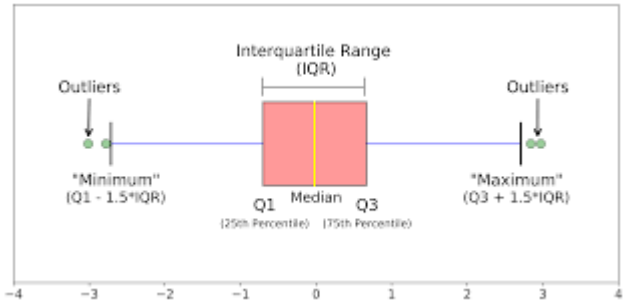
In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
file_path='C:\\Users\\kurre\\OneDrive\\Documents\\Naresh IT\\datafiles\\Vis
visa_df=pd.read_csv(file_path)
visa_df
```

Out[2]:

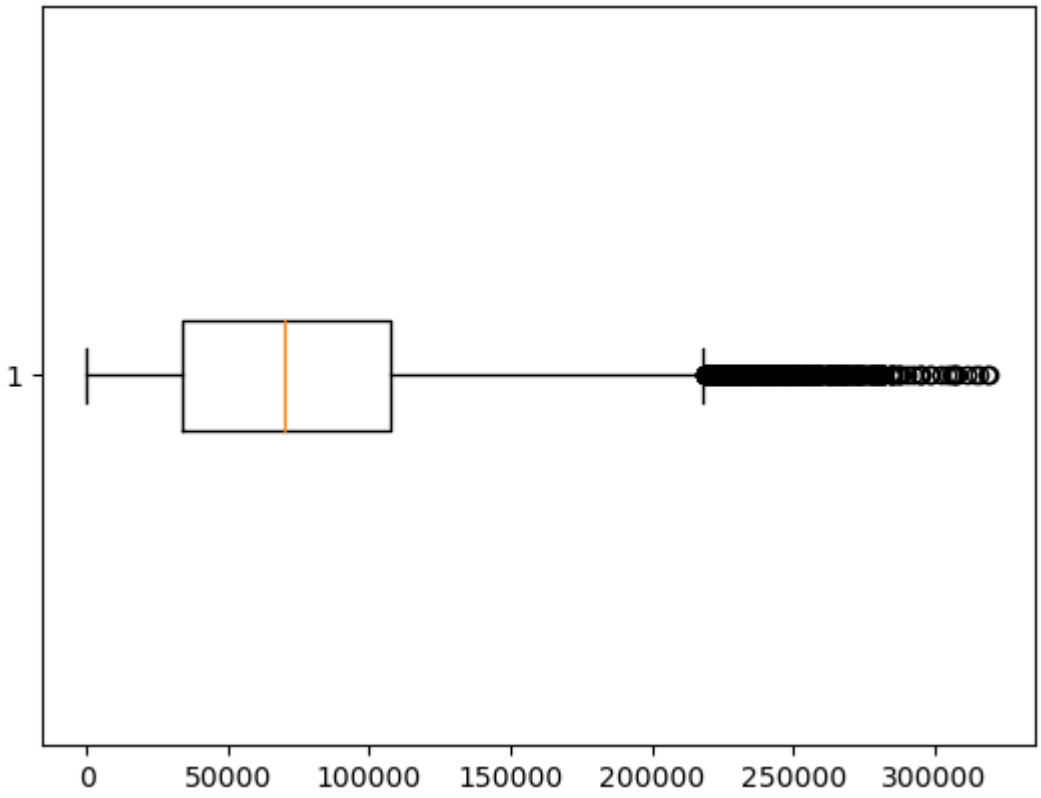
	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
0	EZYV01	Asia	High School	N	
1	EZYV02	Asia	Master's	Y	
2	EZYV03	Asia	Bachelor's	N	
3	EZYV04	Asia	Bachelor's	N	
4	EZYV05	Africa	Master's	Y	
...	...	...	...	...	
25475	EZYV25476	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

25480 rows × 12 columns



In [3]:
```
plt.boxplot(visa_df['prevailing_wage'],
            vert=False)
plt.show()


