWithCurrManager        18
dtype: int64
```

In [9]:
```
df['Age'].unique()
```

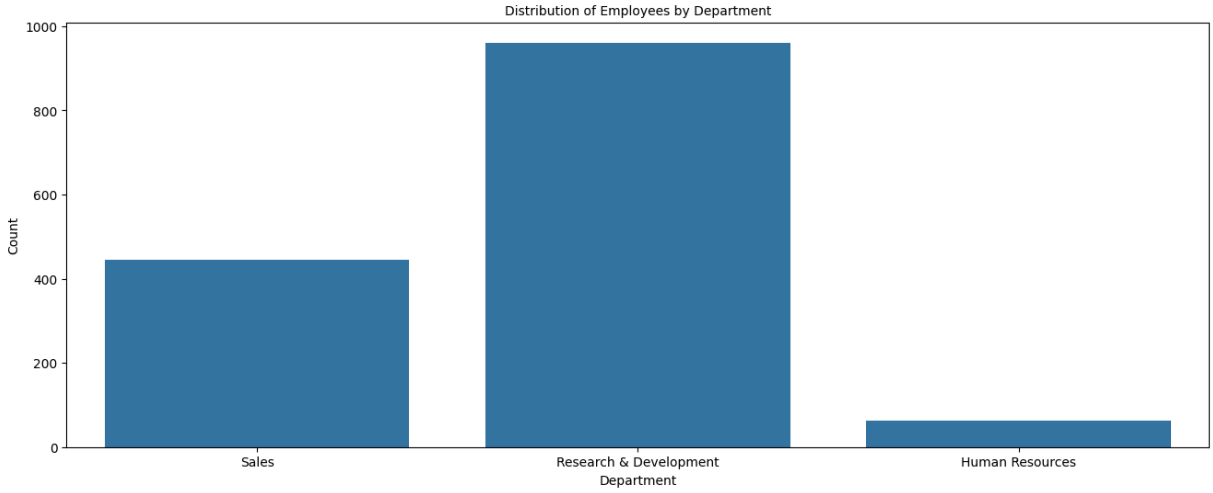
Out[9]:
```
array([41, 49, 37, 33, 27, 32, 59, 30, 38, 36, 35, 29, 31, 34, 28, 22, 53,
       24, 21, 42, 44, 46, 39, 43, 50, 26, 48, 55, 45, 56, 23, 51, 40, 54,
       58, 20, 25, 19, 57, 52, 47, 18, 60])
```

In [15]:
```
plt.figure(figsize=(16,6))
sns.distplot( df['Age'], bins=20, color='blue')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.show()
```

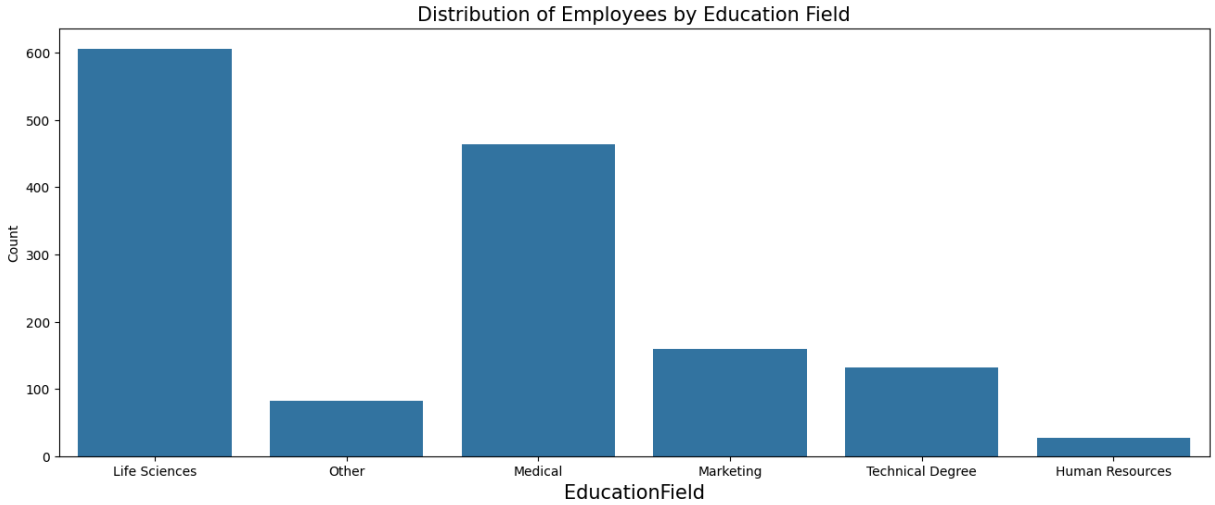
```
C:\Users\goudv\AppData\Local\Temp\ipykernel_10928\1930959742.py:2: UserWarning:

