```
In [1]:  # Import the packages
         # read the data

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

         path=r"C:\Users\omkar\OneDrive\Documents\Data science\Naresh IT\Datafiles\V:
         visa_df=pd.read_csv(path)
         visa_df.head(3)
```

Out[1]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_
0	EZYV01	Asia	High School	N	N	
1	EZYV02	Asia	Master's	Y	N	
2	EZYV03	Asia	Bachelor's	N	Y	

Steps in Outlier analysis

- Step-1: Find the Q1 , Q2 and Q3
    - np.percentile(column data, q)
- Step-2: Calculate the IQR
    - IQR= Q3-Q1
- Step-3: Calculate lower boundary and upper boundary
    - lb: Q1-1.5IQR
    - ub: Q3+1.5IQR
- Step-4: Find the Outliersdf
    - c1: column data < lb
    - c2: column data > ub
    - c: apply the main condition
    - main data[c]

```
In [3]:  #####------------ Step-1------########### 
         Q1=np.percentile(visa_df['prevailing_wage'],25)
         Q2=np.percentile(visa_df['prevailing_wage'],50)
         Q3=np.percentile(visa_df['prevailing_wage'],75)

         #####------------ Step-2------###########
         IQR=Q3-Q1

         #####------------ Step-3------###########
         lb=Q1-1.5*IQR
         ub=Q3+1.5*IQR

         #####------------ Step-4------###########
         c1=visa_df['prevailing_wage']<lb
         c2=visa_df['prevailing_wage']>ub
         con=c1|c2
         #####------------ Step-5------###########
         outliers_df=visa_df[con]
         outliers_df

         #####------------ Step-6------###########
         c1=visa_df['prevailing_wage']>lb
         c2=visa_df['prevailing_wage']<ub
         con=c1&c2
         non_outliers_df=visa_df[c1&c2]
         non_outliers_df
```

Out[3]:

|	case_id	continent	education_of_employee	has_job_experience	requires_job_trainin
0	EZYV01	Asia	High School	N	
1	EZYV02	Asia	Master's	Y	
2	EZYV03	Asia	Bachelor's	N	
3	EZYV04	Asia	Bachelor's	N	
4	EZYV05	Africa	Master's	Y	
...	...	...	...	...	...
25474	EZYV25475	Africa	Doctorate	N	
25475	EZYV25476	Asia	Bachelor's	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