#black dots are outliers
```



Dealing-Outliers

- Removal of outliers
- Impute the outliers with medain value
  - because medain is not impact by Outliers
- Cap the outliers with Q3, which are having more than Q3
- Cap the outliers with Q1, which are having less than Q1

Find the outliers

- Q3+1.5*IQR> and Q1-1.5*IQR
- Step-1: Calculate Q1 Q2 Q3
- Step-2: Calculate IQR=(Q3-Q1)
- Step-3: UB=Q3+1.5*IQR
- Step-4: LB=Q1-1.5*IQR
- Step-5: con1= col>UB
- Step-6: con2= col<LB
- Step-7: con1|con2
- Step-8: col[con1|con2]

In [6]:
```
#Step-1: Calculate Q1 Q2 Q3
q1=np.quantile(visa_df['prevailing_wage'],0.25)
q2=np.quantile(visa_df['prevailing_wage'],0.50)
q3=np.quantile(visa_df['prevailing_wage'],0.75)

#Step-2:Calculate IQR=(Q3-Q1)
IQR=q3-q1

#Step-3: UB=Q3+1.5*IQR
ub=q3+1.5*IQR

#Step-4: LB=Q1-1.5*IQR
lb=q1-1.5*IQR

#Step-5: con1= col>UB
#Step-6: con2= col<LB

con1=visa_df['prevailing_wage']>ub
con2=visa_df['prevailing_wage']<lb

#step-7 and step-8
outliers=visa_df['prevailing_wage'][con1|con2]

# series into array of values by applying a .values
outliers_data=outliers.values
len(outliers_data)
```

Out[6]: 427

In [7]:
```
def outliers():
    q1=np.quantile(visa_df['prevailing_wage'],0.25)
    q2=np.quantile(visa_df['prevailing_wage'],0.50)
    q3=np.quantile(visa_df['prevailing_wage'],0.75)
    IQR=q3-q1
    ub=q3+1.5*IQR
    lb=q1-1.5*IQR
    con1=visa_df['prevailing_wage']>ub
    con2=visa_df['prevailing_wage']<lb
    ################################################
    outliers=visa_df['prevailing_wage'][con1|con2]
    ################################################
    outliers_data=outliers.values
    return(outliers_data)

outliers_data=outliers()
len(outliers_data)
```

Out[7]: 427

In [8]:
```
len(outliers_data),len(visa_df),len(outliers_data)*100/len(visa_df)
```

Out[8]: (427, 25480, 1.6758241758241759)

$Case-1$

Removal of outliers

- we have 427 outliers in pre_wage column
- that means we need to remove 427 rows from entire dataframe

In [9]:
```
q1=np.quantile(visa_df['prevailing_wage'],0.25)
q2=np.quantile(visa_df['prevailing_wage'],0.50)
q3=np.quantile(visa_df['prevailing_wage'],0.75)
IQR=q3-q1
ub=q3+1.5*IQR
lb=q1-1.5*IQR
con1=visa_df['prevailing_wage']<ub
con2=visa_df['prevailing_wage']>lb
################################################
non_outliers_df=visa_df[con1&con2]
################################################
non_outliers_df
```

Out[9]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
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	

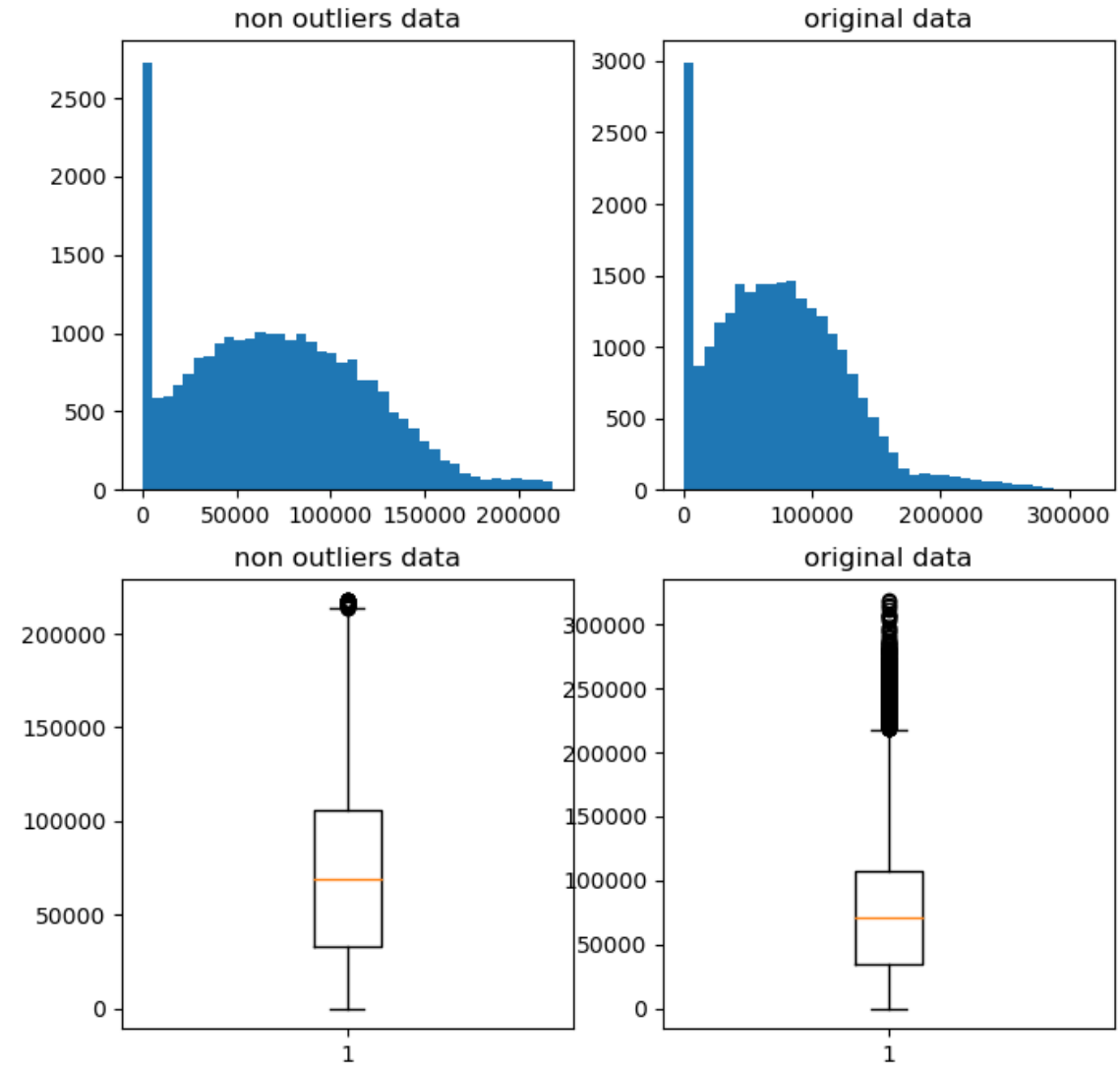
25053 rows × 12 columns

In [10]:
```
plt.figure(figsize=(8,8))
plt.subplot(2,2,1)
plt.title("non outliers data")
plt.hist(non_outliers_df['prevailing_wage'],bins=40)

