HR Analytics

Problem Statement

A large company named XYZ, employs, at any given point of time, around 4000 employees. However, every year, around 15% of its employees leave the company and need to be replaced with the talent pool available in the job market. The management believes that this level of attrition (employees leaving, either on their own or because they got fired) is bad for the company, because of the following reasons -

The former employees’ projects get delayed, which makes it difficult to meet timelines, resulting in a reputation loss among consumers and partners A sizeable department has to be maintained, for the purposes of recruiting new talent More often than not, the new employees have to be trained for the job and/or given time to acclimatise themselves to the company Hence, the management has contracted an HR analytics firm to understand what factors they should focus on, in order to curb attrition. In other words, they want to know what changes they should make to their workplace, in order to get most of their employees to stay. Also, they want to know which of these variables is most important and needs to be addressed right away.

Since you are one of the star analysts at the firm, this project has been given to you.

Goal of the case study You are required to model the probability of attrition. The results thus obtained will be used by the management to understand what changes they should make to their workplace, in order to get most of their employees to stay.

In [1]:

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

In [2]:

```
# Load Dataset

df = pd.read_csv('dataset_ass7.csv')
df.head()
```

Out[2]:

rtment	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	...
Sales	6	2	Life Sciences	1	1	Female	...
arch & ıpment	10	1	Life Sciences	1	2	Female	...
arch & ıpment	17	4	Other	1	3	Male	...
arch & ıpment	2	5	Life Sciences	1	4	Male	...
arch & ıpment	10	1	Medical	1	5	Male	...

In [3]:

```
df.columns
```

Out[3]:

```
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHom
e',
       'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gende
r',
       'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
       'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager'],
      dtype='object')
```

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4410 entries, 0 to 4409
Data columns (total 24 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Age                      4410 non-null   int64  
 1   Attrition                4410 non-null   object 
 2   BusinessTravel           4410 non-null   object 
 3   Department               4410 non-null   object 
 4   DistanceFromHome         4410 non-null   int64  
 5   Education                4410 non-null   int64  
 6   EducationField           4410 non-null   object 
 7   EmployeeCount            4410 non-null   int64  
 8   EmployeeID               4410 non-null   int64  
 9   Gender                   4410 non-null   object 
 10  JobLevel                 4410 non-null   int64  
 11  JobRole                  4410 non-null   object 
 12  MaritalStatus            4410 non-null   object 
 13  MonthlyIncome            4410 non-null   int64  
 14  NumCompaniesWorked       4391 non-null   float64
 15  Over18                   4410 non-null   object 
 16  PercentSalaryHike        4410 non-null   int64  
 17  StandardHours            4410 non-null   int64  
 18  StockOptionLevel         4410 non-null   int64  
 19  TotalWorkingYears        4401 non-null   float64
 20  TrainingTimesLastYear    4410 non-null   int64  
 21  YearsAtCompany           4410 non-null   int64  
 22  YearsSinceLastPromotion  4410 non-null   int64  
 23  YearsWithCurrManager     4410 non-null   int64  
dtypes: float64(2), int64(14), object(8)
memory usage: 827.0+ KB
```

In [5]:

```
df.isnull().any()
```

Out[5]:

```
Age                         False
Attrition                   False
BusinessTravel              False
Department                  False
DistanceFromHome            False
Education                   False
EducationField              False
EmployeeCount               False
EmployeeID                  False
Gender                      False
JobLevel                    False
JobRole                     False
MaritalStatus               False
MonthlyIncome               False
NumCompaniesWorked           True
Over18                      False
PercentSalaryHike           False
StandardHours               False
StockOptionLevel            False
TotalWorkingYears            True
TrainingTimesLastYear       False
YearsAtCompany              False
YearsSinceLastPromotion     False
YearsWithCurrManager        False
dtype: bool
```

In [7]:

```
df.fillna(0,inplace =True)
df.isnull().any() # no null values
```

Out[7]:

```
Age                         False
Attrition                   False
BusinessTravel              False
Department                  False
DistanceFromHome            False
Education                   False
EducationField              False
EmployeeCount               False
EmployeeID                  False
Gender                      False
JobLevel                    False
JobRole                     False
MaritalStatus               False
MonthlyIncome               False
NumCompaniesWorked          False
Over18                      False
PercentSalaryHike           False
StandardHours               False
StockOptionLevel            False
TotalWorkingYears           False
TrainingTimesLastYear       False
YearsAtCompany              False
YearsSinceLastPromotion     False
YearsWithCurrManager        False
dtype: bool
```

In [8]:

```
df.drop(['EmployeeCount','EmployeeID','StandardHours', 'Over18'],axis=1,inplace=True) # rem
df.head() # unnecessary features removed
```

Out[8]:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField
0	51	No	Travel_Rarely	Sales	6	2	Life Sciences
1	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences
2	32	No	Travel_Frequently	Research & Development	17	4	Other
3	38	No	Non-Travel	Research & Development	2	5	Life Sciences
4	32	No	Travel_Rarely	Research & Development	10	1	Medical

Univariate Analysis

In [13]:

```
# get describe of Continuous variables only
df[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSal
    'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrMana
```

Out[13]:

	count	mean	std	min	25%	50%	75
Age	4410.0	36.923810	9.133301	18.0	30.0	36.0	43.
DistanceFromHome	4410.0	9.192517	8.105026	1.0	2.0	7.0	14.
Education	4410.0	2.912925	1.023933	1.0	2.0	3.0	4.
MonthlyIncome	4410.0	65029.312925	47068.888559	10090.0	29110.0	49190.0	83800.
NumCompaniesWorked	4410.0	2.683220	2.499737	0.0	1.0	2.0	4.
PercentSalaryHike	4410.0	15.209524	3.659108	11.0	12.0	14.0	18.
TotalWorkingYears	4410.0	11.256916	7.790928	0.0	6.0	10.0	15.
TrainingTimesLastYear	4410.0	2.799320	1.288978	0.0	2.0	3.0	3.
YearsAtCompany	4410.0	7.008163	6.125135	0.0	3.0	5.0	9.
YearsSinceLastPromotion	4410.0	2.187755	3.221699	0.0	0.0	1.0	3.
YearsWithCurrManager	4410.0	4.123129	3.567327	0.0	2.0	3.0	7.

In [100]:

```
# get median of continuous variables
df[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSal
  'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManage
```

Out[100]:

```
Age                          36.0
DistanceFromHome              7.0
Education                     3.0
MonthlyIncome             49190.0
NumCompaniesWorked            2.0
PercentSalaryHike            14.0
TotalWorkingYears            10.0
TrainingTimesLastYear         3.0
YearsAtCompany                5.0
YearsSinceLastPromotion       1.0
YearsWithCurrManager          3.0
dtype: float64
```

In [102]:

```
# Mode
df[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSal
  'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManage
```

Out[102]:

	0
Age	35.0
DistanceFromHome	2.0
Education	3.0
MonthlyIncome	23420.0
NumCompaniesWorked	1.0
PercentSalaryHike	11.0
TotalWorkingYears	10.0
TrainingTimesLastYear	2.0
YearsAtCompany	5.0
YearsSinceLastPromotion	0.0
YearsWithCurrManager	2.0

In [30]:

```
# Variance
df[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSal
    'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion
```

Out[30]:

```
Age                        8.341719e+01
DistanceFromHome           6.569144e+01
Education                  1.048438e+00
MonthlyIncome              2.215480e+09
NumCompaniesWorked         6.248686e+00
PercentSalaryHike          1.338907e+01
TotalWorkingYears          6.069855e+01
TrainingTimesLastYear      1.661465e+00
YearsAtCompany             3.751728e+01
YearsSinceLastPromotion    1.037935e+01
YearsWithCurrManager       1.272582e+01
dtype: float64
```

In [32]:

```
# Skewness
df[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSal
    'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion
```

Out[32]:

```
Age                         0.413005
DistanceFromHome            0.957466
Education                  -0.289484
MonthlyIncome               1.368884
NumCompaniesWorked          1.029836
PercentSalaryHike           0.820569
TotalWorkingYears           1.113489
TrainingTimesLastYear       0.552748
YearsAtCompany              1.763328
YearsSinceLastPromotion     1.982939
YearsWithCurrManager        0.832884
dtype: float64
```

In [33]:

```
# kurtosis
df[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSal
    'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion
```

Out[33]:

```
Age                        -0.405951
DistanceFromHome           -0.227045
Education                  -0.560569
MonthlyIncome               1.000232
NumCompaniesWorked          0.015084
PercentSalaryHike          -0.302638
TotalWorkingYears           0.909606
TrainingTimesLastYear       0.491149
YearsAtCompany              3.923864
YearsSinceLastPromotion     3.601761
YearsWithCurrManager        0.167949
dtype: float64
```

From above Description we have to see the continuous features that can be causes to Attrition i.e 'Age','DistanceFromHome','MonthlyIncome', 'PercentSalaryHike', 'TotalWorkingYears', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager'

from these features all of them showing positive Skewness

Age, DistanceFromHome, PercentSalaryHike are leptokurtic and all other are platykurtic.

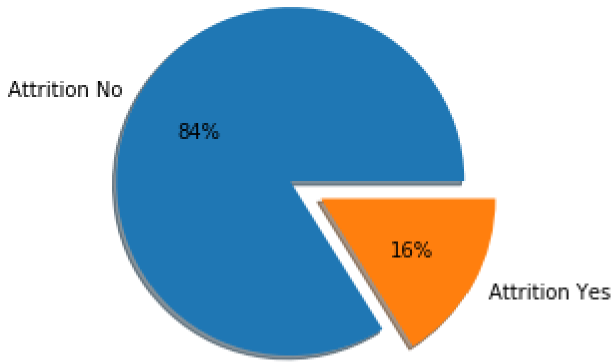
First let's check Attrition rate %

In [142]:

```
plt.pie(df['Attrition'].value_counts(), explode= (0,0.2),
autopct='%1.0f%%', shadow=True, labels= ['Attrition No', 'Attrition Yes'])
```

Out[142]:

```
([<matplotlib.patches.Wedge at 0x178f8488>,
  <matplotlib.patches.Wedge at 0x178f5948>],
 [Text(-0.9618916732177651, 0.5336332157899547, 'Attrition No'),
  Text(1.1367810683482678, -0.6306574368426737, 'Attrition Yes')],
 [Text(-0.5246681853915082, 0.29107266315815705, '84%'),
  Text(0.6995575805220109, -0.3880968842108762, '16%')])
```



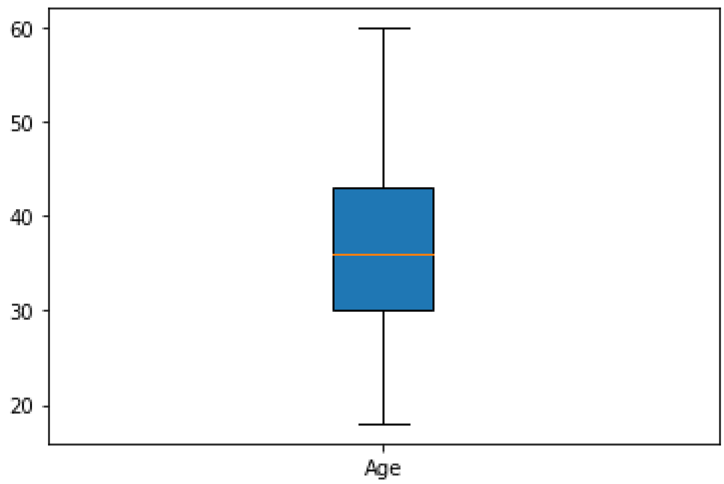
Let's check outliers through box plot

In [112]:

```
plt.boxplot(df.Age, patch_artist= True, labels = ['Age'])
```

Out[112]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x145a46c8>,
  <matplotlib.lines.Line2D at 0x145a4fc8>],
 'caps': [<matplotlib.lines.Line2D at 0x145a7808>,
  <matplotlib.lines.Line2D at 0x145a7fc8>],
 'boxes': [<matplotlib.patches.PathPatch at 0x145a4108>],
 'medians': [<matplotlib.lines.Line2D at 0x145a7f48>],
 'fliers': [<matplotlib.lines.Line2D at 0x145abf88>],
 'means': []}
```

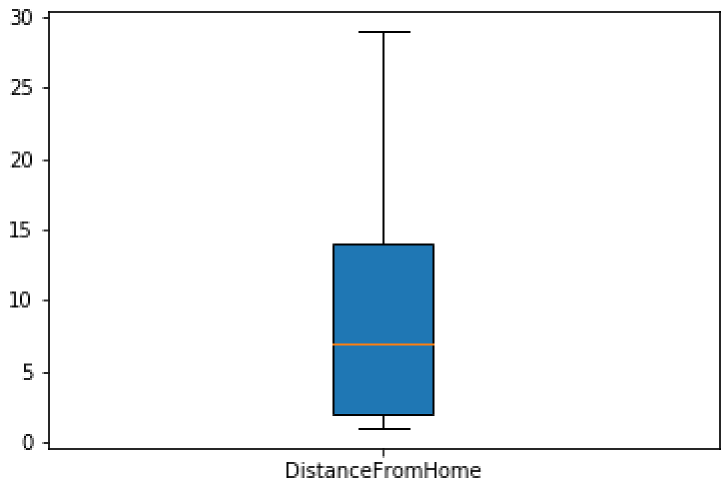


In [81]:

```
plt.boxplot(df.DistanceFromHome, patch_artist= True, labels = ['DistanceFromHome'])
```

Out[81]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0xf0a1048>,
  <matplotlib.lines.Line2D at 0xf8af248>],
 'caps': [<matplotlib.lines.Line2D at 0xf8aff48>,
  <matplotlib.lines.Line2D at 0xf8d3608>],
 'boxes': [<matplotlib.patches.PathPatch at 0xf489e08>],
 'medians': [<matplotlib.lines.Line2D at 0xf8c80c8>],
 'fliers': [<matplotlib.lines.Line2D at 0xf8cf648>],
 'means': []}
```

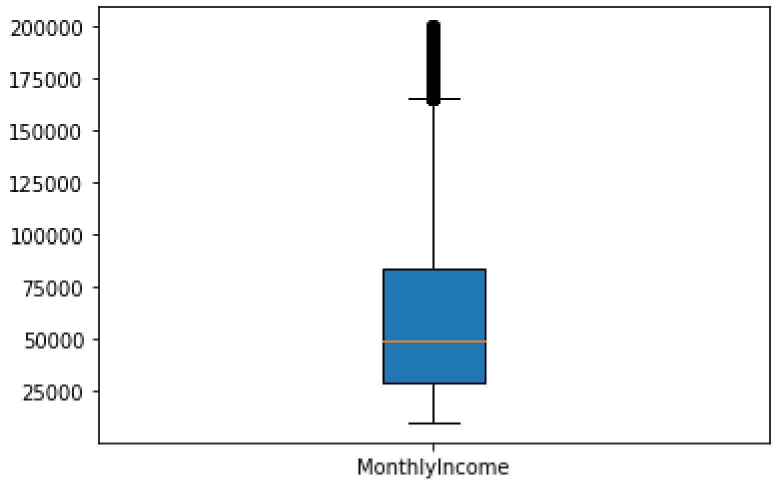


In [82]:

```
plt.boxplot(df.MonthlyIncome, patch_artist= True, labels = ['MonthlyIncome'])
```

Out[82]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0xcbdc5c8>,
  <matplotlib.lines.Line2D at 0xcbdcb08>],
 'caps': [<matplotlib.lines.Line2D at 0xf8e2c88>,
  <matplotlib.lines.Line2D at 0xf5d3fc8>],
 'boxes': [<matplotlib.patches.PathPatch at 0xf8e0bc8>],
 'medians': [<matplotlib.lines.Line2D at 0xf6de308>],
 'fliers': [<matplotlib.lines.Line2D at 0xeb8f7c8>],
 'means': []}
```

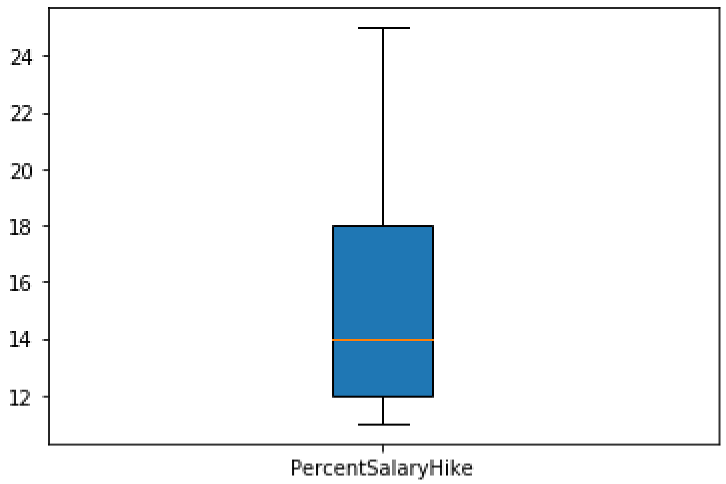


In [83]:

```
plt.boxplot(df.PercentSalaryHike, patch_artist= True, labels = ['PercentSalaryHike'])
```

Out[83]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0xef94d88>,
  <matplotlib.lines.Line2D at 0xed2ac48>],
 'caps': [<matplotlib.lines.Line2D at 0xf6baac8>,
  <matplotlib.lines.Line2D at 0xf92c288>],
 'boxes': [<matplotlib.patches.PathPatch at 0xf533888>],
 'medians': [<matplotlib.lines.Line2D at 0xfeb6348>],
 'fliers': [<matplotlib.lines.Line2D at 0xccd1548>],
 'means': []}
```



In [84]:

```
plt.boxplot(df.TotalWorkingYears, patch_artist= True, labels = ['TotalWorkingYears'])
```

Out[84]:

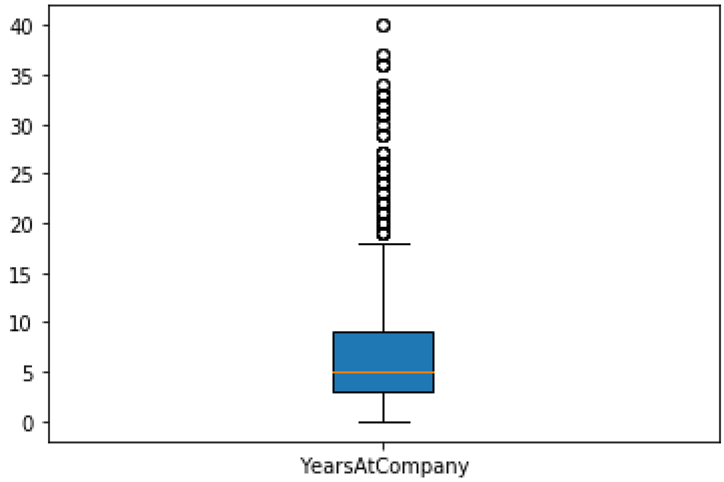
```
{'whiskers': [<matplotlib.lines.Line2D at 0xcbd3b88>,
  <matplotlib.lines.Line2D at 0xeb70788>],
 'caps': [<matplotlib.lines.Line2D at 0xeb70fc8>,
  <matplotlib.lines.Line2D at 0xeb4ba08>],
 'boxes': [<matplotlib.patches.PathPatch at 0xcc86b88>],
 'medians': [<matplotlib.lines.Line2D at 0xff90788>],
 'fliers': [<matplotlib.lines.Line2D at 0xdf75d88>],
 'means': []}
```

In [85]:

```
plt.boxplot(df.YearsAtCompany, patch_artist= True, labels = ['YearsAtCompany'])
```

Out[85]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0xff725c8>,
  <matplotlib.lines.Line2D at 0xeb56d88>],
 'caps': [<matplotlib.lines.Line2D at 0x101f9308>,
  <matplotlib.lines.Line2D at 0xec08d08>],
 'boxes': [<matplotlib.patches.PathPatch at 0xfe0ff88>],
 'medians': [<matplotlib.lines.Line2D at 0x1022d308>],
 'fliers': [<matplotlib.lines.Line2D at 0x1022d988>],
 'means': []}
```

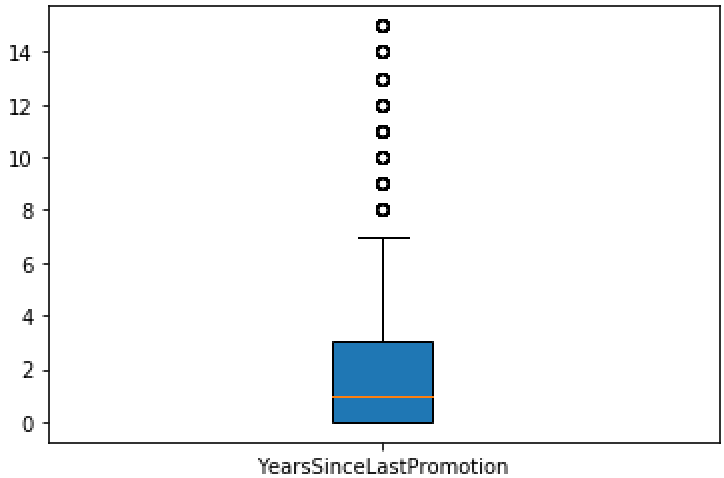


In [95]:

```
plt.boxplot(df.YearsSinceLastPromotion,patch_artist= True, labels = ['YearsSinceLastPromoti
```

Out[95]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x14035ec8>,
  <matplotlib.lines.Line2D at 0x14058b08>],
 'caps': [<matplotlib.lines.Line2D at 0x140ac448>,
  <matplotlib.lines.Line2D at 0x140bfec8>],
 'boxes': [<matplotlib.patches.PathPatch at 0x1407bd48>],
 'medians': [<matplotlib.lines.Line2D at 0x14073b88>],
 'fliers': [<matplotlib.lines.Line2D at 0x1406d108>],
 'means': []}
```

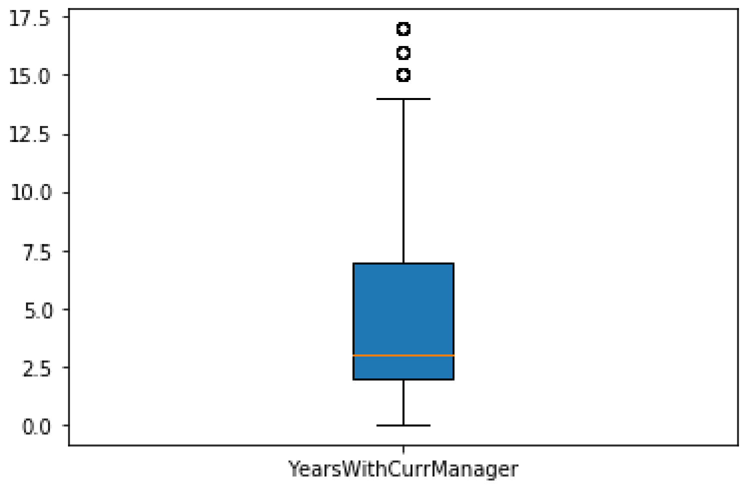


In [96]:

```
plt.boxplot(df.YearsWithCurrManager, patch_artist= True, labels = ['YearsWithCurrManager'])
```

Out[96]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x13fff208>,
  <matplotlib.lines.Line2D at 0x13fc7648>],
 'caps': [<matplotlib.lines.Line2D at 0x13fc7408>,
  <matplotlib.lines.Line2D at 0x1400ce48>],
 'boxes': [<matplotlib.patches.PathPatch at 0x13fff948>],
 'medians': [<matplotlib.lines.Line2D at 0x1402ce08>],
 'fliers': [<matplotlib.lines.Line2D at 0x1402c788>],
 'means': []}
```



From above boxplots we can see that Age, DistanceFromHome, PercentSalaryHike don't have Outliers others have outliers

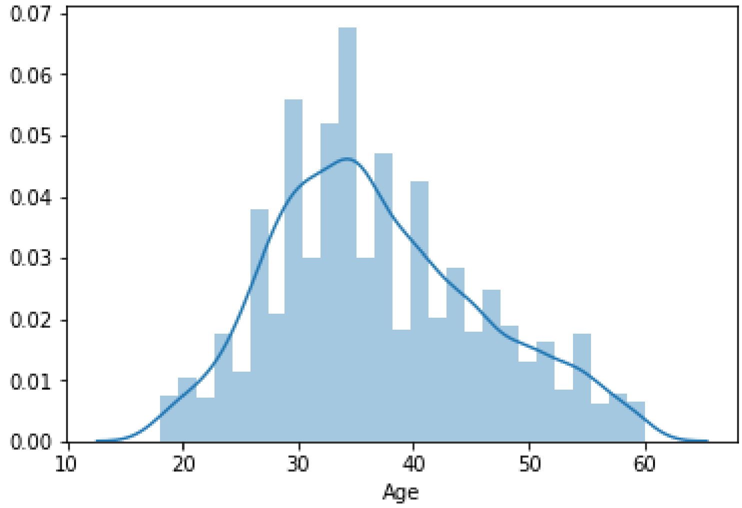
IQR of Age is 30 to 43 years (13 years)

In [109]:

```
sns.distplot(df.Age)
```

Out[109]:

<matplotlib.axes._subplots.AxesSubplot at 0x129237c8>

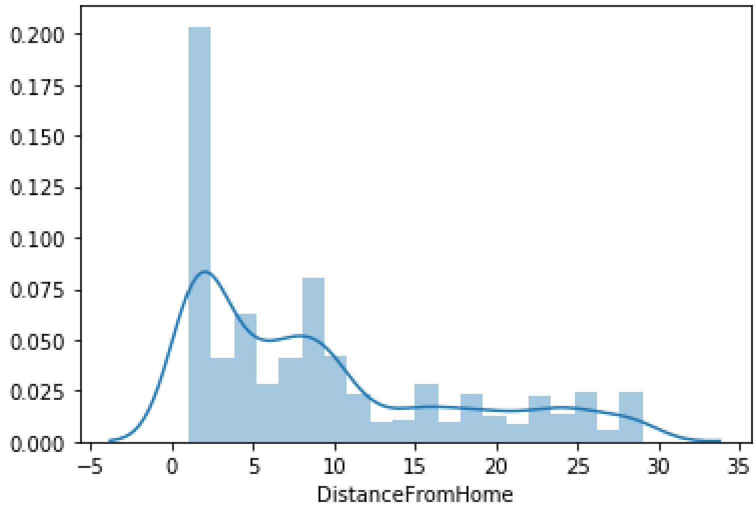


In [113]:

```
sns.distplot(df.DistanceFromHome)
```

Out[113]:

<matplotlib.axes._subplots.AxesSubplot at 0x145cd648>

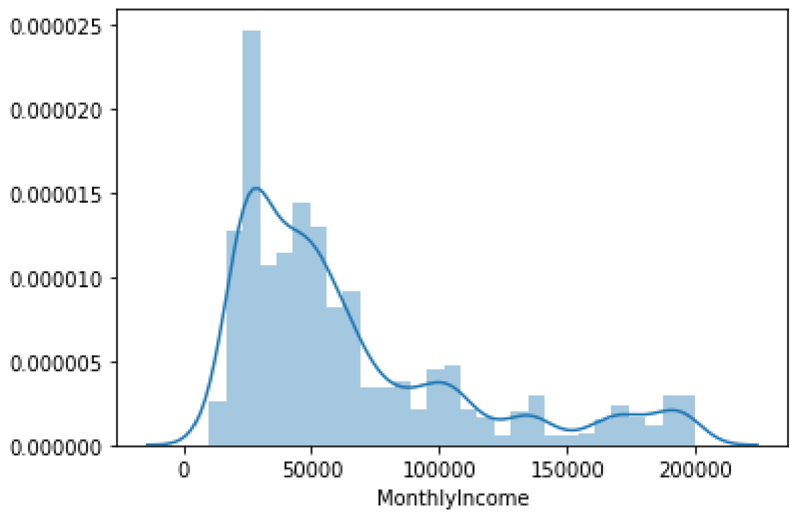


In [115]:

```
sns.distplot(df.MonthlyIncome)
```

Out[115]:

<matplotlib.axes._subplots.AxesSubplot at 0x1476d8c8>

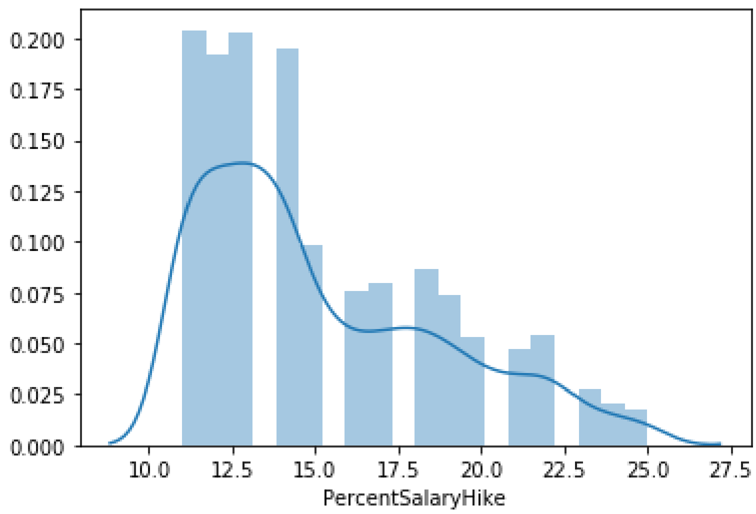


In [116]:

```
sns.distplot(df.PercentSalaryHike)
```

Out[116]:

<matplotlib.axes._subplots.AxesSubplot at 0x14790448>

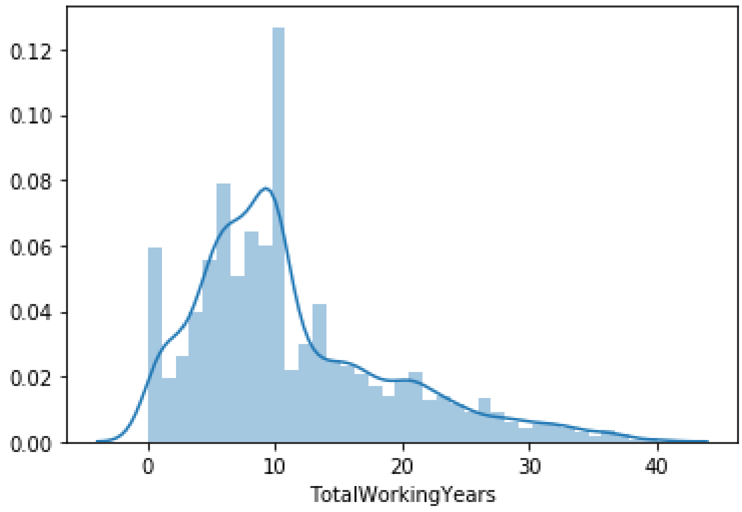


In [117]:

```
sns.distplot(df.TotalWorkingYears)
```

Out[117]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1580bf88>
```

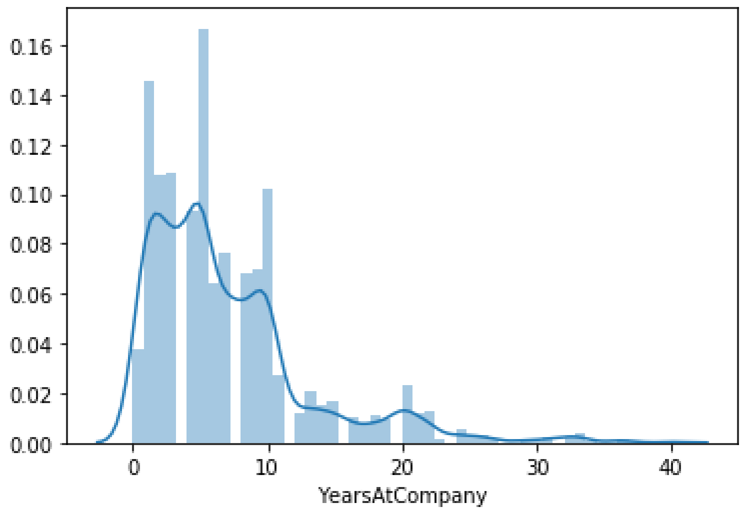


In [118]:

```
sns.distplot(df.YearsAtCompany)
```

Out[118]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1580be48>
```

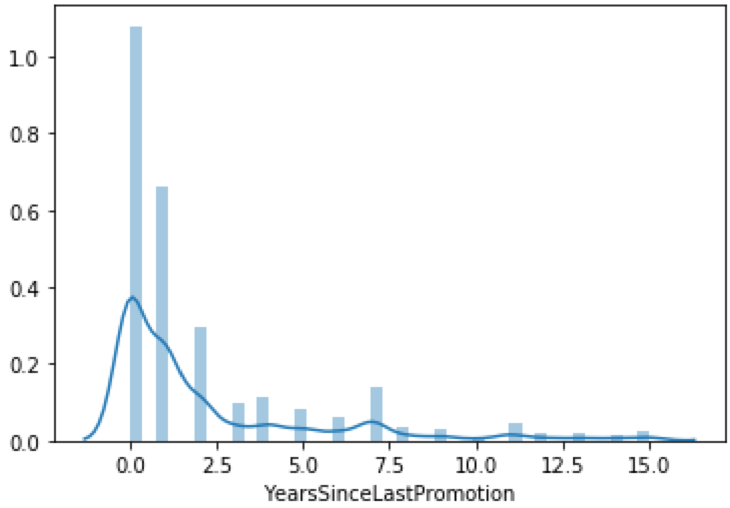


In [119]:

```
sns.distplot(df.YearsSinceLastPromotion)
```

Out[119]:

<matplotlib.axes._subplots.AxesSubplot at 0x15a1b888>

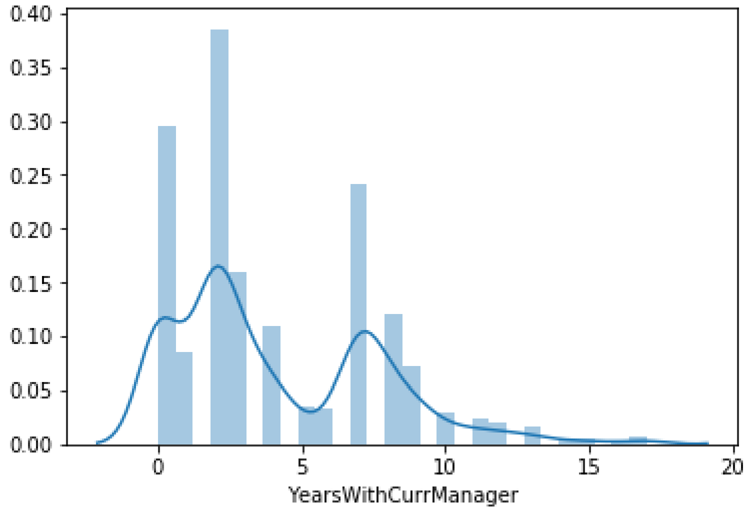


In [121]:

```
sns.distplot(df.YearsWithCurrManager)
```

Out[121]:

<matplotlib.axes._subplots.AxesSubplot at 0x16947a88>



From above distplots we can see Distribution & Skewness of individual variables

Let's see data distribution with Categorical Features

Categorical features which can have dependency on Attrition are: BusinessTravel, Department, Gender, JobLevel, JobRole, MaritalStatus

In [123]:

```
df.columns
```

Out[123]:

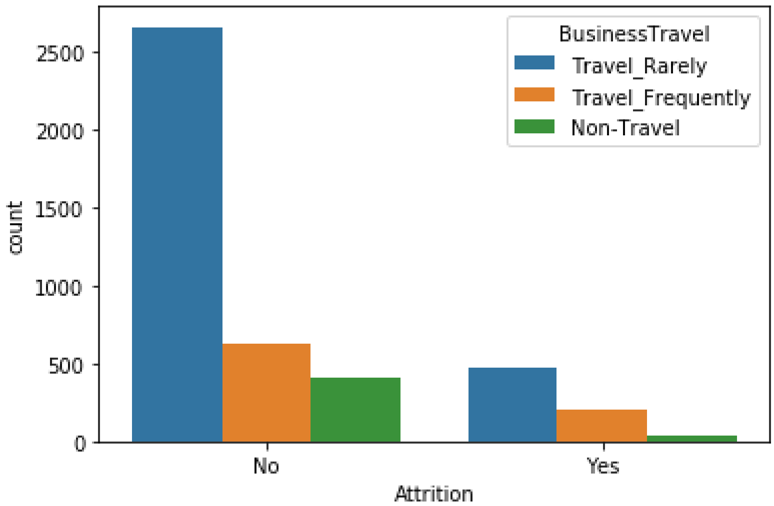
```
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHom
e',
       'Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole',
       'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',
       'PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears',
       'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

In [129]:

```
sns.countplot('Attrition', data= df, hue= 'BusinessTravel')
```

Out[129]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16dd1508>
```

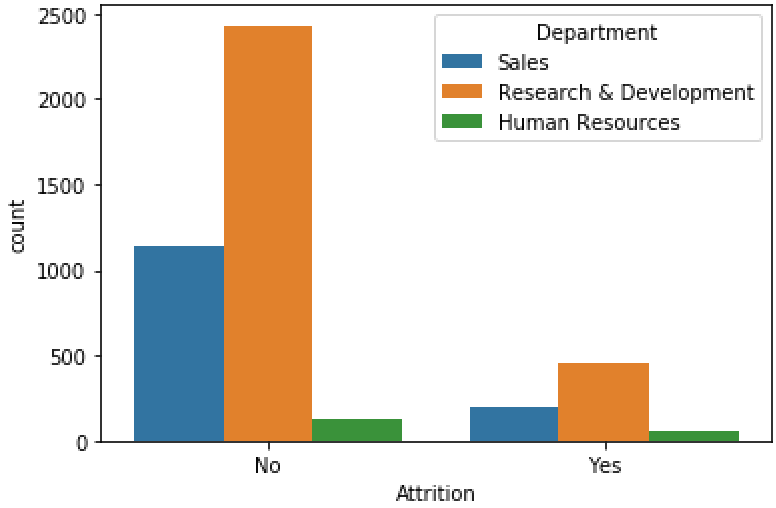


In [135]:

```
sns.countplot('Attrition', data= df, hue= 'Department')
```

Out[135]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16dfe808>
```

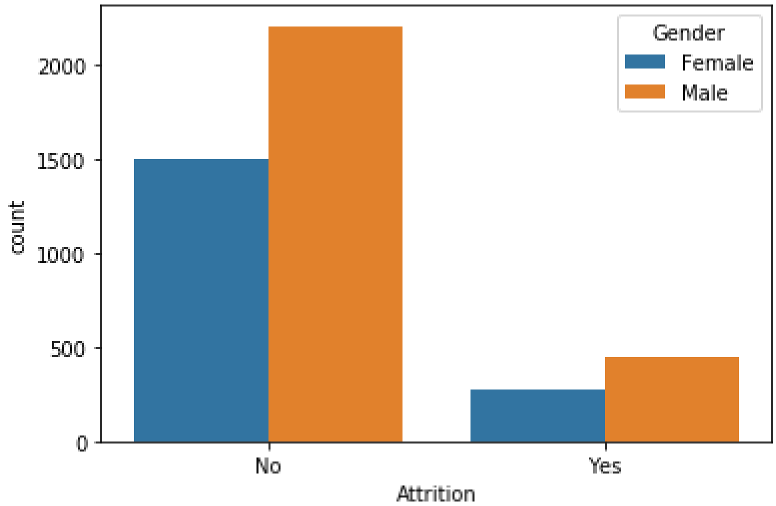


In [136]:

```
sns.countplot('Attrition', data= df, hue= 'Gender')
```

Out[136]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x170f3fc8>
```

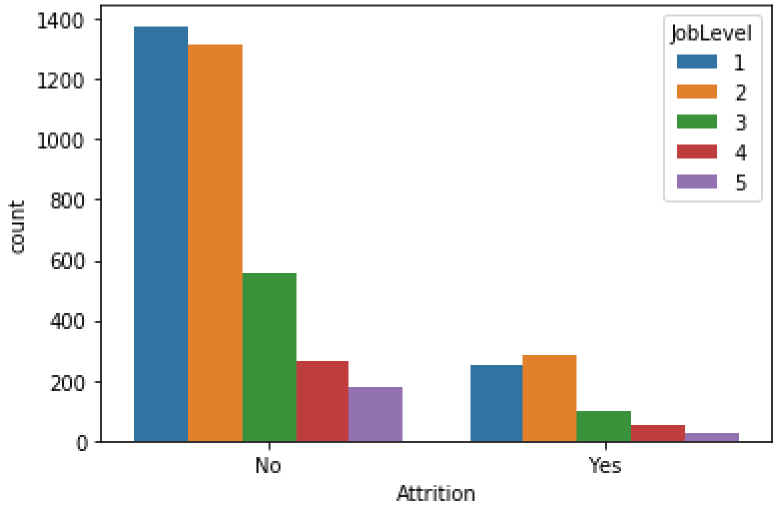


In [137]:

```
sns.countplot('Attrition', data= df, hue= 'JobLevel')
```

Out[137]:

<matplotlib.axes._subplots.AxesSubplot at 0x17169ec8>

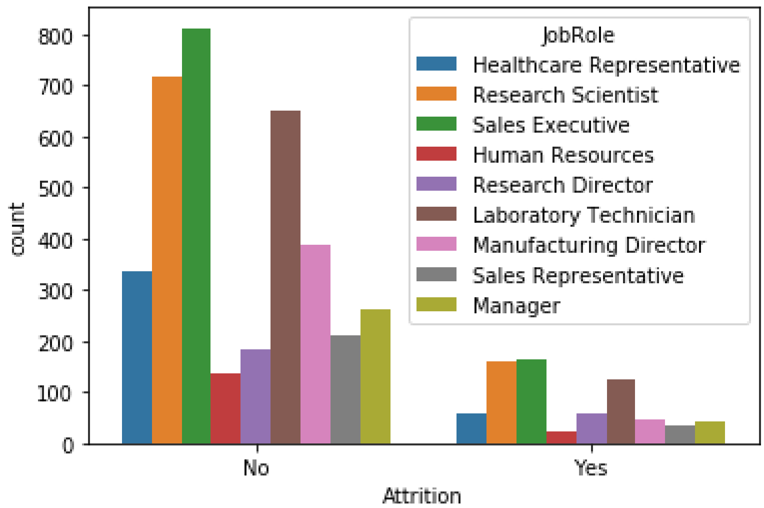


In [138]:

```
sns.countplot('Attrition', data= df, hue= 'JobRole')
```

Out[138]:

<matplotlib.axes._subplots.AxesSubplot at 0x171cc788>

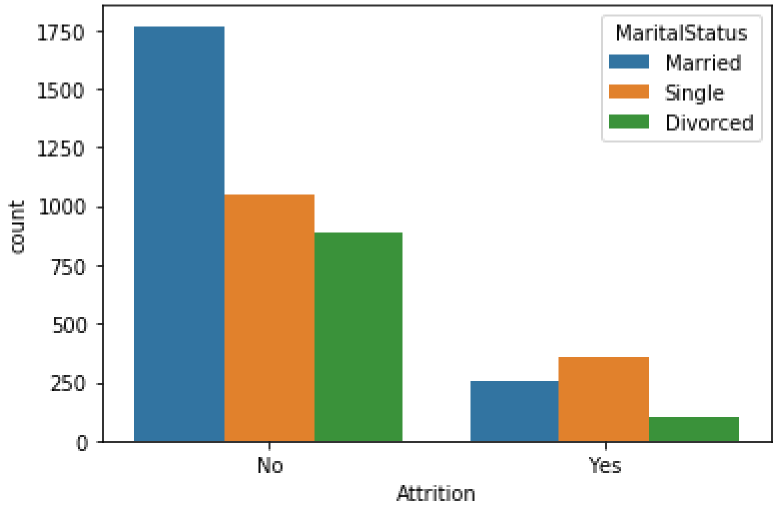


In [139]:

```
sns.countplot('Attrition', data= df, hue= 'MaritalStatus')
```

Out[139]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x17253108>
```



From above plots we can make some conclusions:

1. The employee who Rarely travels have high attrition rate

2. Research & Development Department has high Attrition rate

3. Male Employees have high Attrition rate as compared to Female Employees

4. Job level 1 & 2 has high Attrition rate

5. Research & Development, Sales Executives & Laboratory Technitions has high Attrition rate

6. Single employees has high Attrition rate than married and Divorced Employees

In [143]:

```
#Convert all the Categorical data into numerical data
# get unique values from each categorical feature
print(df['BusinessTravel'].unique())
print(df['EducationField'].unique())
print(df['Gender'].unique())
print(df['Department'].unique())
print(df['JobRole'].unique())
print(df['MaritalStatus'].unique())
```

```
['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']
['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'
 'Human Resources']
['Female' 'Male']
['Sales' 'Research & Development' 'Human Resources']
['Healthcare Representative' 'Research Scientist' 'Sales Executive'
 'Human Resources' 'Research Director' 'Laboratory Technician'
 'Manufacturing Director' 'Sales Representative' 'Manager']
['Married' 'Single' 'Divorced']
```

In [145]:

```
# sklearn used to convert data to numerical
from sklearn.preprocessing import LabelEncoder
cat_x = LabelEncoder()

df['BusinessTravel'] = cat_x.fit_transform(df['BusinessTravel'])
df['Department'] = cat_x.fit_transform(df['Department'])
df['EducationField'] = cat_x.fit_transform(df['EducationField'])
df['Gender'] = cat_x.fit_transform(df['Gender'])
df['JobRole'] = cat_x.fit_transform(df['JobRole'])
df['MaritalStatus'] = cat_x.fit_transform(df['MaritalStatus'])
df['Attrition'] = cat_x.fit_transform(df['Attrition'])
```

In [146]:

```
# checking categorical values again it's changed to numerical
# No = 0, Yes = 1
df.head()
```

Out[146]:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField
0	51	0	2	2	6	2	1
1	31	1	1	1	10	1	1
2	32	0	1	1	17	4	4
3	38	0	0	1	2	5	1
4	32	0	2	1	10	1	3

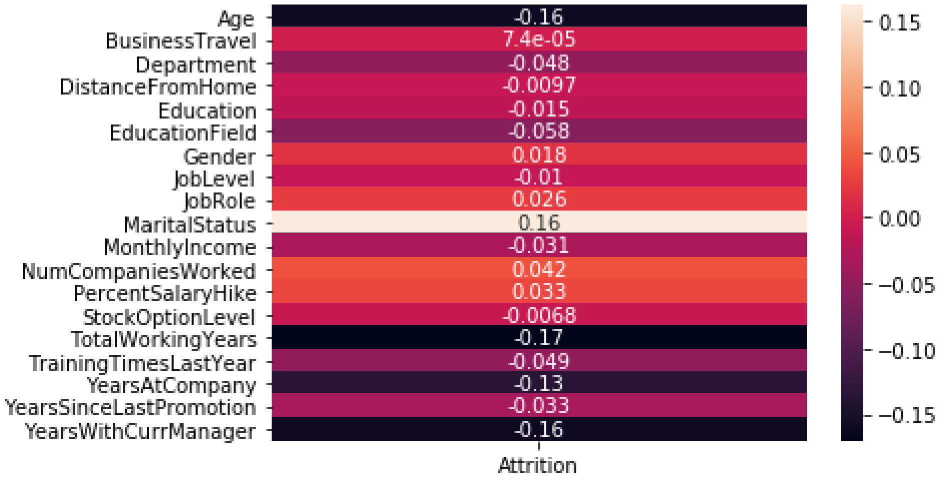
Correlations of all independent features with dependent feature Attrition

In [150]:

```
# Correlation of all columns with Attrition
sns.heatmap(df.corr().iloc[:, 1:2].drop('Attrition', axis= 0), annot = True)
```

Out[150]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x178e7a88>
```



In [151]:

```
# Function to check Accepting or Rejecting Null Hypothesis
def check(Attrition, b, c):
    print('\nNull Hypothesis: There is no Significant Correlation between Attrition and', b
    print('Alternate Hypothesis: There is Significant Correlation between Attrition and', b
    from scipy.stats import pearsonr
    stats, p = pearsonr(df.Attrition, c)
    print('\nCorrelation:', stats, 'P Value:', p,'\n')
    if p < 0.05:
        print('P-Value < 0.05 hence Null Hypothesis is rejected, Accepting Alternate Hypoth
        if stats > 0:
            print('There is Positive Correlation between Attrition and', b)
        else:
            print('There is Negative Correlation between Attrition and', b)
    else:
        print("P-Value >= 0.05 hence Null hypothesis is Accepted")
        print('There is no Significant Correlation between Attrition and', b)
    print('-----------------------------------------------------------------------------
```

In [152]:

```
# Correlation & P - Value of relationship with Attrition and other features
check('Attrition', 'Age', df.Age)
check('Attrition', 'BusinessTravel', df.BusinessTravel)
check('Attrition', 'DistanceFromHome', df.DistanceFromHome)
check('Attrition', 'Education', df.Education)
check('Attrition', 'EducationField', df.EducationField)
check('Attrition', 'Gender', df.Gender)
check('Attrition', 'JobLevel', df.JobLevel)
check('Attrition', 'JobRole', df.JobRole)
check('Attrition', 'MaritalStatus', df.MaritalStatus)
check('Attrition', 'MonthlyIncome', df.MonthlyIncome)
check('Attrition', 'NumCompaniesWorked', df.NumCompaniesWorked)
check('Attrition', 'PercentSalaryHike', df.PercentSalaryHike)
check('Attrition', 'StockOptionLevel', df.StockOptionLevel)
check('Attrition', 'TotalWorkingYears', df.TotalWorkingYears)
check('Attrition', 'TrainingTimesLastYear', df.TrainingTimesLastYear)
check('Attrition', 'YearsAtCompany', df.YearsAtCompany)
check('Attrition', 'YearsSinceLastPromotion', df.YearsSinceLastPromotion)
check('Attrition', 'YearsWithCurrManager', df.YearsWithCurrManager)
```

```
Null Hypothesis: There is no Significant Correlation between Attrition and
DistanceFromHome
Alternate Hypothesis: There is Significant Correlation between Attrition a
nd DistanceFromHome

Correlation: -0.009730141010179674 P Value: 0.5182860428050771

P-Value >= 0.05 hence Null hypothesis is Accepted
There is no Significant Correlation between Attrition and DistanceFromHome
--------------------------------------------------------------------------
---------------------------------------


Null Hypothesis: There is no Significant Correlation between Attrition and
Education
Alternate Hypothesis: There is Significant Correlation between Attrition a
nd Education

Correlation: -0.015111167710968713 P Value: 0.3157293177118575
```

From above calculations we can have some conclusion based on correlation & P value

Features don't have relationship with Attrition : BusinessTravel, DistanceFromHome, Education, Gender, JobLevel, JobRole, StockOptionLevel

Features have relationship with Attrition and have Positive Correlation: MaritalStatus, NumCompaniesWorked, PercentSalaryHike,

Features have relationship with Attrition and have Negative Correlation: Age, EducationField, MonthlyIncome, TotalWorkingYears, TrainingTimesLastYear, YearsAtCompany, YearsSinceLastPromotion, YearsWithCurrManager

Statistical tests:

As we seen in distplots none of features Normally Distributed so we can perform only Non-Parametric tests.

Dependent variable is Attrition and that is Categorical so in Non-Parametric we can perform below Tests:

1. Mann-Whitney Test - 1 Dependent categorical variable and other continuous varibles

2. CHI Square test - Only for Categorical Variables

1. Mann-Whitney Test:

For Mann-Whitney Test we need to separate data as Attrition Yes & Attrition No

In [157]:

```
# attrition = Yes and no seprate
att_yes = df[df['Attrition']== 1]
att_no = df[df['Attrition']== 0]
```

In [158]:

```
att_yes.head()
```

Out[158]:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField
1	31	1	1	1	10	1	1
6	28	1	2	1	11	2	3
13	47	1	0	1	1	1	3
28	44	1	1	1	1	2	3
30	26	1	2	1	4	3	3

In [159]:

```
att_no.head()
```

Out[159]:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField
0	51	0	2	2	6	2	1
2	32	0	1	1	17	4	4
3	38	0	0	1	2	5	1
4	32	0	2	1	10	1	3
5	46	0	2	1	8	3	1

Continuous Variables we should check with Attrition as per above correlation results : Age, MonthlyIncome, TotalWorkingYears, TrainingTimesLastYear, YearsAtCompany, YearsSinceLastPromotion, YearsWithCurrManager, NumCompaniesWorked, PercentSalaryHike

In [160]:

```
# defining function for ManWhitney tests
def manwhitney(stats, p, b):
    print('\nH0 = There is no significant difference between Attrition_yes with',b, 'and At
    print('H1 = There is significant difference between Attrition_yes with',b, 'and Attriti
    print(stats, 'P Value:', p,'\n')
    if p < 0.05:
        print('P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis')
    else:
        print("P-Value >= 0.05 hence H0 Accepted")
    print('--------------------------------------------------------------------------------
```

In [161]:

```
# importing scipy module
from scipy.stats import mannwhitneyu
```

In [166]:

```
stats, p = mannwhitneyu(att_yes.Age, att_no.Age)
manwhitney(stats, p, 'Age')
stats, p = mannwhitneyu(att_yes.MonthlyIncome, att_no.MonthlyIncome)
manwhitney(stats, p, 'MonthlyIncome')
stats, p = mannwhitneyu(att_yes.NumCompaniesWorked, att_no.NumCompaniesWorked)
manwhitney(stats, p, 'NumCompaniesWorked')
stats, p = mannwhitneyu(att_yes.PercentSalaryHike, att_no.PercentSalaryHike)
manwhitney(stats, p, 'PercentSalaryHike')
stats, p = mannwhitneyu(att_yes.TotalWorkingYears, att_no.TotalWorkingYears)
manwhitney(stats, p, 'TotalWorkingYears')
stats, p = mannwhitneyu(att_yes.TrainingTimesLastYear, att_no.TrainingTimesLastYear)
manwhitney(stats, p, 'TrainingTimesLastYear')
stats, p = mannwhitneyu(att_yes.YearsAtCompany, att_no.YearsAtCompany)
manwhitney(stats, p, 'YearsAtCompany')
stats, p = mannwhitneyu(att_yes.YearsSinceLastPromotion, att_no.YearsSinceLastPromotion)
manwhitney(stats, p, 'YearsSinceLastPromotion')
stats, p = mannwhitneyu(att_yes.YearsWithCurrManager, att_no.YearsWithCurrManager)
manwhitney(stats, p, 'YearsWithCurrManager')
```

```
H0 = There is no significant difference between Attrition_yes with Age and A
ttrition_No with Age
H1 = There is significant difference between Attrition_yes with Age and Attr
ition_No with Age

961731.0 P Value: 2.9951588479067175e-30

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
----------------------------------------------------------------------------
-------------------------------------

H0 = There is no significant difference between Attrition_yes with MonthlyIn
come and Attrition_No with MonthlyIncome
H1 = There is significant difference between Attrition_yes with MonthlyIncom
e and Attrition_No with MonthlyIncome

1264900.5 P Value: 0.053577283839938566

P-Value >= 0.05 hence H0 Accepted
----------------------------------------------------------------------------
-------------------------------------

H0 = There is no significant difference between Attrition_yes with NumCompan
iesWorked and Attrition_No with NumCompaniesWorked
H1 = There is significant difference between Attrition_yes with NumCompanies
Worked and Attrition_No with NumCompaniesWorked

1259144.0 P Value: 0.03266173775282211

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
----------------------------------------------------------------------------
-------------------------------------

H0 = There is no significant difference between Attrition_yes with PercentSa
laryHike and Attrition_No with PercentSalaryHike
H1 = There is significant difference between Attrition_yes with PercentSalar
yHike and Attrition_No with PercentSalaryHike
```

```
1250640.0 P Value: 0.018660129917539733

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
--------------------------------------------------------------------------------
-------------------------------------

H0 = There is no significant difference between Attrition_yes with TotalWork
ingYears and Attrition_No with TotalWorkingYears
H1 = There is significant difference between Attrition_yes with TotalWorking
Years and Attrition_No with TotalWorkingYears

907502.5 P Value: 1.0203529765342384e-39

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
--------------------------------------------------------------------------------
-------------------------------------

H0 = There is no significant difference between Attrition_yes with TrainingT
imesLastYear and Attrition_No with TrainingTimesLastYear
H1 = There is significant difference between Attrition_yes with TrainingTime
sLastYear and Attrition_No with TrainingTimesLastYear

1238940.0 P Value: 0.005167954938699059

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
--------------------------------------------------------------------------------
-------------------------------------

H0 = There is no significant difference between Attrition_yes with YearsAtCo
mpany and Attrition_No with YearsAtCompany
H1 = There is significant difference between Attrition_yes with YearsAtCompa
ny and Attrition_No with YearsAtCompany

923238.0 P Value: 6.047598261692858e-37

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
--------------------------------------------------------------------------------
-------------------------------------

H0 = There is no significant difference between Attrition_yes with YearsSinc
eLastPromotion and Attrition_No with YearsSinceLastPromotion
H1 = There is significant difference between Attrition_yes with YearsSinceLa
stPromotion and Attrition_No with YearsSinceLastPromotion

1209366.0 P Value: 0.0002021180346719736

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
--------------------------------------------------------------------------------
-------------------------------------

H0 = There is no significant difference between Attrition_yes with YearsWith
CurrManager and Attrition_No with YearsWithCurrManager
H1 = There is significant difference between Attrition_yes with YearsWithCur
rManager and Attrition_No with YearsWithCurrManager

957253.5 P Value: 1.2365483142169853e-31

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
--------------------------------------------------------------------------------
-------------------------------------
```

From above Mann-Whitney Tests we can have conclusion

There is no significant difference between Attrition Yes Monthly Income & Attrition No Monthly Income

There is significant difference between Attrition Yes & Attrition No with following Variables: Age, TotalWorkingYears, TrainingTimesLastYear, YearsAtCompany, YearsSinceLastPromotion, YearsWithCurrManager, NumCompaniesWorked, PercentSalaryHike

2. CHI Square Test:

For CHI Square test we are checking dependency of Categorical Variables with Attrition

Categorical Variables we should check based on correlation results: BusinessTravel , EducationField , Gender , Department , JobRole , MaritalStatus , JobLevel , StockOptionLevel

In [167]:

```
# defining function for CHI Square tests
def chi2(stats, p, b):
    print('\nHo= There is no dependency betweem Attrition and', b)
    print('H1= There is dependency betweem Attrition and', b, '\n')
    print(chitable,'\n')
    print(stats, 'P Value:', p,'\n')
    if p < 0.05:
        print('P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis')
    else:
        print("P-Value >= 0.05 hence H0 Accepted")
    print('--------------------------------------------------------------------------------
```

In [168]:

```
# impoerting scipy module
from scipy.stats import chi2_contingency
```

In [178]:

```
chitable = pd.crosstab(df.Attrition, df.BusinessTravel)
stats, p, dof, expected = chi2_contingency(chitable)
chi2(stats, p, 'BusinessTravel')
chitable = pd.crosstab(df.Attrition, df.EducationField)
stats, p, dof, expected = chi2_contingency(chitable)
chi2(stats, p, 'EducationField')
chitable = pd.crosstab(df.Attrition, df.Gender)
stats, p, dof, expected = chi2_contingency(chitable)
chi2(stats, p, 'Gender')
chitable = pd.crosstab(df.Attrition, df.Department)
stats, p, dof, expected = chi2_contingency(chitable)
chi2(stats, p, 'Department')
chitable = pd.crosstab(df.Attrition, df.JobRole)
stats, p, dof, expected = chi2_contingency(chitable)
chi2(stats, p, 'JobRole')
chitable = pd.crosstab(df.Attrition, df.MaritalStatus)
stats, p, dof, expected = chi2_contingency(chitable)
chi2(stats, p, 'MaritalStatus')
chitable = pd.crosstab(df.Attrition, df.JobLevel)
stats, p, dof, expected = chi2_contingency(chitable)
chi2(stats, p, 'JobLevel')
chitable = pd.crosstab(df.Attrition, df.StockOptionLevel)
stats, p, dof, expected = chi2_contingency(chitable)
chi2(stats, p, 'StockOptionLevel')
```

```
Ho= There is no dependency betweem Attrition and BusinessTravel
H1= There is dependency betweem Attrition and BusinessTravel

BusinessTravel    0    1     2
Attrition
0               414  624  2661
1                36  207   468

72.54724105696552 P Value: 1.764276972983189e-16

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
---------------------------------------------------------------------------
---------------------------------------

Ho= There is no dependency betweem Attrition and EducationField
H1= There is dependency betweem Attrition and EducationField

EducationField   0     1    2     3    4    5
Attrition
0               48  1515  402  1167  216  351
1               33   303   75   225   30   45

46.194921001730584 P Value: 8.288917469574179e-09

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
---------------------------------------------------------------------------
---------------------------------------

Ho= There is no dependency betweem Attrition and Gender
H1= There is dependency betweem Attrition and Gender

Gender        0     1
```

```
Attrition
0          1494  2205
1           270   441

1.349904410246582 P Value: 0.24529482862926827

P-Value >= 0.05 hence H0 Accepted
--------------------------------------------------------------------------
---------------------------------------

Ho= There is no dependency betweem Attrition and Department
H1= There is dependency betweem Attrition and Department

Department    0     1     2
Attrition
0           132  2430  1137
1            57   453   201

29.090274924488266 P Value: 4.820888218170406e-07

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
--------------------------------------------------------------------------
---------------------------------------

Ho= There is no dependency betweem Attrition and JobRole
H1= There is dependency betweem Attrition and JobRole

JobRole      0    1    2    3    4    5    6    7    8
Attrition
0          336  135  651  264  387  183  717  813  213
1           57   21  126   42   48   57  159  165   36

25.116313674604072 P Value: 0.001485544744815264

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
--------------------------------------------------------------------------
---------------------------------------

Ho= There is no dependency betweem Attrition and MaritalStatus
H1= There is dependency betweem Attrition and MaritalStatus

MaritalStatus    0     1     2
Attrition
0              882  1767  1050
1               99   252   360

138.49102962254608 P Value: 8.45385940605786e-31

P-Value < 0.05 hence H0 rejected, Accepting H1 Hypothesis
--------------------------------------------------------------------------
---------------------------------------

Ho= There is no dependency betweem Attrition and JobLevel
H1= There is dependency betweem Attrition and JobLevel

JobLevel      1     2    3    4    5
Attrition
0          1377  1317  558  267  180
1           252   285   96   51   27

6.2691759264759925 P Value: 0.1799276801337184
```

```
P-Value >= 0.05 hence H0 Accepted
---------------------------------------------------------------------------
---------------------------------------

Ho= There is no dependency betweem Attrition and StockOptionLevel
H1= There is dependency betweem Attrition and StockOptionLevel

StockOptionLevel     0     1    2    3
Attrition
0                 1575  1518  390  216
1                  318   270   84   39

3.046265305068262 P Value: 0.38454683657380506

P-Value >= 0.05 hence H0 Accepted
---------------------------------------------------------------------------
---------------------------------------
```

From above performed CHI Square tests we can have following conclusions

Variables have dependency with Attrition are: BusinessTravel , EducationField, Department , JobRole , MaritalStatus

Variables don't have dependency with Attrition are: Gender, JobLevel , StockOptionLevel