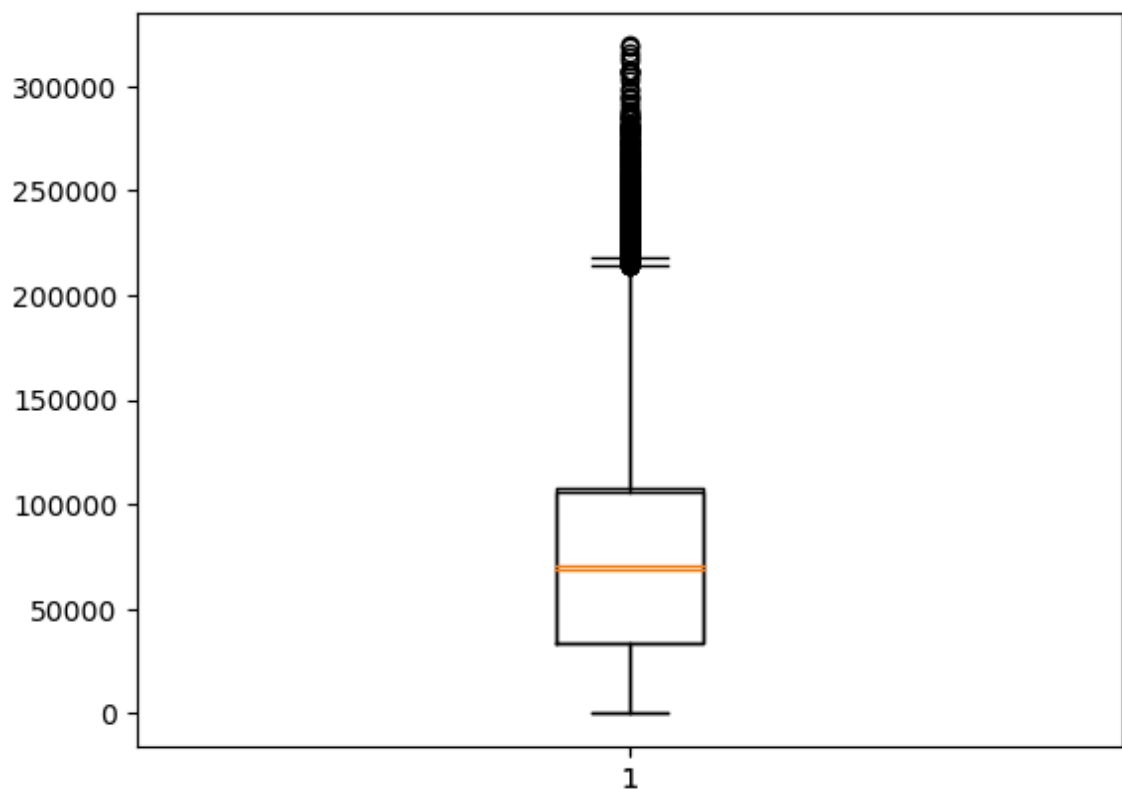25053 rows × 12 columns

Compare original data with non outliers data

will plot histogram and box plot of the both

```
In [5]: plt.hist(visa_df['prevailing_wage'])
        plt.hist(non_outliers_df['prevailing_wage'])
        plt.show()
```
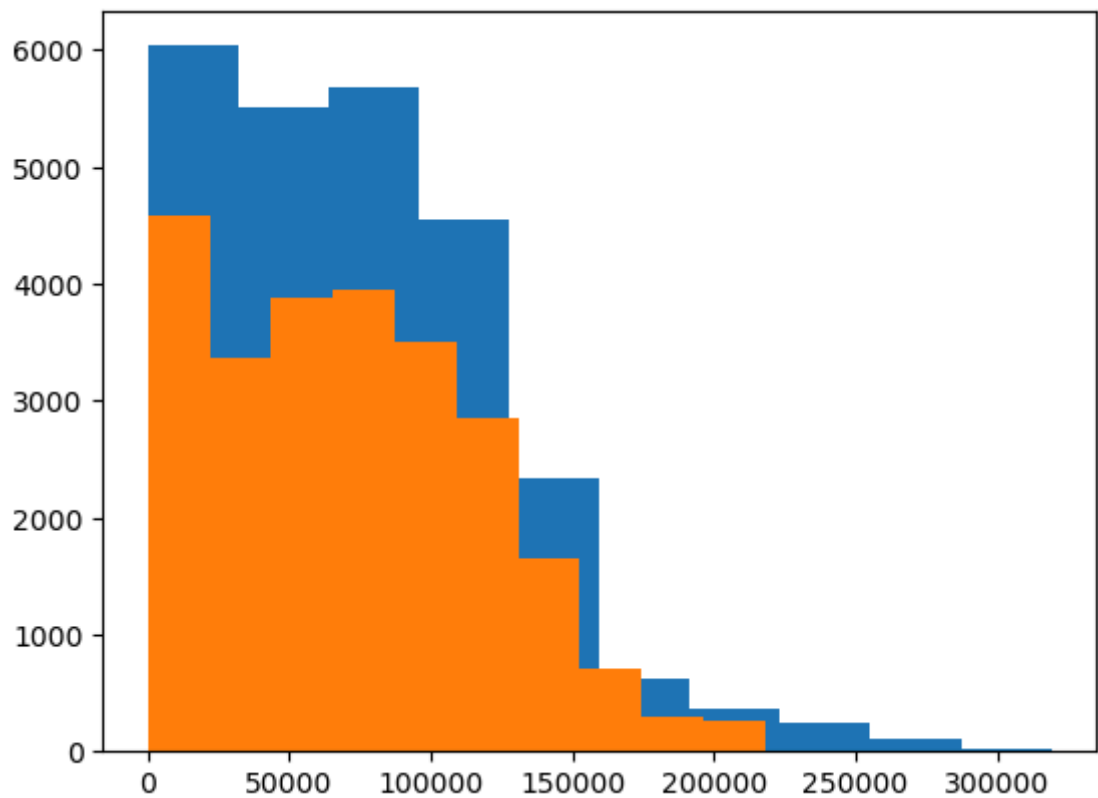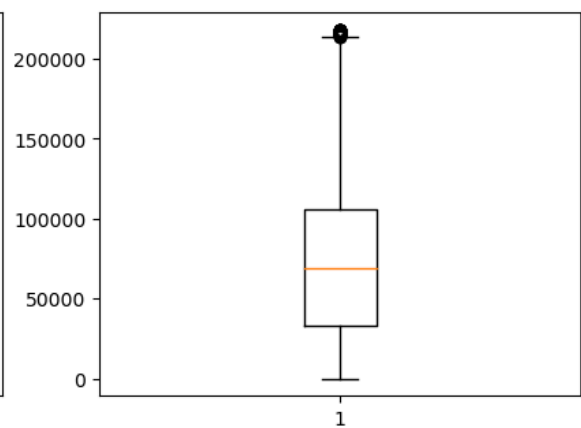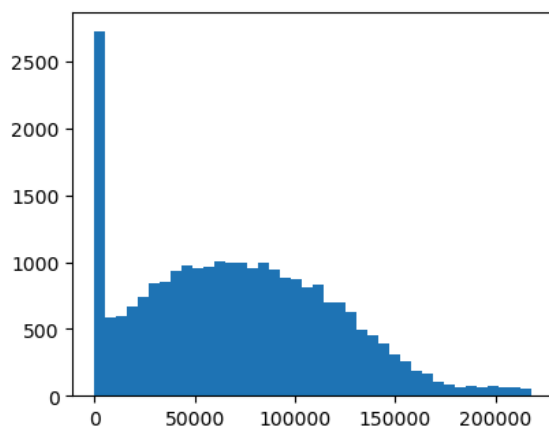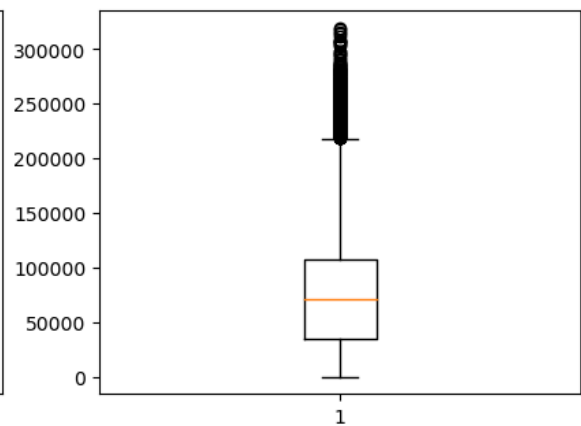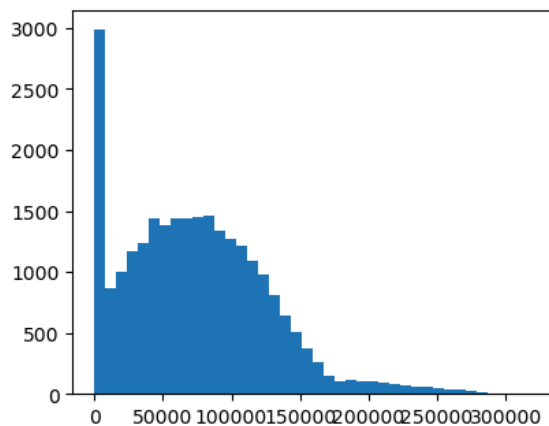


```
In [6]: plt.boxplot(visa_df['prevailing_wage'])
        plt.boxplot(non_outliers_df['prevailing_wage'])
        plt.show()
```

```
In [13]: plt.figure(figsize=(10,8))
         plt.subplot(2,2,1)
         plt.hist(visa_df['prevailing_wage'],bins=40)
         plt.subplot(2,2,2)
         plt.boxplot(visa_df['prevailing_wage'])
         plt.subplot(2,2,3)
         plt.hist(non_outliers_df['prevailing_wage'],bins=40)
         plt.subplot(2,2,4)
         plt.boxplot(non_outliers_df['prevailing_wage'])
         plt.show()
```
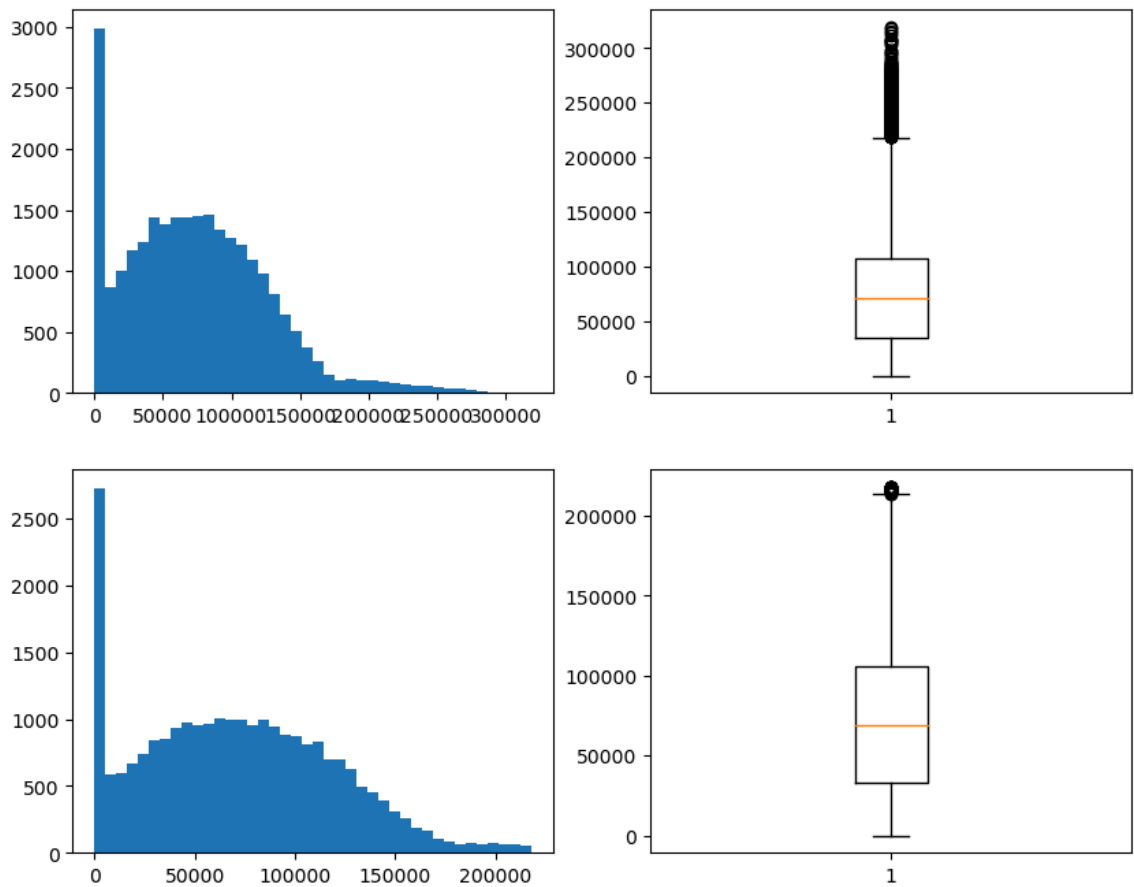
```
In [17]: plt.figure(figsize=(10,8))
         plt.subplot(2,2,1).hist(visa_df['prevailing_wage'],bins=40)
         plt.subplot(2,2,2).boxplot(visa_df['prevailing_wage'])
         plt.subplot(2,2,3).hist(non_outliers_df['prevailing_wage'],bins=40)
         plt.subplot(2,2,4).boxplot(non_outliers_df['prevailing_wage'])
         plt.show()
```


