## Attrition Analysis

#### Import pandas, numpy, seaborn, matplotlib.pyplot packages

In [1]:

**import** numpy **as** np

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**%**matplotlib inline

**import** seaborn **as** sns

**from** warnings **import** filterwarnings

filterwarnings('ignore')

#### Importing Attrition Dataset

In [2]:

df **=** pd.read\_csv('general\_data.csv')

* **Shape of Dataset**

In [3]:

df.shape

Out[3]:

(4410, 24)

* **The dataset has total 4410 rows**
* **View the first 5 records using Head method**

In [11]:

df.head().T

Out[11]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** | **4** |
| **Age** | 51 | 31 | 32 | 38 | 32 |
| **Attrition** | No | Yes | No | No | No |
| **BusinessTravel** | Travel\_Rarely | Travel\_Frequently | Travel\_Frequently | Non-Travel | Travel\_Rarely |
| **Department** | Sales | Research & Development | Research & Development | Research & Development | Research & Development |
| **DistanceFromHome** | 6 | 10 | 17 | 2 | 10 |
| **Education** | 2 | 1 | 4 | 5 | 1 |
| **EducationField** | Life Sciences | Life Sciences | Other | Life Sciences | Medical |
| **EmployeeCount** | 1 | 1 | 1 | 1 | 1 |
| **EmployeeID** | 1 | 2 | 3 | 4 | 5 |
| **Gender** | Female | Female | Male | Male | Male |
| **JobLevel** | 1 | 1 | 4 | 3 | 1 |
| **JobRole** | Healthcare Representative | Research Scientist | Sales Executive | Human Resources | Sales Executive |
| **MaritalStatus** | Married | Single | Married | Married | Single |
| **MonthlyIncome** | 131160 | 41890 | 193280 | 83210 | 23420 |
| **NumCompaniesWorked** | 1 | 0 | 1 | 3 | 4 |
| **Over18** | Y | Y | Y | Y | Y |
| **PercentSalaryHike** | 11 | 23 | 15 | 11 | 12 |
| **StandardHours** | 8 | 8 | 8 | 8 | 8 |
| **StockOptionLevel** | 0 | 1 | 3 | 3 | 2 |
| **TotalWorkingYears** | 1 | 6 | 5 | 13 | 9 |
| **TrainingTimesLastYear** | 6 | 3 | 2 | 5 | 2 |
| **YearsAtCompany** | 1 | 5 | 5 | 8 | 6 |
| **YearsSinceLastPromotion** | 0 | 1 | 0 | 7 | 0 |
| **YearsWithCurrManager** | 0 | 4 | 3 | 5 | 4 |

* **Columns Name**

In [6]:

df.columns

Out[6]:

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

* **Checking Information of Dataset**

In [7]:

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

* Dataset has 2 Float columns, 14 integer columns and 8 object (string) Columns
* **Checking data summary**

In [15]:

df.describe().T

Out[15]:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **Age** | 4410.0 | 36.923810 | 9.133301 | 18.0 | 30.00 | 36.0 | 43.00 | 60.0 |
| **DistanceFromHome** | 4410.0 | 9.192517 | 8.105026 | 1.0 | 2.00 | 7.0 | 14.00 | 29.0 |
| **Education** | 4410.0 | 2.912925 | 1.023933 | 1.0 | 2.00 | 3.0 | 4.00 | 5.0 |
| **EmployeeCount** | 4410.0 | 1.000000 | 0.000000 | 1.0 | 1.00 | 1.0 | 1.00 | 1.0 |
| **EmployeeID** | 4410.0 | 2205.500000 | 1273.201673 | 1.0 | 1103.25 | 2205.5 | 3307.75 | 4410.0 |
| **JobLevel** | 4410.0 | 2.063946 | 1.106689 | 1.0 | 1.00 | 2.0 | 3.00 | 5.0 |
| **MonthlyIncome** | 4410.0 | 65029.312925 | 47068.888559 | 10090.0 | 29110.00 | 49190.0 | 83800.00 | 199990.0 |
| **NumCompaniesWorked** | 4391.0 | 2.694830 | 2.498887 | 0.0 | 1.00 | 2.0 | 4.00 | 9.0 |
| **PercentSalaryHike** | 4410.0 | 15.209524 | 3.659108 | 11.0 | 12.00 | 14.0 | 18.00 | 25.0 |
| **StandardHours** | 4410.0 | 8.000000 | 0.000000 | 8.0 | 8.00 | 8.0 | 8.00 | 8.0 |
| **StockOptionLevel** | 4410.0 | 0.793878 | 0.851883 | 0.0 | 0.00 | 1.0 | 1.00 | 3.0 |
| **TotalWorkingYears** | 4401.0 | 11.279936 | 7.782222 | 0.0 | 6.00 | 10.0 | 15.00 | 40.0 |
| **TrainingTimesLastYear** | 4410.0 | 2.799320 | 1.288978 | 0.0 | 2.00 | 3.0 | 3.00 | 6.0 |
| **YearsAtCompany** | 4410.0 | 7.008163 | 6.125135 | 0.0 | 3.00 | 5.0 | 9.00 | 40.0 |
| **YearsSinceLastPromotion** | 4410.0 | 2.187755 | 3.221699 | 0.0 | 0.00 | 1.0 | 3.00 | 15.0 |
| **YearsWithCurrManager** | 4410.0 | 4.123129 | 3.567327 | 0.0 | 2.00 | 3.0 | 7.00 | 17.0 |

In [8]:

* From the above description : on seeing mean and meadian

We conclude that **EmployeeID , StandardHours are Bell Shaped Curve ( Mean = Median)**

* Checking Median of all the columns

In [10]:

df[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].median()

Out[10]:

Age 36.0  
DistanceFromHome 7.0  
Education 3.0  
EmployeeCount 1.0  
EmployeeID 2205.5  
JobLevel 2.0  
MonthlyIncome 49190.0  
NumCompaniesWorked 2.0  
PercentSalaryHike 14.0  
StandardHours 8.0  
StockOptionLevel 1.0  
TotalWorkingYears 10.0  
TrainingTimesLastYear 3.0  
YearsAtCompany 5.0  
YearsSinceLastPromotion 1.0  
YearsWithCurrManager 3.0  
dtype: float64

* Checking Mean of All the Columns

In [11]:

df[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].mean()

Out[11]:

Age 36.923810  
DistanceFromHome 9.192517  
Education 2.912925  
EmployeeCount 1.000000  
EmployeeID 2205.500000  
JobLevel 2.063946  
MonthlyIncome 65029.312925  
NumCompaniesWorked 2.694830  
PercentSalaryHike 15.209524  
StandardHours 8.000000  
StockOptionLevel 0.793878  
TotalWorkingYears 11.279936  
TrainingTimesLastYear 2.799320  
YearsAtCompany 7.008163  
YearsSinceLastPromotion 2.187755  
YearsWithCurrManager 4.123129  
dtype: float64

In [12]:

df[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].skew()

Out[12]:

Age 0.413005  
DistanceFromHome 0.957466  
Education -0.289484  
EmployeeCount 0.000000  
EmployeeID 0.000000  
JobLevel 1.024703  
MonthlyIncome 1.368884  
NumCompaniesWorked 1.026767  
PercentSalaryHike 0.820569  
StandardHours 0.000000  
StockOptionLevel 0.968321  
TotalWorkingYears 1.116832  
TrainingTimesLastYear 0.552748  
YearsAtCompany 1.763328  
YearsSinceLastPromotion 1.982939  
YearsWithCurrManager 0.832884  
dtype: float64

**The skewness checks the Symmetry of curve :**

* Education Column is negatively skewed ie. Mean < Median
* EmployeeID , StandardHours has Skewness = 0 therefore it is normally distributed (Bell shaped curve)
* Remaining are positively Skewed (Mean > Median)

In [13]:

df[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].kurt()

Out[13]:

Age -0.405951  
DistanceFromHome -0.227045  
Education -0.560569  
EmployeeCount 0.000000  
EmployeeID -1.200000  
JobLevel 0.395525  
MonthlyIncome 1.000232  
NumCompaniesWorked 0.007287  
PercentSalaryHike -0.302638  
StandardHours 0.000000  
StockOptionLevel 0.361086  
TotalWorkingYears 0.912936  
TrainingTimesLastYear 0.491149  
YearsAtCompany 3.923864  
YearsSinceLastPromotion 3.601761  
YearsWithCurrManager 0.167949  
dtype: float64

**The kurtosis checks the peakness of curve :**

* Age , DistanceFromHome , Education, EmployeeID, PercentSalaryHike are **Platykurtic** in nature ie. Flat and spread out
* EmployeeCount, StandardHours are **Mesokurtic** in nature (Bell shaped curve - Normally Distributed)
* Remaining are Leptokurtic in nature

In [14]:

df[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].var()

Out[14]:

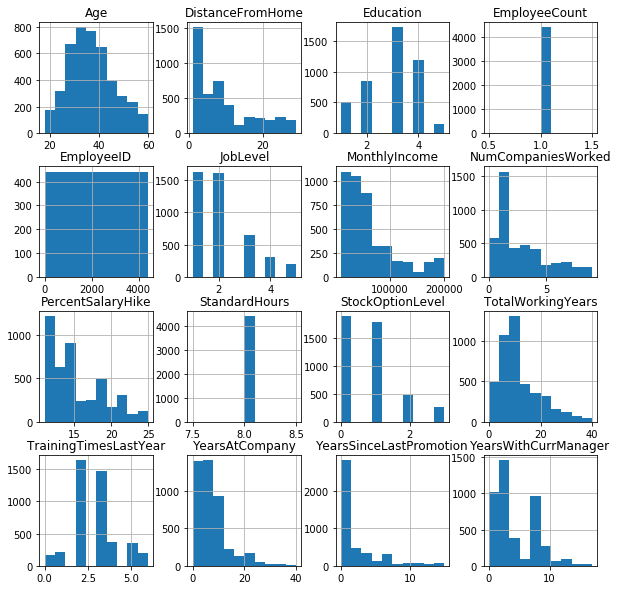
Age 8.341719e+01  
DistanceFromHome 6.569144e+01  
Education 1.048438e+00  
EmployeeCount 0.000000e+00  
EmployeeID 1.621042e+06  
JobLevel 1.224760e+00  
MonthlyIncome 2.215480e+09  
NumCompaniesWorked 6.244436e+00  
PercentSalaryHike 1.338907e+01  
StandardHours 0.000000e+00  
StockOptionLevel 7.257053e-01  
TotalWorkingYears 6.056298e+01  
TrainingTimesLastYear 1.661465e+00  
YearsAtCompany 3.751728e+01  
YearsSinceLastPromotion 1.037935e+01  
YearsWithCurrManager 1.272582e+01  
dtype: float64

* **Variance : It is used to find the Variation in data**

In [3]:

**Ploting Histogram**

df.hist(figsize**=**(10,10))

plt.show(

### Inference from Histogram:

* Education, JobLevel, StockOptional Level are Categorical Variables which are in Encoded format
* Standard Hours, Employee Count, Employee ID are irrelevent columns

In [9]:

**Ploting Distplot to check Distribution**

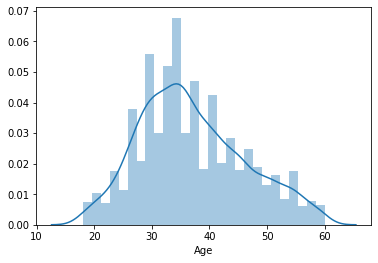
columns **=** ['Age', 'DistanceFromHome', 'MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears',

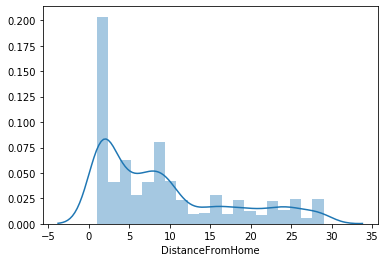
'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']

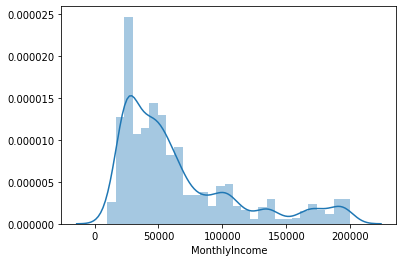
**for** col **in** columns:

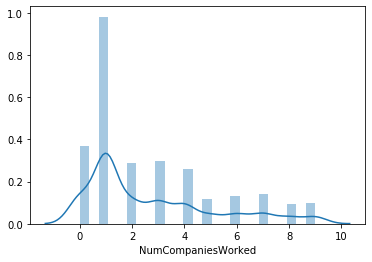
sns.distplot(df[col])

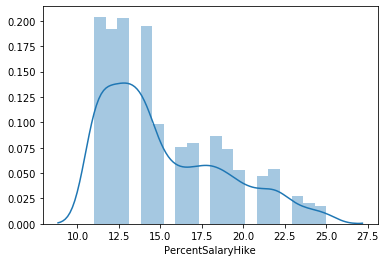
plt.show()

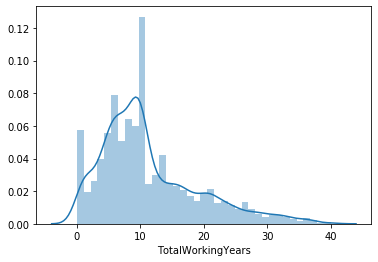


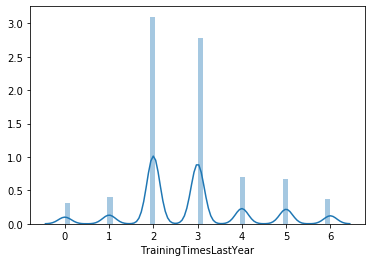


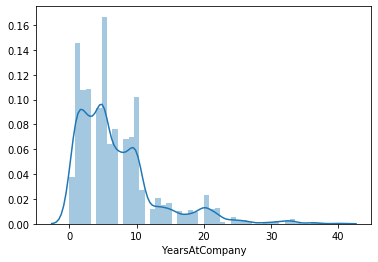


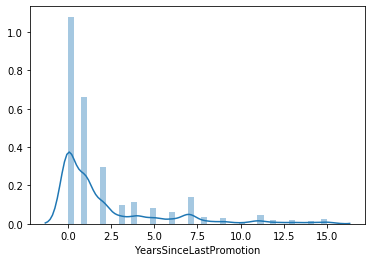


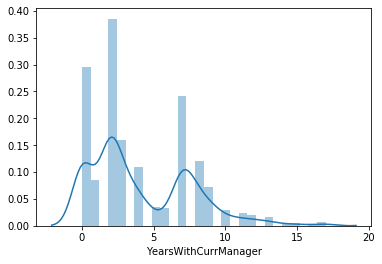












### Inference from Distplot

* Age, NumCompanies Worked, PercentSalaryHike, TotalWorking Years, YearsSinceLastPromotion is Unimodal
* Distance from Home is Bimodal
* Monthly Income, TrainingtimesLastYear, YearsAtCompany, YearswithCurrManager is Multimodall

In [5]:

**Ploting Boxplot to check Outliers:**

col **=** ['Age', 'DistanceFromHome', 'Education', 'EmployeeCount', 'EmployeeID','JobLevel','MonthlyIncome',

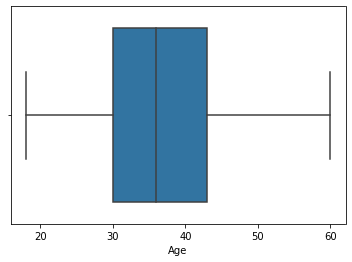
'NumCompaniesWorked', 'PercentSalaryHike', 'StandardHours','StockOptionLevel',

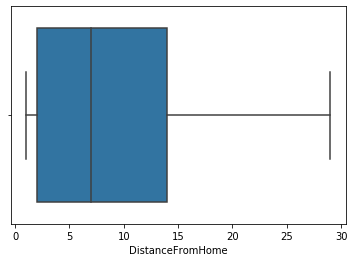
'TotalWorkingYears','TrainingTimesLastYear','YearsAtCompany','YearsSinceLastPromotion','YearsWithCurrManager']

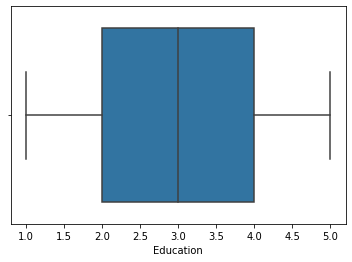
**for** i **in** col:

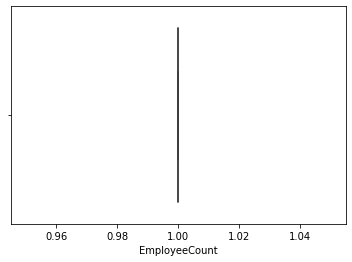
sns.boxplot(df[i])

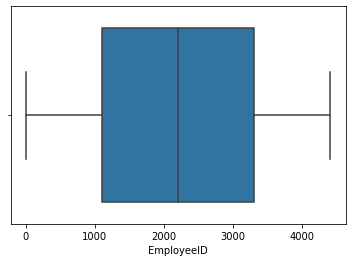
plt.show()

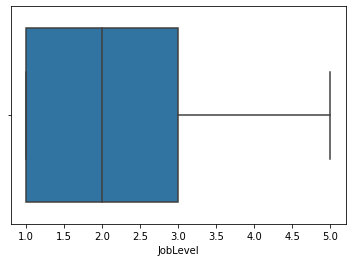


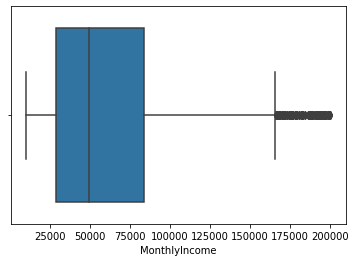


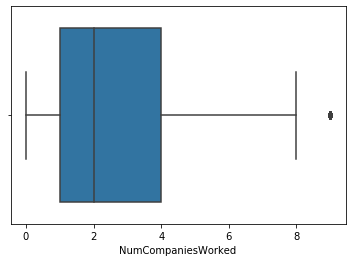


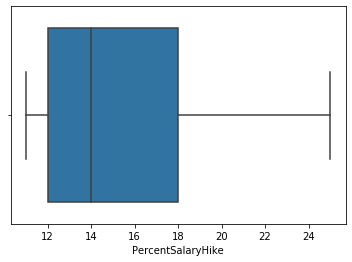


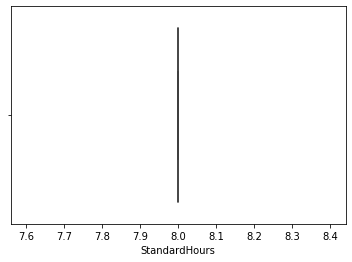


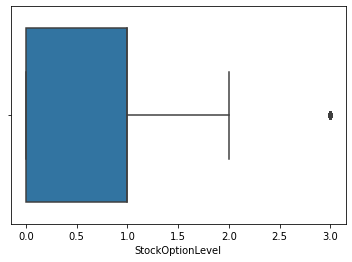


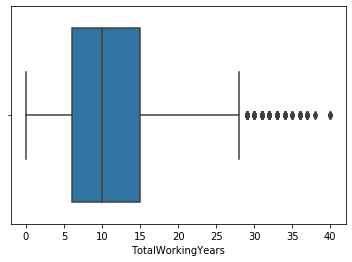


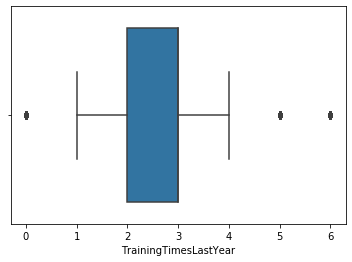


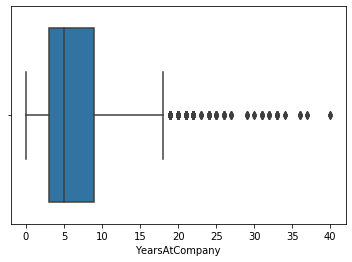


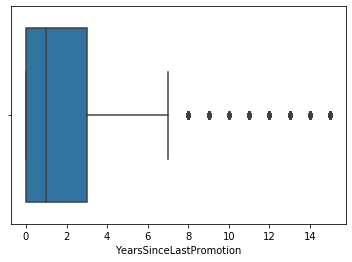


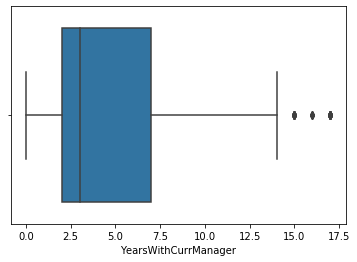












### Inference from BoxPlots

Ploted Box Plot for all the Continous Variable ( int and float not object type variable)

* Age, DistanceFromHome, Education, EmployeeID, JobLevel, PercentSalaryHike : **Donot have any outlier**
* NumCompaniesWorked, StockOptionLevel has **Single Outlier**
* MonthlyIncome has **Many Outliers**
* TotalWorkingYears, YearsAtCompany, YearsSinceLastPromotion have **Moderate Outliers**
* TrainingTimesLastYear, YearsWithCurrManager : **Have Some Outliers**
* EmployeeCount , StandardHours : Just have a line ( **Irrelevant Columns** of DataSet )

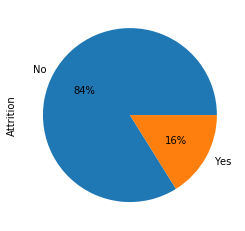
### Attrition Analysis

In [17]:

df['Attrition'].value\_counts().plot(kind**=**'pie', autopct **=** "%1.0f%%")

Out[17]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346e381288>



* From the pie chart we can say : **16% of the employee leaves the company**

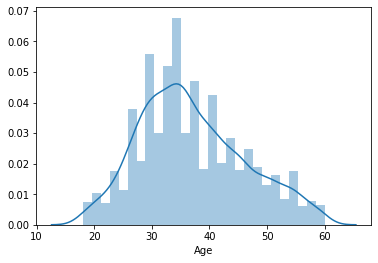
## Analysis 1

In [18]:

sns.distplot(df.Age)

Out[18]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346e3cccc8>

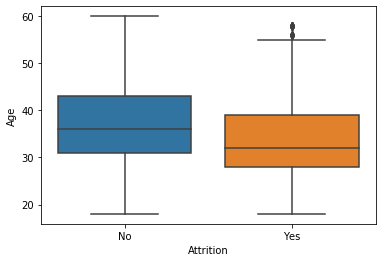


In [19]:

sns.boxplot(x**=**'Attrition', y**=**'Age', data**=**df)

Out[19]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346e46c6c8>



## Inference:

From the plot :

* Mean age of Employee having high Attrition Rate is not 30.

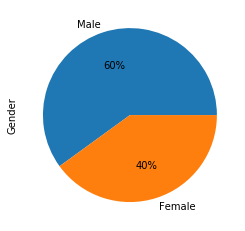
## Analysis 2

In [20]:

df['Gender'].value\_counts().plot(kind**=**'pie', autopct **=** "%1.0f%%")

Out[20]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346e5031c8>



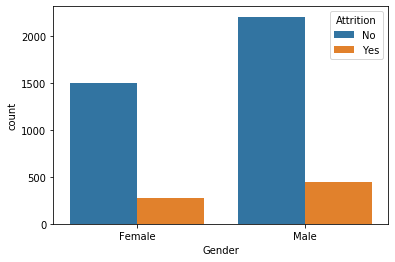
In the dataset : There are total **60% male employee** and **40% female employee**

In [21]:

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

Out[21]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346d18f708>



## Inference:

From the plot :

* Male Employee have high attrition rate than female employee

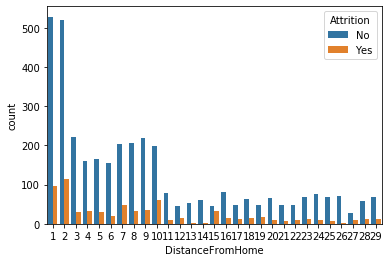
## Analysis 3

In [22]:

sns.countplot(x**=**'DistanceFromHome', hue**=**'Attrition', data**=**df)

Out[22]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346e332848>



## Inference:

From the plot :

* The employee having distance from home less than 10 have high attrition rate

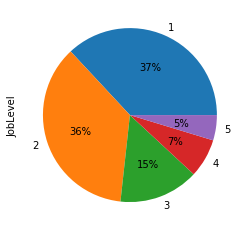
## Analysis 4

In [23]:

df['JobLevel'].value\_counts().plot(kind**=**'pie', autopct **=** "%1.0f%%")

Out[23]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346ca82188>

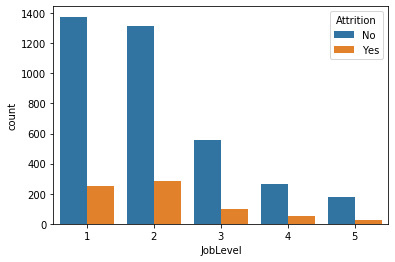


In [24]:

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

Out[24]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346d205508>



## Inference:

From the plot :

* Employee having joblevel less than 3 have high attrition rate

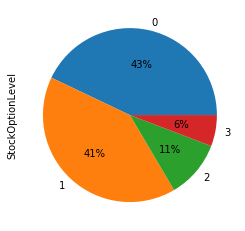
## Analysis 5

In [25]:

df['StockOptionLevel'].value\_counts().plot(kind**=**'pie', autopct **=** "%1.0f%%")

Out[25]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346cdfecc8>

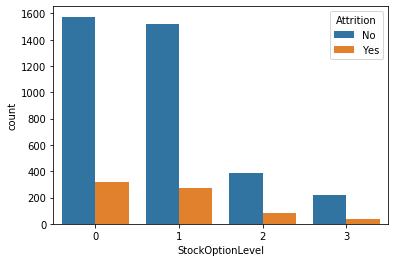


In [26]:

sns.countplot(x**=**'StockOptionLevel', hue**=**'Attrition', data**=**df)

Out[26]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346d12a708>



## Inference:

From the plot :

* The employee having Stack Option Level less than 2 have high attrition rate

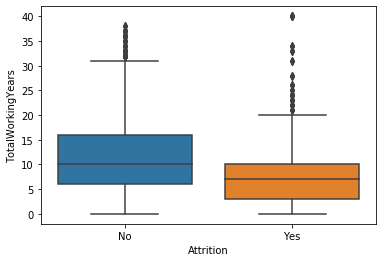
## Analysis 6

In [28]:

sns.boxplot(x**=**'Attrition', y**=**'TotalWorkingYears', data**=**df)

Out[28]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346d045e08>



## Inference:

From the plot :

* The employee having Total Working Years less than 10 have high attrition rate

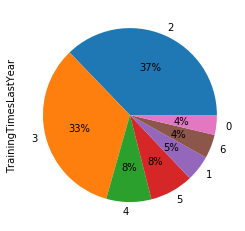
## Analysis 7

In [29]:

df['TrainingTimesLastYear'].value\_counts().plot(kind**=**'pie', autopct **=** "%1.0f%%")

Out[29]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346d088908>

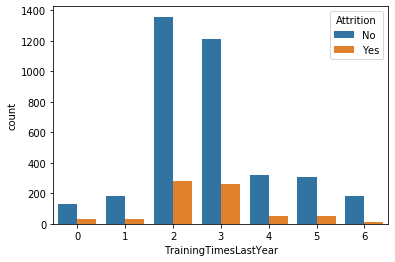


In [30]:

sns.countplot(x**=**'TrainingTimesLastYear', hue**=**'Attrition', data**=**df)

Out[30]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346cfe1f08>



## Inference:

From the plot :

* The Employee who have Training experience of 2 or 3 years have high atrrition rate

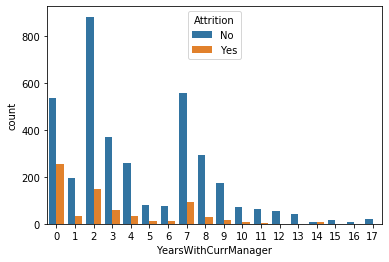
## Analysis 8

In [31]:

sns.countplot(x**=**'YearsWithCurrManager', hue**=**'Attrition', data**=**df)

Out[31]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346e5a4688>



## Inference:

From the plot :

* The employee who have 0 year work experience with Current Manger have high attrition rate than other

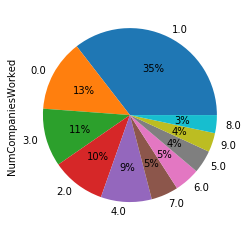
## Analysis 9

In [32]:

df['NumCompaniesWorked'].value\_counts().plot(kind**=**'pie', autopct **=** "%1.0f%%")

Out[32]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346c9c0e88>

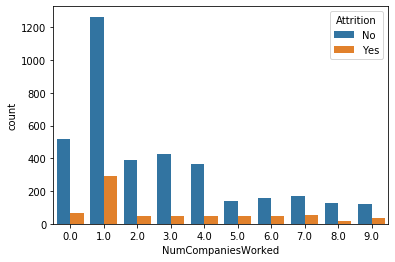


In [33]:

sns.countplot(x**=**'NumCompaniesWorked', hue**=**'Attrition', data**=**df)

Out[33]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346e71ca48>



## Inference:

From the plot :

* The employee Worked in 1 Company have high attrition rate

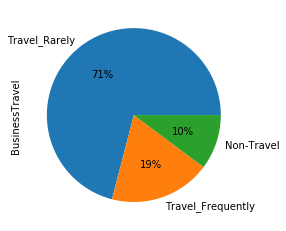
## Analysis 10

In [34]:

df['BusinessTravel'].value\_counts().plot(kind**=**'pie', autopct **=** "%1.0f%%")

Out[34]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346e7a5948>

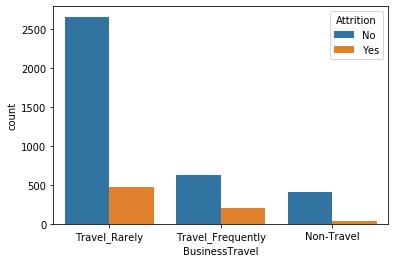


In [35]:

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

Out[35]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2346e812648>



## Inference:

From the plot :

* The employee who Rarely travels have high attrition rate than the employee who Frequently travels

## Correlation Analysis

#### Checking Null Columns

In [38]:

df.isnull().sum()

Out[38]:

Age 0  
Attrition 0  
BusinessTravel 0  
Department 0  
DistanceFromHome 0  
Education 0  
EducationField 0  
EmployeeCount 0  
EmployeeID 0  
Gender 0  
JobLevel 0  
JobRole 0  
MaritalStatus 0  
MonthlyIncome 0  
NumCompaniesWorked 19  
Over18 0  
PercentSalaryHike 0  
StandardHours 0  
StockOptionLevel 0  
TotalWorkingYears 9  
TrainingTimesLastYear 0  
YearsAtCompany 0  
YearsSinceLastPromotion 0  
YearsWithCurrManager 0  
dtype: int64

#### Filling Null Values by mean and median

In [39]:

df['TotalWorkingYears'] **=** df['TotalWorkingYears'].fillna(11.28)

*# 11.28 is the mean of TotalWorkingYears column*

df['NumCompaniesWorked'] **=** df['NumCompaniesWorked'].fillna(2)

*# 2 is the median of NumCompaniesWorked*

#### Droping Null Values

In [40]:

df.isnull().sum()

Out[40]:

Age 0  
Attrition 0  
BusinessTravel 0  
Department 0  
DistanceFromHome 0  
Education 0  
EducationField 0  
EmployeeCount 0  
EmployeeID 0  
Gender 0  
JobLevel 0  
JobRole 0  
MaritalStatus 0  
MonthlyIncome 0  
NumCompaniesWorked 0  
Over18 0  
PercentSalaryHike 0  
StandardHours 0  
StockOptionLevel 0  
TotalWorkingYears 0  
TrainingTimesLastYear 0  
YearsAtCompany 0  
YearsSinceLastPromotion 0  
YearsWithCurrManager 0  
dtype: int64

* No Null values are there now

### Dropping Duplicates

In [41]:

df **=** df.drop\_duplicates()

* Resultant Shape of Dataset

In [42]:

df.shape

Out[42]:

(4410, 24)

* As the shape remains same , so there is no duplicate Column in this Dataset

#### Checking information of Dataset

In [43]:

df.info()

<class 'pandas.core.frame.DataFrame'>  
Int64Index: 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 4410 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 4410 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: 861.3+ KB

## Correlation Matrix

In [9]:

plt.figure(figsize**=**(15,10))

sns.heatmap(df.corr(), annot **=** **True**)

Out[9]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x1bcfdcf2108>

