# HR ANALYTICS - EMPLOYEE ATTRITION

## 1. Loading Libraries

In [1]:
```
# Pandas and Numpy for Data Manipulation & Calculation
# Matplotlib and Seaborn for Visualization

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
pd.set_option('display.max_rows',2000)
pd.set_option('display.max_columns',500)
pd.set_option('display.width',1000)
```

## 2. Loading data

In [74]:
```
df = pd.read_csv(r"D:\Projects\HR Churn Analysis\WA_Fn-UseC_-HR-Employee-Attrition
df.head()
```

Out[74]:

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

## 2.1 Columns

In [4]:
```
df.columns
```

Out[4]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department', 'DistanceF
romHome', 'Education', 'EducationField', 'EmployeeCount', 'EmployeeNumber', 'Envir
onmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement', 'JobLevel', 'JobRol
e', 'JobSatisfaction', 'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompan
iesWorked', 'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating', 'Relat
ionshipSatisfaction', 'StandardHours', 'StockOptionLevel', 'TotalWorkingYears', 'T
rainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
'YearsSinceLastPromotion', 'YearsWithCurrManager'], dtype='object')
```

In [5]:
```
df.describe() # description of data
```

Out[5]:

| | Age | DailyRate | DistanceFromHome | Education | EmployeeCount | EmployeeNum |
|---|---|---|---|---|---|---|
| **count** | 1470.000000 | 1470.000000 | 1470.000000 | 1470.000000 | 1470.0 | 1470.000 |
| **mean** | 36.923810 | 802.485714 | 9.192517 | 2.912925 | 1.0 | 1024.865 |
| **std** | 9.135373 | 403.509100 | 8.106864 | 1.024165 | 0.0 | 602.024 |
| **min** | 18.000000 | 102.000000 | 1.000000 | 1.000000 | 1.0 | 1.000 |
| **25%** | 30.000000 | 465.000000 | 2.000000 | 2.000000 | 1.0 | 491.250 |
| **50%** | 36.000000 | 802.000000 | 7.000000 | 3.000000 | 1.0 | 1020.500 |
| **75%** | 43.000000 | 1157.000000 | 14.000000 | 4.000000 | 1.0 | 1555.750 |
| **max** | 60.000000 | 1499.000000 | 29.000000 | 5.000000 | 1.0 | 2068.000 |

# 2.2 Understanding data

## 2.2.1 Checking for NULL values

In [6]:

```
df.isna().sum() # Getting the number of missing values in each column
```

Out[6]:

```
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
```

### There are no missing values found, we will move to further steps.

## 2.2.2 Checking for data types

In [7]:
```
df.dtypes
```

Out[7]:
```
Age                          int64
Attrition                   object
BusinessTravel              object
DailyRate                    int64
Department                  object
DistanceFromHome             int64
Education                    int64
EducationField              object
EmployeeCount                int64
EmployeeNumber               int64
EnvironmentSatisfaction      int64
Gender                      object
HourlyRate                   int64
JobInvolvement               int64
JobLevel                     int64
JobRole                     object
JobSatisfaction              int64
MaritalStatus               object
MonthlyIncome                int64
MonthlyRate                  int64
NumCompaniesWorked           int64
Over18                      object
OverTime                    object
PercentSalaryHike            int64
PerformanceRating            int64
RelationshipSatisfaction     int64
StandardHours                int64
StockOptionLevel             int64
TotalWorkingYears            int64
TrainingTimesLastYear        int64
WorkLifeBalance              int64
YearsAtCompany               int64
YearsInCurrentRole           int64
YearsSinceLastPromotion      int64
YearsWithCurrManager         int64
dtype: object
```

# 3. Univariate Analysis

## 3.1. Age

In [8]:
```
print(df['Age'].describe())
```

```
count    1470.000000
mean       36.923810
std         9.135373
min        18.000000
25%        30.000000
50%        36.000000
75%        43.000000
max        60.000000
Name: Age, dtype: float64
```

In [9]:
```
#frequency table
tab = pd.crosstab(df.Age, columns = 'Frequency')
tab
```

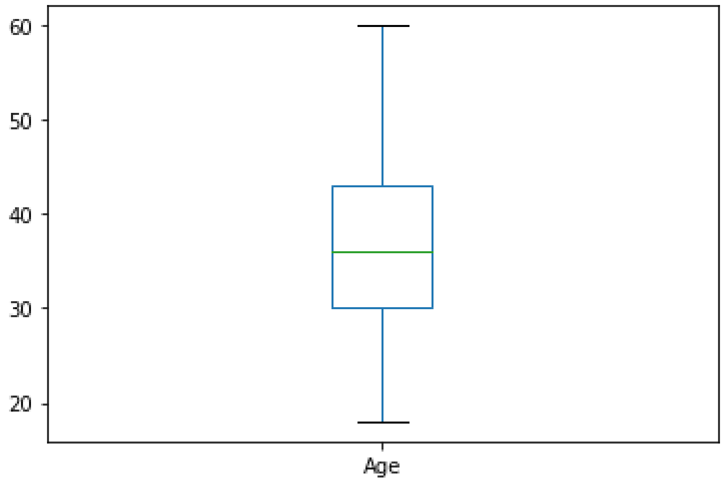
Out[9]:

| col_0 | Frequency |
|---|---|
| **Age** | |
| **18** | 8 |
| **19** | 9 |
| **20** | 11 |
| **21** | 13 |
| **22** | 16 |
| **23** | 14 |
| **24** | 26 |
| **25** | 26 |
| **26** | 39 |
| **27** | 48 |
| **28** | 48 |
| **29** | 68 |
| **30** | 60 |
| **31** | 69 |
| **32** | 61 |
| **33** | 58 |
| **34** | 77 |
| **35** | 78 |
| **36** | 69 |
| **37** | 50 |
| **38** | 58 |
| **39** | 42 |
| **40** | 57 |
| **41** | 40 |
| **42** | 46 |
| **43** | 32 |
| **44** | 33 |
| **45** | 41 |
| **46** | 33 |
| **47** | 24 |
| **48** | 19 |
| **49** | 24 |
| **50** | 30 |
| **51** | 19 |
| **52** | 18 |

| col_0 | Frequency |
|---|---|
| **Age** | |
| **53** | 19 |
| **54** | 18 |
| **55** | 22 |
| **56** | 14 |
| **57** | 4 |
| **58** | 14 |
| **59** | 10 |
| **60** | 5 |

In [10]:
```
df.boxplot(column = 'Age',
       grid = False,
       figsize = (6,4),
       )

plt.show()
```




In [11]:
```
iqr = df.Age.quantile(0.75) - df.Age.quantile(0.25)
print("Interquartile range:",iqr)
ub = df.Age.quantile(0.75) + 1.5*iqr
lb = df.Age.quantile(0.25) - 1.5*iqr

print("upper bound:",ub)
print("lower bound:",lb)
```

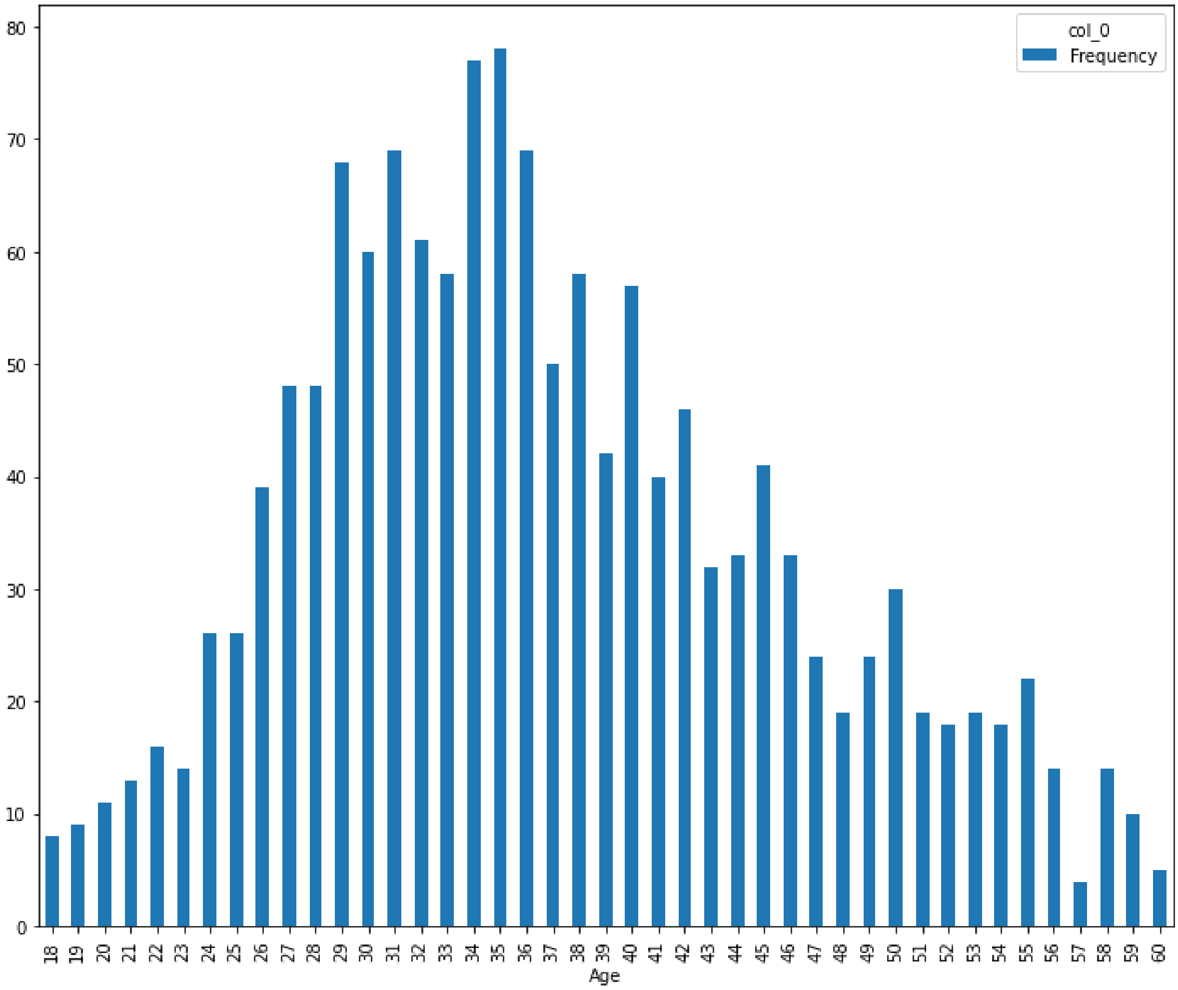
```
Interquartile range: 13.0
upper bound: 62.5
lower bound: 10.5
```

In [12]:
```
below_lb = np.sum(df['Age']<10.5)
print("No. of outliers below lower bound:",below_lb)
above_ub = np.sum(df['Age']>62.5)
print("No. of outliers above upper bound:",above_ub)
total_outliers = above_ub + below_lb
print("Total No. of outliers:",total_outliers)
```

```
No. of outliers below lower bound: 0
No. of outliers above upper bound: 0
Total No. of outliers: 0
```

In [13]:
```
tab.plot(kind='bar', figsize=(12,10))
```

Out[13]:
```
<AxesSubplot:xlabel='Age'>
```




### Observations

- Maximum frequency of people are from age 35 and minimum frequency is of age 60
- Mean age of the sample is 36.92 ~ 37 years.
- The data is symmetric in nature.

## 3.2 Attrition

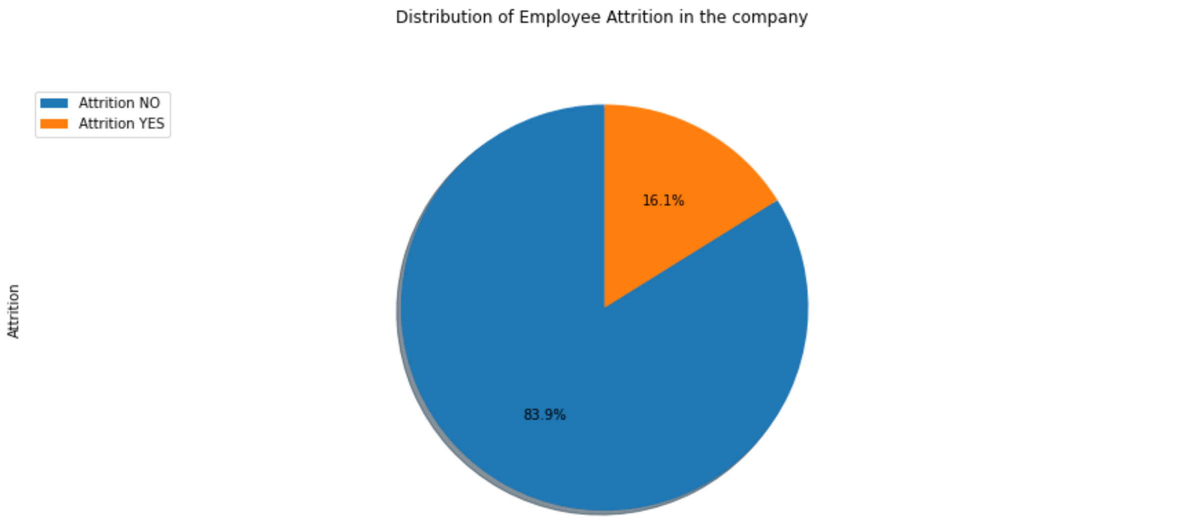
In [15]:
```
tab = pd.crosstab(df.Attrition, columns = 'Frequency')
tab
```

Out[15]:

| col_0 | Frequency |
|---|---|
| **Attrition** | |
| **No** | 1233 |
| **Yes** | 237 |

In [16]:
```
labels = 'Attrition NO', 'Attrition YES'
df['Attrition'].astype(str).value_counts().plot(kind='pie',figsize = (15,6), autop
plt.title('Distribution of Employee Attrition in the company ', y=1.12)
plt.axis('equal')
```

```
plt.legend(labels=labels, loc='upper left')
plt.show()
```




### Observations

- 237 People have left the company
- Attrition rate is 16.1%

# 3.3. Business Travel

In [18]:
```
tab = pd.crosstab(df.BusinessTravel, columns = 'Frequency')
tab
```

Out[18]:

| col_0 | Frequency |
|---|---|
| **BusinessTravel** | |
| **Non-Travel** | 150 |
| **Travel_Frequently** | 277 |
| **Travel_Rarely** | 1043 |

**Hypothesis:**

- The majority of the sample consists of individuals who have limited travel experience.

**Observation**

- People doesnt travel much often and account for almost 75%

# 3.4.Deparment

In [19]:
```
tab = pd.crosstab(df.Department, columns = 'Frequency')
tab
```

Out[19]:

| col_0 | Frequency |
|---|---|
| **Department** | |
| **Human Resources** | 63 |
| **Research & Development** | 961 |
| **Sales** | 446 |

### Hyposthesis:

- The sales department has a larger headcount compared to other departments.

### Observation

- The R & D dept has more employees than any other departments which shows about the nature of the company is "Researh-Oriented".

# 3.5. Education Field

In [20]:
```
tab = pd.crosstab(df.EducationField, columns = 'Frequency')
tab
```

Out[20]:

| col_0 | Frequency |
|---|---|
| **EducationField** | |
| **Human Resources** | 27 |
| **Life Sciences** | 606 |
| **Marketing** | 159 |
| **Medical** | 464 |
| **Other** | 82 |
| **Technical Degree** | 132 |

### Hypothesis

- As the company seems to be research oriented, it should have more people from **Technical background**

### *Observation*

- Max number of people are from **Life Sciences** Background and minimum are from **HR**, which shows that the company is research focused more in Life sciences rather than in technical aspects.

# 3.6. Environment Satisfaction

In [21]:
```
tab = pd.crosstab(df.EnvironmentSatisfaction, columns = 'Frequency')
tab
```

Out[21]:

| col_0 | Frequency |
|---|---|
| **EnvironmentSatisfaction** | |
| **1** | 284 |
| **2** | 287 |
| **3** | 453 |
| **4** | 446 |

### Hypothesis:

- People are satisfied as per attrition rate!

### Observation

- More than 800 people has rated 3 or 4 out of 4 for environment satisfaction, which shows people are satisfied with the working environment.

# 3.7. Gender

In [22]:

```
tab = pd.crosstab(df.Gender, columns = 'Frequency')
tab
```

Out[22]:

| col_0 | Frequency |
|---|---|
| **Gender** | |
| **Female** | 588 |
| **Male** | 882 |

### Hypothesis:

- As the company is research oriented, the gender diversity will be unbalanced.

### Observation

- Female to male ratio is 66.66% and Female accounts for 40% of the sample.

# 3.8. Job Involvement

In [23]:

```
tab = pd.crosstab(df.JobInvolvement, columns = 'Frequency')
tab
```

Out[23]:

| col_0 | Frequency |
|---|---|
| **JobInvolvement** | |
| **1** | 83 |
| **2** | 375 |
| **3** | 868 |
| **4** | 144 |

### Hypothesis:

- Job involvement should be higher as per the company profile.

### Observation

- More than 900 people finds themselves involved in Job

# 3.9. Job Level

In [24]:
```
tab = pd.crosstab(df.JobLevel, columns = 'Frequency')
tab
```

Out[24]:

| col_0 | Frequency |
|---|---|
| **JobLevel** | |
| **1** | 543 |
| **2** | 534 |
| **3** | 218 |
| **4** | 106 |
| **5** | 69 |

### Hypothesis

- More people should be involved in research and management and top level jobs are less

### Observation

- More than 1000 People are low level and only 10% are involved in tier 4 or 5 level jobs

# 3.10. Job Satisfacion

In [25]:
```
tab = pd.crosstab(df.JobSatisfaction, columns = 'Frequency')
tab
```

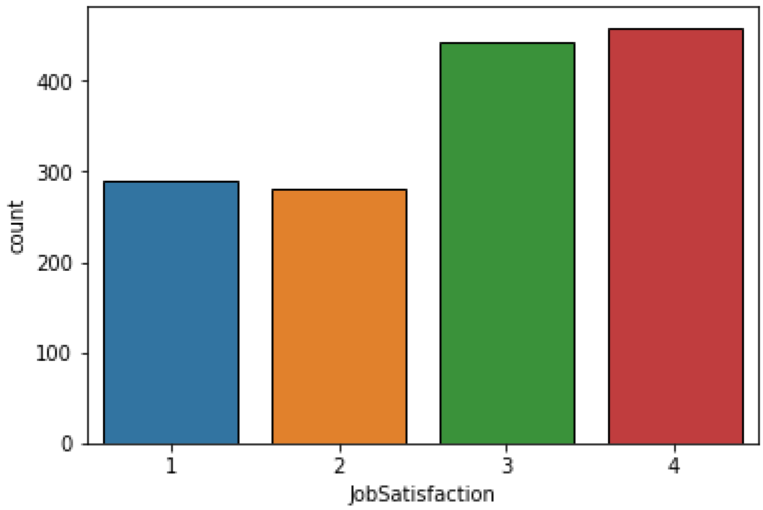
Out[25]:

| col_0 | Frequency |
|---|---|
| **JobSatisfaction** | |
| **1** | 289 |
| **2** | 280 |
| **3** | 442 |
| **4** | 459 |

In [30]:

```
sns.countplot(x='JobSatisfaction', data=df, edgecolor= "Black")
```

Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x177cac8ccf8>




### Hypothesis:

- People are mostly unsatisfied due to imbalance in work life as its a research oriented company.

### Observation

- Only 289 people are unsatisfied with the job and has rated 1 in Job Satisfaction

# 3.11. Marital Status

In [26]:

```
tab = pd.crosstab(df.MaritalStatus, columns = 'Frequency')
tab
```

Out[26]:

| col_0 | Frequency |
|---|---|
| **MaritalStatus** | |
| **Divorced** | 327 |
| **Married** | 673 |
| **Single** | 470 |

### Hypothesis:

- Most of the people are in R&D hence there should be maximum number of married couples.

**Observation**

- **Married** people account for the maximum of the sample followed by Single and Divorced respectively

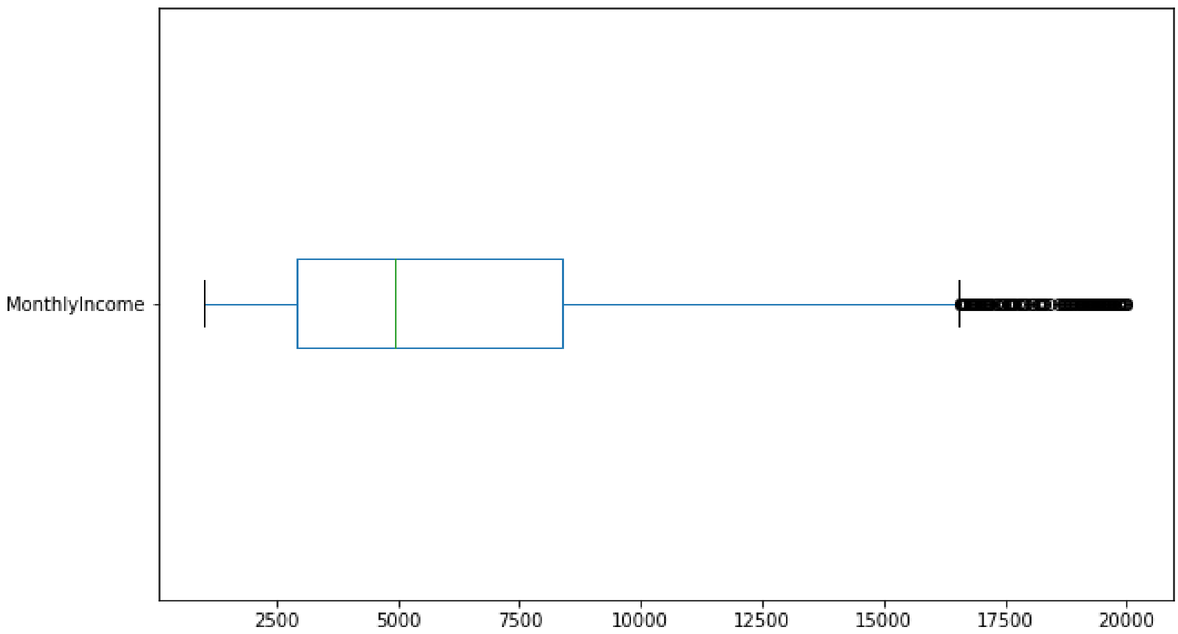
# 3.12. Monthly Income

In [27]:
```
df['MonthlyIncome'].describe()
```

Out[27]:
```
count     1470.000000
mean      6502.931293
std       4707.956783
min       1009.000000
25%       2911.000000
50%       4919.000000
75%       8379.000000
max      19999.000000
Name: MonthlyIncome, dtype: float64
```

In [28]:
```
import matplotlib as plt
df.boxplot(column = 'MonthlyIncome',
            grid = False,
            figsize = (10,6),
            vert = False)
```

Out[28]: <AxesSubplot:>




In [29]:
```
iqr = df.MonthlyIncome.quantile(0.75) - df.MonthlyIncome.quantile(0.25)
print("Interquartile range:",iqr)
ub = df.MonthlyIncome.quantile(0.75) + 1.5*iqr
lb = df.MonthlyIncome.quantile(0.25) - 1.5*iqr
print("upper bound:",ub)
print("lower bound:",lb)
```

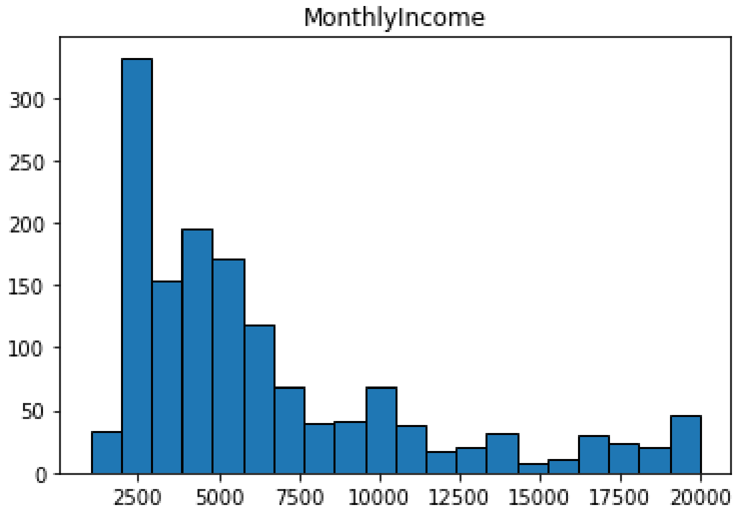
```
Interquartile range: 5468.0
upper bound: 16581.0
lower bound: -5291.0
```

In [30]:
```
below_lb = np.sum(df['MonthlyIncome']<-5291)
print("No. of outliers below lower bound:",below_lb)
upper_ub = np.sum(df['MonthlyIncome']>16581)
print("No. of outliers above upper bound:",upper_ub)
total_outliers = upper_ub + below_lb
print("Total No. of outliers:",total_outliers)
```

```
No. of outliers below lower bound: 0
No. of outliers above upper bound: 114
Total No. of outliers: 114
```

In [31]:
```
df.hist(column = "MonthlyIncome",
         grid=False,
         figsize = (6,4), bins =20,edgecolor = 'black')
```

Out[31]:
```
array([[<AxesSubplot:title={'center':'MonthlyIncome'}>]], dtype=object)
```




**Hypothesis:**

- As the company is research oriented, people are well paid and the mean would lie somewhere around the middle or the data should be left skewed.

**Observation**

- The Monthly income data is right skewed which tells us that mean of the dataset is at the left meaning less income group has more frequency than higher income group.
- The **mean (6502.93)** and **median (4919)** has a huge gap of lumpsum 1500
- Most of the sample **fall below upper quartile**

# 3.13. Overtime

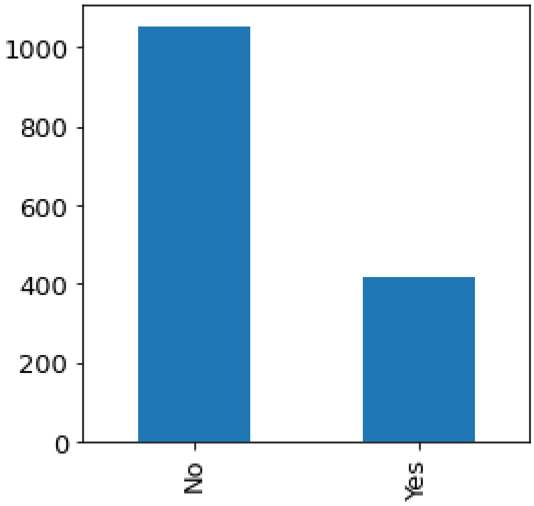
In [32]:
```
tab = pd.crosstab(df.OverTime, columns = 'Frequency')
tab
```

Out[32]:

| col_0 | Frequency |
|---|---|
| **OverTime** | |
| **No** | 1054 |
| **Yes** | 416 |

In [33]:
```
df['OverTime'].value_counts().plot(kind = 'bar', figsize = (4,4),
                                   fontsize =13)
```

Out[33]:
```
<AxesSubplot:>
```




**Hypothesis:**

- Looking at the Attrition rate, overtime should be low.

**Observation**

- More than quarter which is **28.29% of the sample works overtime**.

# 3.14. Total working years

In [34]:
```
df['TotalWorkingYears'].describe()
```

Out[34]:
```
count    1470.000000
mean       11.279592
std         7.780782
min         0.000000
25%         6.000000
50%        10.000000
75%        15.000000
max        40.000000
Name: TotalWorkingYears, dtype: float64
```

In [35]:
```
tab = pd.crosstab(df.TotalWorkingYears, columns='Frequency')
tab
```

Out[35]:

| col_0 | Frequency |
|---|---|
| **TotalWorkingYears** | |
| **0** | 11 |
| **1** | 81 |
| **2** | 31 |
| **3** | 42 |
| **4** | 63 |
| **5** | 88 |
| **6** | 125 |
| **7** | 81 |
| **8** | 103 |
| **9** | 96 |
| **10** | 202 |
| **11** | 36 |
| **12** | 48 |
| **13** | 36 |
| **14** | 31 |
| **15** | 40 |
| **16** | 37 |
| **17** | 33 |
| **18** | 27 |
| **19** | 22 |
| **20** | 30 |
| **21** | 34 |
| **22** | 21 |
| **23** | 22 |
| **24** | 18 |
| **25** | 14 |
| **26** | 14 |
| **27** | 7 |
| **28** | 14 |
| **29** | 10 |
| **30** | 7 |
| **31** | 9 |
| **32** | 9 |
| **33** | 7 |
| **34** | 5 |

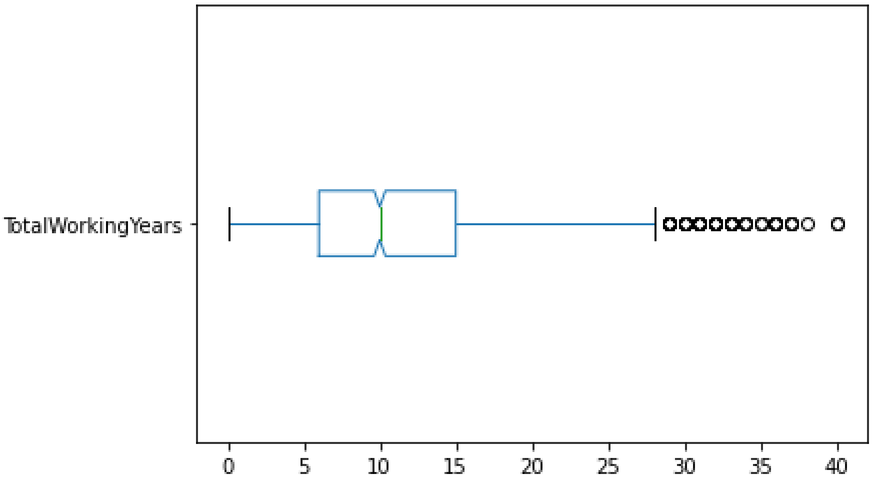
| col_0 | Frequency |
|---|---|
| **TotalWorkingYears** | |
| **35** | 3 |
| **36** | 6 |
| **37** | 4 |
| **38** | 1 |
| **40** | 2 |

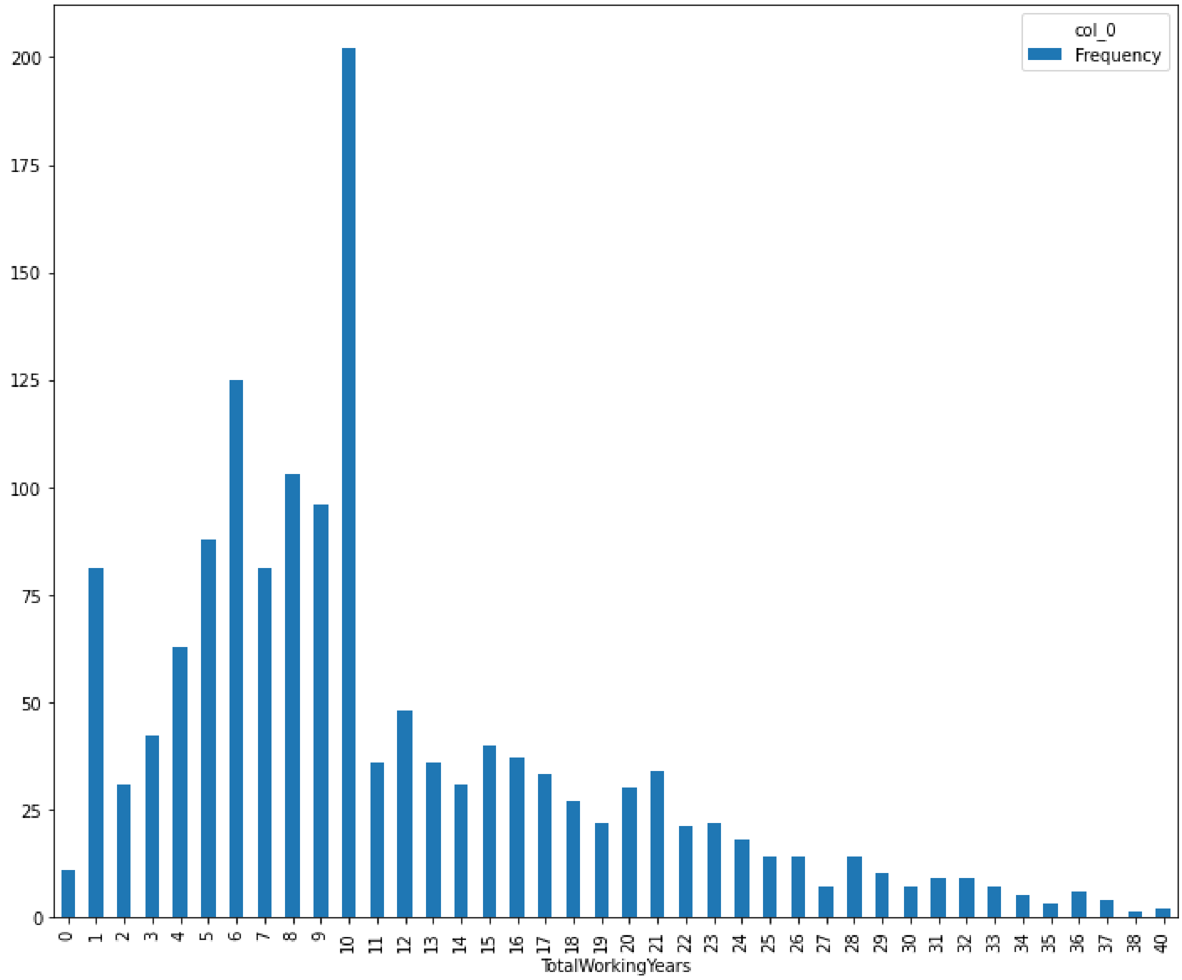
In [36]:

```
df.boxplot(column= 'TotalWorkingYears',
          grid = False,
          vert = False,
          figsize = (6,4),
          notch = True)

tab.plot(kind='bar',
        figsize = (12,10),
       grid = False)
```

Out[36]: <AxesSubplot:xlabel='TotalWorkingYears'>






In [37]:
```
iqr = df.TotalWorkingYears.quantile(0.75) - df.TotalWorkingYears.quantile(0.25)
print("IQR:",iqr)
ub = df.TotalWorkingYears.quantile(0.75) + 1.5*iqr
lb = df.TotalWorkingYears.quantile(0.25) - 1.5*iqr
print("upper bound:",ub)
print("lower bound:",lb)
```

```
IQR: 9.0
upper bound: 28.5
lower bound: -7.5
```

In [38]:
```
below_lb = np.sum(df['TotalWorkingYears']<lb)
print("No. of outliers below lower bound:",below_lb)
upper_ub = np.sum(df['TotalWorkingYears']>ub)
print("No. of outliers above upper bound:",upper_ub)
total_outliers = below_lb + upper_ub
print("Total no. of outliers:",total_outliers)
```

```
No. of outliers below lower bound: 0
No. of outliers above upper bound: 63
Total no. of outliers: 63
```

**Observation**

- Total Working years is rightly skewed meaning, most of the samples has less experience.
- More than **50% of samples are below median** of the data.

# 3.15. Work Life Balance

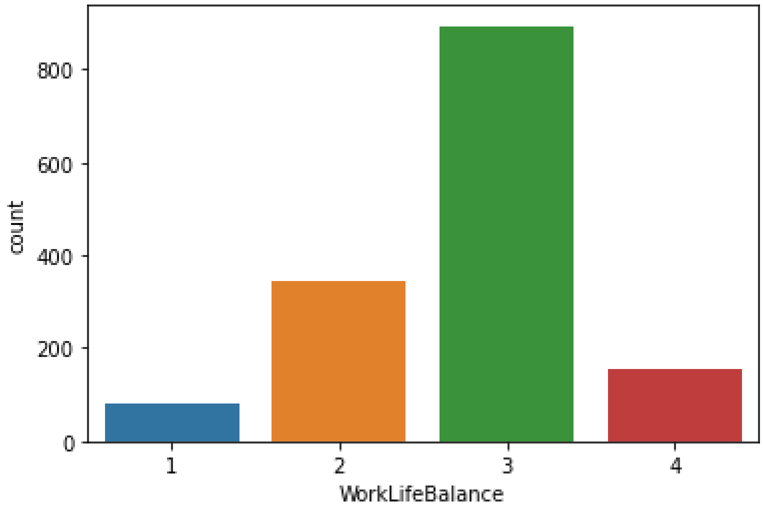
In [39]:
```
tab = pd.crosstab(df.WorkLifeBalance, columns = 'Frequency')
tab
```

Out[39]:

| col_0 | Frequency |
|---|---|
| **WorkLifeBalance** | |
| **1** | 80 |
| **2** | 344 |
| **3** | 893 |
| **4** | 153 |

In [40]:
```
sns.countplot(x = 'WorkLifeBalance', data=df)
```

Out[40]:
```
<AxesSubplot:xlabel='WorkLifeBalance', ylabel='count'>
```




**Hypothesis:**

- As there is minimal number of people doing overtime, Work life balance should be good.

**Observation**

- Only 80 samples has given rating as 1 and maximum has given 3 as a rating.

# 3.16. Years at Company

In [41]:
```
df['YearsAtCompany'].describe()
```

Out[41]:
```
count    1470.000000
mean        7.008163
std         6.126525
min         0.000000
25%         3.000000
50%         5.000000
75%         9.000000
max        40.000000
Name: YearsAtCompany, dtype: float64
```

In [42]:
```
tab = pd.crosstab(df.YearsAtCompany, columns='Frequency')
tab
```

Out[42]:

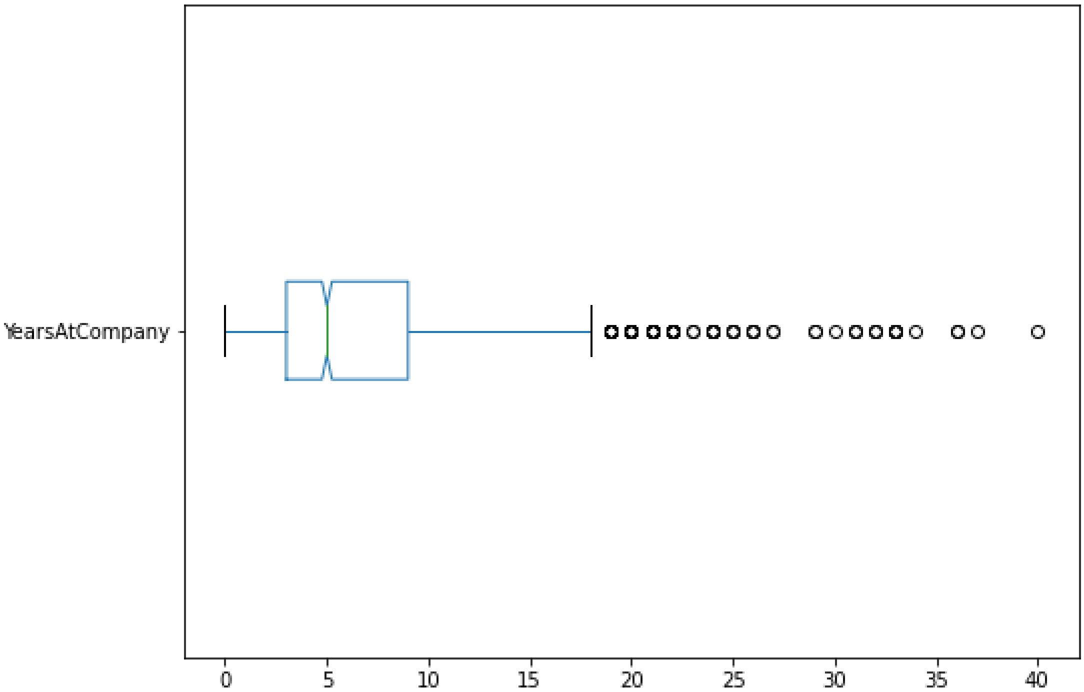
| col_0 | Frequency |
|---|---|
| **YearsAtCompany** | |
| **0** | 44 |
| **1** | 171 |
| **2** | 127 |
| **3** | 128 |
| **4** | 110 |
| **5** | 196 |
| **6** | 76 |
| **7** | 90 |
| **8** | 80 |
| **9** | 82 |
| **10** | 120 |
| **11** | 32 |
| **12** | 14 |
| **13** | 24 |
| **14** | 18 |
| **15** | 20 |
| **16** | 12 |
| **17** | 9 |
| **18** | 13 |
| **19** | 11 |
| **20** | 27 |
| **21** | 14 |
| **22** | 15 |
| **23** | 2 |
| **24** | 6 |
| **25** | 4 |
| **26** | 4 |
| **27** | 2 |
| **29** | 2 |
| **30** | 1 |
| **31** | 3 |
| **32** | 3 |
| **33** | 5 |
| **34** | 1 |
| **36** | 2 |

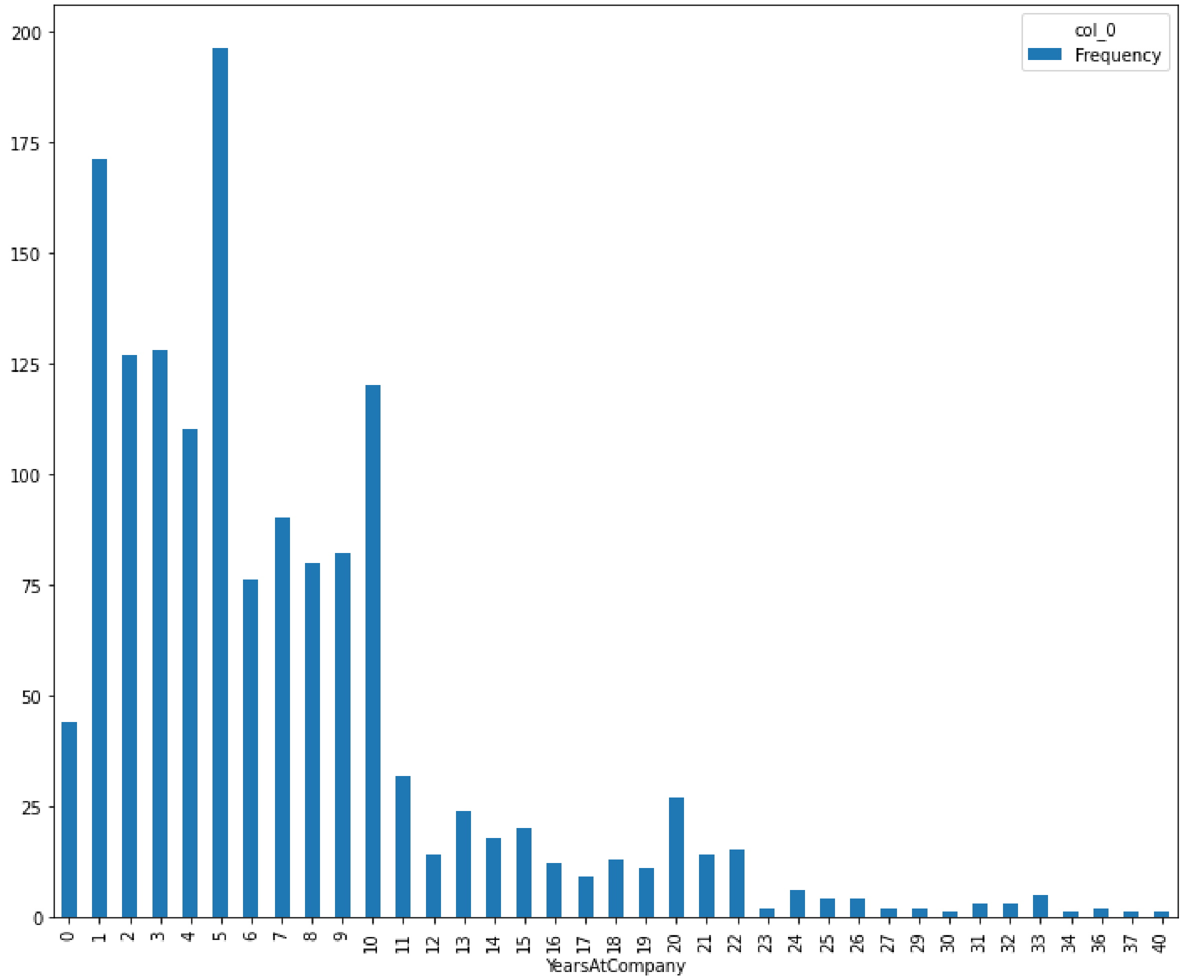
| col_0 | Frequency |
|---|---|
| **YearsAtCompany** | |
| **37** | 1 |
| **40** | 1 |

In [43]:
```
df.boxplot(column = 'YearsAtCompany',
           grid = False,
          vert = False,
          figsize = (8,6),
          notch = True)


tab.plot(kind='bar',
        figsize = (12,10),
       grid = False)
```

Out[43]:
```
<AxesSubplot:xlabel='YearsAtCompany'>
```






In [44]:
```python
iqr = df.YearsAtCompany.quantile(0.75) - df.YearsAtCompany.quantile(0.25)
print("IQR:",iqr)
ub = df.YearsAtCompany.quantile(0.75) + 1.5*iqr
lb = df.YearsAtCompany.quantile(0.25) - 1.5*iqr
print("upper bound:",ub)
print("lower bound:",lb)
```

```
IQR: 6.0
upper bound: 18.0
lower bound: -6.0
```

In [45]:
```python
below_lb = np.sum(df['YearsAtCompany']<lb)
print("No. of outliers below lower bound:",below_lb)
above_ub = np.sum(df['YearsAtCompany']>ub)
print("No. of outliers above upper bound:",above_ub)
total_outliers = above_ub + below_lb
print("Total no. of outliers:",total_outliers)
```

```
No. of outliers below lower bound: 0
No. of outliers above upper bound: 104
Total no. of outliers: 104
```

### Hypothesis

- Small number of people should be there after the mean.

**Observation**

- **Mean of the sample is 7 years** but there are very few people who has served the company for 40 years.
- The sample is right skewed.

# 3.17. Years in current role

In [46]:
```
df['YearsInCurrentRole'].describe()
```

Out[46]:
```
count    1470.000000
mean        4.229252
std         3.623137
min         0.000000
25%         2.000000
50%         3.000000
75%         7.000000
max        18.000000
Name: YearsInCurrentRole, dtype: float64
```

In [47]:
```
tab = pd.crosstab(df.YearsInCurrentRole, columns='Frequency')
tab
```

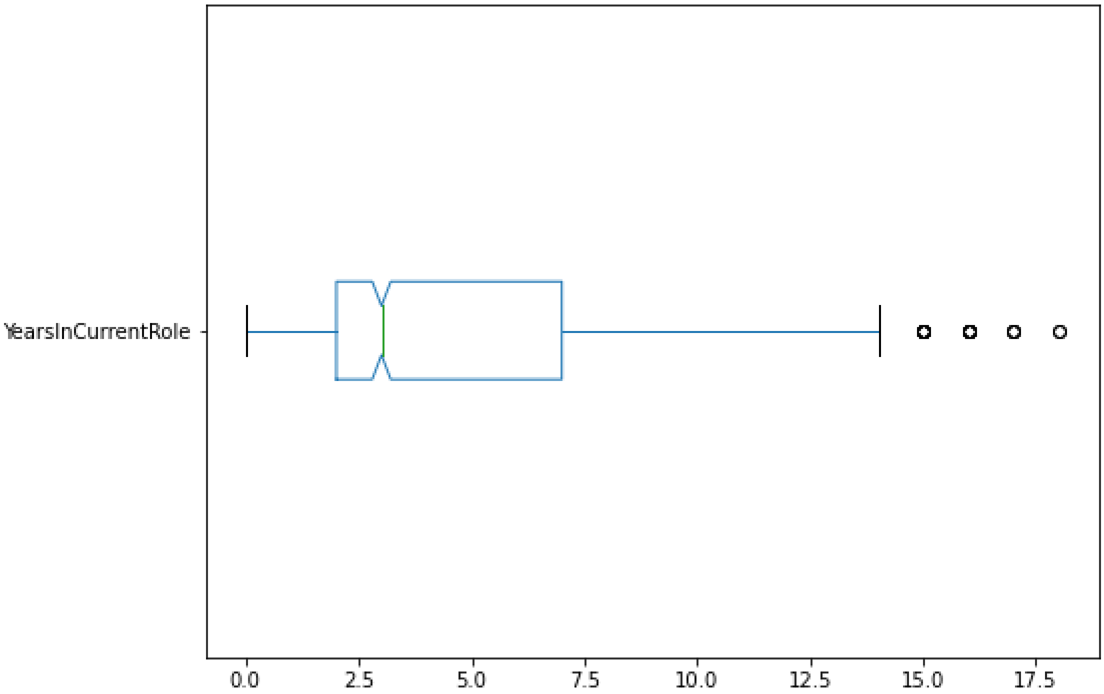
Out[47]:

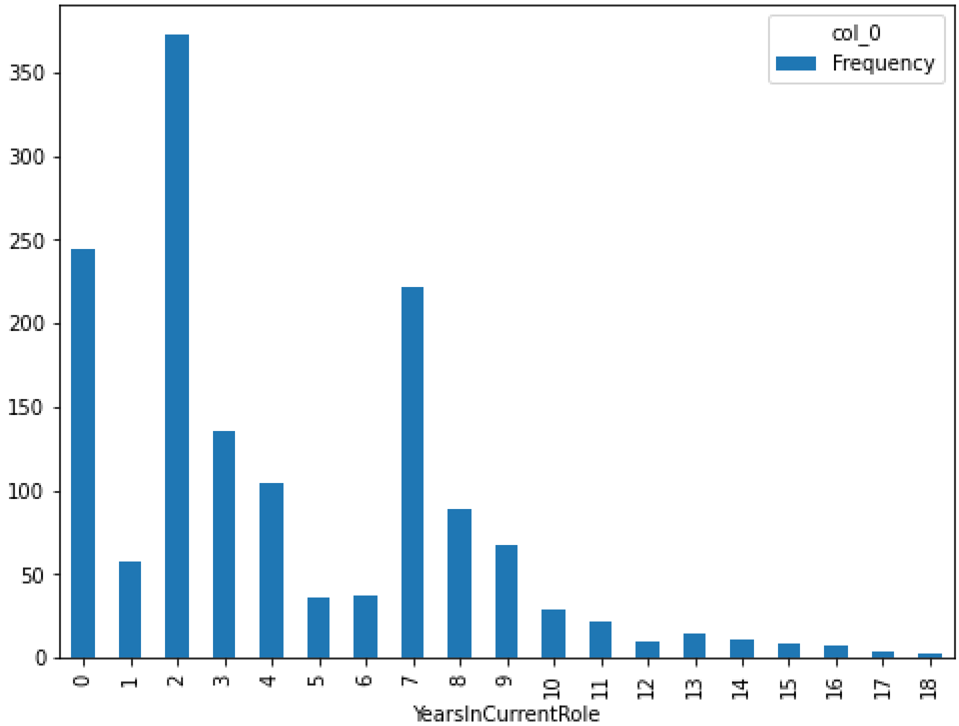
| col_0 | Frequency |
|---|---|
| **YearsInCurrentRole** | |
| **0** | 244 |
| **1** | 57 |
| **2** | 372 |
| **3** | 135 |
| **4** | 104 |
| **5** | 36 |
| **6** | 37 |
| **7** | 222 |
| **8** | 89 |
| **9** | 67 |
| **10** | 29 |
| **11** | 22 |
| **12** | 10 |
| **13** | 14 |
| **14** | 11 |
| **15** | 8 |
| **16** | 7 |
| **17** | 4 |
| **18** | 2 |

In [48]:
```
df.boxplot(column = 'YearsInCurrentRole',
           grid = False,
          vert = False,
          figsize = (8,6),
          notch = True)


tab.plot(kind='bar',
        figsize = (8,6),
       grid = False)
```

Out[48]: <AxesSubplot:xlabel='YearsInCurrentRole'>




In [49]:
```
iqr = df.YearsInCurrentRole.quantile(0.75) - df.YearsInCurrentRole.quantile(0.25)
print("IQR:",iqr)
ub = df.YearsInCurrentRole.quantile(0.75) + 1.5*iqr
lb = df.YearsInCurrentRole.quantile(0.25) - 1.5*iqr
```

```
print("upper bound:",ub)
print("lower bound:",lb)
```

```
IQR: 5.0
upper bound: 14.5
lower bound: -5.5
```

In [50]:
```
below_lb = np.sum(df['YearsInCurrentRole']<lb)
print("No. of outliers below lower bound:",below_lb)
above_ub = np.sum(df['YearsInCurrentRole']>ub)
print("No. of outliers above upper bound:",above_ub)
total_outliers = above_ub + below_lb
print("Total no. of outliers:",total_outliers)
```

```
No. of outliers below lower bound: 0
No. of outliers above upper bound: 21
Total no. of outliers: 21
```

### Hypothesis

- As the company is onvolved in research, there must be less internal job mobility.

### Observation

- 50% of the data has 3 years of experience in the current role.
- The sample is right skewed.

# 3.18. Years Since last proportion

In [51]:
```
df['YearsSinceLastPromotion'].describe()
```

Out[51]:
```
count    1470.000000
mean        2.187755
std         3.222430
min         0.000000
25%         0.000000
50%         1.000000
75%         3.000000
max        15.000000
Name: YearsSinceLastPromotion, dtype: float64
```

In [52]:
```
tab = pd.crosstab(df.YearsSinceLastPromotion, columns='Frequency')
tab
```

Out[52]:

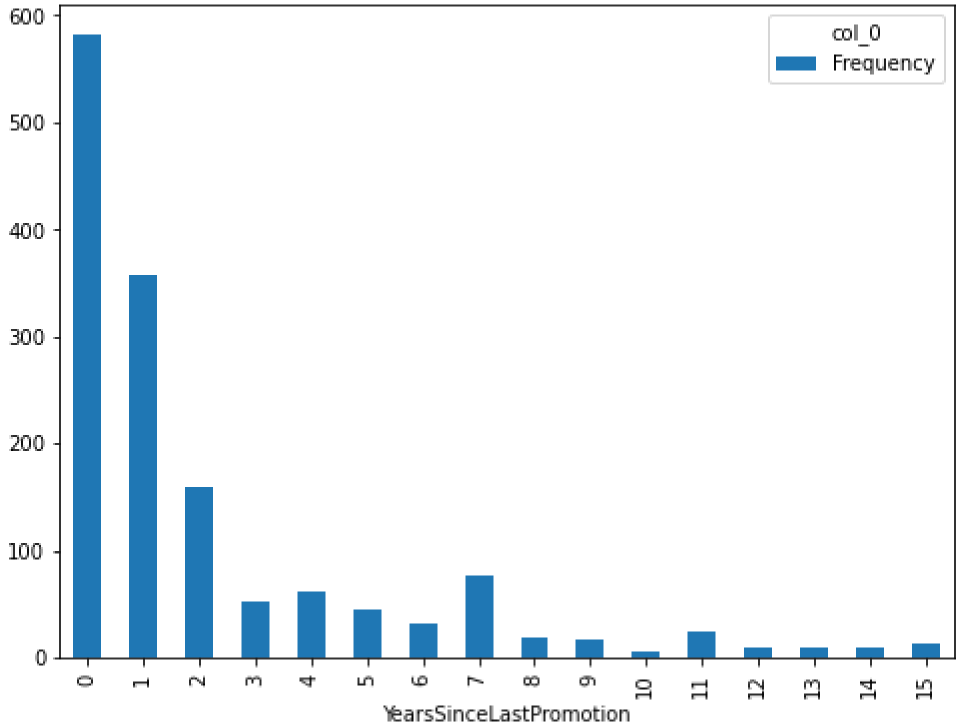
| col_0 | Frequency |
|---|---|
| **YearsSinceLastPromotion** | |
| **0** | 581 |
| **1** | 357 |
| **2** | 159 |
| **3** | 52 |
| **4** | 61 |
| **5** | 45 |
| **6** | 32 |
| **7** | 76 |
| **8** | 18 |
| **9** | 17 |
| **10** | 6 |
| **11** | 24 |
| **12** | 10 |
| **13** | 10 |
| **14** | 9 |
| **15** | 13 |

In [53]:

```
tab.plot(kind='bar', figsize=(8,6))
```

Out[53]:

```
<AxesSubplot:xlabel='YearsSinceLastPromotion'>
```



**Hypothesis:**

- Only few people due to performance problem wouldnot be able to get promotion

**Observation**

- There are almost 100 people who had no promotion since last promotion.

# 3.19. Years with current manager

In [54]:
```
df['YearsWithCurrManager'].describe()
```

Out[54]:
```
count    1470.000000
mean        4.123129
std         3.568136
min         0.000000
25%         2.000000
50%         3.000000
75%         7.000000
max        17.000000
Name: YearsWithCurrManager, dtype: float64
```

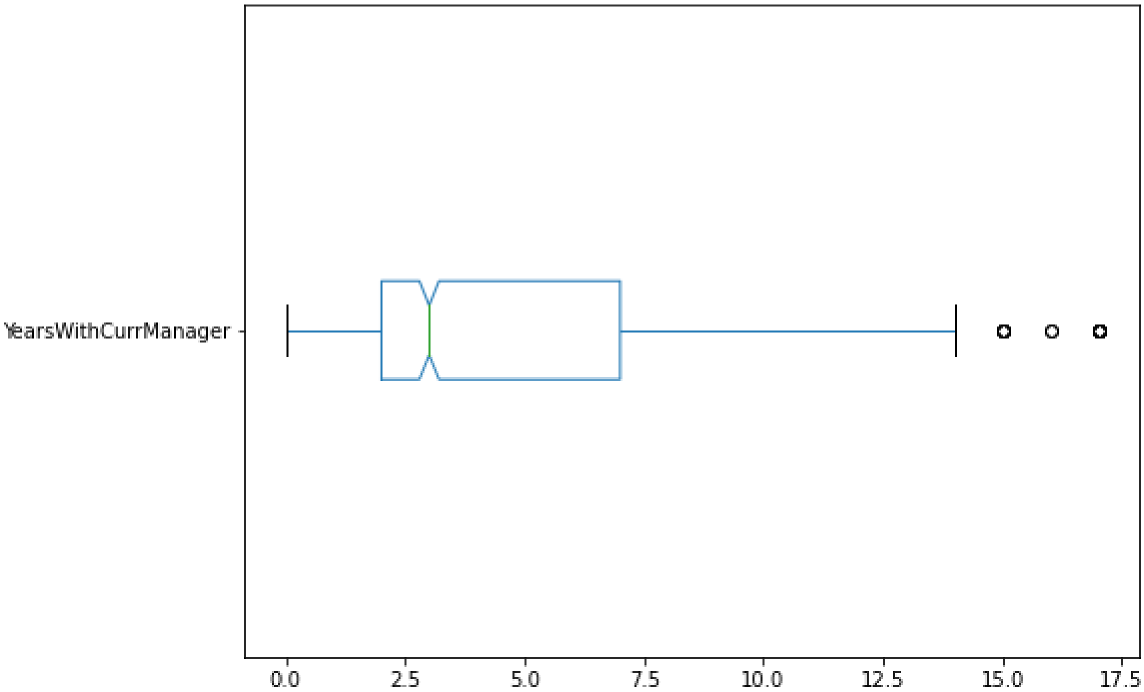
In [55]:
```
tab = pd.crosstab(df.YearsWithCurrManager, columns='Frequency')
tab
```

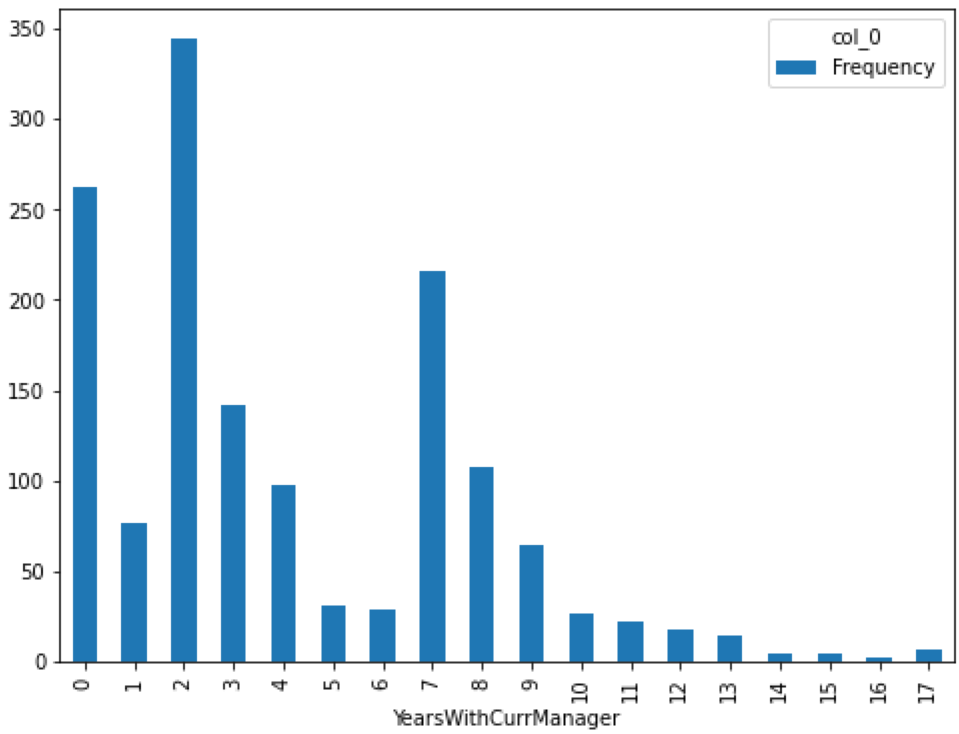
Out[55]:

| col_0 | Frequency |
|---|---|
| **YearsWithCurrManager** | |
| **0** | 263 |
| **1** | 76 |
| **2** | 344 |
| **3** | 142 |
| **4** | 98 |
| **5** | 31 |
| **6** | 29 |
| **7** | 216 |
| **8** | 107 |
| **9** | 64 |
| **10** | 27 |
| **11** | 22 |
| **12** | 18 |
| **13** | 14 |
| **14** | 5 |
| **15** | 5 |
| **16** | 2 |
| **17** | 7 |

In [56]:
```
df.boxplot(column = 'YearsWithCurrManager',
           grid = False,
           figsize = (8,6),
           vert = False,
           notch = True)

tab.plot(kind='bar', figsize=(8,6));
```




In [57]:
```
iqr = df.YearsWithCurrManager.quantile(0.75) - df.YearsWithCurrManager.quantile(0.
print("IQR:",iqr)
ub = df.YearsWithCurrManager.quantile(0.75) + 1.5*iqr
lb = df.YearsWithCurrManager.quantile(0.25) - 1.5*iqr
print("upper bound:",ub)
print("lower bound:",lb)
```

```
IQR: 5.0
upper bound: 14.5
lower bound: -5.5
```

In [58]:
```
below_lb = np.sum(df['YearsWithCurrManager']<lb)
print("No. of outliers below lower bound:",below_lb)
above_ub = np.sum(df['YearsWithCurrManager']>ub)
print("No. of outliers above upper bound:",above_ub)
total_outliers = above_ub + below_lb
print("Total no. of outliers:",total_outliers)
```

```
No. of outliers below lower bound: 0
No. of outliers above upper bound: 14
Total no. of outliers: 14
```

**Hypothesis**

- Since very few people are there who have got no promotion from very long time should be with current manager

**Observation**

- Almost near to 500 People has shown a shift to different managers afte working for 2.5 years.
- Almost 300 people has shown shift to different managers in 0 to 1 year.

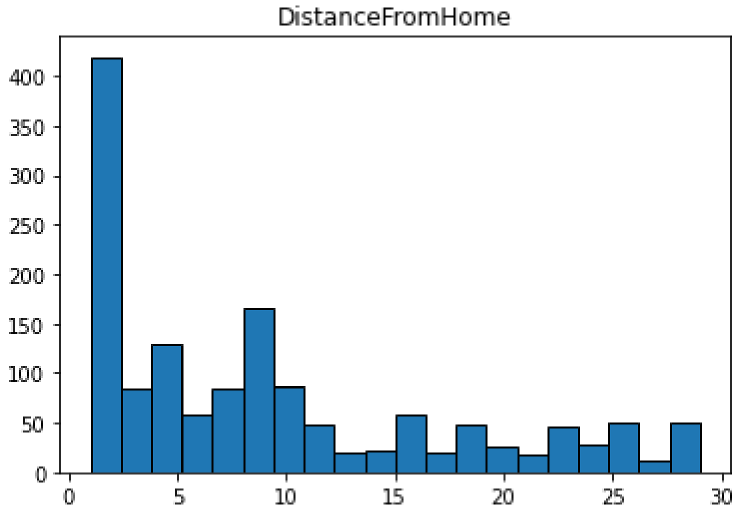
# 3.20. Distance from Home

In [59]:
```
df['DistanceFromHome'].describe()
```

Out[59]:
```
count    1470.000000
mean        9.192517
std         8.106864
min         1.000000
25%         2.000000
50%         7.000000
75%        14.000000
max        29.000000
Name: DistanceFromHome, dtype: float64
```

In [60]:
```
df.hist(column = "DistanceFromHome",
         grid=False,
         figsize = (6,4), bins =20,edgecolor = 'black')
```

Out[60]:
```
array([[<AxesSubplot:title={'center':'DistanceFromHome'}>]], dtype=object)
```




#### Hypothesis

- Most People will prefer to live near the company.

#### Observation

- 500 people live in 2 km range of the office followed by 250 people in the range of 7-8 Km range.

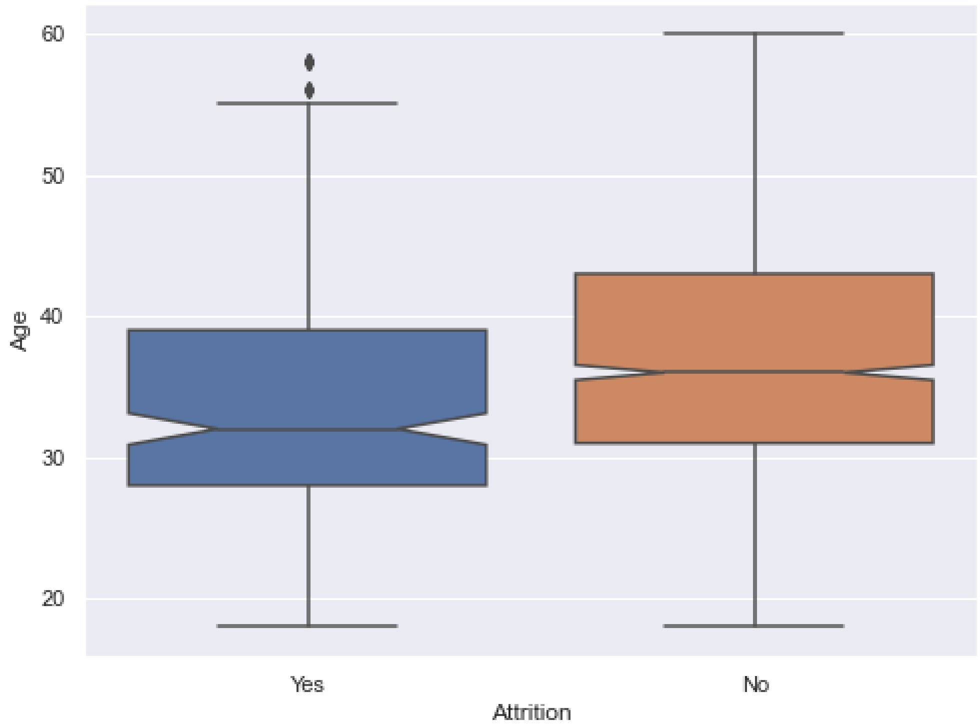
# 4. Bivariate Analysis

## 4.1. Age - Attrition

In [61]:
```
df['Age'].describe()
```

Out[61]:
```
count    1470.000000
mean       36.923810
std         9.135373
min        18.000000
25%        30.000000
50%        36.000000
75%        43.000000
max        60.000000
Name: Age, dtype: float64
```

In [75]:
```
sns.boxplot(x = "Attrition", y = "Age", data=df, notch = True);
```




In [77]:
```
a=np.quantile(df[df["Attrition"]=='Yes']["Age"],0.25)
b=np.quantile(df[df["Attrition"]=='Yes']["Age"],0.75)

iqr = b-a
print("IQR: ",iqr)

ub = b + 1.5*iqr
lb = a - 1.5*iqr
print("Upper bound: ",ub)
print("Lower bound: ",lb)
```

```
IQR:  11.0
Upper bound:  55.5
Lower bound:  11.5
```

#### Hypothesis

- People who has age less than sample mean tend to leave the company!

#### Fact

- People of all the ages are leaving the company, but people within age 28 to 38 are more prone to leave the company.

# 4.2. Business Travel - Attrition

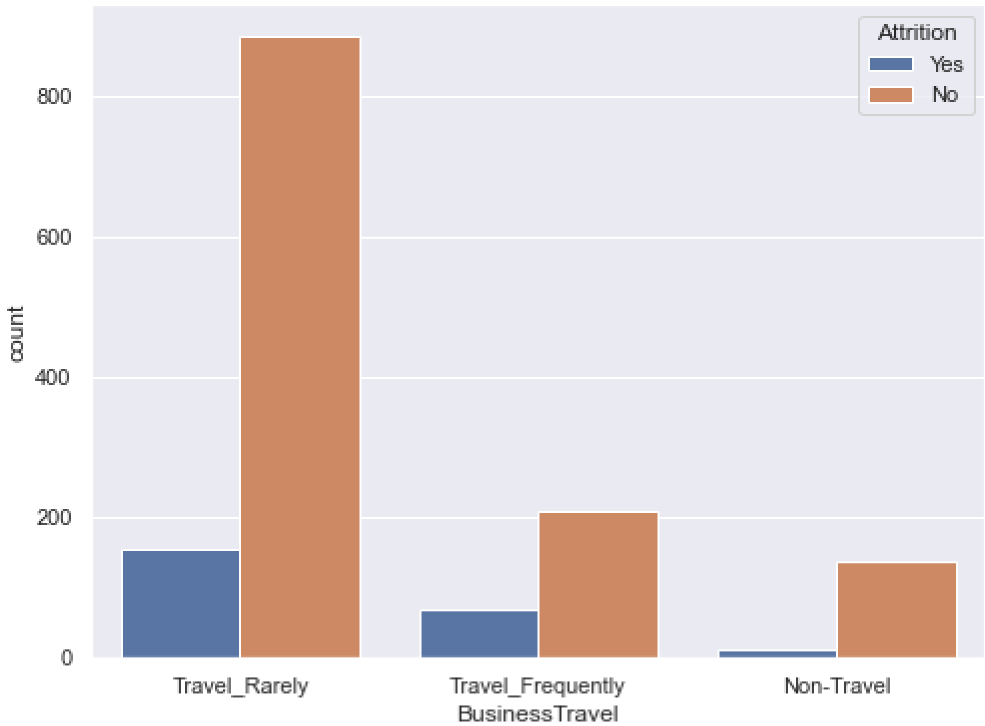
In [78]:
```
tab = pd.crosstab(df.Attrition, columns = df.BusinessTravel)
tab
```

Out[78]:

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

In [79]:
```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='BusinessTravel', data=df, hue='Attrition')
```

Out[79]: <AxesSubplot:xlabel='BusinessTravel', ylabel='count'>




In [80]:
```
# Attrition rate:
Non_travel = 12/140
Travel_frequently = 69/(208+69)
Travel_rarely = 156/(156+887)
print("Attrition rate for Non_travel: " , round(Non_travel*100, 2))
print("Attrition rate for Travel_frequently: " ,round(Travel_frequently*100, 2))
print("Attrition rate for Travel_rarely: " , round(Travel_rarely*100, 2))
```

```
Attrition rate for Non_travel:  8.57
Attrition rate for Travel_frequently:  24.91
Attrition rate for Travel_rarely:  14.96
```

**Hypothesis -** Travelling Frequently leads to unbalanced work life which should result in attrition!

**Fact -** We can observe that the attrition rate has prominent connection with Business Travel.

- Highest attrition rate i.e. 24.91% ~ 25% can be observed in the records who travel frequently.

# 4.3. Department - Attrition

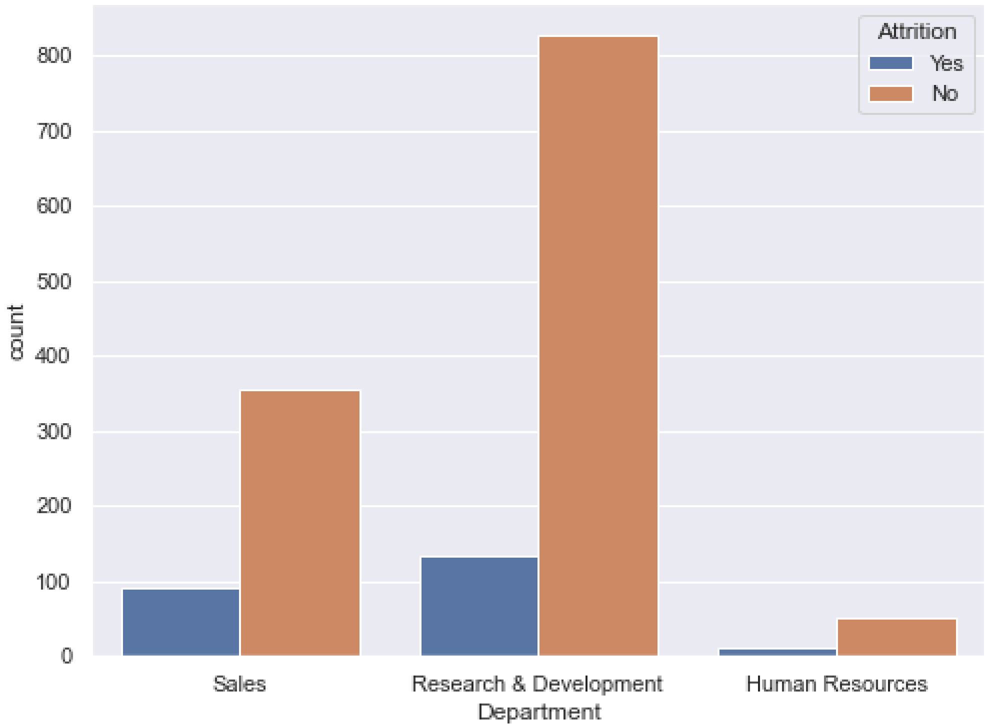
In [81]:
```
tab = pd.crosstab(df.Attrition, columns = df.Department)
tab
```

Out[81]:

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

In [82]:
```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='Department', data=df, hue='Attrition')
```

Out[82]:
```
<AxesSubplot:xlabel='Department', ylabel='count'>
```




In [83]:
```
print("Attrition rate in Human resources: ", round((12/63)*100,2))
print("Attrition rate in Research & Developement: ", round((133/961)*100,2))
print("Attrition rate in Sales: ", round((12/446)*100,2))
```

```
Attrition rate in Human resources:  19.05
Attrition rate in Research & Developement:  13.84
Attrition rate in Sales:  2.69
```

**Hypothesis -** Sales department should have more attrition than any other department due to huge work pressure!

**Fact -** We can observe that the attrition rate has prominent connection with Department.

### Highest attrition rate i.e. 19.05 which can be observed from HR Department which has the lowest count which nullify our hypothesis.

## 4.4. Environment Satisfaction - Attrition

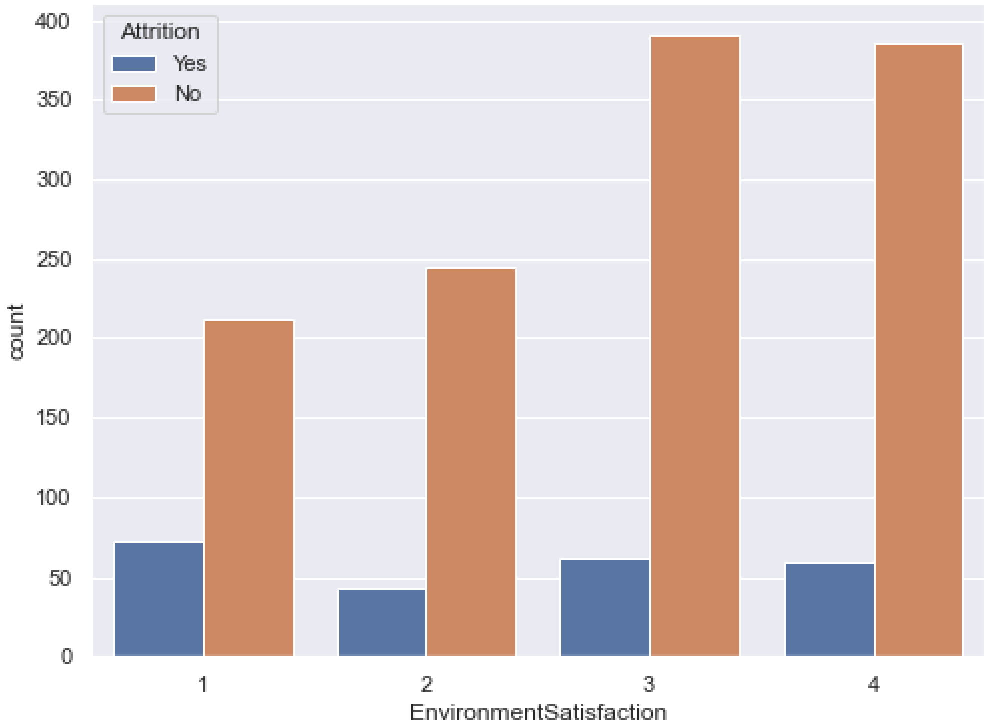
In [84]:
```
tab = pd.crosstab(df.Attrition, columns = df.EnvironmentSatisfaction)
tab
```

Out[84]:

| EnvironmentSatisfaction | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **Attrition** | | | | |
| **No** | 212 | 244 | 391 | 386 |
| **Yes** | 72 | 43 | 62 | 60 |

In [85]:
```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='EnvironmentSatisfaction', data=df, hue='Attrition')
```

Out[85]: <AxesSubplot:xlabel='EnvironmentSatisfaction', ylabel='count'>




In [86]:
```
print("Attrition rate for voting 1: ", round((72/284)*100,2))
print("Attrition rate for voting 2: ", round((43/287)*100,2))
print("Attrition rate for voting 3: ", round((62/453)*100,2))
print("Attrition rate for voting 4: ", round((60/446)*100,2))
```

```
Attrition rate for voting 1:  25.35
Attrition rate for voting 2:  14.98
Attrition rate for voting 3:  13.69
Attrition rate for voting 4:  13.45
```

**Hypothesis -** People who has voted 1 for Environment satisfaction are more prone to leaving company!

**Fact -** We can observe that the attrition rate has prominent connection with votings for Environment Satisfaction.

### Highest attrition rate i.e. **25.35%** can be observed by the people who are dissatisfied with the environment and has voted 1.

## 4.5. Gender - Attrition

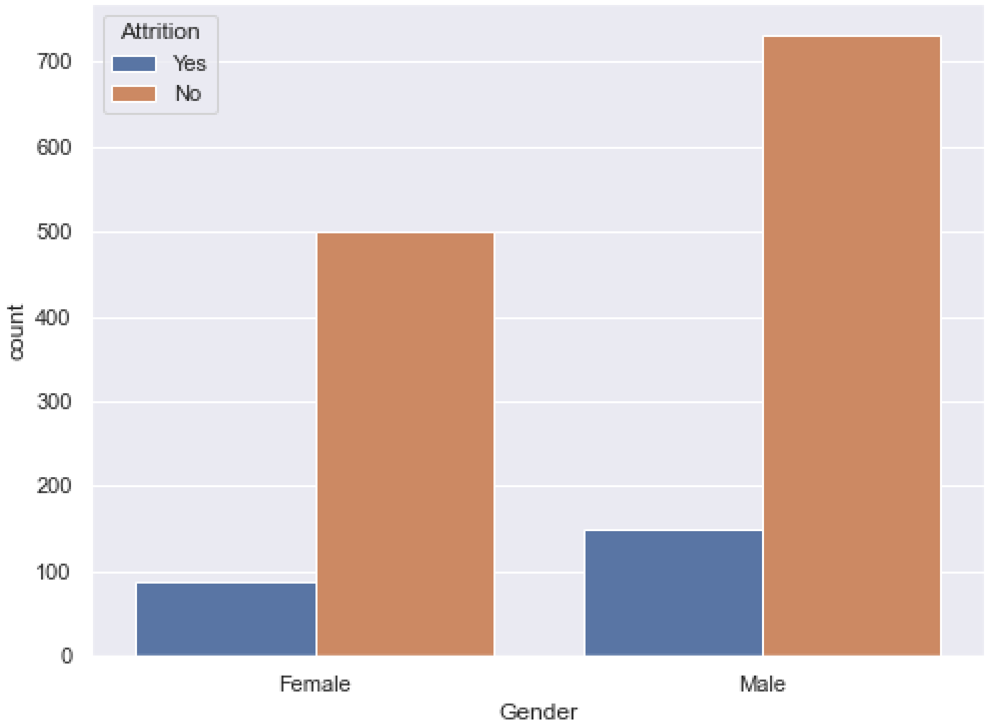
In [87]:
```
tab = pd.crosstab(df.Attrition, columns = df.Gender)
tab
```

Out[87]:

| Gender | Female | Male |
|---|---|---|
| **Attrition** | | |
| **No** | 501 | 732 |
| **Yes** | 87 | 150 |

In [88]:
```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='Gender', data=df, hue='Attrition')
```

Out[88]: <AxesSubplot:xlabel='Gender', ylabel='count'>




In [89]:
```
print("Attrition rate for Female: ", round((87/588)*100,2))
print("Attrition rate for Male: ", round((150/732)*100,2))
```

```
Attrition rate for Female:  14.8
Attrition rate for Male:  20.49
```

**Hypothesis -** Male population tends to leave the firm more frequently than Female!

**Fact -** Male attrition rate is higher than Female which is 20.49% .

# 4.6. Job Involvement - Attrition

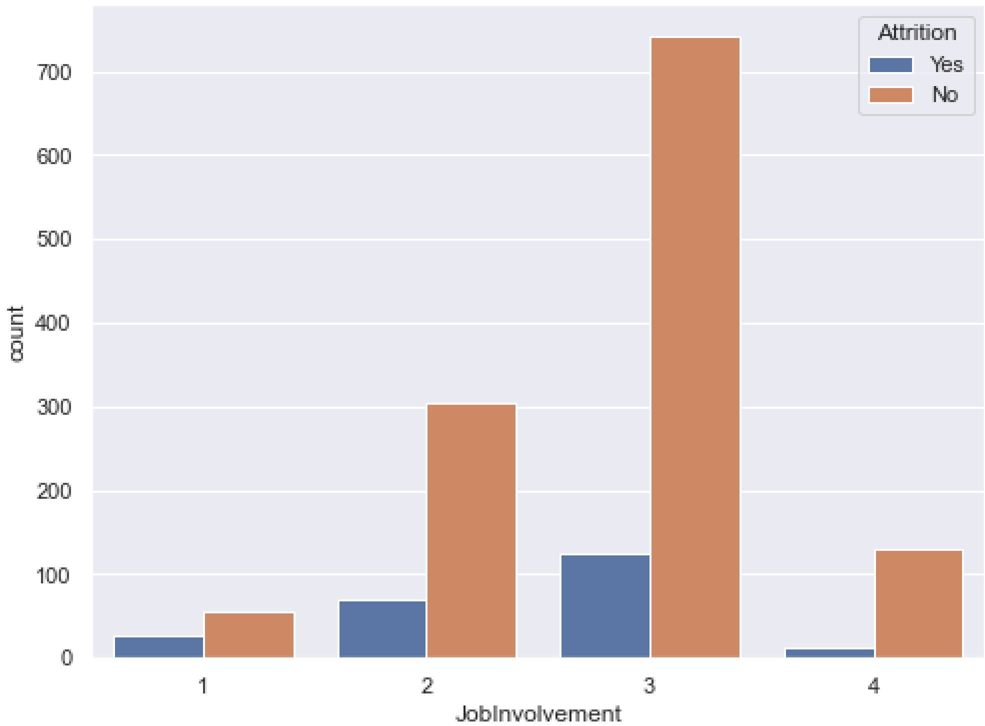
In [90]:
```
tab = pd.crosstab(df.Attrition, columns = df.JobInvolvement)
tab
```

Out[90]:

| JobInvolvement | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **Attrition** | | | | |
| **No** | 55 | 304 | 743 | 131 |
| **Yes** | 28 | 71 | 125 | 13 |

In [91]:
```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='JobInvolvement', data=df, hue='Attrition')
```

Out[91]: <AxesSubplot:xlabel='JobInvolvement', ylabel='count'>




In [92]:
```
print("Attrition rate for JI 1: ", round((28/83)*100,2))
print("Attrition rate for JI 2: ", round((71/375)*100,2))
print("Attrition rate for JI 3: ", round((125/868)*100,2))
print("Attrition rate for JI 4: ", round((13/144)*100,2))
```

```
Attrition rate for JI 1:  33.73
Attrition rate for JI 2:  18.93
Attrition rate for JI 3:  14.4
Attrition rate for JI 4:  9.03
```

**Hypothesis -** People who have voted less in the Job Involvement index are more prone to leave the company!

**Fact -** People who have Job involvement index of 1, have an attrition rate of 33.73% which supports the Hypothesis.

# 4.7. Job Level - Attrition

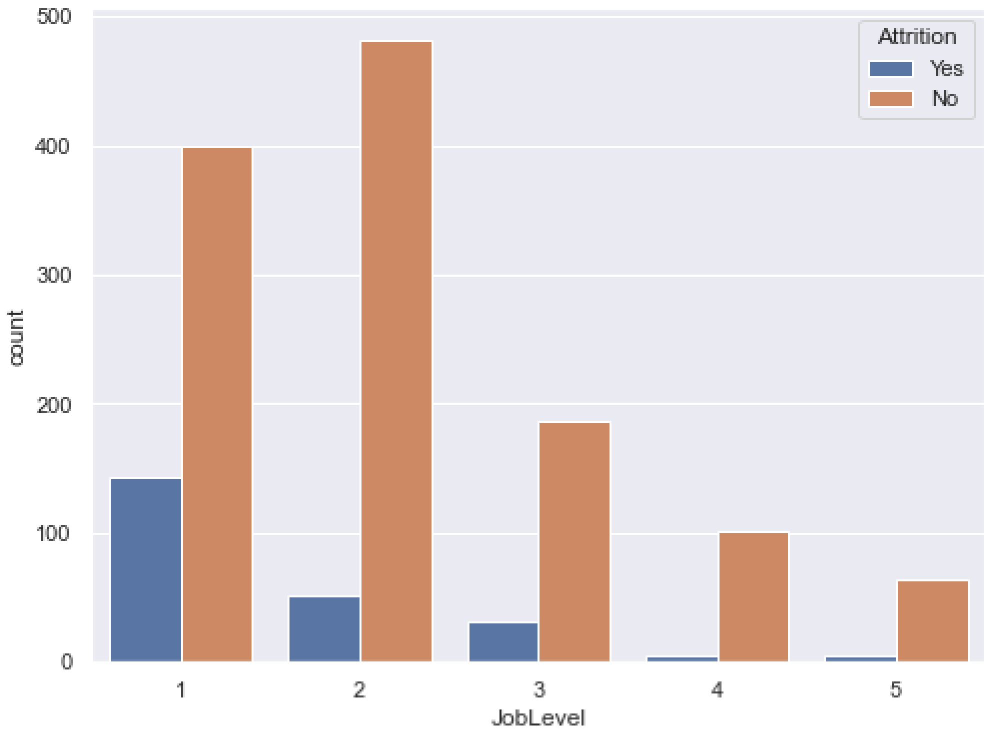
In [93]:
```
tab = pd.crosstab(df.Attrition, columns = df.JobLevel)
tab
```

Out[93]:

| JobLevel | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **Attrition** | | | | | |
| **No** | 400 | 482 | 186 | 101 | 64 |
| **Yes** | 143 | 52 | 32 | 5 | 5 |

In [94]:
```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='JobLevel', data=df, hue='Attrition')
```

Out[94]: <AxesSubplot:xlabel='JobLevel', ylabel='count'>




In [95]:
```
print("Attrition rate for JL 1: ", round((143/543)*100,2))
print("Attrition rate for JL 2: ", round((52/534)*100,2))
print("Attrition rate for JL 3: ", round((32/218)*100,2))
print("Attrition rate for JL 4: ", round((5/106)*100,2))
print("Attrition rate for JL 5: ", round((5/69)*100,2))
```

```
Attrition rate for JL 1:  26.34
Attrition rate for JL 2:  9.74
Attrition rate for JL 3:  14.68
Attrition rate for JL 4:  4.72
Attrition rate for JL 5:  7.25
```

**Hypothesis -** People who have lesser Job Level are more prone to leave the company!

**Fact -** People who are involved in Job Level 1 has the highest probability of leaving the job followed by Level 3.

### Interestingly Job Level 2 people are more stable compared to Job level 3.

# 4.8. Job Satisfaction - Attrition

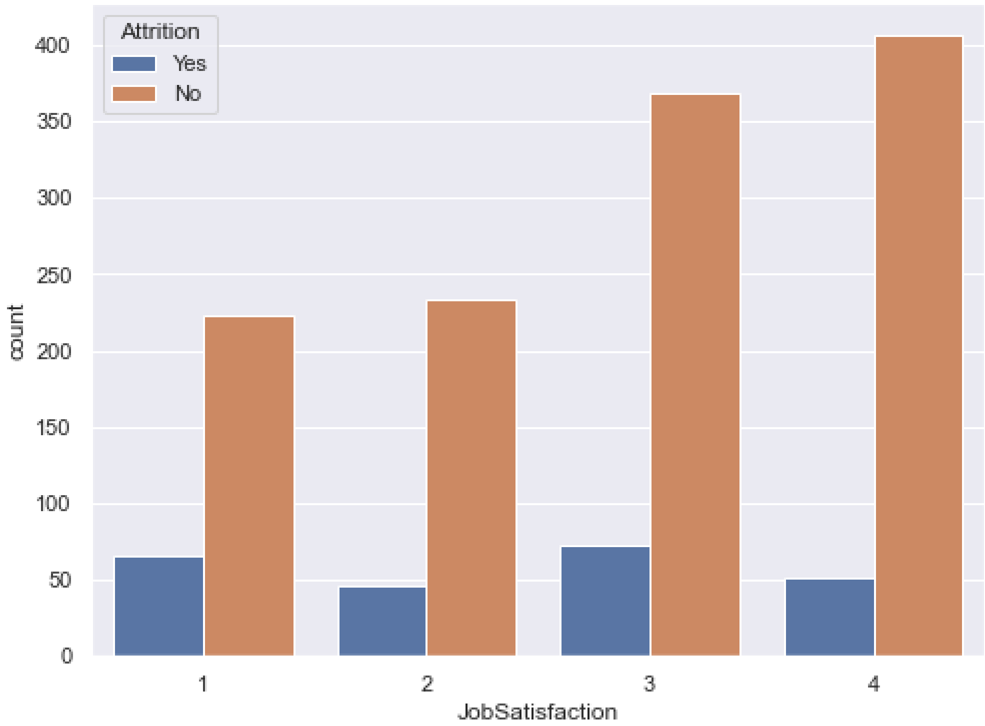
In [96]:
```
tab = pd.crosstab(df.Attrition, columns = df.JobSatisfaction)
tab
```

Out[96]:

| JobSatisfaction | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **Attrition** | | | | |
| **No** | 223 | 234 | 369 | 407 |
| **Yes** | 66 | 46 | 73 | 52 |

In [97]:
```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='JobSatisfaction', data=df, hue='Attrition')
```

Out[97]: <AxesSubplot:xlabel='JobSatisfaction', ylabel='count'>




In [98]:
```
print("Attrition rate for voting 1: ", round((66/289)*100,2))
print("Attrition rate for voting 2: ", round((46/280)*100,2))
print("Attrition rate for voting 3: ", round((73/442)*100,2))
print("Attrition rate for voting 4: ", round((52/459)*100,2))
```

```
Attrition rate for voting 1:  22.84
Attrition rate for voting 2:  16.43
Attrition rate for voting 3:  16.52
Attrition rate for voting 4:  11.33
```

**Hypothesis -** People who has voted less in the Job Satisfaction index are more prone to leave the company!

**Fact -** People who have voted 1 in Job Satisfaction has 22.84% chances of leaving the company.

- Interesting fact is that even people who has voted highest in satisfaction index has a chance of 11.33% of leaving the company.

# 4.9. Marital Status - Attrition

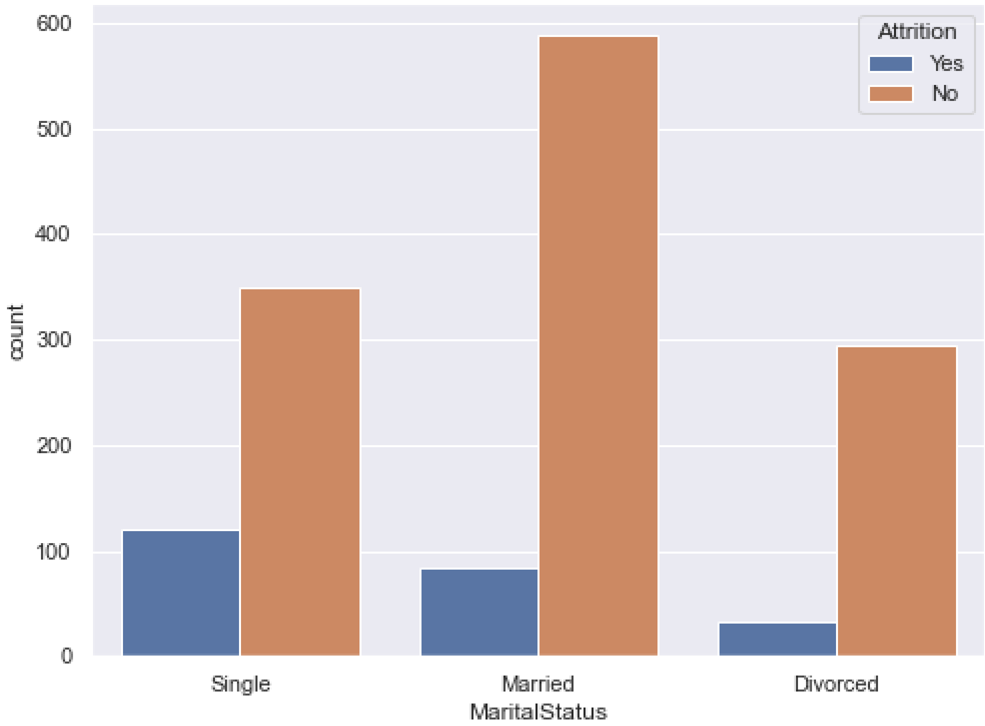
In [99]:
```
tab = pd.crosstab(df.Attrition, columns = df.MaritalStatus)
tab
```

Out[99]:

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

In [100…
```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='MaritalStatus', data=df, hue='Attrition')
```

Out[100]: <AxesSubplot:xlabel='MaritalStatus', ylabel='count'>




In [101…

```
print("Attrition rate for Divorced: ", round((33/327)*100,2))
print("Attrition rate for Married: ", round((84/673)*100,2))
print("Attrition rate for Single: ", round((120/470)*100,2))
```

```
Attrition rate for Divorced:  10.09
Attrition rate for Married:  12.48
Attrition rate for Single:  25.53
```

**Hypothesis -** People who are single has more attrition rate!

**Fact -** People who are single has more attrition.

- Interesting fact is that Married people are more prone to attrition than divorced people.

# 4.10. Monthly Income - Attrition

In [102…

```
df["MonthlyIncome"].describe()
```

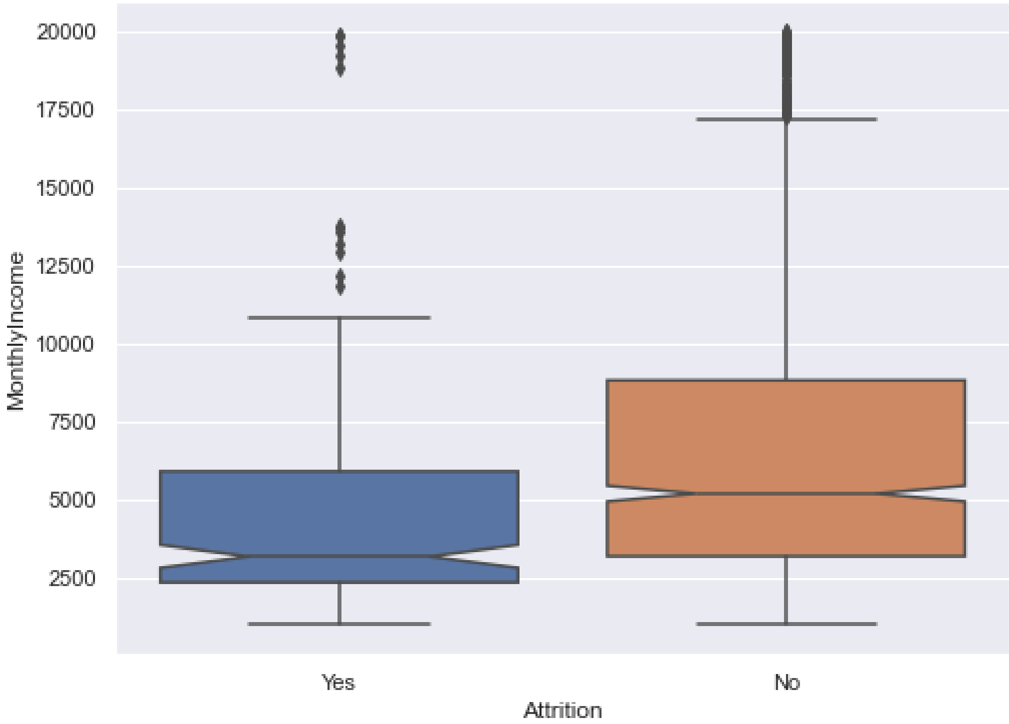
Out[102]:

```
count     1470.000000
mean      6502.931293
std       4707.956783
min       1009.000000
25%       2911.000000
50%       4919.000000
75%       8379.000000
max      19999.000000
Name: MonthlyIncome, dtype: float64
```

In [103…

```
sns.boxplot(x = "Attrition", y = "MonthlyIncome", data=df, notch = True)
```

Out[103]: <AxesSubplot:xlabel='Attrition', ylabel='MonthlyIncome'>

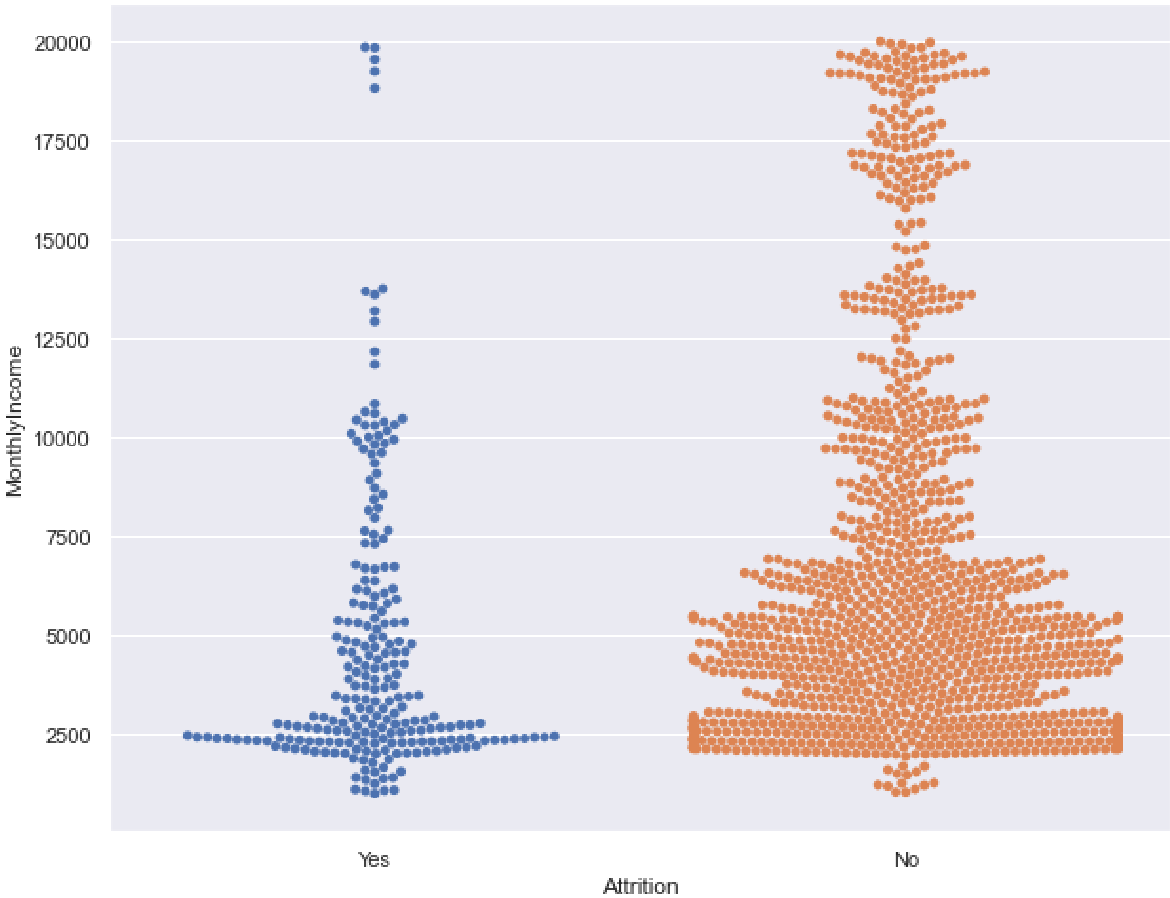



In [104…

```
sns.set(rc={'figure.figsize':(10,8)})
sns.swarmplot(x = "Attrition", y = "MonthlyIncome", data=df)
```

```
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 6.7% of the points cannot be placed; you may want to decrease the size of the m
arkers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[104]: <AxesSubplot:xlabel='Attrition', ylabel='MonthlyIncome'>



In [106…

```
a=np.quantile(df[df['Attrition']=='Yes']['MonthlyIncome'],0.25)
b=np.quantile(df[df['Attrition']=='Yes']['MonthlyIncome'],0.75)
iqr = b-a
print("IQR:",iqr)
ub = b + 1.5*iqr
lb = a - 1.5*iqr
print("Upper bound:",ub)
print("Lower bound:",lb)
```

```
IQR: 3543.0
Upper bound: 11230.5
Lower bound: -2941.5
```

**Hypothesis**

- People with low income group,must be leaving the comapany frequently.

**Inferences:**

- We can see in the above plots that the frequency of people leaving the company is more in between 1000 to 2500.

# 4.11. Overtime - Attrition

In [107…

```
tab = pd.crosstab(df.Attrition, columns= df.OverTime)
tab
```

Out[107]:

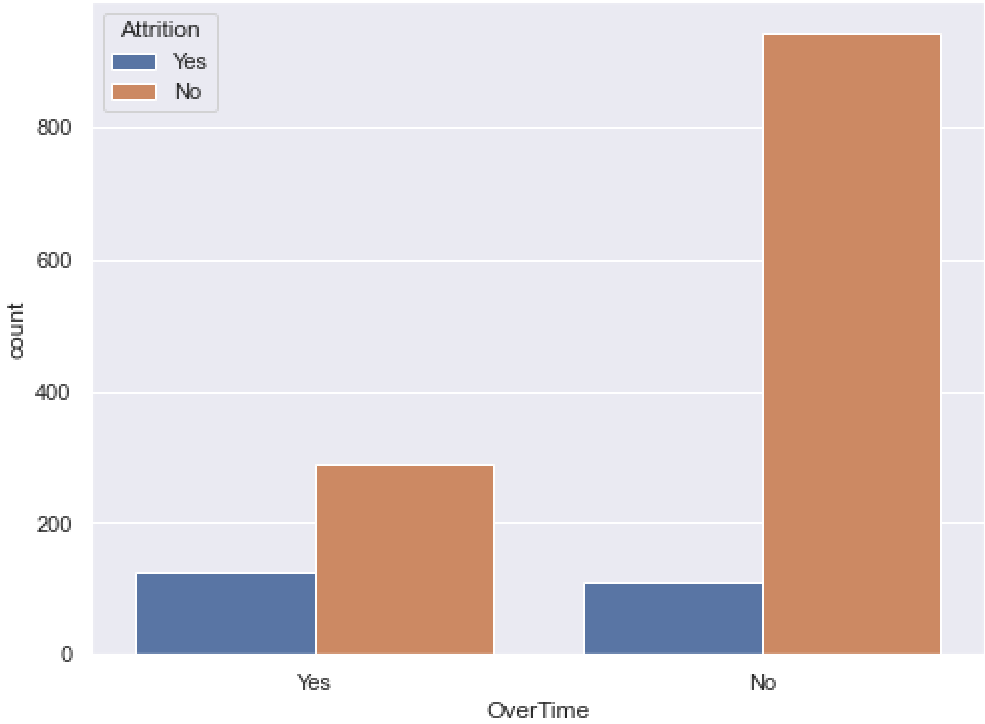
| OverTime | No | Yes |
|---|---|---|
| **Attrition** | | |
| **No** | 944 | 289 |
| **Yes** | 110 | 127 |

In [108…

```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='OverTime', data=df, hue='Attrition')
```

Out[108]:

```
<AxesSubplot:xlabel='OverTime', ylabel='count'>
```




In [109…
```
print("Attrition rate for Overtime: ", round((127/416)*100,2))
print("Attrition rate for no overtime: ", round((110/1054)*100,2))
```

```
Attrition rate for Overtime:  30.53
Attrition rate for no overtime:  10.44
```

**Hypothesis:**

- People doing overtime are leaving the comapny frequently.

**Inference:**

- People doing overtime has 30.53% probability of leaving the comapany as compared to the people who are not doing overtime and still are prone to leave the company with 10.44% probability

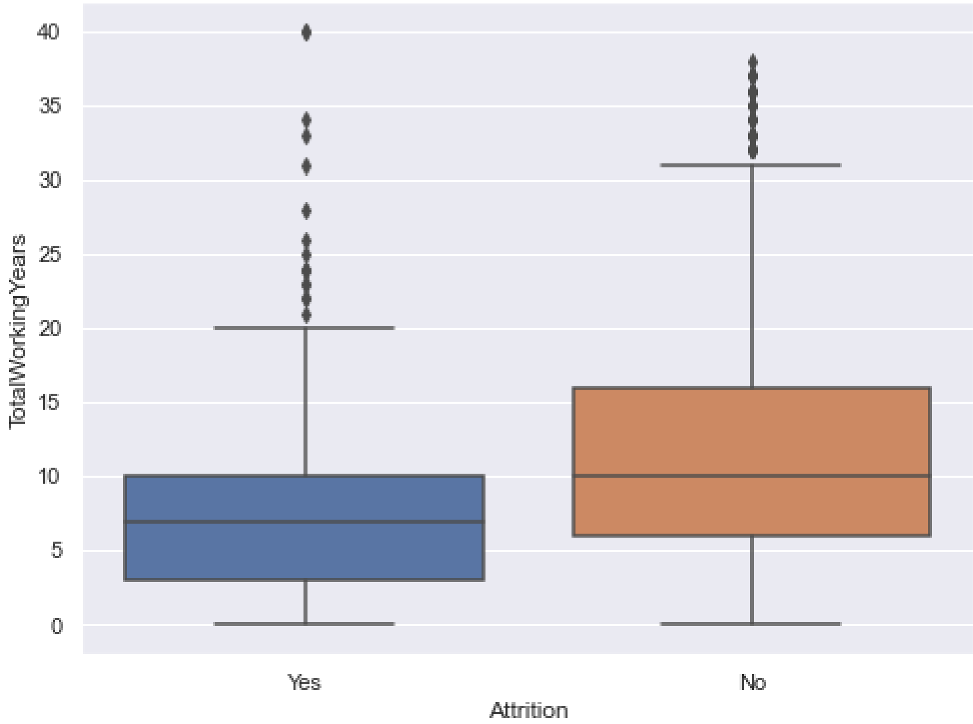
# 4.12. Total working years - Attrition

In [110…
```
df["TotalWorkingYears"].describe()
```

Out[110]:
```
count    1470.000000
mean       11.279592
std         7.780782
min         0.000000
25%         6.000000
50%        10.000000
75%        15.000000
max        40.000000
Name: TotalWorkingYears, dtype: float64
```

In [111…
```
sns.boxplot(x = "Attrition", y = "TotalWorkingYears", data=df)
```

Out[111]:
```
<AxesSubplot:xlabel='Attrition', ylabel='TotalWorkingYears'>
```




In [119…

```
sns.set(rc={'figure.figsize':(10,8)})
sns.swarmplot(x = "Attrition", y = "TotalWorkingYears", data=df)
```

```
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 27.4% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[119]: `<AxesSubplot:xlabel='Attrition', ylabel='TotalWorkingYears'>`

In [120…
```
a=np.quantile(df[df['Attrition']=='Yes']['TotalWorkingYears'],0.25)
b=np.quantile(df[df['Attrition']=='Yes']['TotalWorkingYears'],0.75)
iqr = b-a
print("IQR:",iqr)
ub = b + 1.5*iqr
lb = a - 1.5*iqr
print("Upper bound:",ub)
print("Lower bound:",lb)
```

```
IQR: 7.0
Upper bound: 20.5
Lower bound: -7.5
```

**Hypothesis:**

- People who are new to the company and has gained experience of 1 year or so, they are more tending to leave the company.

**Inferences:**

- We can notice in the plots that People with experience in range 0-2 years are more prone to quit, but we can see people with either 5 or 10 years of experience quitting as well.

# 4.13. Work Life Balance - Attrition

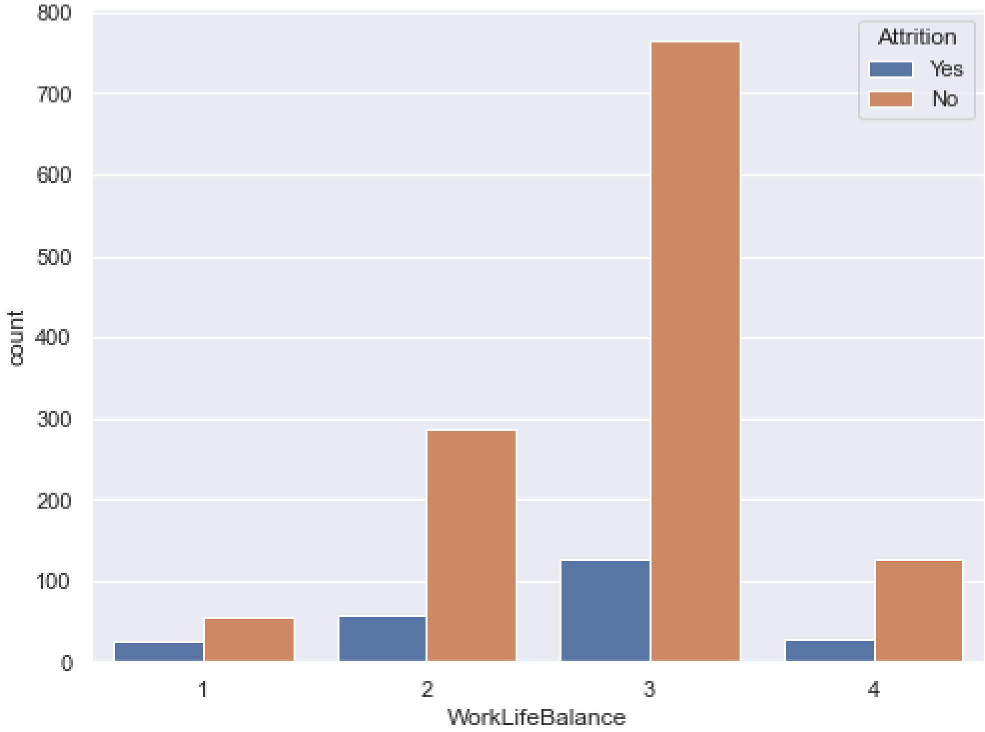
In [121…
```
tab = pd.crosstab(df.Attrition, columns=df.WorkLifeBalance)
tab
```

Out[121]:

| WorkLifeBalance | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **Attrition** | | | | |
| **No** | 55 | 286 | 766 | 126 |
| **Yes** | 25 | 58 | 127 | 27 |

In [122…
```
sns.set(rc={'figure.figsize':(8,6)})
sns.countplot(x='WorkLifeBalance', data=df, hue='Attrition')
```

Out[122]: <AxesSubplot:xlabel='WorkLifeBalance', ylabel='count'>




In [123…

```
print("Attrition rate for voting 1: ", round((25/80)*100,2))
print("Attrition rate for voting 2: ", round((58/344)*100,2))
print("Attrition rate for voting 3: ", round((127/839)*100,2))
print("Attrition rate for voting 4: ", round((27/153)*100,2))
```

```
Attrition rate for voting 1:  31.25
Attrition rate for voting 2:  16.86
Attrition rate for voting 3:  15.14
Attrition rate for voting 4:  17.65
```

### Hypothesis

- People who have rated 1 for work life balance has high attrition rate.

### Observation

- People who have rated 1 has the highest attrition rate. ##### Interestingly the second highest rate of attrition i.e. 17.65% can be observed from the people who have voted 4.

# 4.14. Years at company - Attrition

In [124…

```
df['YearsAtCompany'].describe()
```

Out[124]:

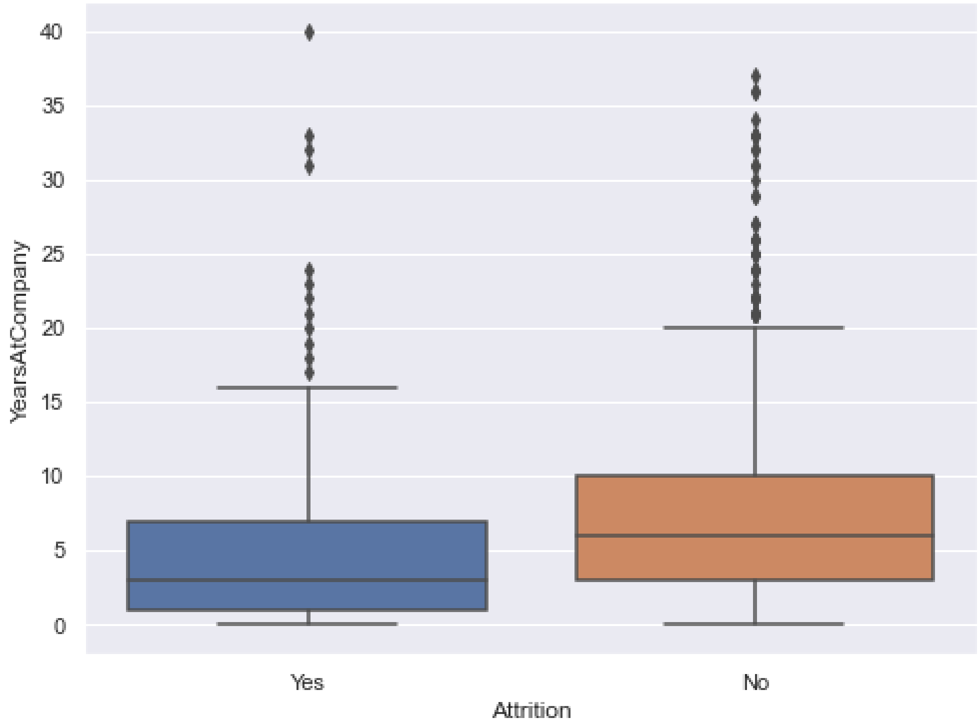
```
count    1470.000000
mean        7.008163
std         6.126525
min         0.000000
25%         3.000000
50%         5.000000
75%         9.000000
max        40.000000
Name: YearsAtCompany, dtype: float64
```

In [125…

```
sns.boxplot(x = 'Attrition', y = 'YearsAtCompany', data = df)
```

Out[125]:

```
<AxesSubplot:xlabel='Attrition', ylabel='YearsAtCompany'>
```




In [126…

```
sns.set(rc={'figure.figsize':(10,8)})
sns.swarmplot(x = 'Attrition', y = 'YearsAtCompany', data = df)
```

```
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 6.8% of the points cannot be placed; you may want to decrease the size of the m
arkers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 44.5% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[126]:

```
<AxesSubplot:xlabel='Attrition', ylabel='YearsAtCompany'>
```


In [128…

```
a=np.quantile(df[df['Attrition']=='Yes']['YearsAtCompany'],0.25)
b=np.quantile(df[df['Attrition']=='Yes']['YearsAtCompany'],0.75)
iqr = b-a
print("IQR:",iqr)
ub = b + 1.5*iqr
lb = a - 1.5*iqr
print("Upper bound:",ub)
print("Lower bound:",lb)
```

```
IQR: 6.0
Upper bound: 16.0
Lower bound: -8.0
```

**Hypothesis:**

- People who are new to the company has high probability of leaving it.

**Observation**

- The maximum frequency of people leaving the firm can be seen in the range of 0-2 years, although we have people leaving the firm at 40 years as well

# 4.15. Years in Current role - Attrition

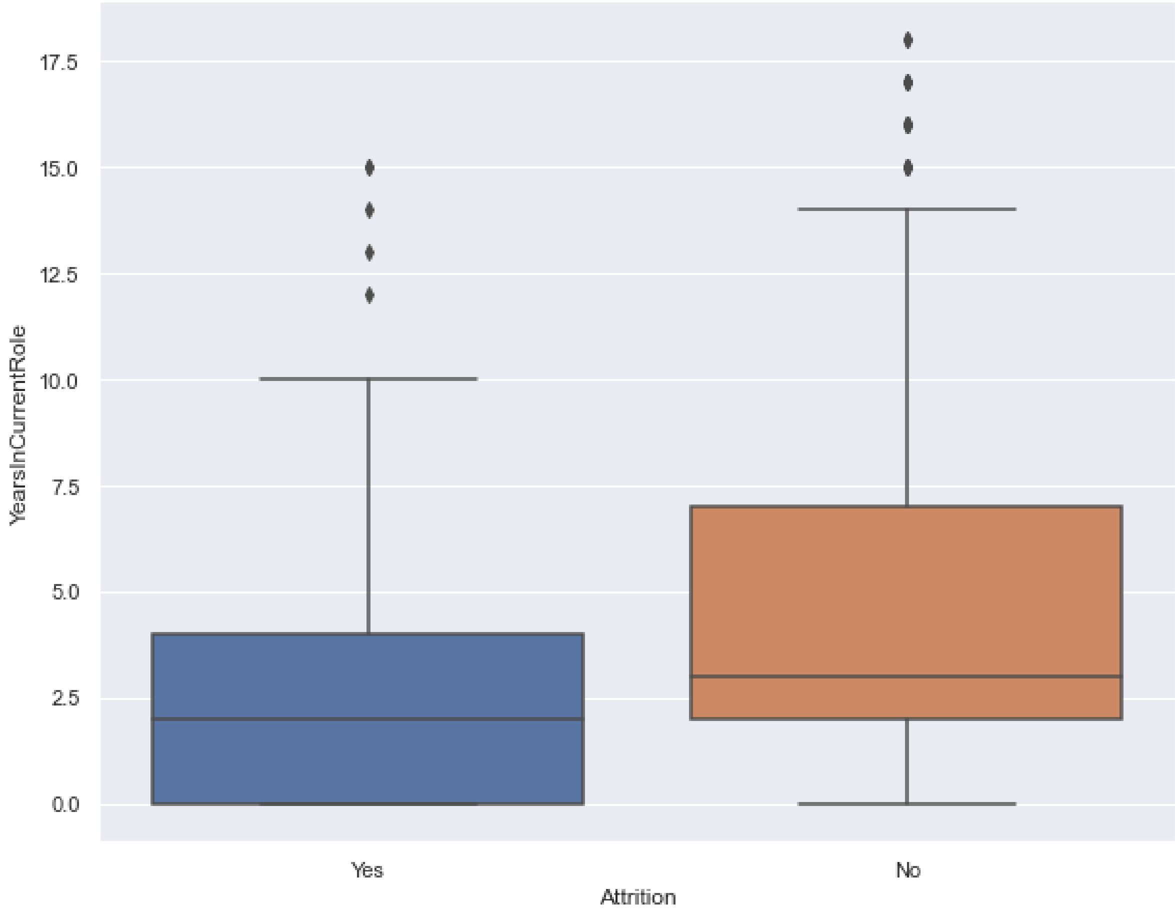
In [129…

```
df['YearsInCurrentRole'].describe()
```

Out[129]:
```
count    1470.000000
mean        4.229252
std         3.623137
min         0.000000
25%         2.000000
50%         3.000000
75%         7.000000
max        18.000000
Name: YearsInCurrentRole, dtype: float64
```

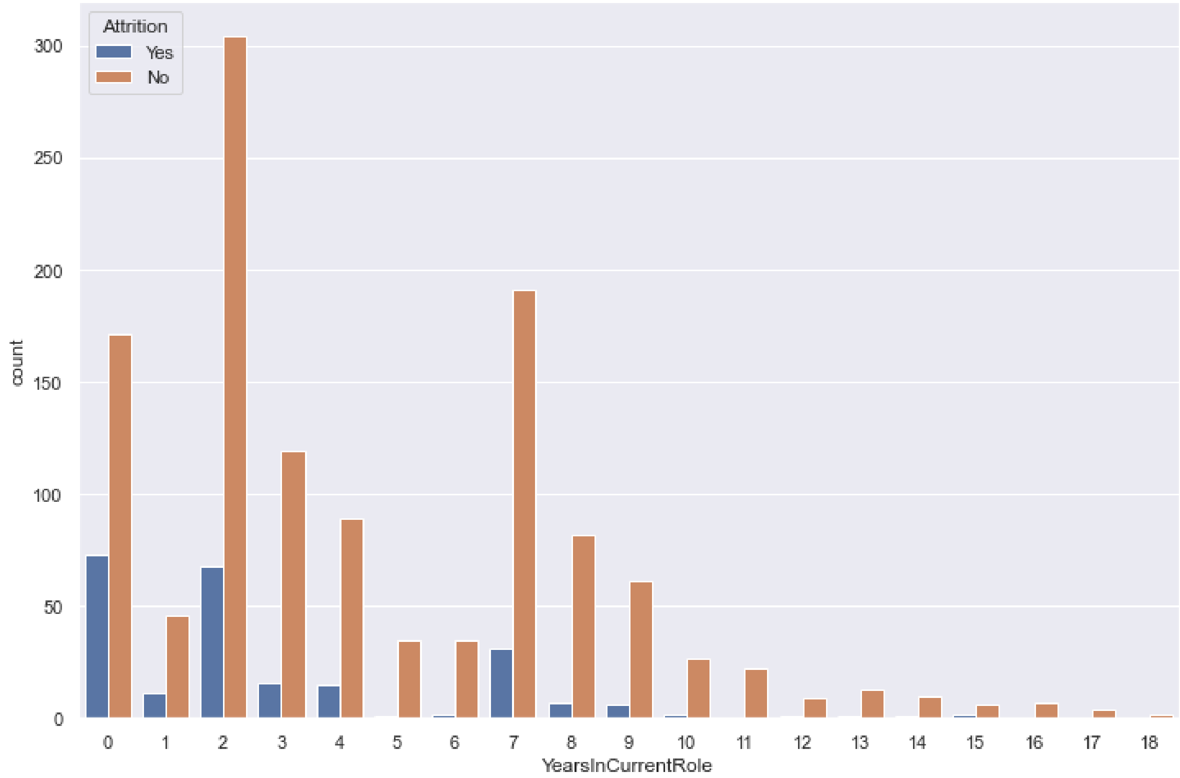
In [130…
```
sns.boxplot(x = 'Attrition', y = 'YearsInCurrentRole', data = df)
```

Out[130]:
```
<AxesSubplot:xlabel='Attrition', ylabel='YearsInCurrentRole'>
```




In [131…
```
sns.set(rc={'figure.figsize':(12,8)})
sns.countplot(x='YearsInCurrentRole',hue="Attrition", data=df)
```

Out[131]:
```
<AxesSubplot:xlabel='YearsInCurrentRole', ylabel='count'>
```




In [134…

```
a=np.quantile(df[df['Attrition']=='Yes']['YearsInCurrentRole'],0.25)
b=np.quantile(df[df['Attrition']=='Yes']['YearsInCurrentRole'],0.75)
iqr = b-a
print("IQR:",iqr)
ub = b + 1.5*iqr
lb = a - 1.5*iqr
print("Upper bound:",ub)
print("Lower bound:",lb)
```

```
IQR: 4.0
Upper bound: 10.0
Lower bound: -6.0
```

**Hypothesis:**

- People who has spent more years in current role will leave the company.

**'Observation:**

- People with even 0 years in current role (maybe they have shifted from another role to a new role) accounts to 74 in total count and has left the company.
- There are only 4-5 people who has left the company for spending around 12 -15 years in the current role, so this nullify our hypothesis.

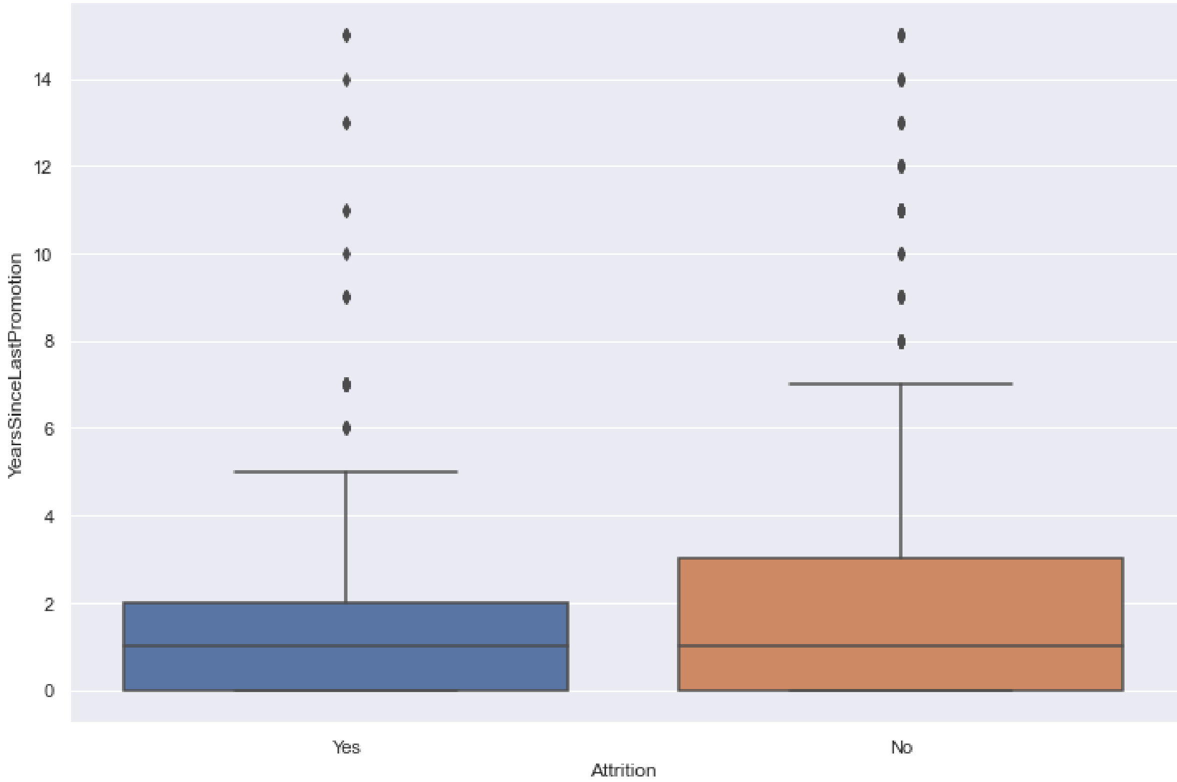
# 4.16. Years since last promotion - Attrition

In [135…

```
sns.boxplot(x = 'Attrition', y = 'YearsSinceLastPromotion', data = df)
```

Out[135]:

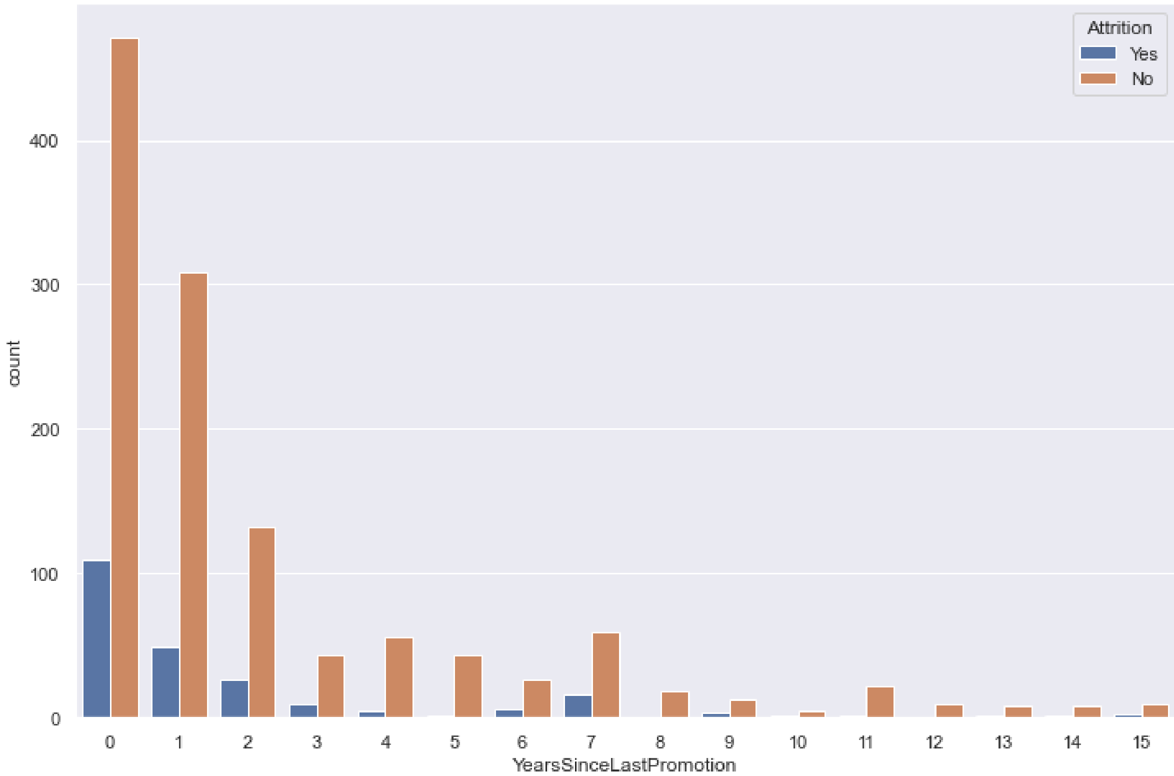
```
<AxesSubplot:xlabel='Attrition', ylabel='YearsSinceLastPromotion'>
```




In [136…

```
sns.set(rc={'figure.figsize':(12,8)})
sns.countplot(x='YearsSinceLastPromotion',hue="Attrition", data=df)
```

Out[136]: <AxesSubplot:xlabel='YearsSinceLastPromotion', ylabel='count'>




In [137…

```
a=np.quantile(df[df['Attrition']=='Yes']['YearsSinceLastPromotion'],0.25)
b=np.quantile(df[df['Attrition']=='Yes']['YearsSinceLastPromotion'],0.75)
iqr = b-a
print("IQR:",iqr)
ub = b + 1.5*iqr
lb = a - 1.5*iqr
print("Upper bound:",ub)
print("Lower bound:",lb)
```

```
IQR: 2.0
Upper bound: 5.0
Lower bound: -3.0
```

**Hypothesis:**

- People who have longer time interval since last promotion should be leaving the company.

**Observation**

- People with 0-2 years since last promotion has more frequeny to leave the company, in comparison to people who had long time interval since last promotion which nullify our hypothesis.

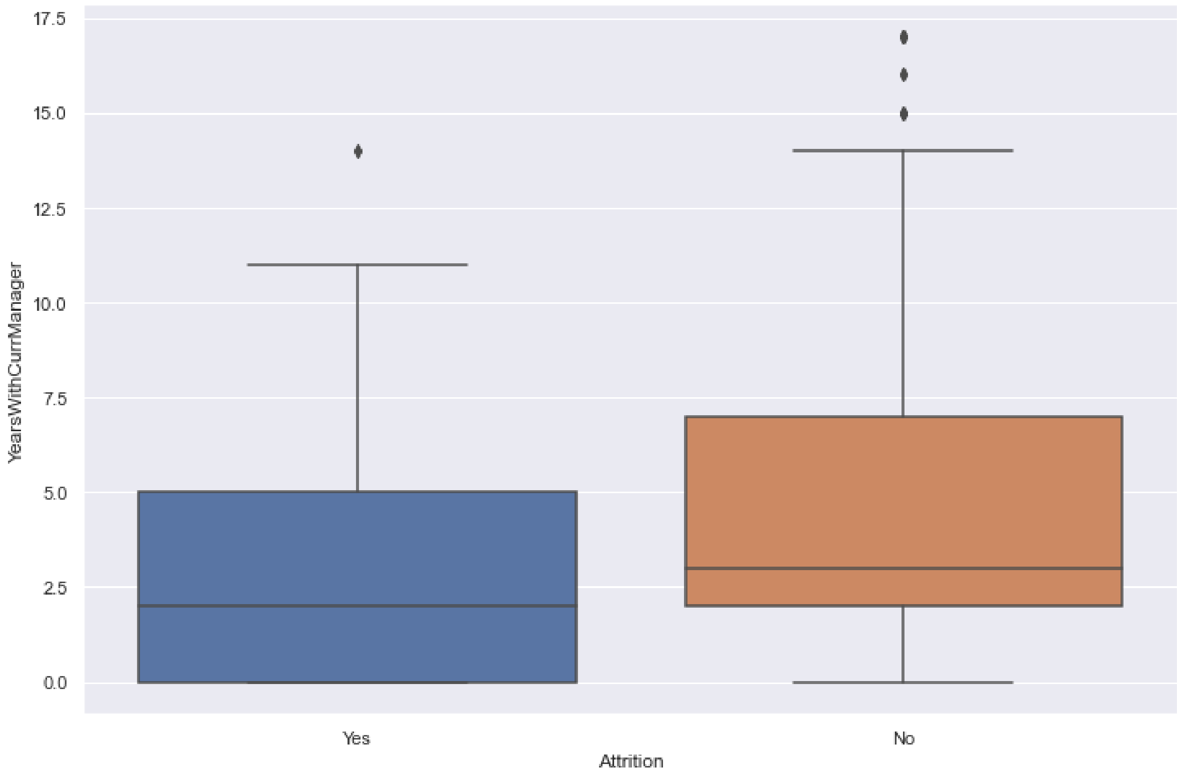
## 4.17. Years with current manager - Attrition

In [138…

```
sns.boxplot(x = 'Attrition', y = 'YearsWithCurrManager', data = df)
```

Out[138]:

```
<AxesSubplot:xlabel='Attrition', ylabel='YearsWithCurrManager'>
```

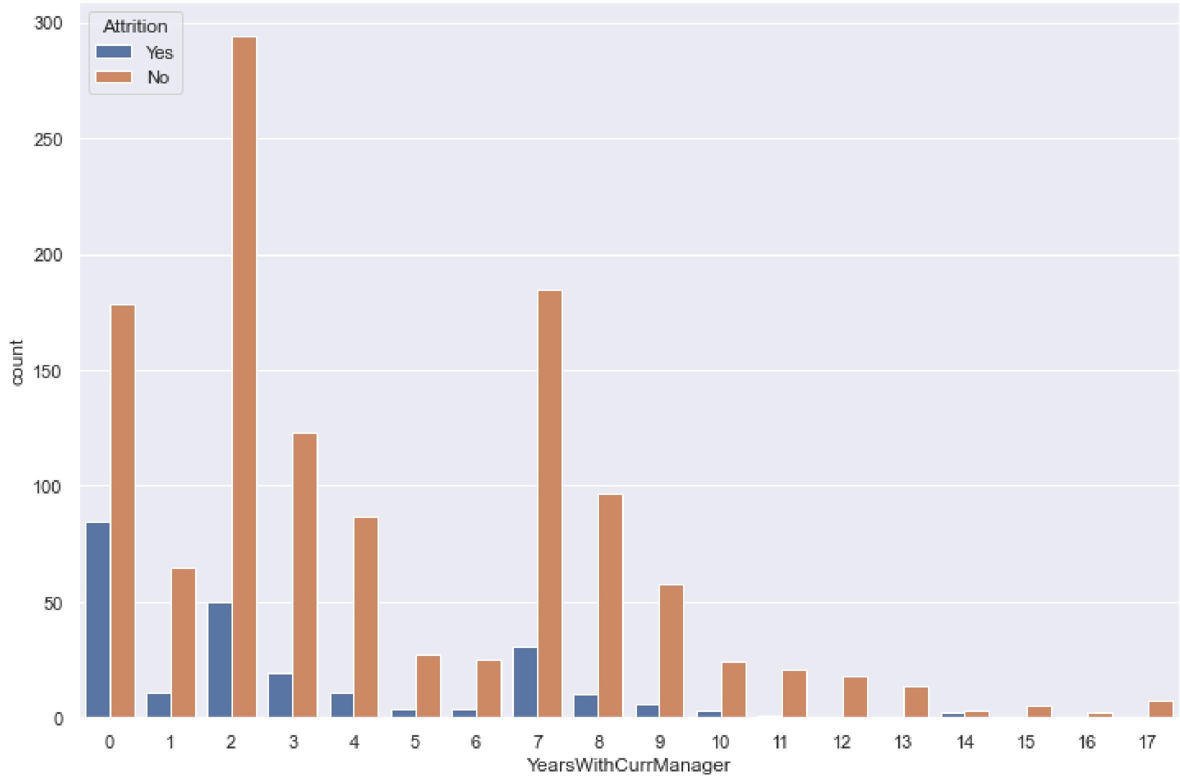



In [139…

```
sns.set(rc={'figure.figsize':(12,8)})
sns.countplot(x='YearsWithCurrManager',hue="Attrition", data=df)
```

Out[139]:

```
<AxesSubplot:xlabel='YearsWithCurrManager', ylabel='count'>
```




In [141...

```
a=np.quantile(df[df['Attrition']=='Yes']['YearsWithCurrManager'],0.25)
b=np.quantile(df[df['Attrition']=='Yes']['YearsWithCurrManager'],0.75)
iqr = b-a
print("IQR:",iqr)
ub = b + 1.5*iqr
lb = a - 1.5*iqr
print("Upper bound:",ub)
print("Lower bound:",lb)
```

```
IQR: 5.0
Upper bound: 12.5
Lower bound: -7.5
```

**Hypothesis:**

- People with longer time period will be leaving the company frequently than other people.

**Observation:**

- In this we can see, mostly people within 7 years of experience with current manager are more prone to leave the company as compared to the people who have already been with current manager for more than 7 years.

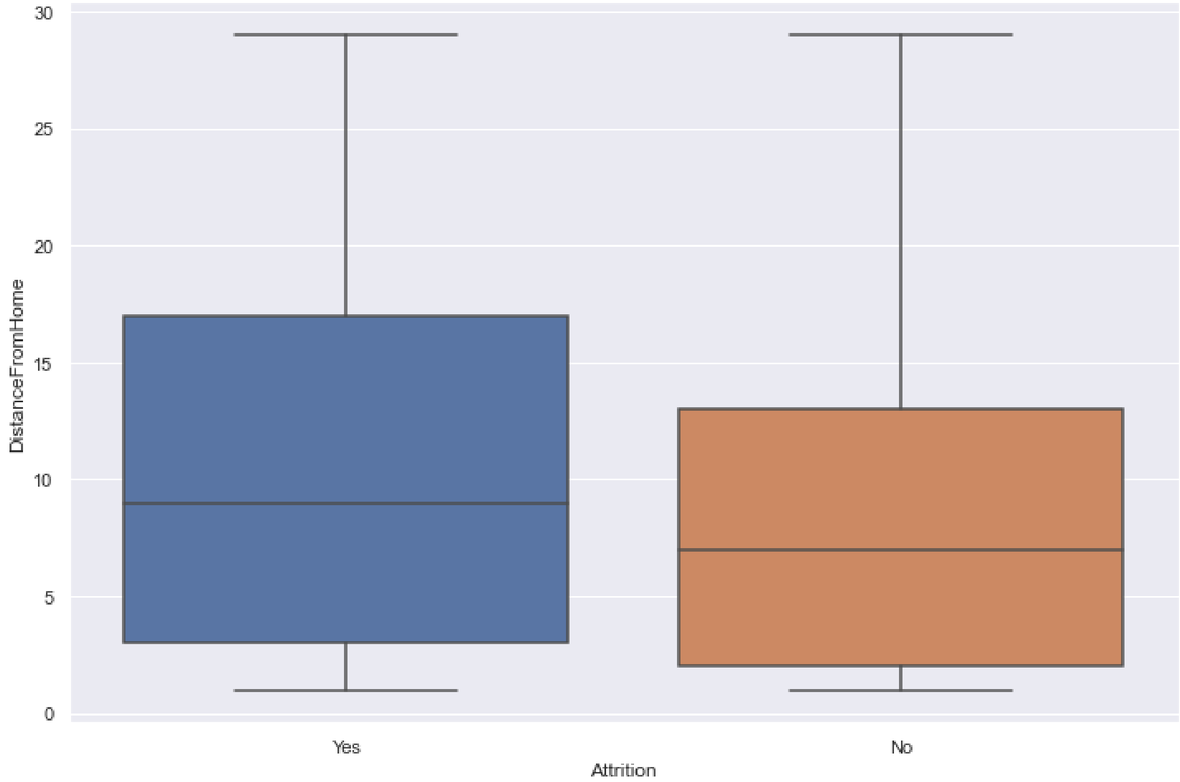
# 4.18. Distance from Home - Attrition

In [142...

```
sns.boxplot(x = 'Attrition', y = 'DistanceFromHome', data = df)
```

Out[142]:

```
<AxesSubplot:xlabel='Attrition', ylabel='DistanceFromHome'>
```

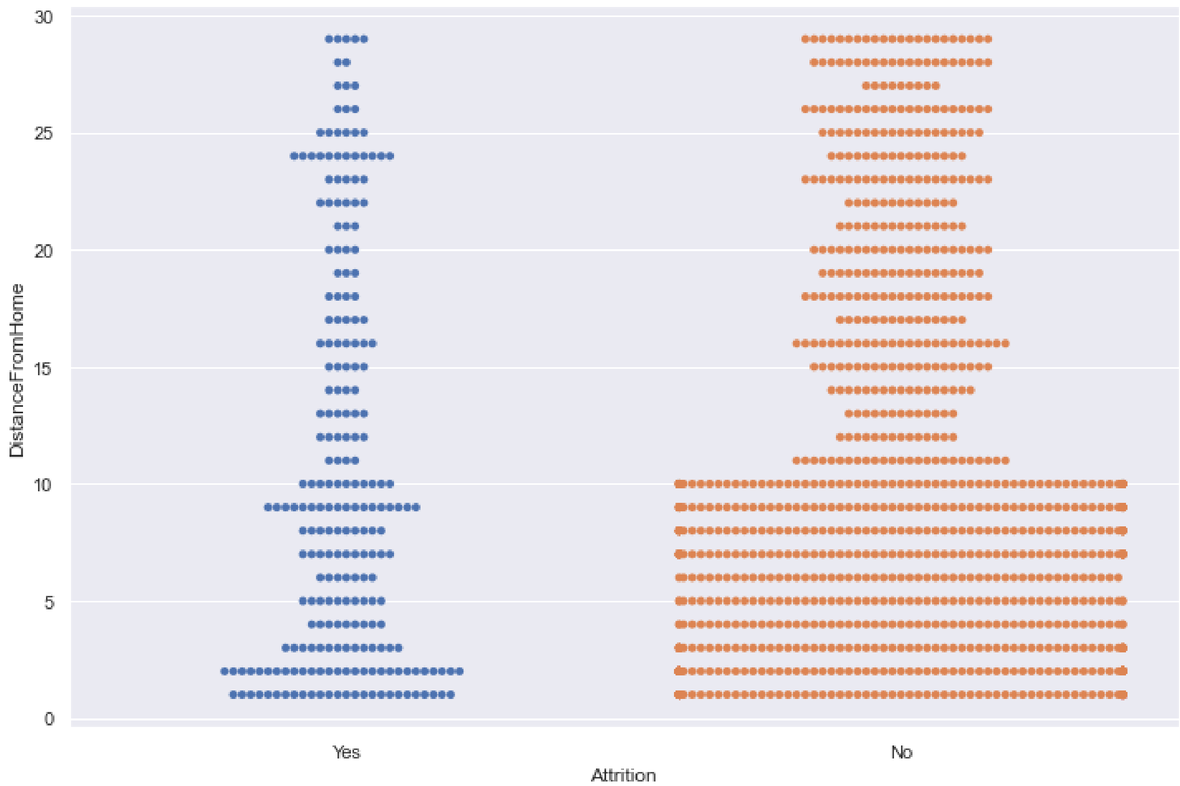



In [143…

```
sns.swarmplot(x = 'Attrition', y = 'DistanceFromHome', data = df)
```

```
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 30.2% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[143]: <AxesSubplot:xlabel='Attrition', ylabel='DistanceFromHome'>




In [144…

```
a=np.quantile(df[df['Attrition']=='Yes']['DistanceFromHome'],0.25)
b=np.quantile(df[df['Attrition']=='Yes']['DistanceFromHome'],0.75)
iqr = b-a
print("IQR:",iqr)
```

```
ub = b + 1.5*iqr
lb = a - 1.5*iqr
print("Upper bound:",ub)
print("Lower bound:",lb)
```

```
IQR: 14.0
Upper bound: 38.0
Lower bound: -18.0
```

**Hypothesis**

- People living far away will be leaving the company more often due to lead time increase in reaching to work.

**Observation:**

- People who are living within the radius of 0-2 kms are leaving the company more frequently than the people living far off.

# 5. Multivariate Analysis

## 5.1. Plotting correlation matrix

In [145…

```
sns.set(rc={'figure.figsize':(20,15)})
sns.heatmap(df.corr(), annot = True, vmin=-1, vmax=1, center= 0)
```

Out[145]:

```
<AxesSubplot:>
```

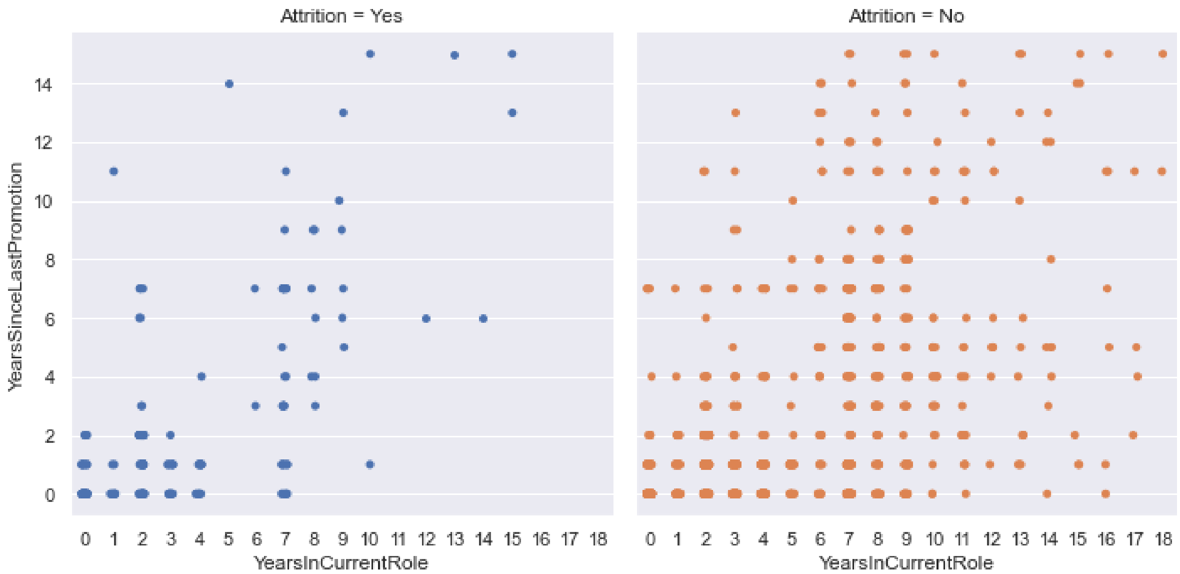
# 5.2. Comparing Attrition, Years since Last Promotion and Years in Current Role.

## Hypothesis: Employees having spent more than 5 years in Current Role yet not getting Promotion for >=5 years, are leaving the Company

In [146…

```
sns.factorplot(x="YearsInCurrentRole",
                   y="YearsSinceLastPromotion",
                   data=df,
                   hue='Attrition',  # Color by stage
                   col='Attrition',  # Separate by stage
                   kind='strip');
```

```
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:3717: UserWarnin
g: The `factorplot` function has been renamed to `catplot`. The original name will
be removed in a future release. Please update your code. Note that the default `ki
nd` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)
```




**Fact:**

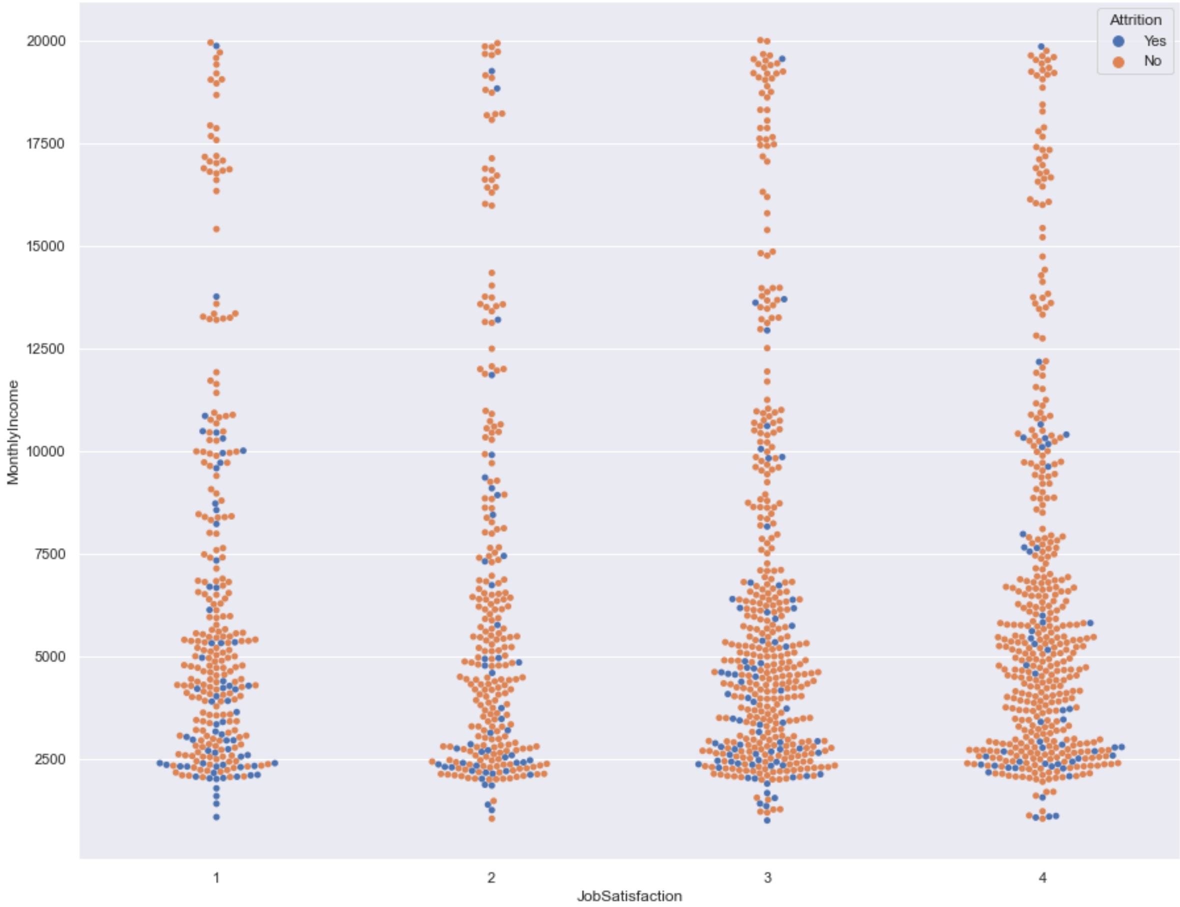
- Employees are leaving the job even within 0 to 3 Years in current Role with less than or equal to 2 years since last promotion. However, Maximum no. of the Employees leaving the job are seen to be the ones who haven't got promoted in 3 to 9 years while being in Current Role for about 6-9 years. Above this, employees are less seemingly to leave the Company.

# 5.3 Comparing Attrition, Job Satisfaction and Monthly income

## Hypothesis: Employees with Lower Monthly Income and with Less Job satisfaction are leaving the Company.

In [147…

```
sns.set(rc={'figure.figsize':(15,12)})
sns.swarmplot(x="JobSatisfaction", y="MonthlyIncome", hue="Attrition", data=df);
```




**Fact:**

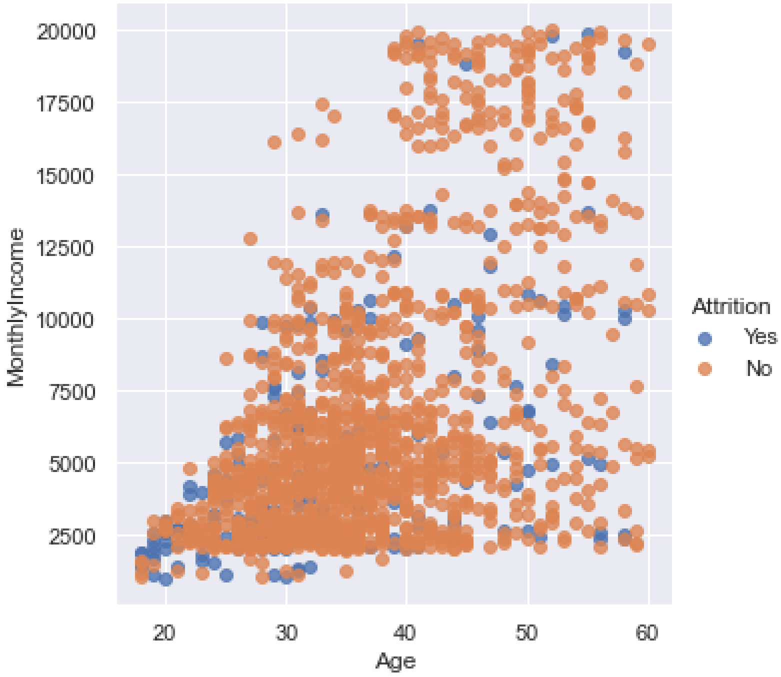
- Employee with Lower monthly income & lowest Job Satisfaction are more prone to leave the Company but Employees with lower Monthly Income but Satisfaction level as good as 3, are also leaving the company at higher rates. In a few exceptional cases Employees with highest salary and with lowest to highest Job Satisfaction are also leaving the Company.

# 5.4. Comparing Attrition, Age and Monthly Income.

## Hypothesis: Employees at younger age with Lower Monthly Income are more vulnerable, leading to Employee Attrition.

In [148…

```
sns.lmplot(x = 'Age', y ='MonthlyIncome', fit_reg=False, data = df, hue= 'Attritio
```




**Fact:**

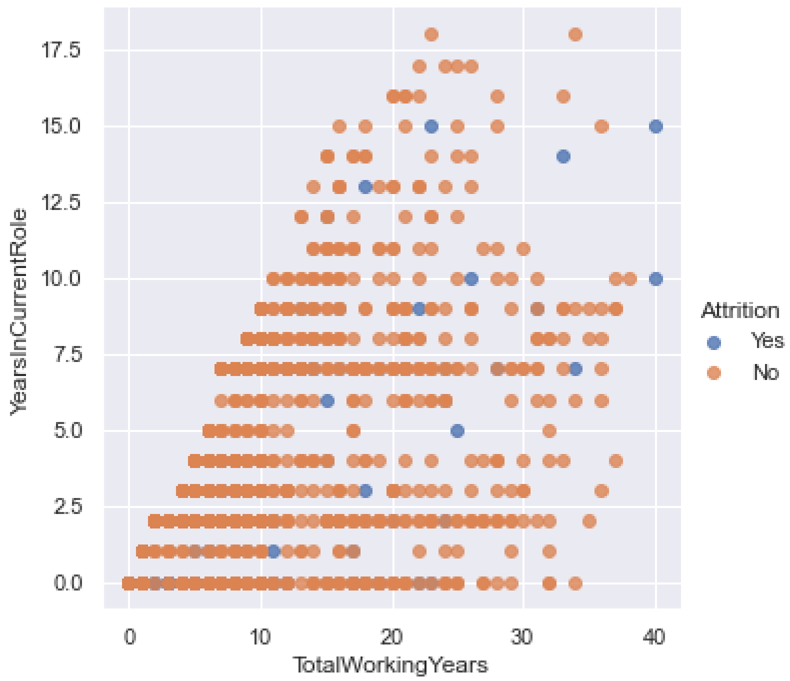
- The highest number of Attrition is seen among Employees of age-group 20-35 under Monthly Income Rs.10000. Our hypothesis holds true.

# 5.5. Comparing Attrition, Years in Current Role and Total Working Years.

## Hypothesis: Employees with more than 15 years of Work Experience and more than 5 years in the Current Role are leaving the company.

In [149…

```
sns.lmplot(x = 'TotalWorkingYears', y ='YearsInCurrentRole', fit_reg=False, data =
```



**Fact:**

- Employees under 14years of Work Experience and having spent upto 0-7 years in Current Role are mostly seen to be leaving the Company than Employees with around 15+ years of Total Working Years, so the hypothesis fails.

## 5.6. Comparing Attrition, Years with Current Manager and Years in Current Role.

### Hypothesis: Employees with 10+ years in Current role and 10+ years under same Manager are attriting.

In [150…

```
sns.factorplot(x="YearsInCurrentRole",
                   y="YearsWithCurrManager",
                   data=df,
                   hue='Attrition',  # Color by stage
                   col='Attrition',  # Separate by stage
                   kind='swarm');
```

```
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:3717: UserWarnin
g: The `factorplot` function has been renamed to `catplot`. The original name will
be removed in a future release. Please update your code. Note that the default `ki
nd` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 89.0% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 18.2% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 70.6% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 25.0% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 6.7% of the points cannot be placed; you may want to decrease the size of the m
arkers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 41.9% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 28.6% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 84.2% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 52.2% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 91.4% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 79.8% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 68.5% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 42.9% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 28.6% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 78.0% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
```

```
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 63.4% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 47.5% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 18.5% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 22.7% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\UMAIR\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 7.7% of the points cannot be placed; you may want to decrease the size of the m
arkers or use stripplot.
  warnings.warn(msg, UserWarning)
```


**Fact:**

- Our Hypothesis fails to hold true as we can clearly see that most number of Employees leaving the job belong to <=10 years with the current manager and around <=5 years in Current role!

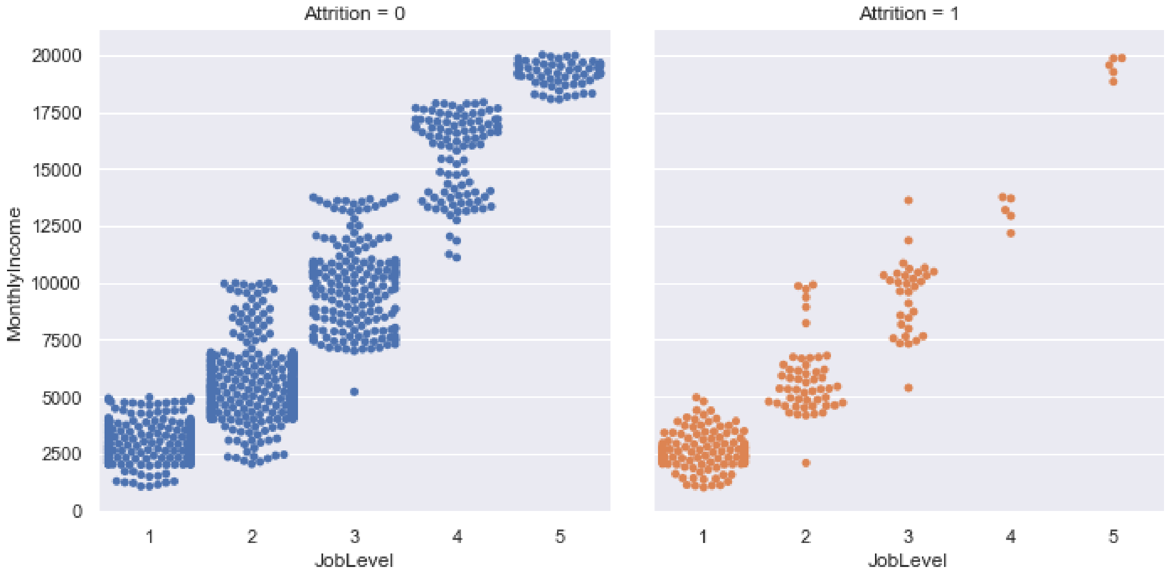
# 5.7. Comparing Job Level, Monthly Income and Attrition.

## Employess at Higher Job Level and Lower Monthly Income are leaving the Company.

In [130…

```
sns.set(rc={'figure.figsize':(12,8)})
sns.factorplot(x='JobLevel',
                   y='MonthlyIncome',
                   data=df,
                   hue='Attrition',   # Color by stage
                   col='Attrition',   # Separate by stage
                   kind='swarm');
```

```
C:\Users\Pranjal Shandilya\Anaconda3\lib\site-packages\seaborn\categorical.py:366
6: UserWarning: The `factorplot` function has been renamed to `catplot`. The origi
nal name will be removed in a future release. Please update your code. Note that t
he default `kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)
```




**Fact:**

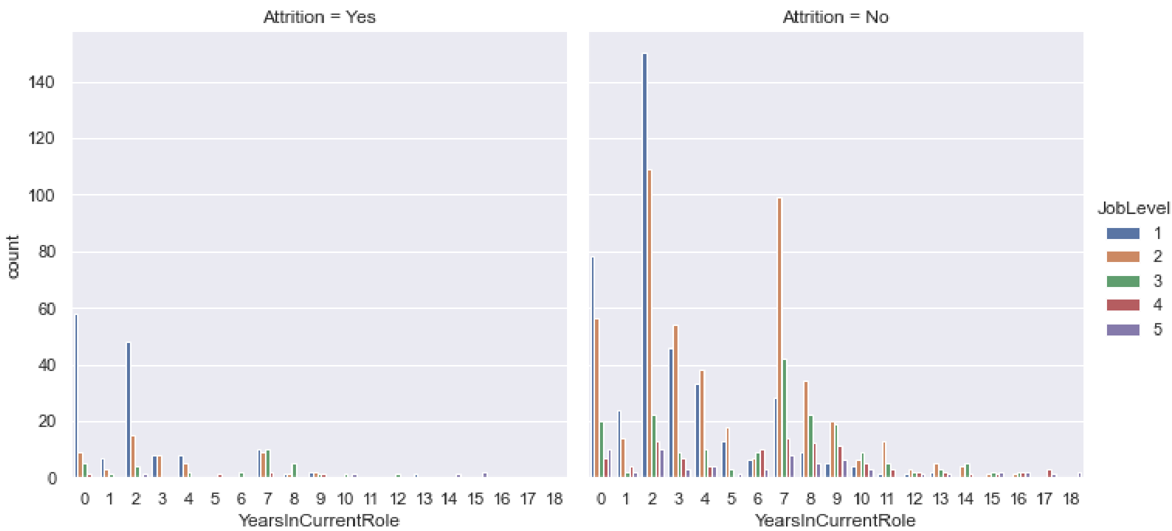
- Employees at Job Level 1, with monthly income less than 5000 are more prone to leave the Company against our Hypothesis of Employess at Higher Job Level and Lower Monthly Income are leaving the Company.

# 5.8. Comparing Attrition, Job Level and Years in Current Role.

## Employees having spent more than 5 years in the Current Role at lower level, are leaving the Company.

In [151…

```
sns.set(rc={'figure.figsize':(12,8)})
sns.catplot(x='YearsInCurrentRole', data=df, hue="JobLevel", col="Attrition", kind
```




**Fact:**

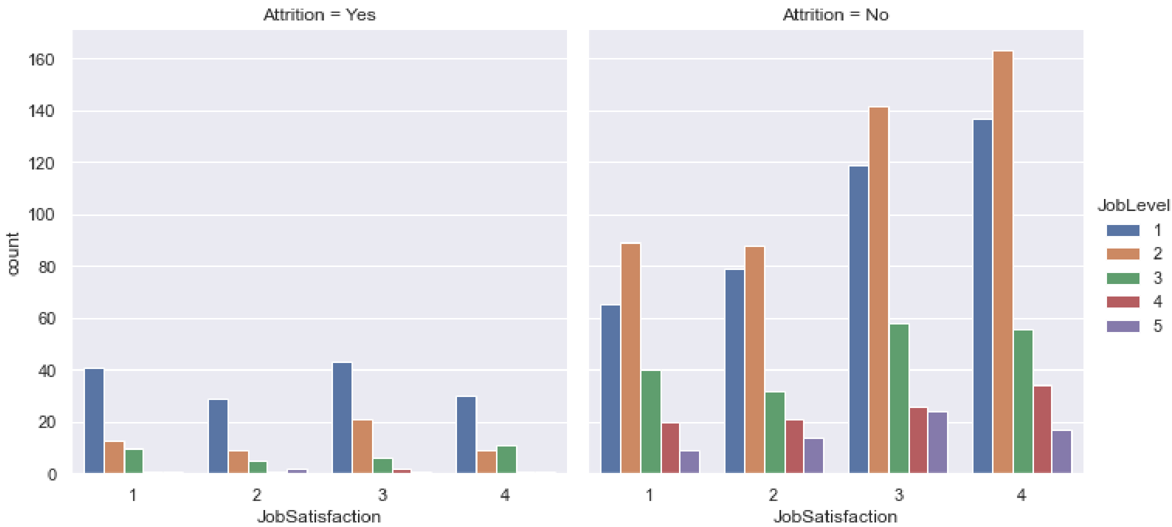
- It is clearly evident from the graphs that Employees at lower level with less than 4 years in Current role are leaving the company. So our hypothesis fails to hold true.

# 5.9. Comparing Attrition, Job Satisfaction and Job Level.

## Hypothesis : Employees at Lower Job Level with Low Job Satisfaction are leaving the Company.

In [152…

```
sns.set(rc={'figure.figsize':(12,8)})
sns.catplot(x='JobSatisfaction', data=df, hue="JobLevel", col="Attrition", kind="c
```




**Fact:**

- We can clearly see that attrition is more in case of Employees at Lower Job levels at any level of Job Satisfaction. Our Hypothesis is partially correct.

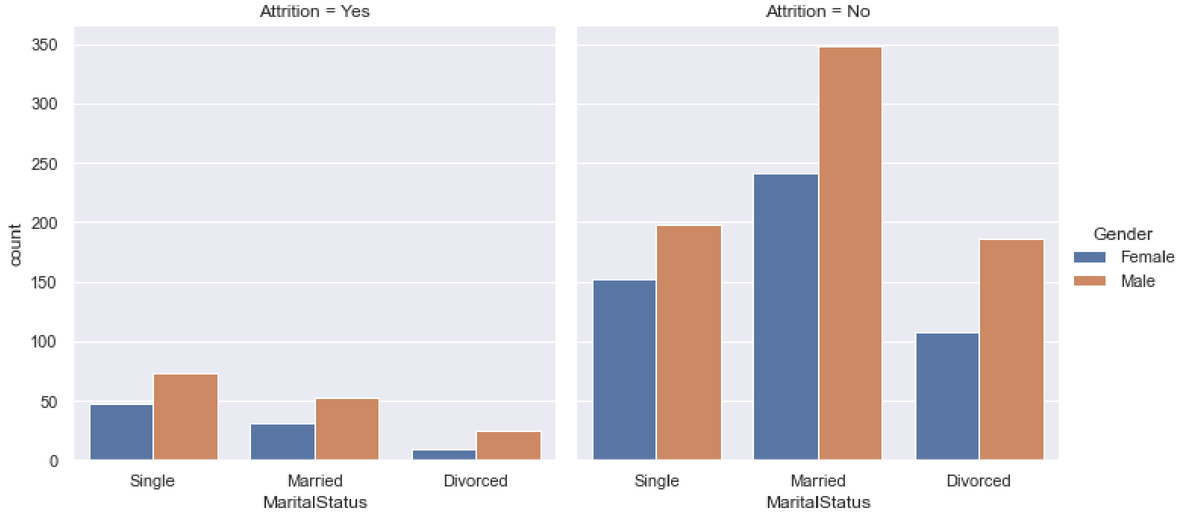
# 5.10. Comparing Attrition, Gender and Marital Status.

## Hypothesis: 1. Married-Female Employees are more prone to leave the Company.

## Hypothesis: 2. Unmarried-Male Employees are more prone to leave the Company.

In [153…

```
sns.catplot(x='MaritalStatus', data=df, hue="Gender", col="Attrition", kind="coun
```




**Fact**

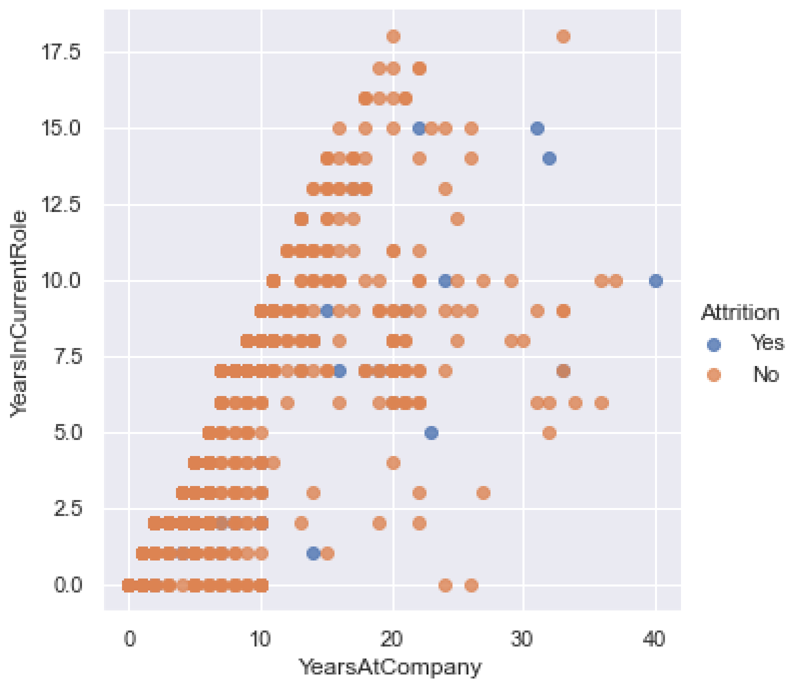
- The rate of attrition is mostly seen in Employees with Marital Status as Single, so our Hypothesis 1 fails. However among singles, male employees are leaving more so our 2nd Hypothesis that Unmarried-Male employees are more prone to leave the Company holds true!

# 5.11. Comparing Attrition, Years in company and Years in current role.

## Employees with more than 5 years in the Company in their Current Role are leaving the Company

In [154…

```
sns.lmplot(x = 'YearsAtCompany', y ='YearsInCurrentRole', fit_reg=False, data = df
```




**Fact:**

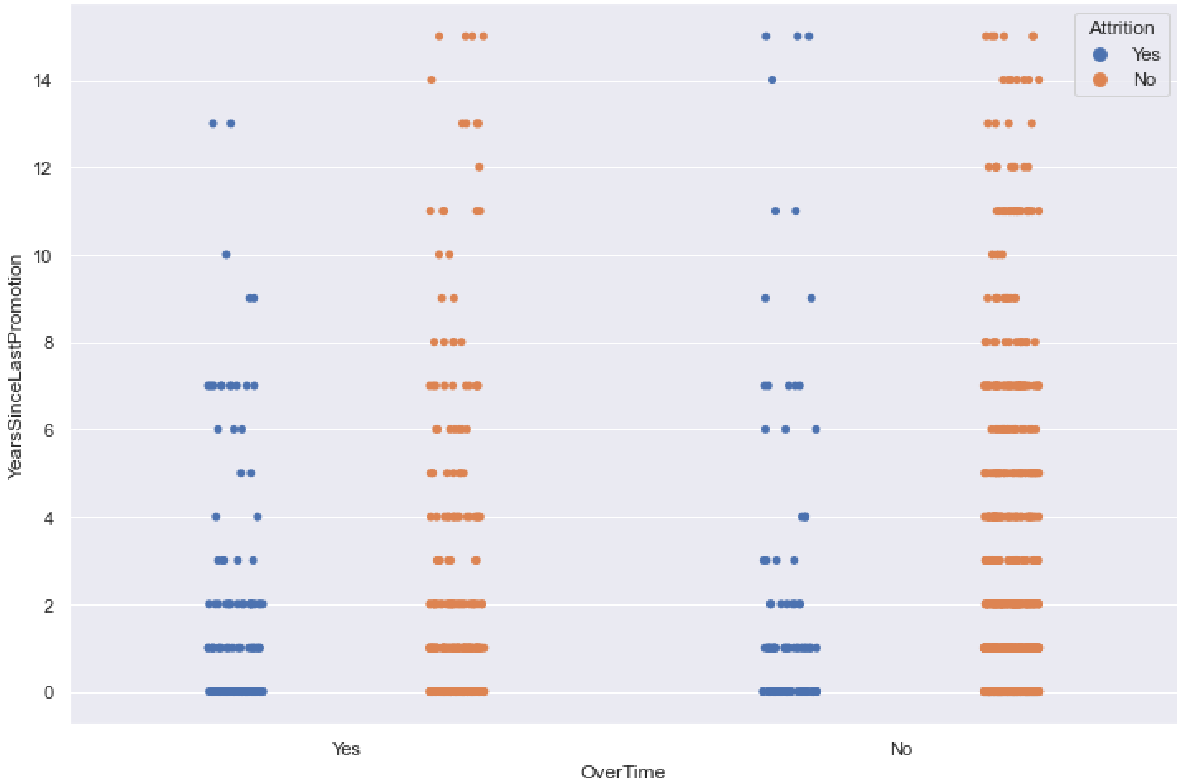
- Most of the attrition is seen among Employees under 6 years At Company while being in their Current Role for about 0-4 years.

# 5.12. Comparing Attrition, Overtime and Years Since Last Promotion.

## Hypothesis: Employees doing Overtime but not getting Promoted for past 5 years are leaving the Company.

In [155…
```
sns.set(rc={'figure.figsize':(12,8)})
sns.stripplot(x = 'OverTime', y = 'YearsSinceLastPromotion', hue = 'Attrition', da
```

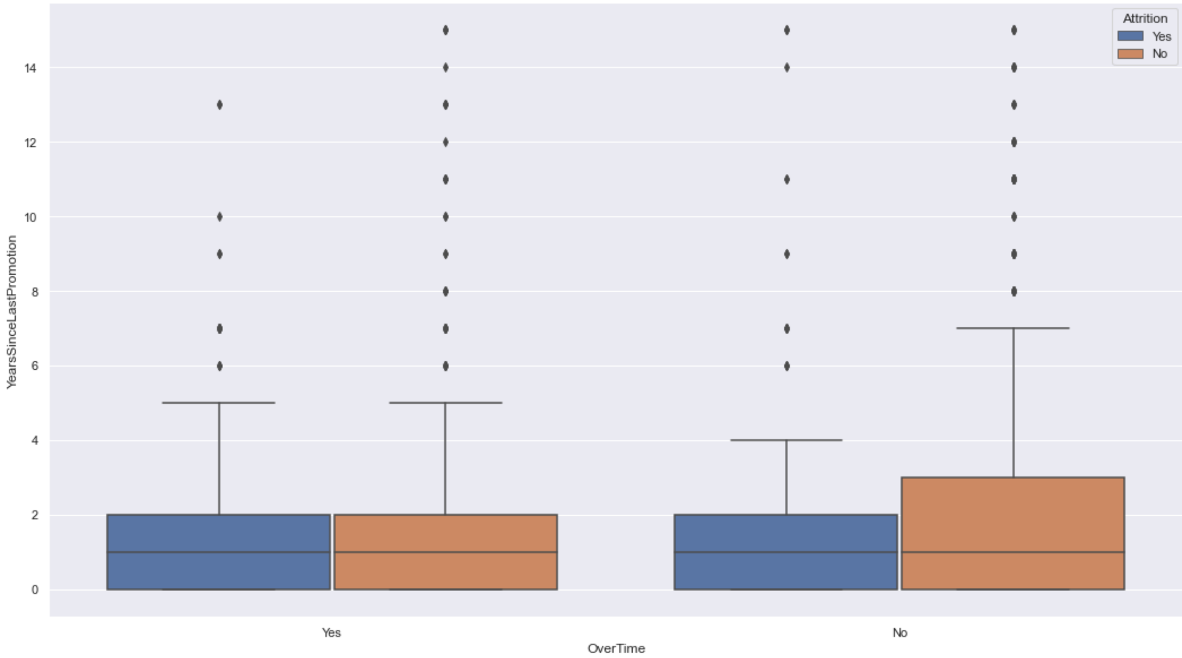
Out[155]: <AxesSubplot:xlabel='OverTime', ylabel='YearsSinceLastPromotion'>




In [158…
```
a= sns.boxplot( y = "YearsSinceLastPromotion",
                x = "OverTime",
                hue= df['Attrition'],
                data = df)

a.figure.set_size_inches(18,10)
plt
```

Out[158]:
```
<module 'matplotlib' from 'C:\\Users\\UMAIR\\anaconda3\\lib\\site-packages\\matplo
tlib\\__init__.py'>
```




**Fact:**

- Most Employees with around 0-3 years since promotion are leaving the company irrespective of Overtime done or not. Here our hypothesis fails to hold true!

# SUMMARY

## Top 5 Variables which has affected Attrition:

**1. OverTime**

- Attrition rate for Overtime shows a promising 30.53%, so its one of the important variables to look for.

**2. Job Level**

- It is negatively Correlated to Attrition. Lower the Job Level, higher the Attrition Rate. Attrition rate for lowest Job Level i.e. 1, shows 26.34%.

**3. Monthly Income**

- It is negatively correlated to attrition. Lesser the Monthly Income, higher the attrition rate. Employees with salary around 2500 are more prone to leave the company.

**4. Total Working Years**

- Its negatively Correlated to Attrition. Lesser the Total Working Years, higher the Attrition Rate. People with 3 to 10 years of experience before joining the company are more prone to leave.

**5. Years in Current Role**

- Its negatively correlated to Attrition. Lesser the Years in Current role, Higher the attrition rate. People with 0 to 2 years in current role show maximum frequency of attrition.

## To reduce the attrition rate I would recommend:

1. **Offer support**: Provide work-life balance programs and flexible work arrangements that help employees manage their workload and reduce overtime. Offer mentorship and coaching programs that support employees in their current roles and help them develop the skills required for future roles.

2. **Encourage career growth**: Provide career advancement opportunities, training programs, and mentorship to support employee progression to higher job levels.

3. **Offer competitive compensation**: Offer competitive salaries and benefits that align with the market standards and recognize and reward long-serving employees for their commitment to the organization.

4. **Foster a positive work environment**: Provide a positive and inclusive work environment that encourages employee engagement and job satisfaction.

5. **Gather employee feedback**: Conduct regular employee engagement surveys to understand the underlying reasons for employee turnover and take corrective actions accordingly.