How to deal outliers

Drop the outliers

- we can drop the outliers if outlier percentage < 2%
- But this is not recommended , we lost other columns data also

Impute with Median values

- As we know that Median does not affect by outliers
- So it is good prcatice we can impute outliers with Median value

Cap with Q3 or Q1 value

- If outliers are presnt less than lower bound then fill with Q1
- If outliers are more than upper bound then fill with Q3

```
# Task 3
# Read the each observation from prevailing wage
# if that observation <lb or >ub : fill with median value
# else: keep as it is


# take empty list =[]
# median= visa_df['pwage'].medain()
# for  i in visa_df['pwage']:
#     if i<lb or i>ub:
#         emptylist.append(median)
#     else:
#         emptylist.append(i)

# 25480
```

```
# Import the packages
# Read the data


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

path=r"C:\Users\omkar\OneDrive\Documents\Data science\Naresh IT\Datafiles\V:
visa_df=pd.read_csv(path)
visa_df.head(3)
```

Out[1]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_
0	EZYV01	Asia	High School	N	N	
1	EZYV02	Asia	Master's	Y	N	
2	EZYV03	Asia	Bachelor's	N	Y	

```
In [3]:  #####------------ Step-1------###########
         Q1=np.percentile(visa_df['prevailing_wage'],25)
         Q2=np.percentile(visa_df['prevailing_wage'],50)
         Q3=np.percentile(visa_df['prevailing_wage'],75)

         #####------------ Step-2------###########
         IQR=Q3-Q1

         #####------------ Step-3------###########
         lb=Q1-1.5*IQR
         ub=Q3+1.5*IQR

         median=visa_df['prevailing_wage'].median()

         list1=[]
         for i in visa_df['prevailing_wage']:
             if i<lb or i>ub:
                 list1.append(median)
             else:
                 list1.append(i)

         visa_df['prevailing_wage_new']=list1
```
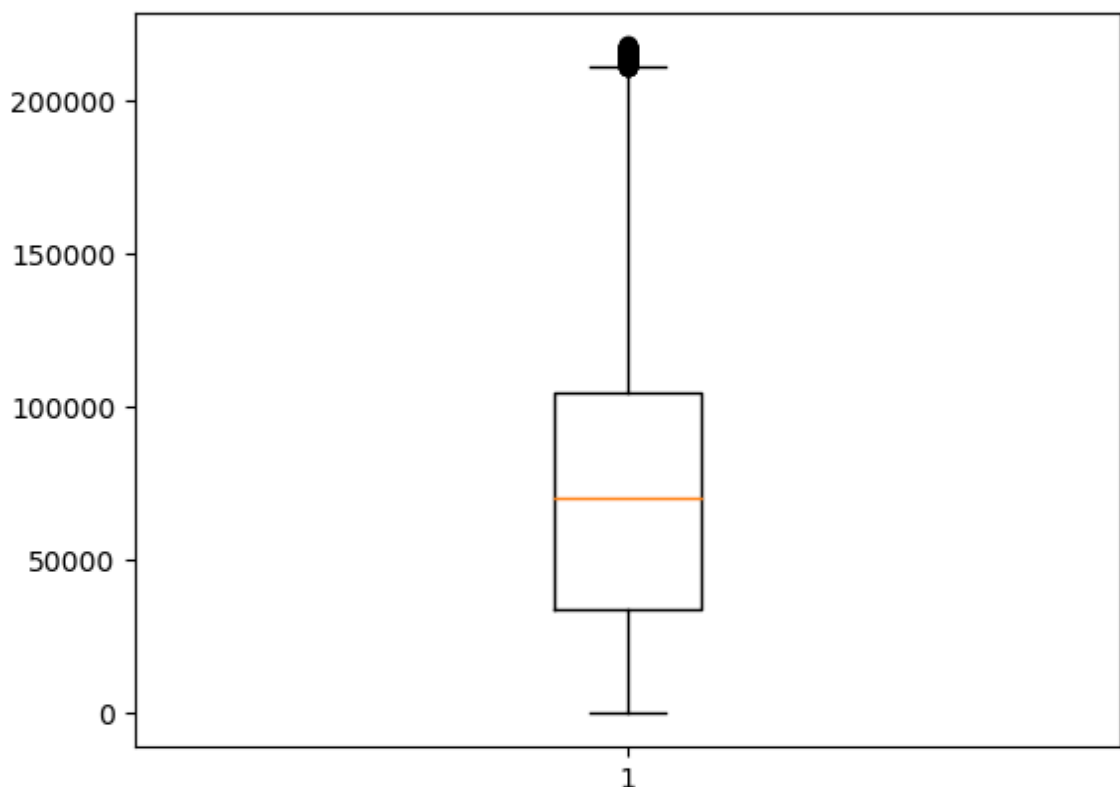
```
In [5]:  plt.boxplot(visa_df['prevailing_wage_new'])
         plt.show()
```


np.where

```
In [ ]:  # Read the data again

         path=r"C:\Users\omkar\OneDrive\Documents\Data science\Naresh IT\Datafiles\V
         visa_df=pd.read_csv(path)
         visa_df.head(3)
```

- above replace one we use a traditional approach
- for loop, list, if-else
- the same we can get by using np.where method

```
In [12]:  dict1={'Name':['A','B','C','D'],
                 'Num':[1,2,3,4]}
          d=pd.DataFrame(dict1)
          d
          d['Num']>2
```

```
Out[12]:  0    False
          1    False
          2     True
          3     True
          Name: Num, dtype: bool
```

```
In [10]:  # I want to replace with 100 num which has >2
          # other wise keep same number
          l=[]
          for i in d['Num']:
              if i>2:
                  l.append(100)
              else:
                  l.append(i)
          d['Num']=l
          d

          # How many condition  i>2 if it is True ==== one value
          #                          if it is false ==== another
```

Out[10]:

	Name	Num
0	A	1
1	B	2
2	C	100
3	D	100

np.where(con,True,False)

- Will take 3 arguments
  - Condition
    - con= d['Num']>2
  - True value
    - t=100
  - False value
    - f= d['Num']

```
In [14]: l=np.where(d['Num']>2,100,d['Num'])

d['Num']=l
d
```

Out[14]:

	Name	Num
0	A	1
1	B	2
2	C	100
3	D	100

```
In [16]:  #####------------- Step-1------###########
          Q1=np.percentile(visa_df['prevailing_wage'],25)
          Q2=np.percentile(visa_df['prevailing_wage'],50)
          Q3=np.percentile(visa_df['prevailing_wage'],75)

          #####------------- Step-2------###########
          IQR=Q3-Q1

          #####------------- Step-3------###########
          lb=Q1-1.5*IQR
          ub=Q3+1.5*IQR

          median=visa_df['prevailing_wage'].median()

          c1=visa_df['prevailing_wage']<lb
          c2=visa_df['prevailing_wage']>ub
          con=c1|c2

          t=median
          f=visa_df['prevailing_wage']

          visa_df['prevailing_wage']=np.where(con,t,f)
          plt.boxplot(visa_df['prevailing_wage'])
          plt.show()
```
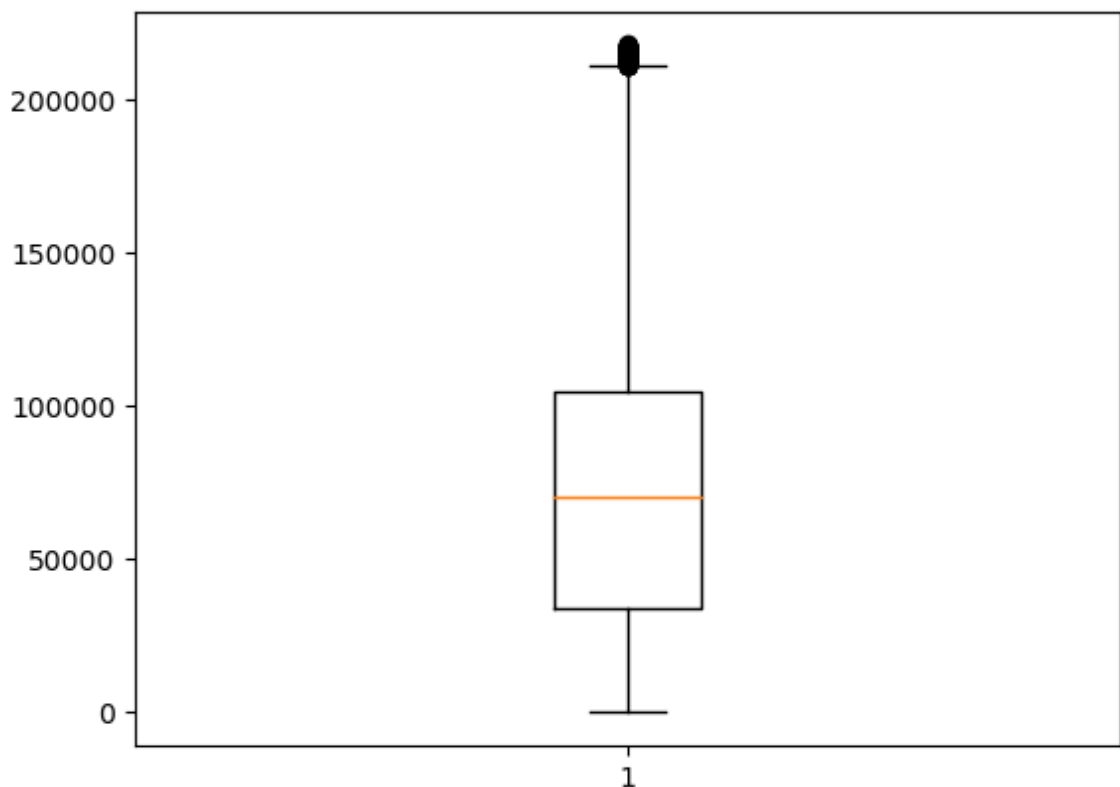
```
In [15]: dict1={'Name':['A','B','C','D'],'Num':[1,2,3,4]}
         d1=pd.DataFrame(dict1)
         d1
         l=[]
         for i in d1['Num']:
             if i >2:
                 l.append(100)
             else:
                 l.append(i)
         d1['Num']=l
         d1
```

Out[15]:

	Name	Num
0	A	1
1	B	2
2	C	100
3	D	100

In [ ]: