solution for outlier detection and identification of continuous and categorical variables

importing pandas library ¶

In [2]:

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 %matplotlib inline
4 import numpy as np
```

Solution1 - Jupyter Notebook

jupyter Solution1 Logout

Menu Not Trusted | Python 3

Run Markdown

reading csv file

In [3]:

```
1 df=pd.read_csv("Data4.csv")
```

reading initial 5 rows

In [4]:

```
1 df.head()
```

Out[4]:

	duration
0	20
1	372
2	676
3	65
4	111

4	111

reading last 5 rows

In [5]:

```
1 df.tail()
```

Out[5]:

	duration
179	142
180	258
181	149
182	173
183	101

shape of dataset

shape of dataset

In [6]:

```
1 df.shape
```

Out[6]:

```
(184, 1)
```

describe function is used to show dataset parameters

In [7]:

```
1 df.describe()
```

Out[7]:

describe function is used to show dataset parameters ¶

In [7]:

```
1 df.describe()
```

Out[7]:

	duration
count	184.000000
mean	280.326087
std	277.395463
min	8.000000
25%	115.000000
50%	192.500000
75%	323.250000
max	1809.000000

isnull function is used to show missing values in terms of boolean value such as true or false

In [8]:

```
1 df.isnull()
```

Out[8]:

	duration
0	False
1	False
2	False
3	False
4	False
...	...
179	False

```
1 df.isnull()
```

Out[8]:

	duration
0	False
1	False
2	False
3	False
4	False
...	...
179	False
180	False
181	False
182	False
183	False

184 rows × 1 columns

2	False
3	False
4	False
...	...
179	False
180	False
181	False
182	False
183	False

184 rows × 1 columns

In [ ]:

```
1 # all values are false means
2 # no missing values
```

sum function is used to show count of missing

sum function is used to show count of missing values

In [9]:

```
1 df.isnull().sum()
```

Out[9]:

```
duration    0
dtype: int64
```

In [ ]:

```
1 # 0 means no missing values in da
```

following syntax is used to fill missing values if any

following syntax is used to fill missing values if any

In [ ]:

```
df.fillna(value,inplace=True)
```

In [ ]:

```
# value indicates numerical
# number to be filled in
dataset
```

syntax to drop missing values in data set

In [10]:

```
df.dropna()
```

Out[10]:

syntax to drop missing values in data set

In [10]:

```
1 df.dropna()
```

Out[10]:

	duration
0	20
1	372
2	676
3	65
4	111
...	...
179	142
180	258
181	149

12:47

Solution1 - Jupyter Notebook

jupyter Solution1 Logout

Menu Not Trusted | Python 3

Run Code

In [10]:

```
1 df.dropna()
```

Out[10]:

	duration
0	20
1	372
2	676
3	65
4	111
...	...
179	142
180	258
181	149
182	173
183	101

184 rows × 1 columns

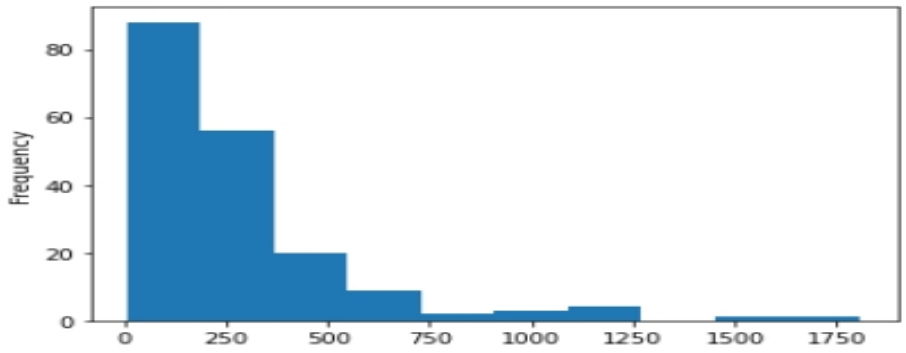
plotting histograms on dataset

In [11]:

```
1 df['duration'].plot.hist()
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x70e9e19640>
```



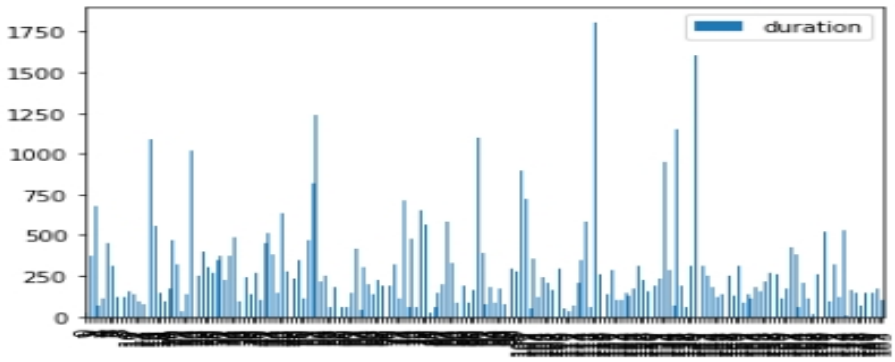
plotting bargraph

In [12]:

```
1 df.plot.bar()
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x70e88f0670>
```



plotting boxplot

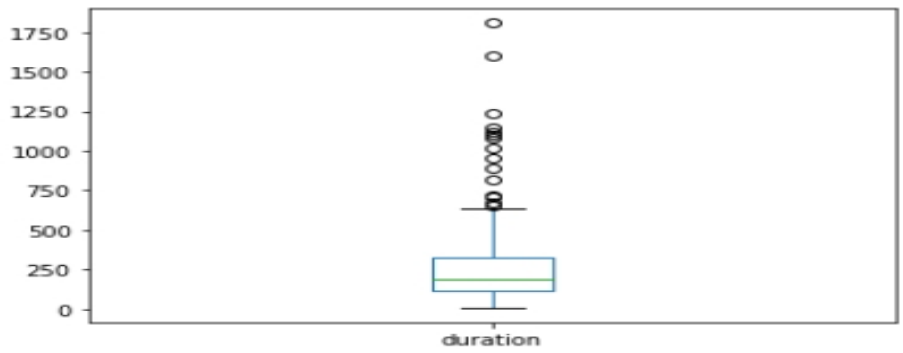
plotting boxplot

In [16]:

```
1 df['duration'].plot.box()
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x70e83ae370>
```

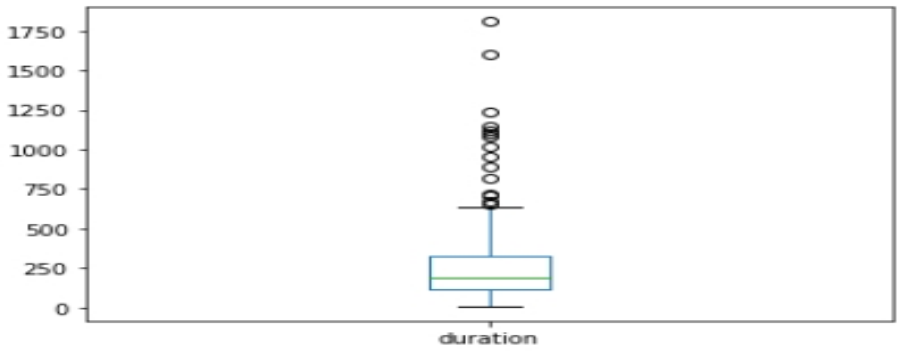


In [ ]:

```
In [16]:
1 df['duration'].plot.box()
```

Out[16]:

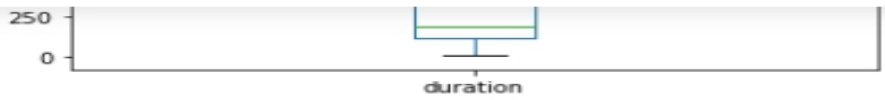
```
<matplotlib.axes._subplots.AxesSubplot at 0x70e83ae370>
```



In [ ]:

```
1 dots indicate outliers in dataset
```

code to eliminate outliers in univariate analysis



In [ ]:

```
1 dots indicate outliers in dataset
```

code to eliminate outliers in univariate analysis

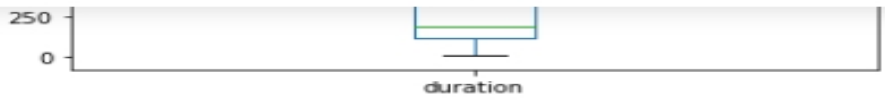
In [21]:

```
1 df.loc[df['duration']>500,'duratic
```

checking outlier elimination

In [22]:

```
1 df.plot.box()
```



In [ ]:

```
1 dots indicate outliers in dataset
```

code to eliminate outliers in univariate analysis

In [21]:

```
1 'duration']=np.mean(df['duration']
```

checking outlier elimination

In [22]:

```
1 df.plot.box()
```

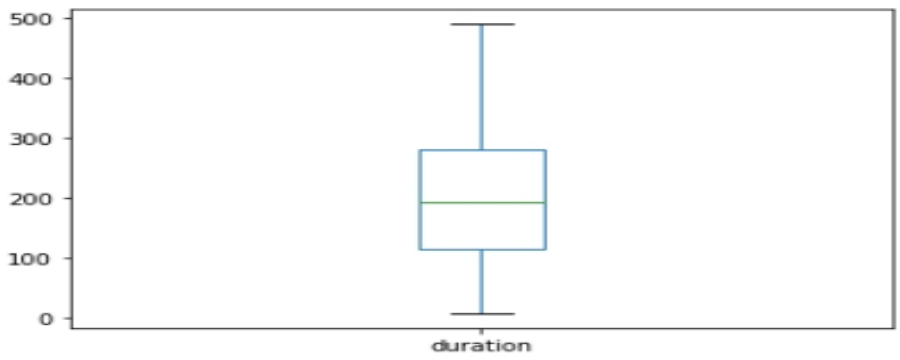
checking outlier elimination

In [22]:

```
1 df.plot.box()
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x70e7c665b0>
```



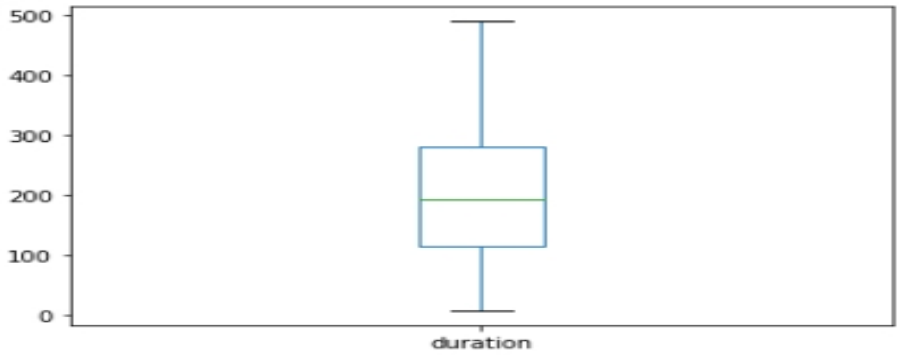
In [ ]:

In [22]:

```
1 df.plot.box()
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x70e7c665b0>
```



In [ ]:

```
1 # no outliers in dataset
```

In [25]:

```
1 df.dtypes
```

In [25]:

```
1 df.dtypes
```

Out[25]:

```
duration    float64
dtype: object
```

In [ ]:

```
1 # dtypes function is used to
2 check datatypes of variables in
3 dataset which is all float
```

syntax for outlier removal of bivariate analysis

In [26]:

```
1 df=df[df['duration']>500]
```

syntax for outlier removal of bivariate analysis

In [26]:

```
df=df[df['duration']>500]
```

reading another dataset

In [3]:

```
df1=pd.read_csv("Data45.csv")
```

In [12]:

```
df1.head()
```

Out[12]:

Solution1 - Jupyter Notebook

jupyter Solution1 Logout

Menu Not Trusted | Python 3

Run Code

reading another dataset

In [3]:

```
1 df1=pd.read_csv("Data45.csv")
```

In [12]:

```
1 df1.head()
```

Out[12]:

	PROPERTY. Income
0	Urban. 5849
1	Rural. 4583
2	Urban. 3000
3	Urban. 2583
4	Urban. 6000

In [5]:

In [5]:

```
1 df1.shape
```

Out[5]:

```
(10, 1)
```

In [11]:

```
1 df1.columns
```

Out[11]:

```
Index(['PROPERTY.
       Income'], dtype='o
bject')
```

In [ ]:

```
1 # columns function is used to
2 # give list of columns as shown
3 which helps to identify continuou
4 and categorical variables
```