plt.subplot(2,2,2)
plt.title("original data")
plt.hist(visa_df['prevailing_wage'],bins=40)

plt.subplot(2,2,3)
plt.title("non outliers data")
plt.boxplot(non_outliers_df['prevailing_wage'])

plt.subplot(2,2,4)
plt.title("original data")
plt.boxplot(visa_df['prevailing_wage'])
plt.show()
```



Case – 2:

Impute with Median

- We got pre_wage has 427 outliers

In [11]:
```
ub,lb
```

Out[11]:
```
(218315.56125000003, -76564.56875000002)
```

In [ ]:
```
# iterate through pre_wages as i
# if a value>ub or <lb ===== > median
# else: i
```

Task

In [25]:
```
new_values=[]
for i in visa_df['prevailing_wage'].values:
    # if condition:
    #   append median
    #else:
    #   append.i
```

```
  Cell In[25], line 6
    #   append.i
                ^
SyntaxError: incomplete input
```

In [26]:
```
# Import pacakages
# Read the data

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

In [27]:
```
file_path='C:\\Users\\kurre\\OneDrive\\Documents\\Naresh IT\\datafiles\\Vis
visa_df=pd.read_csv(file_path)
visa_df
```

Out[27]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
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	
...	...	...	...	...	
25475	EZYV25476	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

25480 rows × 12 columns

In [28]:
```
q1=np.quantile(visa_df['prevailing_wage'],0.25)
q2=np.quantile(visa_df['prevailing_wage'],0.50)
q3=np.quantile(visa_df['prevailing_wage'],0.75)
IQR=q3-q1
ub=q3+1.5*IQR
lb=q1-1.5*IQR
con1=visa_df['prevailing_wage']>ub
con2=visa_df['prevailing_wage']<lb
################################################
outliers=visa_df['prevailing_wage'][con1|con2]
################################################
len(outliers)
```

Out[28]: 427

In [29]:
```
new_data=[]
for i in visa_df['prevailing_wage']:
    if i>ub or i<lb:
        new_data.append(visa_df['prevailing_wage'].median)
    else:
        new_data.append(i)


len(new_data)


# We are iterate trough pre_wage data
# if any datapoint >ub or <lb means it is a outliers so in that postition
#                   we are keeping medain value of the column

# otherwise we are keeping the same value
```

Out[29]: 25480

np. where

In [30]:
```
dict1={'Col1':[1,2,3,4],
      'Col2':['A','B','C','D']}

data=pd.DataFrame(dict1)
data

# I want to impute with a value 100 in the col1
# which are having values >2

# Col1  Col2
#  1     A
#  2     B
#  100   C
#  100   D
```

Out[30]:

	Col1	Col2
0	1	A
1	2	B
2	3	C
3	4	D

- np.where will take 3 argument values
- Condition : con=data['Col1']>2
- If that condition is True will provide the value:100
- If that condition is False will keep the same value: data['Col1']
- np.where(,,)

In [31]:
```
con=data['Col1']>2
np.where(con,100,data['Col1'])

# binary conditions
# True False
# if else
```

Out[31]:
```
array([  1,   2, 100, 100], dtype=int64)
```

In [32]:
```
data
```

Out[32]:

	Col1	Col2
0	1	A
1	2	B
2	3	C
3	4	D

$Case-1$

Create a column

In [33]:
```
data['new_col']=[100,200,300,400]
data
```

Out[33]:

	Col1	Col2	new_col
0	1	A	100
1	2	B	200
2	3	C	300
3	4	D	400

In [34]:
```
con=data['Col1']>2
data['Col3']=np.where(con,100,data['Col1'])
data
```

Out[34]:

	Col1	Col2	new_col	Col3
0	1	A	100	1
1	2	B	200	2
2	3	C	300	100
3	4	D	400	100

$Case-2$

Overwrite the column values

In [36]:
```
con=data['Col1']>2
data['Col1']=np.where(con,100,data['Col1'])
data
```

Out[36]:

	Col1	Col2	new_col	Col3
0	1	A	100	1
1	2	B	200	2
2	100	C	300	100
3	100	D	400	100

In [38]:
```
#Drop unwanted columns
data.drop(['new_col','Col3'],
          axis=1,
          inplace=True)
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
Cell In[38], line 2
      1 #Drop unwanted columns
----> 2 data.drop(['new_col','Col3'],
      3           axis=1,
      4           inplace=True)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:5258, in DataFram
e.drop(self, labels, axis, index, columns, level, inplace, errors)
   5110 def drop(
   5111     self,
   5112     labels: IndexLabel = None,
   (...)
   5119     errors: IgnoreRaise = "raise",
   5120 ) -> DataFrame | None:
   5121     """
   5122     Drop specified labels from rows or columns.
   5123
   (...)
   5256             weight  1.0     0.8
   5257     """
-> 5258     return super().drop(
   5259         labels=labels,
   5260         axis=axis,
   5261         index=index,
   5262         columns=columns,
   5263         level=level,
   5264         inplace=inplace,
   5265         errors=errors,
   5266     )

File ~\anaconda3\Lib\site-packages\pandas\core\generic.py:4549, in NDFram
e.drop(self, labels, axis, index, columns, level, inplace, errors)
   4547 for axis, labels in axes.items():
   4548     if labels is not None:
-> 4549         obj = obj._drop_axis(labels, axis, level=level, errors=err
ors)
   4551 if inplace:
   4552     self._update_inplace(obj)

File ~\anaconda3\Lib\site-packages\pandas\core\generic.py:4591, in NDFram
e._drop_axis(self, labels, axis, level, errors, only_slice)
   4589         new_axis = axis.drop(labels, level=level, errors=errors)
   4590     else:
-> 4591         new_axis = axis.drop(labels, errors=errors)
   4592     indexer = axis.get_indexer(new_axis)
   4594 # Case for non-unique axis
   4595 else:

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6699, in In
dex.drop(self, labels, errors)
   6697 if mask.any():
   6698     if errors != "ignore":
-> 6699         raise KeyError(f"{list(labels[mask])} not found in axis")
   6700     indexer = indexer[~mask]
   6701 return self.delete(indexer)
```

```
KeyError: "['new_col', 'Col3'] not found in axis"
```

In [ ]:
```
data
```

Task

Implement the same thing for prevailing wage

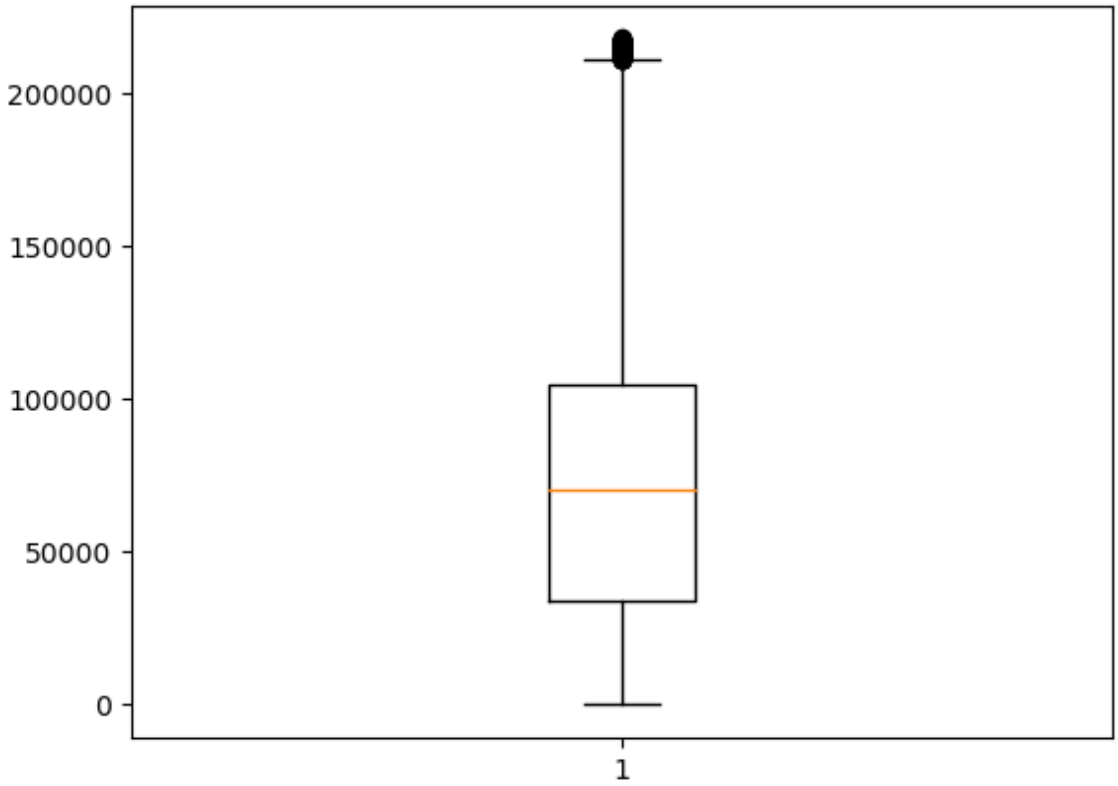
In [ ]:
```
# step-1: write the condition
# step-2: True value: Medain value
# Step-3: False value: same column values
# Step-4: implment np.where(<con1>,<True_vale>,<False_vale>)
# Step-5: Overwrite in the same column name
# Step-6: Draw the boxplot for p_Wage
# Step-7: Daraw the histogram  p_wage
```

In [41]:
```
###################### Read the data ###################################
file_path='C:\\Users\\kurre\\OneDrive\\Documents\\Naresh IT\\datafiles\\Vis
visa_df=pd.read_csv(file_path)
visa_df

##########################np.where#############################################
con1=visa_df['prevailing_wage']>ub
con2=visa_df['prevailing_wage']<lb
con=con1|con2
wage_median=visa_df['prevailing_wage'].median()
visa_df['prevailing_wage']=np.where(con,
                                    wage_median,
                                    visa_df['prevailing_wage'])
```

In [42]:
```
plt.boxplot(visa_df['prevailing_wage'])
```

Out[42]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x29e424c8d50>,
  <matplotlib.lines.Line2D at 0x29e424ca7d0>],
 'caps': [<matplotlib.lines.Line2D at 0x29e424bb450>,
  <matplotlib.lines.Line2D at 0x29e424b9190>],
 'boxes': [<matplotlib.lines.Line2D at 0x29e424c9790>],
 'medians': [<matplotlib.lines.Line2D at 0x29e424b8b50>],
 'fliers': [<matplotlib.lines.Line2D at 0x29e424c46d0>],
 'means': []}
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: