HR EMPLOYEE ATTRITION ANALYSIS

In [1]:
```python
# import libraries
import pandas as pd
import numpy as np
```

In [2]:
```python
df = pd.read_csv('HR-Employee-Attrition.csv')
```

In [3]:
```python
df.head()
```

Out[3]:

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

5 rows × 35 columns

In [4]:
```python
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 [5]:

```
# checking for duplicate values
df.duplicated().sum()
```

Out[5]:

```
0
```

In [6]:

```
# summary statistics
df.describe()
```

Out[6]:

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

8 rows × 26 columns

In [7]:

```
# Check for unique values in the categorical columns

for col in df.columns:
    if df[col].dtype == 'object':
        print(f'{col} unique values are :',df[col].unique())
        print('\n')
```

```
Attrition unique values are : ['Yes' 'No']


BusinessTravel unique values are : ['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']


Department unique values are : ['Sales' 'Research & Development' 'Human Resources']


EducationField unique values are : ['Life Sciences' 'Other' 'Medical' 'Marketing' 'Techn
ical Degree'
 'Human Resources']


Gender unique values are : ['Female' 'Male']


JobRole unique values are : ['Sales Executive' 'Research Scientist' 'Laboratory Technici
an'
 'Manufacturing Director' 'Healthcare Representative' 'Manager'
```

```
 'Sales Representative' 'Research Director' 'Human Resources']


MaritalStatus unique values are : ['Single' 'Married' 'Divorced']


Over18 unique values are : ['Y']


OverTime unique values are : ['Yes' 'No']
```

OBSERVATION:

- `Over18` has only one unique value (`Y`) which makes it irrelevant for our analysis

DATA CLEANING

In [8]:
```
# drop redundant columns
df.drop(['Over18', 'EmployeeCount', 'EmployeeNumber', 'StandardHours'],
        axis=1, inplace=True)
```

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

Out[9]:
```
(1470, 31)
```

In [10]:
```
# descretization of the Age column

bins = [17, 29, 39, 49, 60]
group_names = ['Youth', 'YoungAdult', 'MiddleAged', 'OldAdults']
df['AgeCategory'] = pd.cut(df['Age'], bins, labels=group_names)
```

In [11]:
```
df['AgeCategory'].value_counts()
```

Out[11]:
```
YoungAdult    622
MiddleAged    349
Youth         326
OldAdults     173
Name: AgeCategory, dtype: int64
```

In [12]:
```
# Encodes the Attrition column
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
df['AttritionEncoded'] = label_encoder.fit_transform(df['Attrition'])
```

Exploratory Data Analysis(EDA)

In [13]:
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid')
```

1. UNIVARIATE

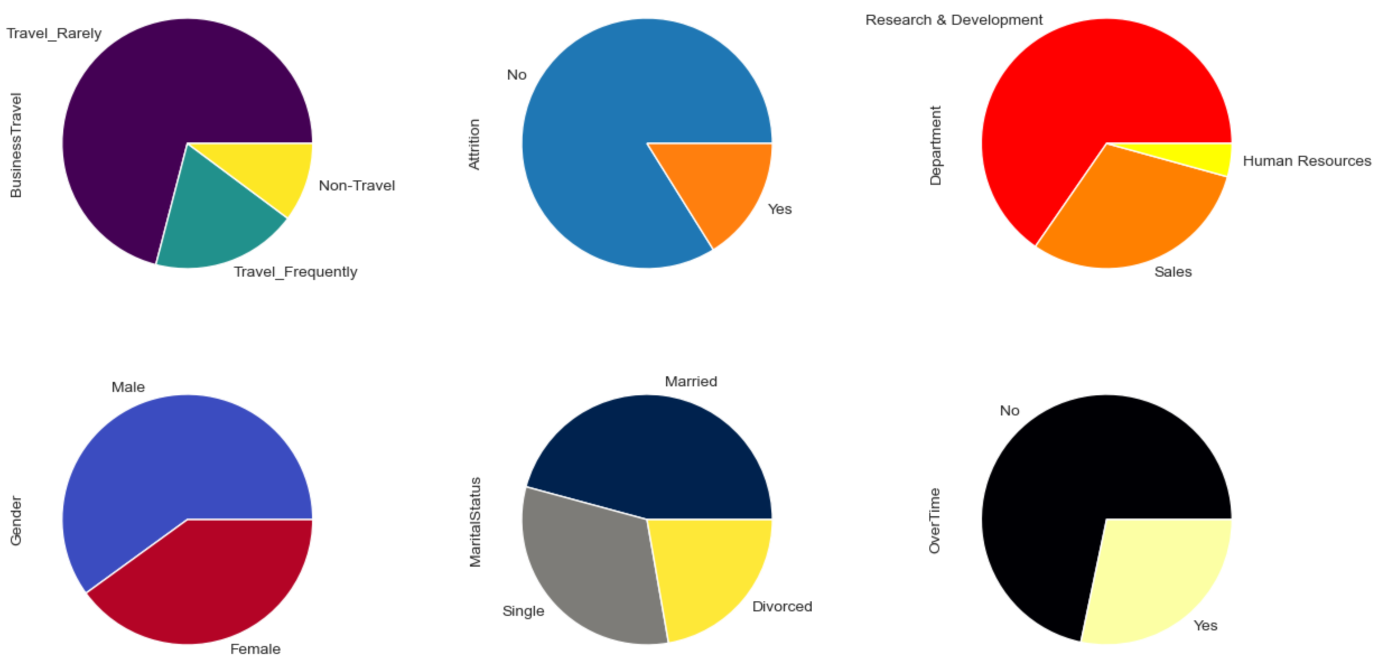
PIE CHART

In [14]:
```
fig, axs = plt.subplots(2,3, figsize=(15, 8))
```

```
#Pie chart for Business Travel
df['BusinessTravel'].value_counts().plot(kind='pie',cmap='viridis',ax=axs[0][0])
#Pie chart for Attrition
df['Attrition'].value_counts().plot(kind='pie',ax=axs[0][1])
#Pie chart for Department
df['Department'].value_counts().plot(kind='pie',cmap='autumn',ax=axs[0][2])
#Pie chart for Gender
df['Gender'].value_counts().plot(kind='pie',cmap='coolwarm',ax=axs[1][0])
#Pie chart for Marital Status
df['MaritalStatus'].value_counts().plot(kind='pie',cmap='cividis',ax=axs[1][1])
#Pie chart for Overtime
df['OverTime'].value_counts().plot(kind='pie',cmap='inferno',ax=axs[1][2])
```

Out[14]:

```
&lt;AxesSubplot:ylabel='OverTime'&gt;
```



OBSERVATIONS:

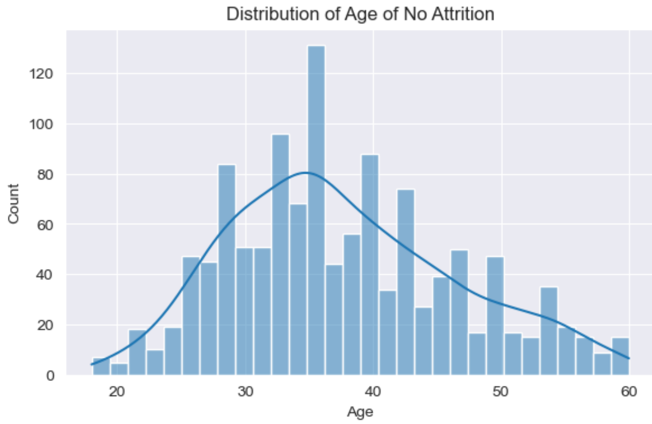
- Most of employee rarely travel and most of them dont travel
- Most of the employee have No Attrition
- Most of the employee are from the Research and Development department followed by Sales then Human Resources
- There are more Male than Female
- Most of the employee are married
- Most of the employee dont work overtime

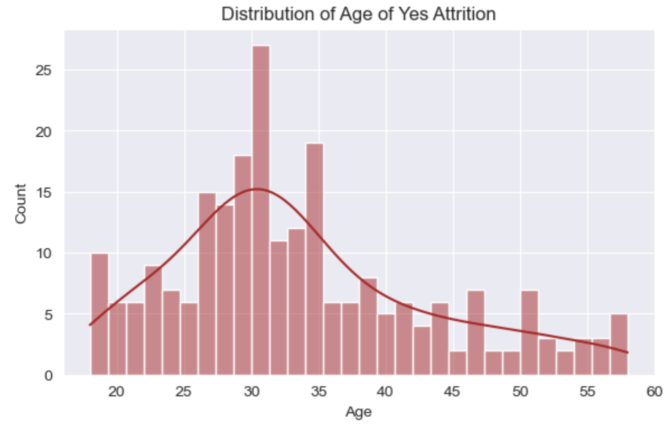
In [15]:

```
fig, axs = plt.subplots(1,2, figsize=(15, 4))

sns.histplot(df[df['Attrition'] == 'No']['Age'], bins=30, kde=True, ax=axs[0])
sns.histplot(df[df['Attrition'] == 'Yes']['Age'], bins=30, kde=True, ax=axs[1],color='br

axs[0].set_title('Distribution of Age of No Attrition')
axs[1].set_title('Distribution of Age of Yes Attrition')
```

Out[15]:

```
Text(0.5, 1.0, 'Distribution of Age of Yes Attrition')
```

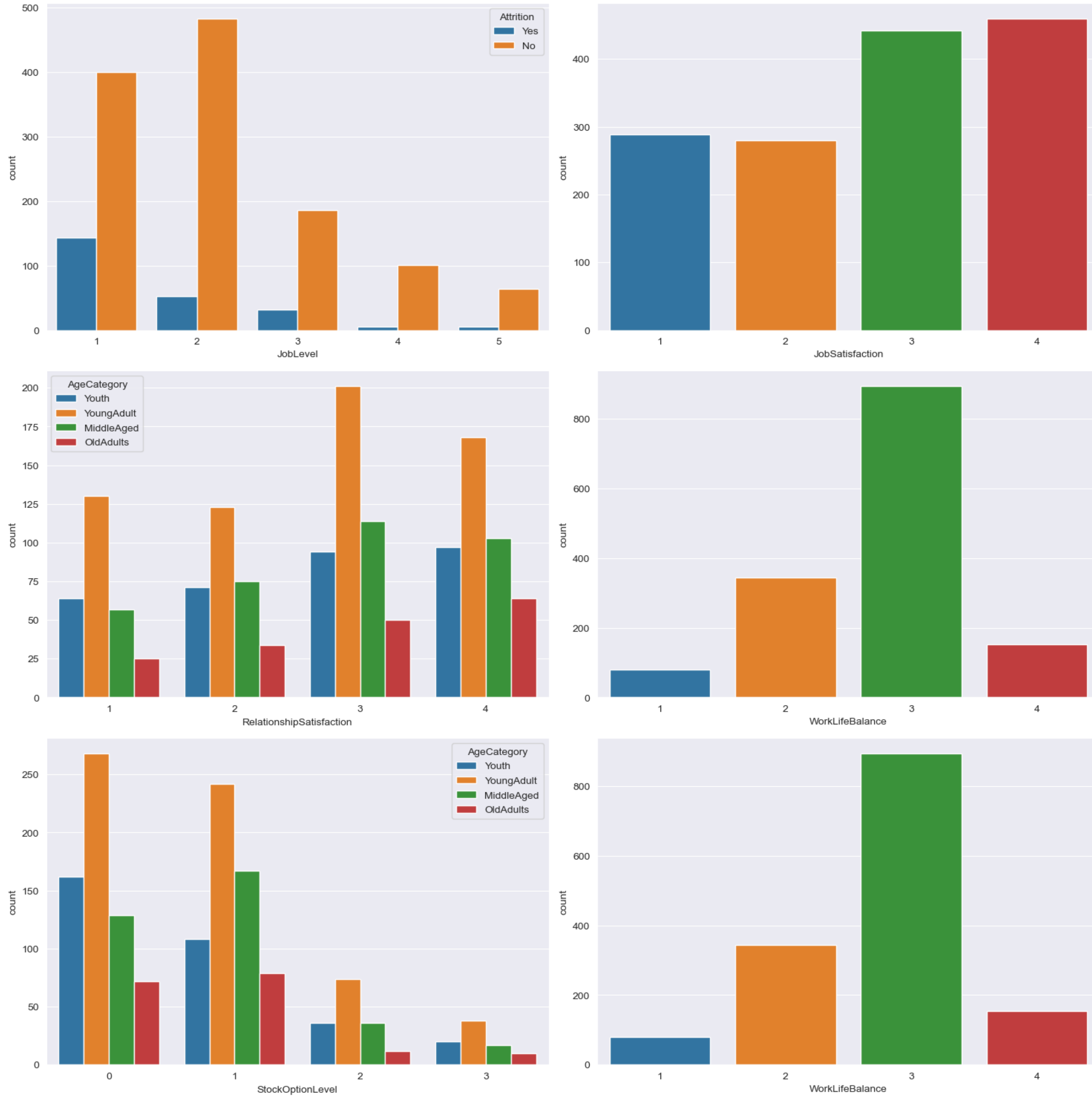




OBSERVATION:

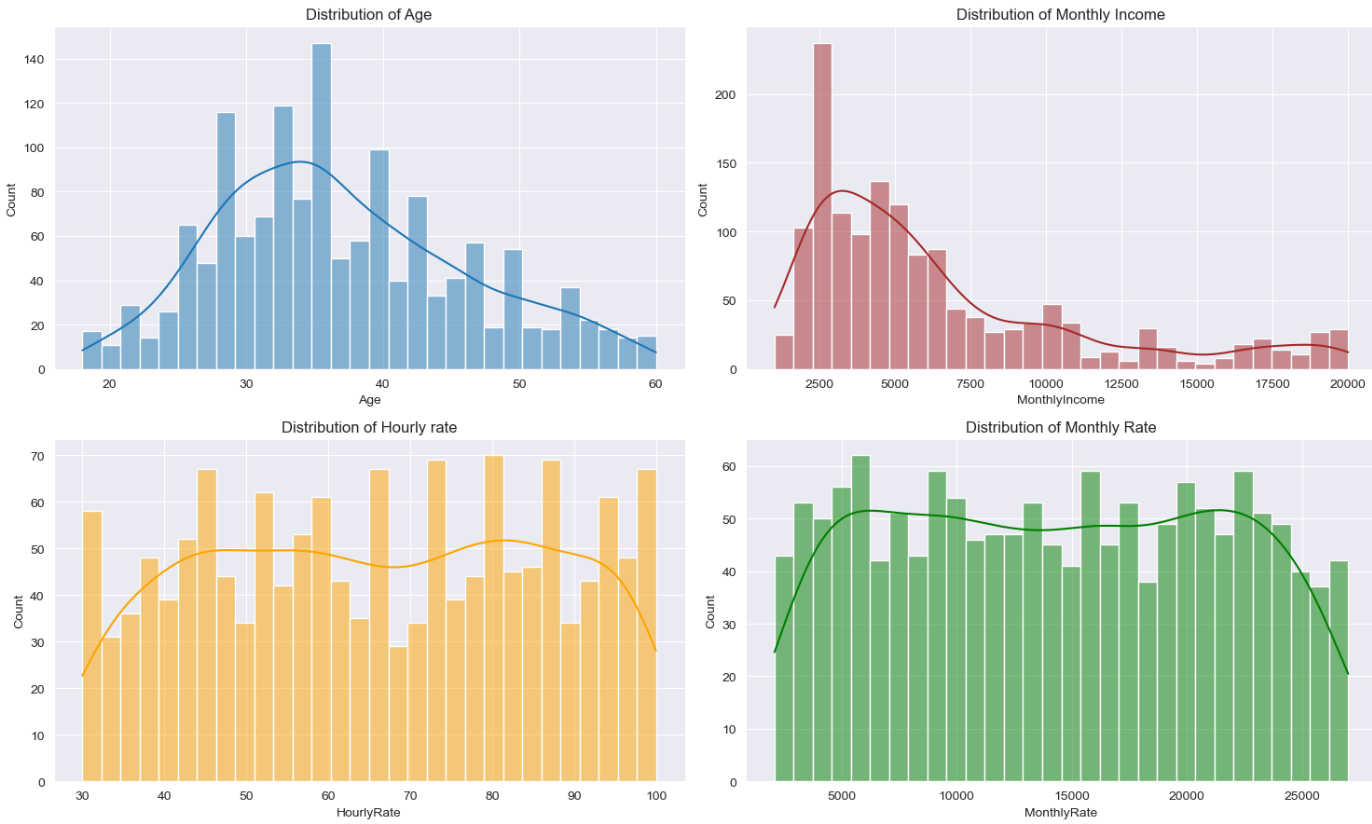
The Age Distribution looks the same for both category of Attrition

In [16]:
```
fig, axs = plt.subplots(3,2, figsize=(15, 15))

sns.countplot(x=df['JobLevel'],hue=df['Attrition'],ax=axs[0][0])
sns.countplot(x=df['JobSatisfaction'],ax=axs[0][1])
sns.countplot(x=df['RelationshipSatisfaction'],hue=df['AgeCategory'], ax=axs[1][0])
sns.countplot(x=df['WorkLifeBalance'],ax=axs[1][1])
sns.countplot(x=df['StockOptionLevel'],hue=df['AgeCategory'],ax=axs[2][0])
sns.countplot(x=df['WorkLifeBalance'],ax=axs[2][1])


plt.tight_layout()
```



In [17]:

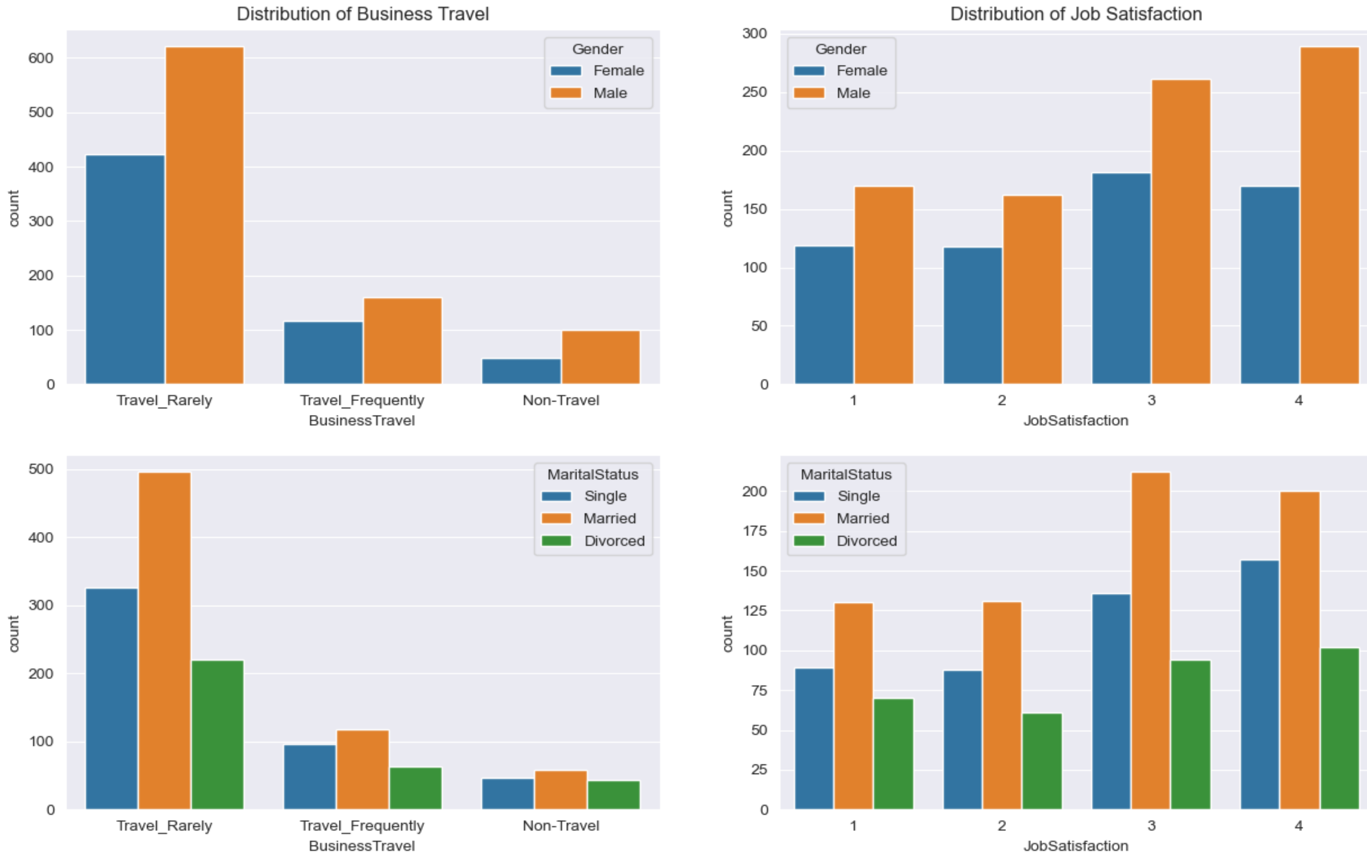
```
fig, axs = plt.subplots(2,2, figsize=(15, 9))

sns.histplot(df['Age'], bins=30, kde=True, ax=axs[0][0])
sns.histplot(df['MonthlyIncome'], bins=30, kde=True, ax=axs[0][1], color='brown')
sns.histplot(df['HourlyRate'], bins=30, kde=True, ax=axs[1][0],color='orange')
sns.histplot(df['MonthlyRate'],  bins=30, kde=True, ax=axs[1][1],color='green')

axs[0][0].set_title('Distribution of Age')
axs[0][1].set_title('Distribution of Monthly Income')
axs[1][0].set_title('Distribution of Hourly rate')
axs[1][1].set_title('Distribution of Monthly Rate')

plt.tight_layout()
```



2. Bivariate

In [18]:
```
fig, axs = plt.subplots(2,2, figsize=(15, 9))

sns.countplot(x=df['BusinessTravel'],hue=df['Gender'], ax=axs[0][0])
sns.countplot(x=df['JobSatisfaction'], hue=df['Gender'], ax=axs[0][1])
sns.countplot(x=df['BusinessTravel'], hue=df['MaritalStatus'], ax=axs[1][0])
sns.countplot(x=df['JobSatisfaction'],hue=df['MaritalStatus'], ax=axs[1][1])

axs[0][0].set_title('Distribution of Business Travel')
axs[0][1].set_title('Distribution of Job Satisfaction')
```

Out[18]:
```
Text(0.5, 1.0, 'Distribution of Job Satisfaction')
```

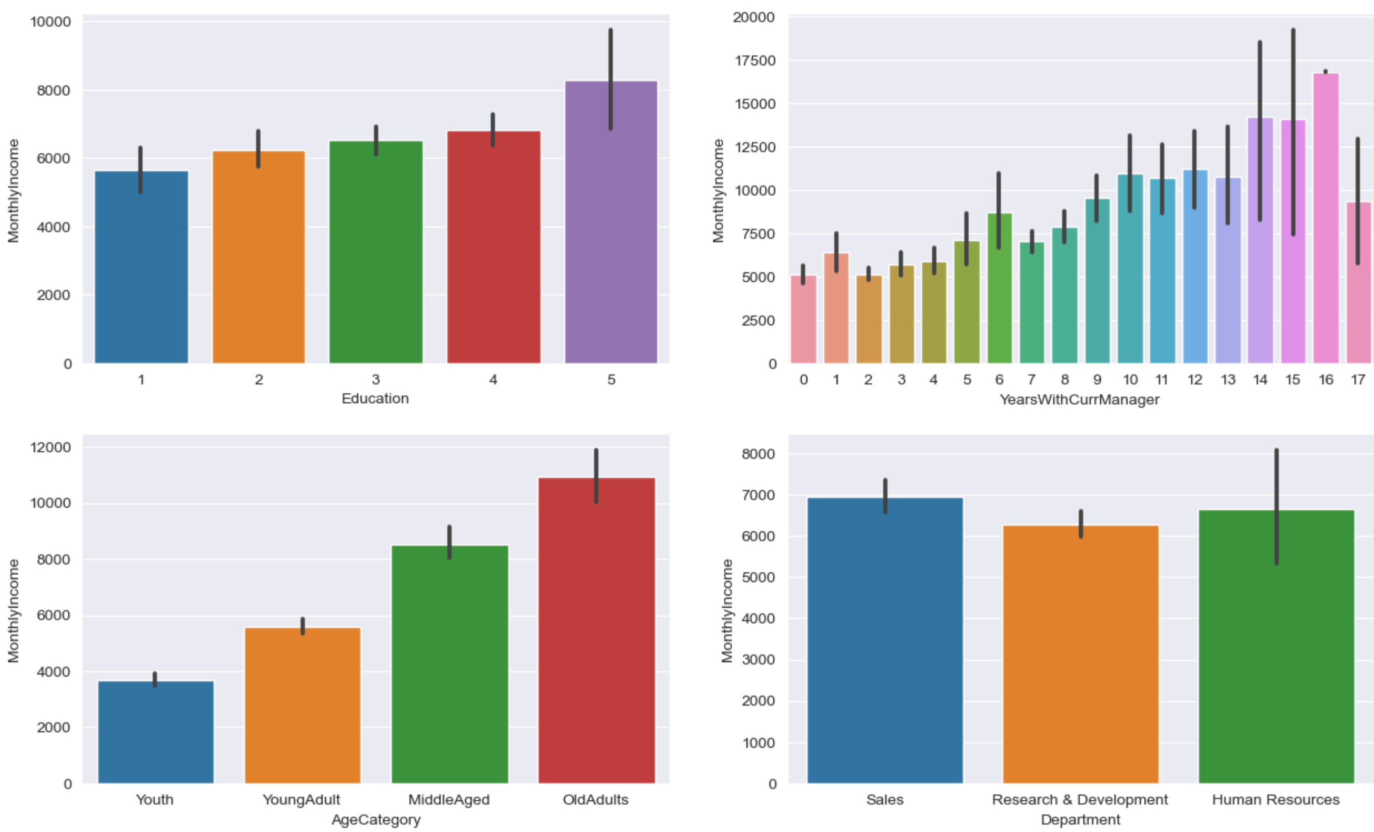


In [19]:

```
fig, axs = plt.subplots(2,2, figsize=(15, 9))

sns.barplot(x=df['Education'], y=df['MonthlyIncome'],estimator=np.mean, ax=axs[0][0])
sns.barplot(x=df['YearsWithCurrManager'], y=df['MonthlyIncome'],estimator=np.mean,ax=axs
sns.barplot(x=df['AgeCategory'], y=df['MonthlyIncome'],estimator=np.mean, ax=axs[1][0])
sns.barplot(x=df['Department'], y=df['MonthlyIncome'],estimator=np.mean, ax=axs[1][1])
```

Out[19]:

```
<AxesSubplot:xlabel='Department', ylabel='MonthlyIncome'>
```

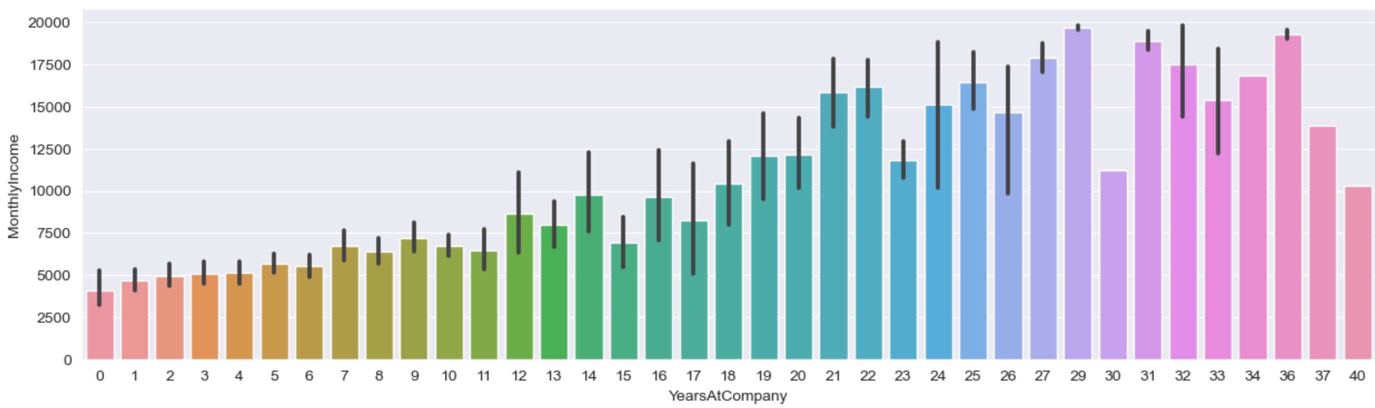


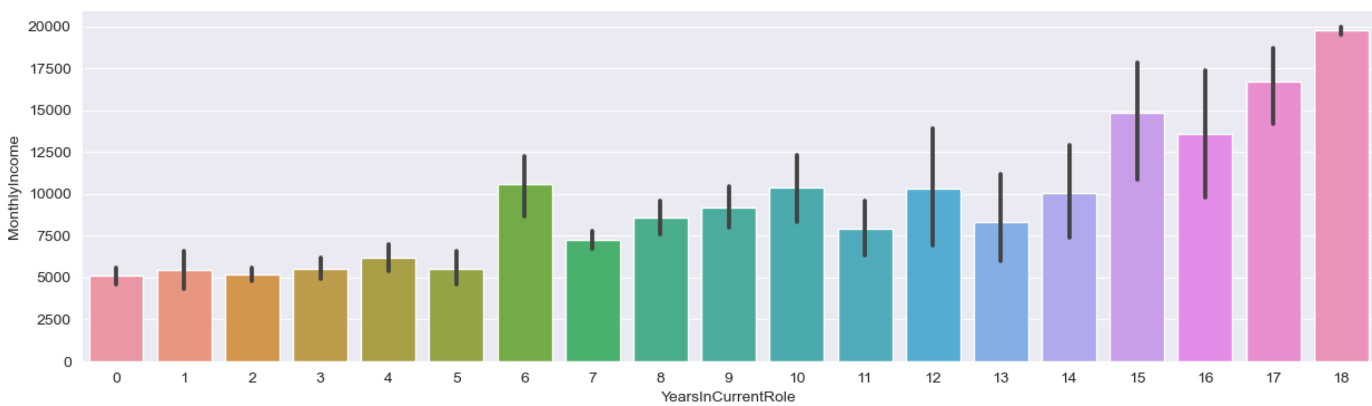
OBSERVATION:

- The mean Monthly income increases with `Education`, `Age` and `Years with current manager`
- The sales department has the mean highest monthly income followed by Human resources then Research and development

In [20]:
```
fig, axs = plt.subplots(2,1, figsize=(15, 9))

sns.barplot(x=df['YearsAtCompany'], y=df['MonthlyIncome'], estimator=np.mean, ax=axs[0])
sns.barplot(x=df['YearsInCurrentRole'], y=df['MonthlyIncome'], estimator=np.mean, ax=axs
```

Out[20]:
```
<AxesSubplot:xlabel='YearsInCurrentRole', ylabel='MonthlyIncome'>
```





OBSERVATION:

- The mean monthly income increases as the number of years in the company increases
- The mean monthly income increases as the number of years in the current role increases

STATISTICAL ANALYSIS

In [21]:
```
import scipy.stats as stats
```

In [22]:
```
alpha = 0.05
```

CHI-SQUARE TEST OF INDEPENDENCE

`1. Attrition vs Gender`

- H0: There is no association between Attrition and Gender
- H1: There is an association between Attrition and Gender

In [23]:
```
# Create a contingency table of Attrition and Gender
crosstab = pd.crosstab(df['Attrition'], df['Gender'])
```

```
crosstab
```

Out[23]:

Gender	Female	Male
Attrition		
No	501	732
Yes	87	150

In [24]:

```
# Perform a chi-square test for association between Attrition and Gender
chi2, p_value, _, _ = stats.chi2_contingency(crosstab)

print("Chi-Square Test Results:")
print("Chi-Square:", chi2)
print("p-value:", p_value)

#reject H0?
print('\nReject H0?: ')
if p_value < alpha:
    print('True')
else:
    print('False')
```

```
Chi-Square Test Results:
Chi-Square: 1.1169671241970975
p-value: 0.29057244902890855

Reject H0?:
False
```

CONCLUSION:

Since we do not reject the null hypothesis, the conclusion is that there is enough evidence to support the claim that Attrition does not depend on gender

`2.Attrition vs Business Travel`

- H0: There is no association between Attrition and Business Travel
- H1: There is an association between Attrition and Business Travel

In [25]:

```
# Create a contingency table of Attrition and Business Travel
crosstab = pd.crosstab(df['Attrition'], df['BusinessTravel'])
crosstab
```

Out[25]:

BusinessTravel	Non-Travel	Travel_Frequently	Travel_Rarely
Attrition			
No	138	208	887
Yes	12	69	156

In [26]:

```
# Perform a chi-square test for association between Attrition and Business Travel
chi2, p_value, _, _ = stats.chi2_contingency(crosstab)

print("Chi-Square Test Results:")
print("Chi-Square:", chi2)
print("p-value:", p_value)

#reject H0?
```

```python
print('\nReject H0?: ')
if p_value < alpha:
    print('True')
else:
    print('False')
```

```
Chi-Square Test Results:
Chi-Square: 24.182413685655174
p-value: 5.608614476449931e-06

Reject H0?:
True
```

CONCLUSION:

The conclusion is that there is enough evidence to support the claim that Attrition is related to how frequent an employee travels

`3. ATTRITION vs DEPARTMENT`

- H0: There is no association between Attrition and Department
- H1: There is an association between Attrition and Department

In [27]:

```python
# Create a contingency table of Attrition and Department
crosstab = pd.crosstab(df['Attrition'], df['Department'])
crosstab
```

Out[27]:

Department	Human Resources	Research & Development	Sales
Attrition			
No	51	828	354
Yes	12	133	92

In [28]:

```python
# Perform a chi-square test for association between Attrition and Department
chi2, p_value, _, _ = stats.chi2_contingency(crosstab)

print("Chi-Square Test Results:")
print("Chi-Square:", chi2)
print("p-value:", p_value)

#reject H0?
print('\nReject H0?: ')
if p_value < alpha:
    print('True')
else:
    print('False')
```

```
Chi-Square Test Results:
Chi-Square: 10.79600732241067
p-value: 0.004525606574479633

Reject H0?:
True
```

CONCLUSION:

The conclusion is that there is enough evidence to support the claim that Attrition is related to the department where the employee works

`ATTRITION VS MARITAL STATUS`

- H0: There is no association between Attrition and Marital Status
- H1: There is an association between Attrition and Marital Status

In [29]:
```
# Create a contingency table of Attrition and Marital Status
crosstab = pd.crosstab(df['Attrition'], df['MaritalStatus'])
crosstab
```

Out[29]:

MaritalStatus	Divorced	Married	Single
Attrition			
No	294	589	350
Yes	33	84	120

In [30]:
```
# Perform a chi-square test for association between Attrition and Marital Status
chi2, p_value, _, _ = stats.chi2_contingency(crosstab)

print("Chi-Square Test Results:")
print("Chi-Square:", chi2)
print("p-value:", p_value)

#reject H0?
print('\nReject H0?: ')
if p_value < alpha:
    print('True')
else:
    print('False')
```

```
Chi-Square Test Results:
Chi-Square: 46.163676540848705
p-value: 9.45551106034083e-11

Reject H0?:
True
```

CONCLUSION:

The conclusion is that there is enough evidence to support the claim that Attrition is related to the Martital Status of the employee

`ATTRITION VS OVERTIME`

- H0: There is no association between Attrition and Overtime
- H1: There is an association between Attrition and Overtime

In [31]:
```
# Create a contingency table of Attrition and Overtime
crosstab = pd.crosstab(df['Attrition'], df['OverTime'])
crosstab
```

Out[31]:

OverTime	No	Yes
Attrition		
No	944	289
Yes	110	127

In [32]:

```python
# Perform a chi-square test for association between Attrition and Overtime
chi2, p_value, _, _ = stats.chi2_contingency(crosstab)

print("Chi-Square Test Results:")
print("Chi-Square:", chi2)
print("p-value:", p_value)

#reject H0?
print('\nReject H0?: ')
if p_value < alpha:
    print('True')
else:
    print('False')
```

```
Chi-Square Test Results:
Chi-Square: 87.56429365828768
p-value: 8.15842372153832e-21

Reject H0?:
True
```

CONCLUSION:

The conclusion is that there is enough evidence to support the claim that Attrition is related to whether the employee works overtime or not