`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( df['Age'], bins=20, color='blue')
```



In [10]:
```
plt.figure(figsize=(16,6))
sns.countplot(data=df, x='Department')
plt.xlabel('Department',fontsize=10)
plt.ylabel('Count', fontsize=10)
plt.title('Distribution of Employees by Department', fontsize=10)
plt.show()
```

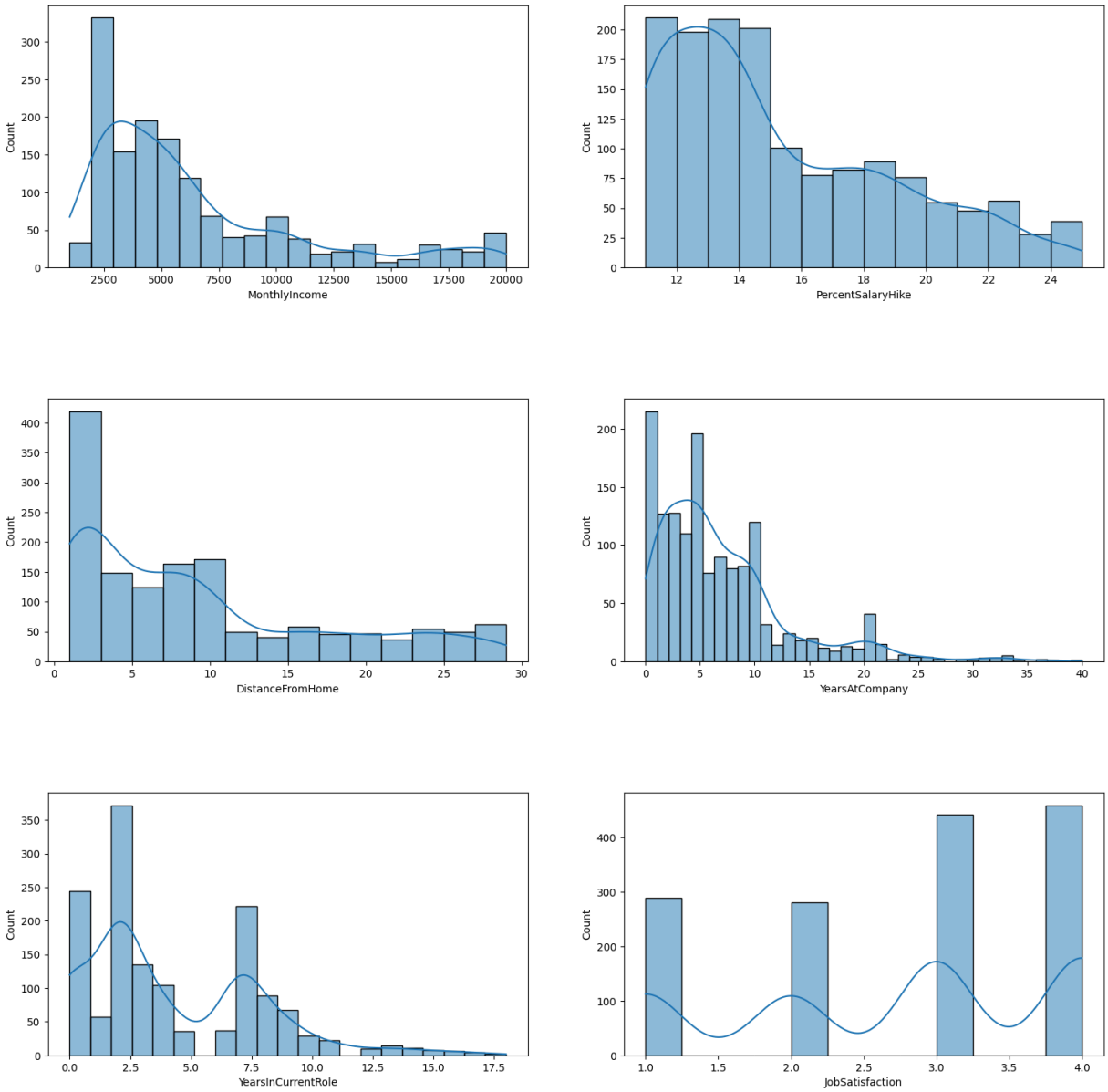


In [11]:
```
plt.figure(figsize=(16, 6))
sns.countplot(data=df , x='EducationField')
plt.xlabel('EducationField', fontsize=15)
plt.ylabel('Count', fontsize=10)
plt.title('Distribution of Employees by Education Field', fontsize=15)
plt.show()
```

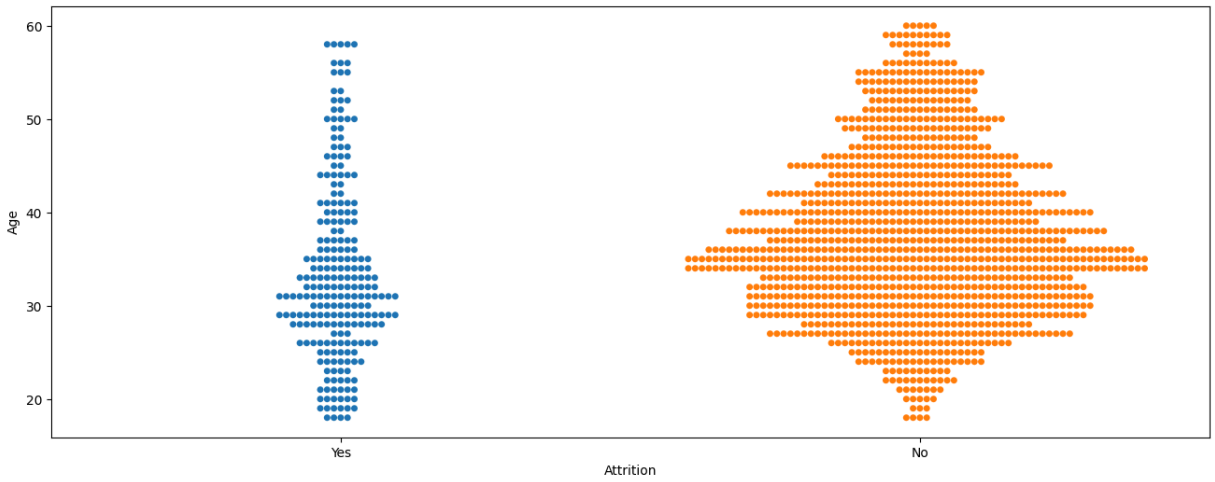


In [16]:
```
fig, axes = plt.subplots(3, 2, figsize=(18, 18), gridspec_kw={'hspace': 0.5})
fig.suptitle('Histograms for All Numerical Variables in the Dataset')

sns.histplot(df['MonthlyIncome'], ax=axes[0, 0], kde=True)
sns.histplot(df['PercentSalaryHike'], ax=axes[0, 1], kde=True)
sns.histplot(df['DistanceFromHome'], ax=axes[1, 0], kde=True)
sns.histplot(df['YearsAtCompany'], ax=axes[1, 1], kde=True)
sns.histplot(df['YearsInCurrentRole'], ax=axes[2, 0], kde=True)
sns.histplot(df['JobSatisfaction'], ax=axes[2, 1], kde=True)
```

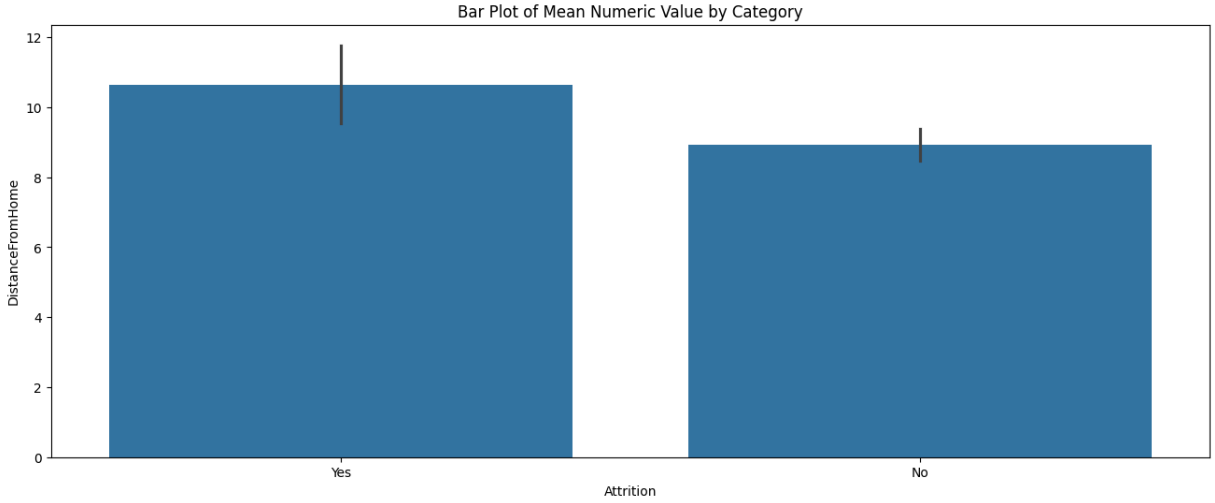
Out[16]:
```
<Axes: xlabel='JobSatisfaction', ylabel='Count'>
```



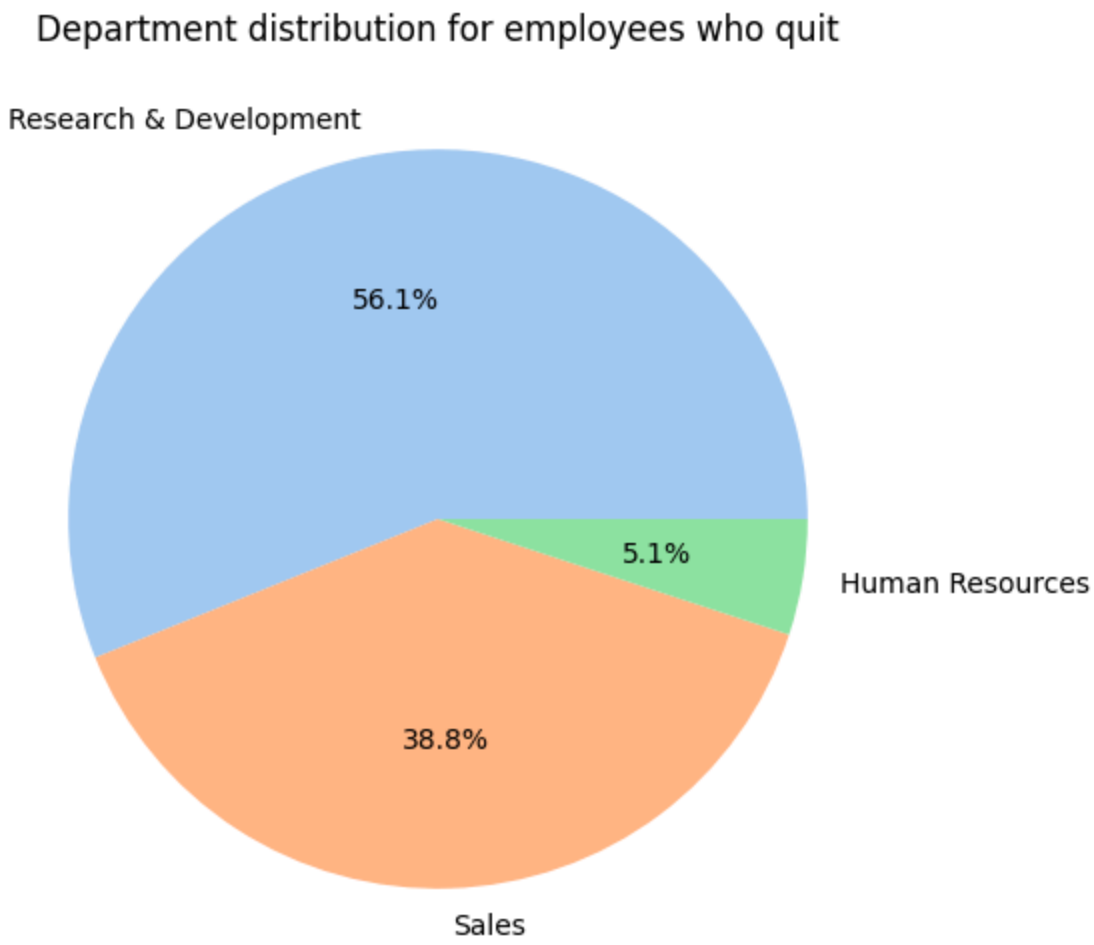
In [17]:
```
plt.figure(figsize = (16,6))
sns.swarmplot(y = 'Age', x = 'Attrition', data = df, hue = 'Attrition')
plt.show()
```



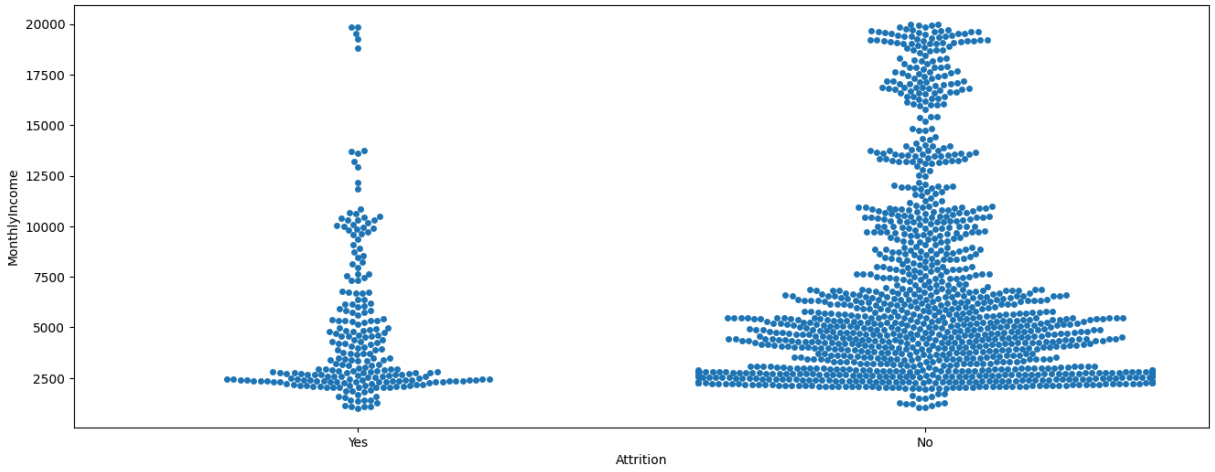
In [18]:
```
plt.figure(figsize=(16, 6))
sns.barplot(x='Attrition', y='DistanceFromHome', data=df)
plt.xlabel('Attrition')
plt.ylabel('DistanceFromHome')
plt.title('Bar Plot of Mean Numeric Value by Category')
plt.show()
```



In [19]:
```
plt.figure(figsize=(16,6))
department_count = df[df['Attrition'] == 'Yes']['Department'].value_counts()
label = department_count.index.tolist()

# Creating a pie chart using the value counts of the 'Department' column
plt.pie(department_count, labels=label, autopct='%1.1f%%', colors=sns.color_palette

# Setting the title for the pie chart
plt.title('Department distribution for employees who quit')

# Display the pie chart
plt.show()
```

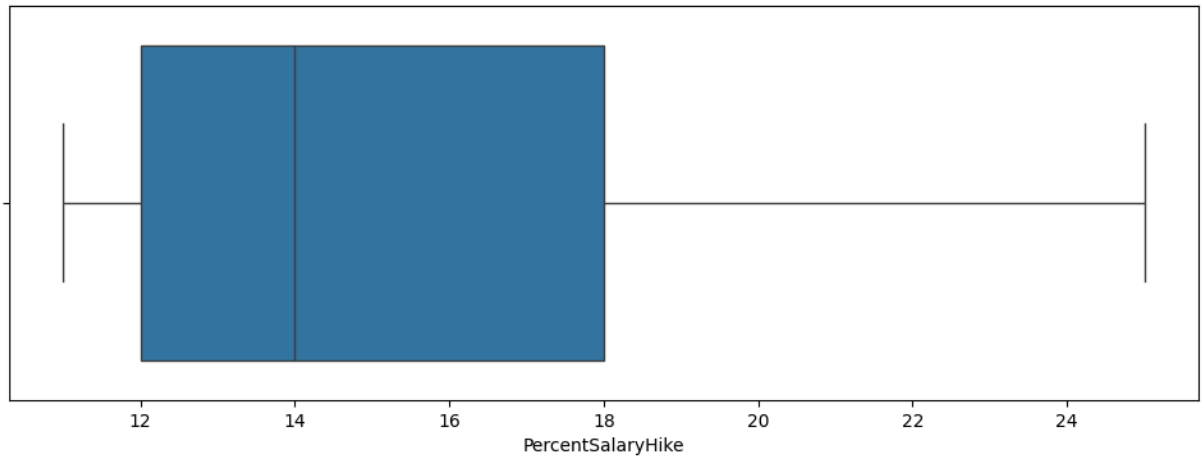


In [22]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming df is your DataFrame containing the data
plt.figure(figsize=(16, 6))
sns.swarmplot(x='Attrition', y='MonthlyIncome', data=df)
plt.show()
```

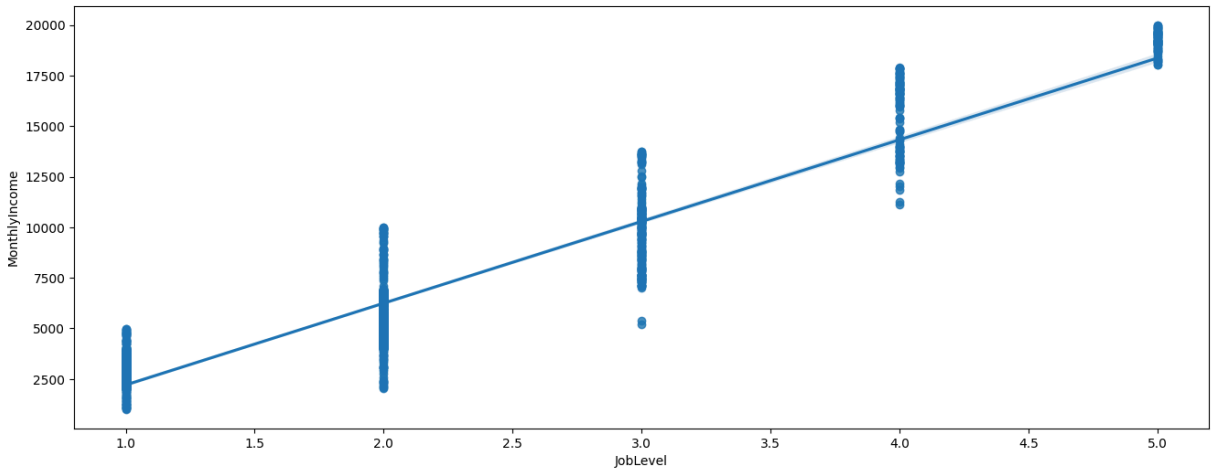


In [23]:
```
plt.figure(figsize = (12,4))
sns.boxplot(x='PercentSalaryHike', data =df)
```

```
plt.show()
```



In [25]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming df is your DataFrame containing the data
plt.figure(figsize=(16, 6))
sns.regplot(x='JobLevel', y='MonthlyIncome', data=df)
plt.show()
```



In [26]:
```
plt.figure(figsize =(17,10))
sns.heatmap(df.corr() ,annot=True, cmap='rainbow')
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[26], line 2
      1 plt.figure(figsize =(17,10))
----> 2 sns.heatmap(df.corr() ,annot=True, cmap='rainbow')
      3 plt.show()

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py:11049, in DataFrame.corr(self, method, min_periods, numeric_only)
  11047 cols = data.columns
  11048 idx = cols.copy()
> 11049 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  11051 if method == "pearson":
  11052     correl = libalgos.nancorr(mat, minp=min_periods)

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py:1993, in DataFrame.to_numpy(self, dtype, copy, na_value)
   1991 if dtype is not None:
   1992     dtype = np.dtype(dtype)
-> 1993 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
   1994 if result.dtype is not dtype:
   1995     result = np.asarray(result, dtype=dtype)

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\internals\managers.py:1694, in BlockManager.as_array(self, dtype, copy, na_value)
   1692         arr.flags.writeable = False
   1693 else:
-> 1694     arr = self._interleave(dtype=dtype, na_value=na_value)
   1695     # The underlying data was copied within _interleave, so no need
   1696     # to further copy if copy=True or setting na_value
   1698 if na_value is lib.no_default:

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\internals\managers.py:1753, in BlockManager._interleave(self, dtype, na_value)
   1751     else:
   1752         arr = blk.get_values(dtype)
-> 1753     result[rl.indexer] = arr
   1754     itemmask[rl.indexer] = 1
   1756 if not itemmask.all():

ValueError: could not convert string to float: 'Yes'
```


In [27]:
```
print(df.columns)
```

```
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 [28]:
```
numeric_columns = df.select_dtypes(include=np.number).columns  # Select numeric col
plt.figure(figsize=(17, 10))
plt.show()
```