Step 1 – Launching

```
import pandas as pd
dataset = pd.read_csv("general_data.csv")
dataset
```

Out[3]:

```
      Age Attrition  ... YearsSinceLastPromotion YearsWithCurrManager
0     51       No  ...                    0                  0
1     31      Yes  ...                    1                  4
2     32       No  ...                    0                  3
3     38       No  ...                    7                  5
4     32       No  ...                    0                  4
...   ...      ...  ...                  ...                ...
4405  42       No  ...                    0                  2
4406  29       No  ...                    0                  2
4407  25       No  ...                    1                  2
4408  42       No  ...                    7                  8
4409  40       No  ...                    3                  9

[4410 rows x 24 columns]
```

head() - To get first 5 records of dataset we use head()

```
dataset.head()
```

Out[4]:

```
   Age Attrition  ... YearsSinceLastPromotion YearsWithCurrManager
0  51       No  ...                    0                  0
1  31      Yes  ...                    1                  4
2  32       No  ...                    0                  3
3  38       No  ...                    7                  5
```

4 32 No ... 0 4

[5 rows x 24 columns]

columns - To get columns in dataset we use columns

dataset.columns

Out[5]:

```
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')
```

Step 2 – Data Treatment

isnull() - To check null records in dataset we use isnull()

dataset.isnull()

Out[6]:

	Age	Attrition	...	YearsSinceLastPromotion	YearsWithCurrManager
0	False	False	...	False	False
1	False	False	...	False	False
2	False	False	...	False	False
3	False	False	...	False	False
4	False	False	...	False	False
...	...	...	...	...	...
4405	False	False	...	False	False

4406 False False ... False False

4407 False False ... False False

4408 False False ... False False

4409 False False ... False False

[4410 rows x 24 columns]

duplicated() - To check duplicate records in dataset we use duplicated()

dataset.duplicated()

Out[7]:

0 False

1 False

2 False

3 False

4 False

4405 False

4406 False

4407 False

4408 False

4409 False

Length: 4410, dtype: bool

drop_duplicates() - To drop duplicate records in dataset we use drop_duplicates()

dataset.drop_duplicates()

Out[8]:

	Age	Attrition	...	YearsSinceLastPromotion	YearsWithCurrManager
0	51	No	...	0	0
1	31	Yes	...	1	4
2	32	No	...	0	3
3	38	No	...	7	5

4 32 No ... 0 4

... ...

4405 42 No ... 0 2

4406 29 No ... 0 2

4407 25 No ... 1 2

4408 42 No ... 7 8

4409 40 No ... 3 9

[4410 rows x 24 columns]

Step 3 - Univariate Analysis

describe() - gives all results like count, mean, std, min, 25%, 50%, 75%, max etc.

dataset[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].describe()

Out[9]:

	Age	...	YearsWithCurrManager
count	4410.000000	...	4410.000000
mean	36.923810	...	4.123129
std	9.133301	...	3.567327
min	18.000000	...	0.000000
25%	30.000000	...	2.000000
50%	36.000000	...	3.000000
75%	43.000000	...	7.000000
max	60.000000	...	17.000000

[8 rows x 11 columns]

mean() – finds average values

```
dataset[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked',
'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear',
'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].mean()
```

Out[10]:

```
Age                        36.923810
DistanceFromHome           9.192517
Education                  2.912925
MonthlyIncome              65029.312925
NumCompaniesWorked         2.694830
PercentSalaryHike          15.209524
TotalWorkingYears          11.279936
TrainingTimesLastYear      2.799320
YearsAtCompany             7.008163
YearsSinceLastPromotion    2.187755
YearsWithCurrManager       4.123129
dtype: float64
```

median() – Finds middle value

```
dataset[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked',
'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear',
'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].median()
```

Out[11]:

```
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

mode() – Finds most repeated value

```
dataset[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked',
'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear',
'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].mode()
```

Out[12]:

	Age	DistanceFromHome	...	YearsSinceLastPromotion	YearsWithCurrManager
0	35	2	...	0	2

[1 rows x 11 columns]

var() – Measures the variability of data

```
dataset[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked',
'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear',
'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].var()
```

Out[13]:

Age 8.341719e+01

DistanceFromHome 6.569144e+01

Education 1.048438e+00

MonthlyIncome 2.215480e+09

NumCompaniesWorked 6.244436e+00

PercentSalaryHike 1.338907e+01

TotalWorkingYears 6.056298e+01

TrainingTimesLastYear 1.661465e+00

YearsAtCompany 3.751728e+01

YearsSinceLastPromotion 1.037935e+01

YearsWithCurrManager 1.272582e+01

dtype: float64

std() – Finds consistency of data

dataset[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].std()

Out[14]:

Age 9.133301

DistanceFromHome 8.105026

Education 1.023933

MonthlyIncome 47068.888559

NumCompaniesWorked 2.498887

PercentSalaryHike 3.659108

TotalWorkingYears 7.782222

TrainingTimesLastYear 1.288978

YearsAtCompany 6.125135

YearsSinceLastPromotion 3.221699

YearsWithCurrManager 3.567327

dtype: float64

skew() – finds symmentricness of data

dataset[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].skew()

Out[15]:

Age 0.413005

DistanceFromHome 0.957466

Education -0.289484

MonthlyIncome 1.368884

NumCompaniesWorked 1.026767

PercentSalaryHike 0.820569

TotalWorkingYears 1.116832

TrainingTimesLastYear 0.552748

YearsAtCompany 1.763328

YearsSinceLastPromotion 1.982939

YearsWithCurrManager 0.832884

dtype: float64

kurt() – finds peakness

```
dataset[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked',
'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear',
'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].kurt()
```

Out[17]:

Age -0.405951

DistanceFromHome -0.227045

Education -0.560569

MonthlyIncome 1.000232

NumCompaniesWorked 0.007287

PercentSalaryHike -0.302638

TotalWorkingYears 0.912936

TrainingTimesLastYear 0.491149

YearsAtCompany 3.923864

YearsSinceLastPromotion 3.601761

YearsWithCurrManager 0.167949

dtype: float64

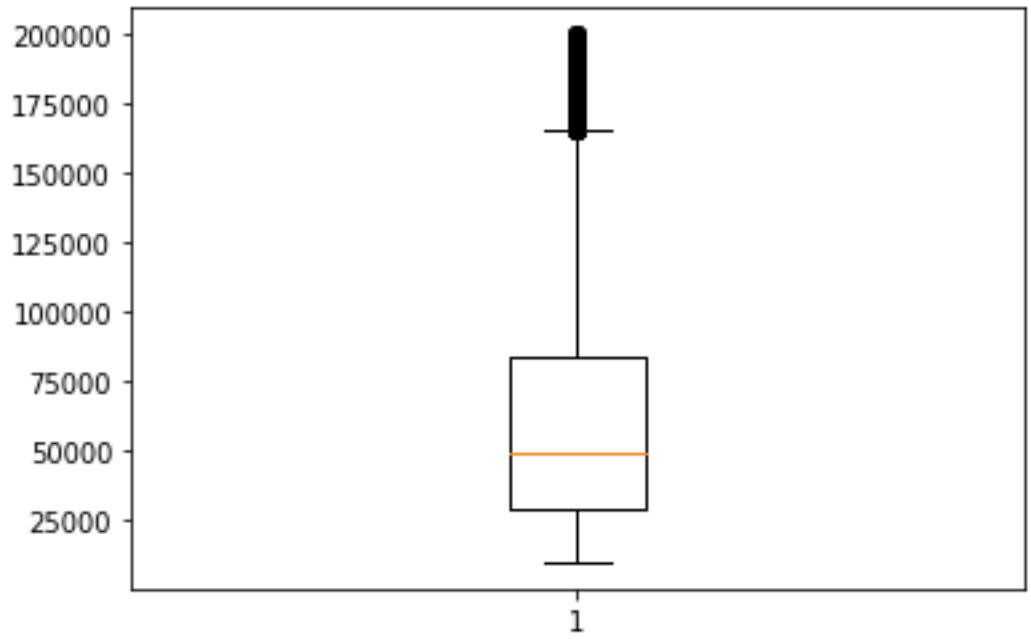
Outliers –

There's no regression found while plotting Age, MonthlyIncome, YearsAtCompany on a scatter plot.

```
import matplotlib.pyplot as plt
box_plot = dataset.MonthlyIncome
plt.boxplot(box_plot)
```

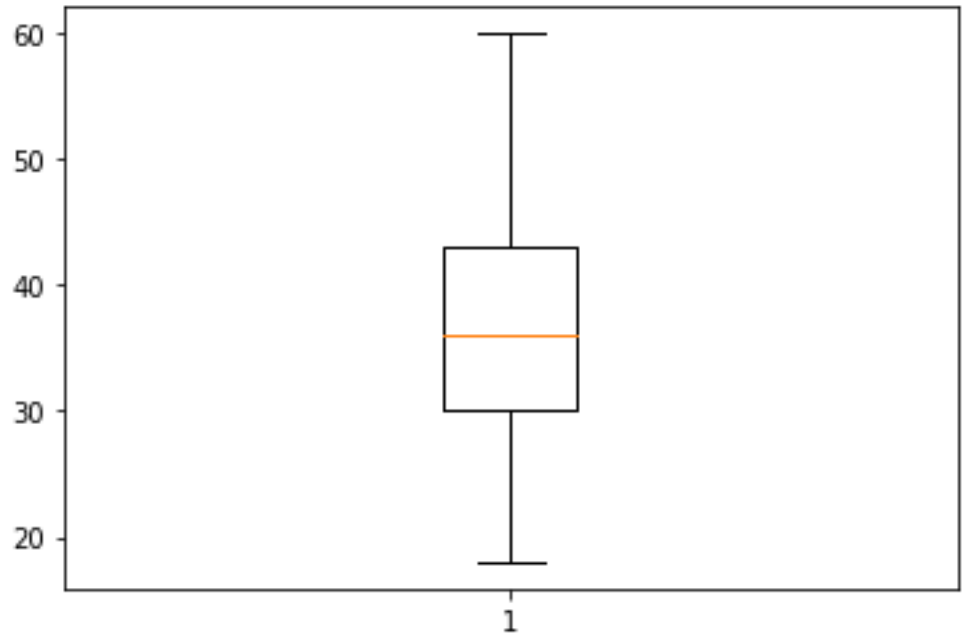
Out[22]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x22f2587bc88>,
  <matplotlib.lines.Line2D at 0x22f2587bf48>],
 'caps': [<matplotlib.lines.Line2D at 0x22f258ca048>,
  <matplotlib.lines.Line2D at 0x22f250d71c8>],
 'boxes': [<matplotlib.lines.Line2D at 0x22f2587bcc8>],
 'medians': [<matplotlib.lines.Line2D at 0x22f25882a08>],
 'fliers': [<matplotlib.lines.Line2D at 0x22f258c9808>],
 'means': []
```



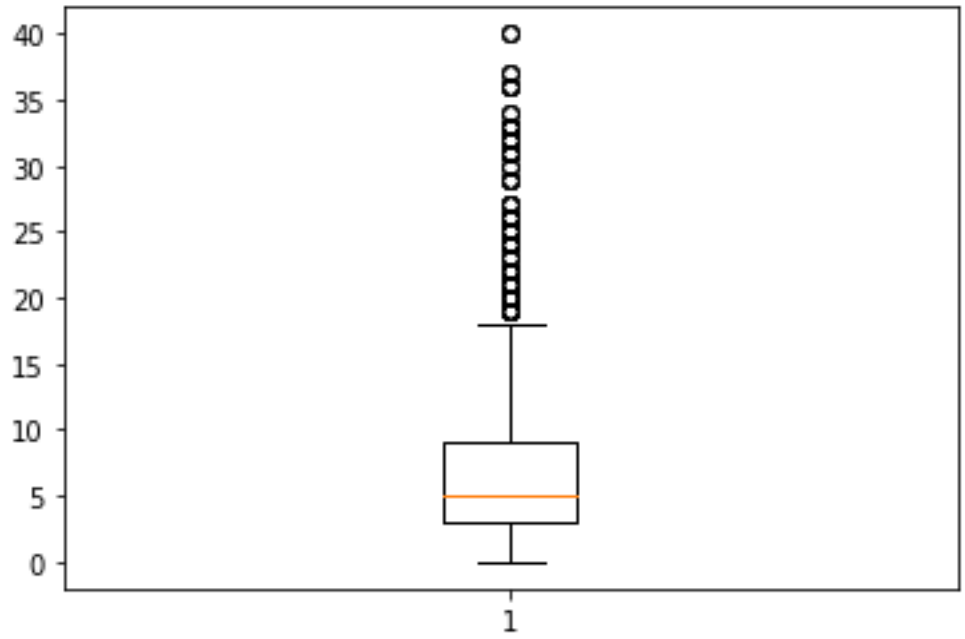
MonthlyIncome is Right skewed with several outliers.

```
box_plot = dataset.Age
plt.boxplot(box_plot)
Out[24]:
{'whiskers': [<matplotlib.lines.Line2D at 0x22f271fcb08>,
  <matplotlib.lines.Line2D at 0x22f27200a88>],
 'caps': [<matplotlib.lines.Line2D at 0x22f27200b88>,
  <matplotlib.lines.Line2D at 0x22f27205a08>],
 'boxes': [<matplotlib.lines.Line2D at 0x22f271fc988>],
 'medians': [<matplotlib.lines.Line2D at 0x22f27205b08>],
 'fliers': [<matplotlib.lines.Line2D at 0x22f27209988>],
 'means': []}
```



Age is normally distributed without any outliers.

```
box_plot = dataset.YearsAtCompany
plt.boxplot(box_plot)
Out[26]:
{'whiskers': [<matplotlib.lines.Line2D at 0x22f27262dc8>,
  <matplotlib.lines.Line2D at 0x22f27266d48>],
 'caps': [<matplotlib.lines.Line2D at 0x22f27266e48>,
  <matplotlib.lines.Line2D at 0x22f2726acc8>],
 'boxes': [<matplotlib.lines.Line2D at 0x22f27262c48>],
 'medians': [<matplotlib.lines.Line2D at 0x22f2726adc8>],
 'fliers': [<matplotlib.lines.Line2D at 0x22f2726ec48>],
 'means': []}
```



YearsAtCompany is right skewed with several outliers.