In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from warnings import filterwarnings
filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv('general_data.csv')
```

- Shape of Dataset

In [3]:

```
df.shape
```

Out[3]:

```
(4410, 24)
```

- View the first 5 and last 5 records using Head and Tail method

In [4]:

```
df.head()
```

Out[4]:

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

5 rows × 24 columns

In [5]:

```
df.tail()
```

Out[5]:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gende
4405	42	No	Travel_Rarely	Research & Development	5	4	Medical	1	4406	Fema
4406	29	No	Travel_Rarely	Research & Development	2	4	Medical	1	4407	Ma
4407	25	No	Travel_Rarely	Research & Development	25	2	Life Sciences	1	4408	Ma
4408	42	No	Travel_Rarely	Sales	18	2	Medical	1	4409	Ma
4409	40	No	Travel_Rarely	Research & Development	28	3	Medical	1	4410	Ma

5 rows × 24 columns

- 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
```

- As the dataset has total 4410 rows and there is no missing Value in it

- Checking Mean, Median of all float and integer columns

In [8]:

```
df.describe()
```

Out[8]:

	Age	DistanceFromHome	Education	EmployeeCount	EmployeeID	JobLevel	MonthlyIncome	NumCompaniesWorke

	Age	DistanceFromHome	Education	EmployeeCount	EmployeeID	JobLevel	MonthlyIncome	NumCompaniesWorke
count	4410.000000	4410.000000	4410.000000	4410.0	4410.000000	4410.000000	4410.000000	4391.00000
mean	36.923810	9.192517	2.912925	1.0	2205.500000	2.063946	65029.312925	2.69483
std	9.133301	8.105026	1.023933	0.0	1273.201673	1.106689	47068.888559	2.49888
min	18.000000	1.000000	1.000000	1.0	1.000000	1.000000	10090.000000	0.00000
25%	30.000000	2.000000	2.000000	1.0	1103.250000	1.000000	29110.000000	1.00000
50%	36.000000	7.000000	3.000000	1.0	2205.500000	2.000000	49190.000000	2.00000
75%	43.000000	14.000000	4.000000	1.0	3307.750000	3.000000	83800.000000	4.00000
max	60.000000	29.000000	5.000000	1.0	4410.000000	5.000000	199990.000000	9.00000

- 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 [9]:

```
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[9]:

```
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 [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']].mean()
```

Out[10]:

```
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
```

```
StockOptionLevel            0.793878
TotalWorkingYears          11.279936
TrainingTimesLastYear       2.799320
YearsAtCompany              7.008163
YearsSinceLastPromotion     2.187755
YearsWithCurrManager        4.123129
dtype: float64
```

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']].skew()
```

Out[11]:

```
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 [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']].kurt()
```

Out[12]:

```
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 [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']].var()
```

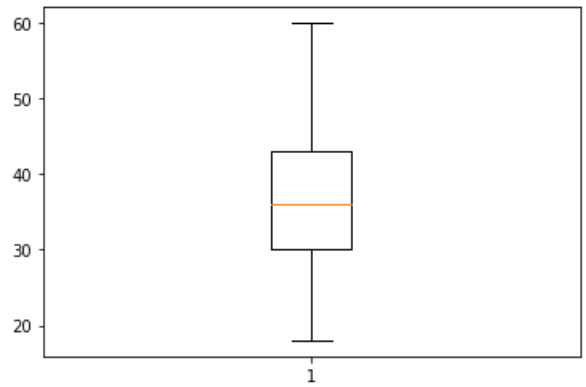
Out[13]:

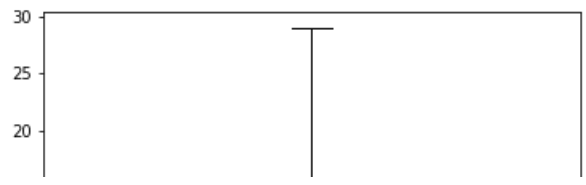
```
Age                         8.341719e+01
DistanceFromHome            6.569144e+01
Education                   1.048438e+00
EmployeeCount               0.000000e+00
EmployeeID                  1.621042e+06
JobLevel                    1.224760e+00
MonthlyIncome               2.215480e+09
NumCompaniesWorked          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
```

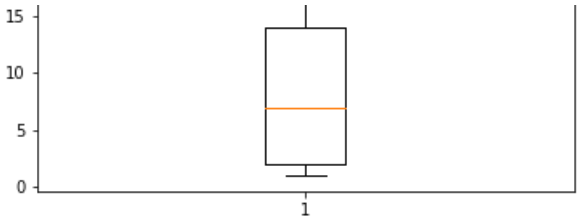
- Variance : It is used to find the Variation in data

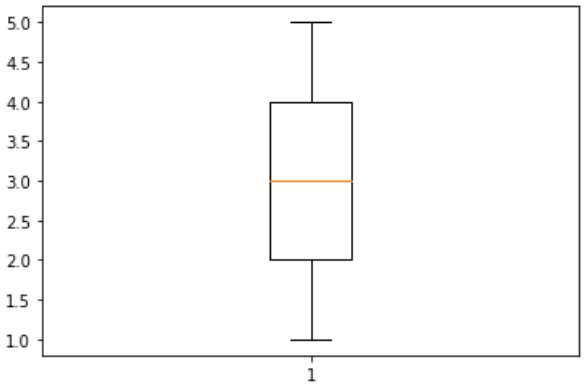
In [14]:

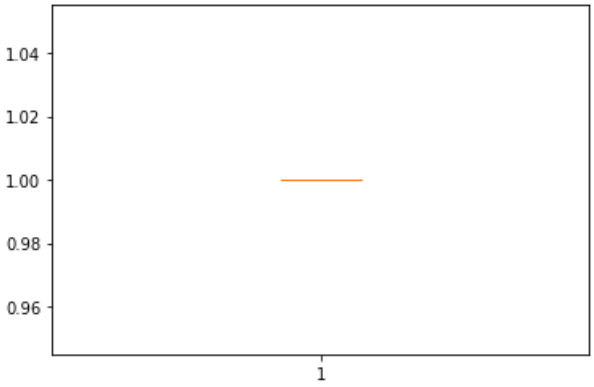
```
col = ['Age', 'DistanceFromHome', 'Education', 'EmployeeCount', 'EmployeeID','JobLevel','MonthlyInc
ome',
       'NumCompaniesWorked', 'PercentSalaryHike', 'StandardHours','StockOptionLevel',
       'TotalWorkingYears','TrainingTimesLastYear','YearsAtCompany','YearsSinceLastPromotion','Year
sWithCurrManager']

for i in col:
    fig, ax = plt.subplots()
    ax.boxplot(df[i])
```

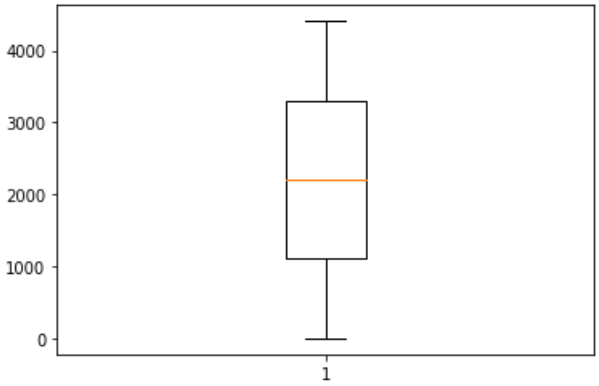


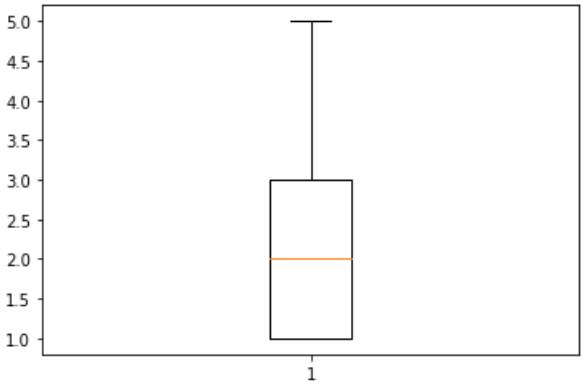



15
10
5
0
1

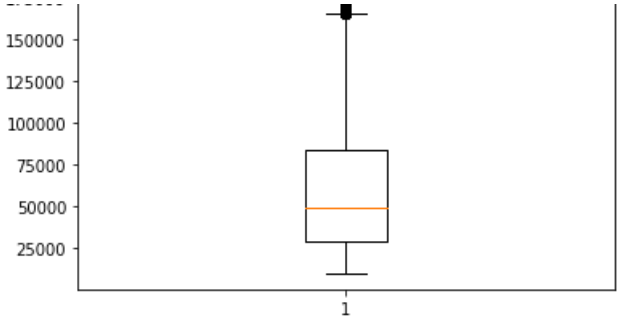

5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
1

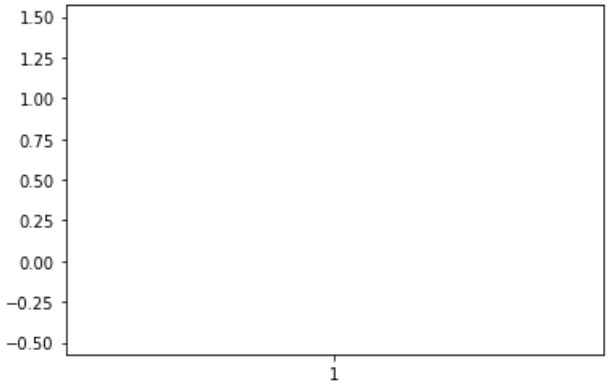

1.04
1.02
1.00
0.98
0.96
1

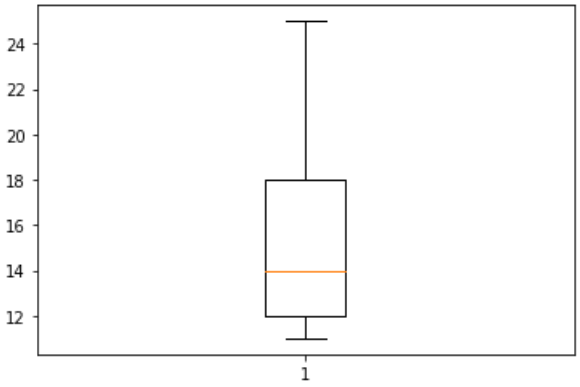

4000
3000
2000
1000
0
1

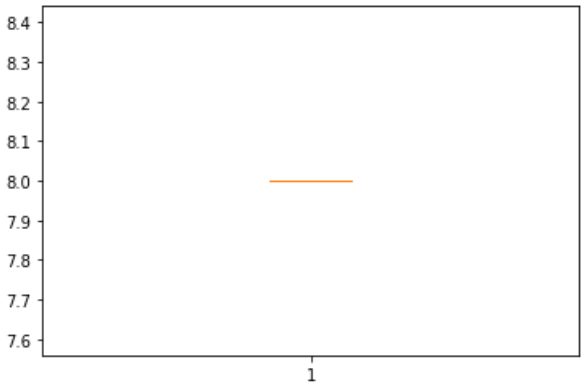

5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
1

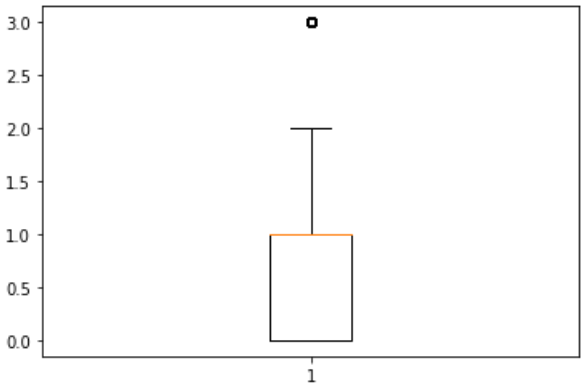

200000
175000

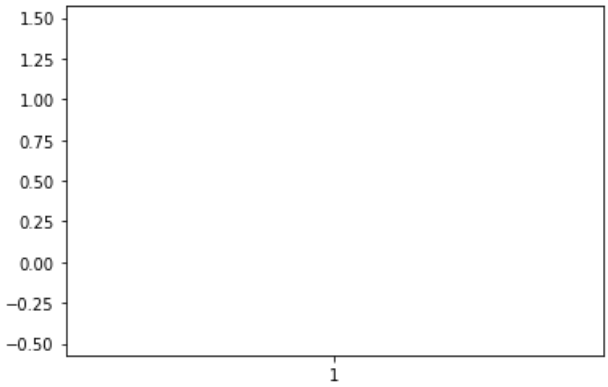

150000
125000
100000
75000
50000
25000
1

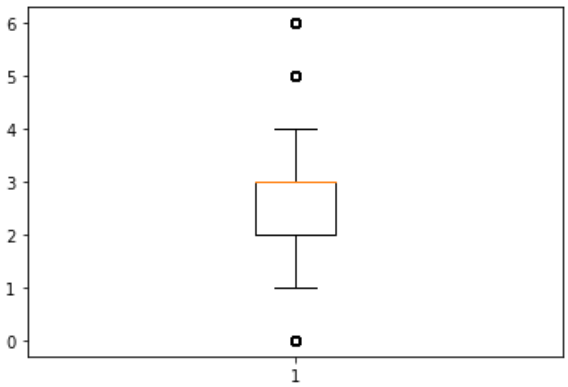

1.50
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
1

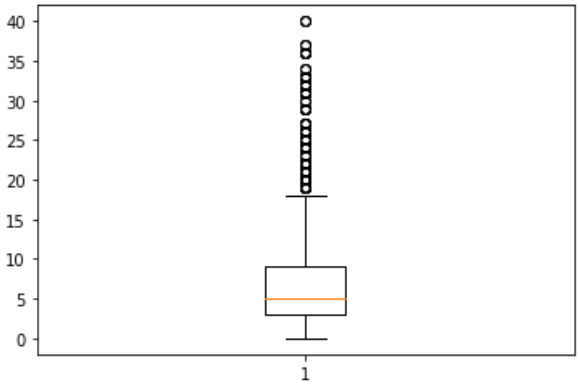

24
22
20
18
16
14
12
1

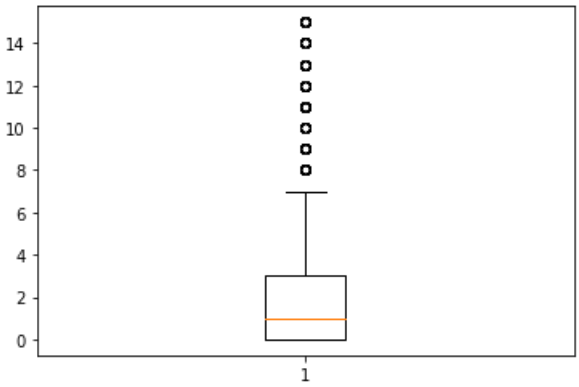

8.4
8.3
8.2
8.1
8.0
7.9
7.8
7.7
7.6
1

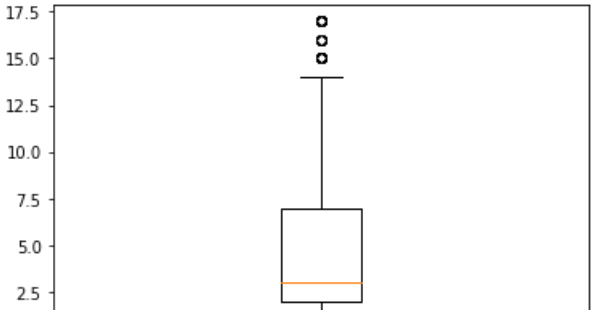

3.0
2.5
2.0
1.5
1.0
0.5
0.0
1


1.50
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
1


6
5
4
3
2
1
0
1


40
35
30
25
20
15
10
5
0
1


14
12
10
8
6
4
2
0
1


17.5
15.0
12.5
10.0
7.5
5.0
2.5

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

- Age, DistanceFromHome, Education, EmployeeID, JobLevel : **Donot have any outlier**
- MonthlyIncome, PercentSalaryHike, StockOptionLevel, TrainingTimesLastYear, YearsAtCompany, YearsSinceLastPromotion, YearsWithCurrManager : **Have Some Outliers**
- EmployeeCount , StandardHours : Just have a line
- NumCompaniesWorked , TotalWorkingYears : **cant be ploted**