Import required packages

In [1]:

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

Read the data

In [2]:

```
file_location="C:\\Users\\omkar\\OneDrive\\Documents\\Data science\\Naresh I
visa_df=pd.read_csv(file_location)
visa_df.head()
```

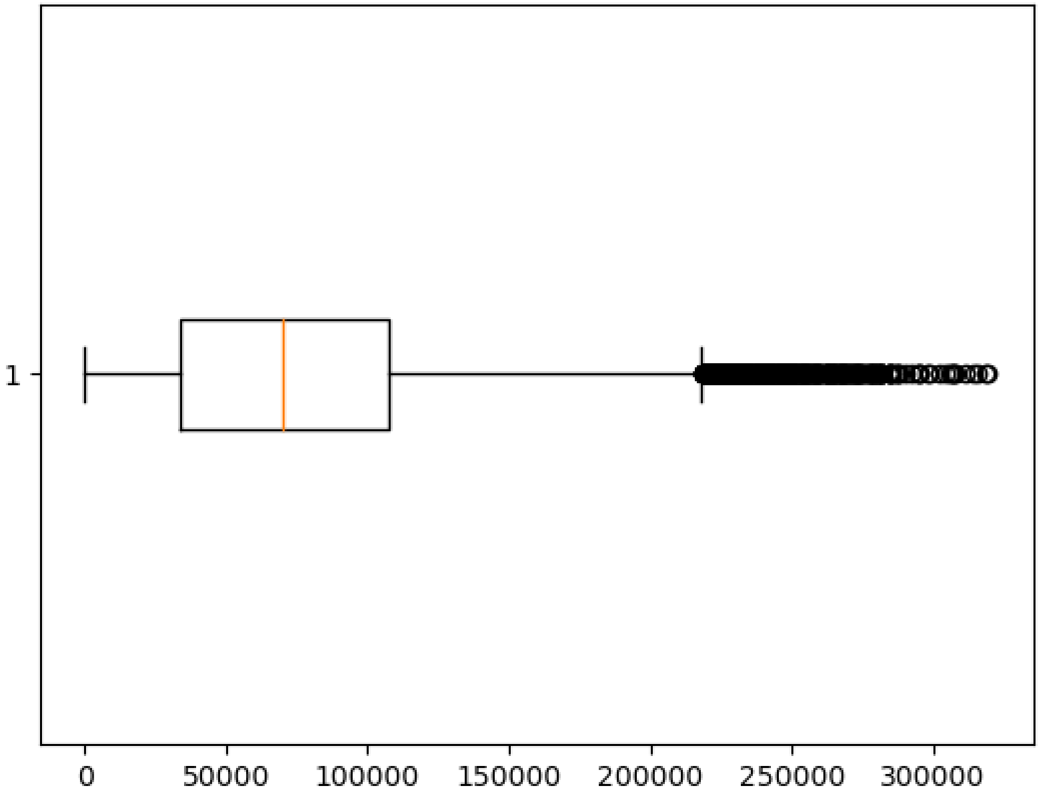
Out[2]:

	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	
3	EZYV04	Asia	Bachelor's	N	N	
4	EZYV05	Africa	Master's	Y	N	

In [6]:

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

# orange line : Median value = 50P data
# Black dots all are outliers
```



```
      Q1-1.5IQR   Q1   median  Q3   Q3+1.5IQR
                  |-----:-----|
  o      |--------|     :     |--------|    o  o
                  |-----:-----|
flier             <----------->            fliers
                       IQR
```

Procedure to find the Outliers

$Step - 1:$

- Calculate Q1 Q2 and Q3

$Step - 2:$

- Calculate IQR=(Q3-Q1)

$Step - 3:$

- Calculate UB=Q3+1.5*IQR
- Calculate LB=Q1-1.5*IQR

$Step - 4:$

- Find the outliers which are having greater than UB
- Find the outliers which are having less than LB

In [13]:

```python
# Step-1
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
IQR=Q3-Q1

#step-3
UB=Q3+1.5*IQR
LB=Q1-1.5*IQR
UB,LB

#Step-4
#>UB  <LB are the outliers
con1=visa_df['prevailing_wage']>UB
con2=visa_df['prevailing_wage']<LB

#Step-5
# if you apply | with outlier
outliers_df=visa_df[con1|con2]
```

In [19]:

```
def outliers(col):
    Q1=np.quantile(visa_df[col],0.25)
    Q2=np.quantile(visa_df[col],0.50)
    Q3=np.quantile(visa_df[col],0.75)
    IQR=Q3-Q1
    UB=Q3+1.5*IQR
    LB=Q1-1.5*IQR
    con1=visa_df[col]>UB
    con2=visa_df[col]<LB
    outliers_df=visa_df[con1|con2]
    print(f'{col} has {len(outliers_df)} outliers')
    print('{} has {} outliers'.format(col,len(outliers_df)))


num_col=visa_df.select_dtypes(exclude='object').columns
for col in num_col:
    outliers(col)
```

```
no_of_employees has 1556 outliers
no_of_employees has 1556 outliers
yr_of_estab has 3260 outliers
yr_of_estab has 3260 outliers
prevailing_wage has 427 outliers
prevailing_wage has 427 outliers
```

In [20]:

```
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
############ Outliers df #################
con1=visa_df['prevailing_wage']>UB
con2=visa_df['prevailing_wage']<LB
outliers_df=visa_df[con1|con2]
########## Non outliers df ###############
con11=visa_df['prevailing_wage']<UB
con22=visa_df['prevailing_wage']>LB
non_outliers_df=visa_df[con11&con22]
```

In [22]:

```
len(non_outliers_df),len(outliers_df)
```

Out[22]:

```
(25053, 427)
```

In [23]:

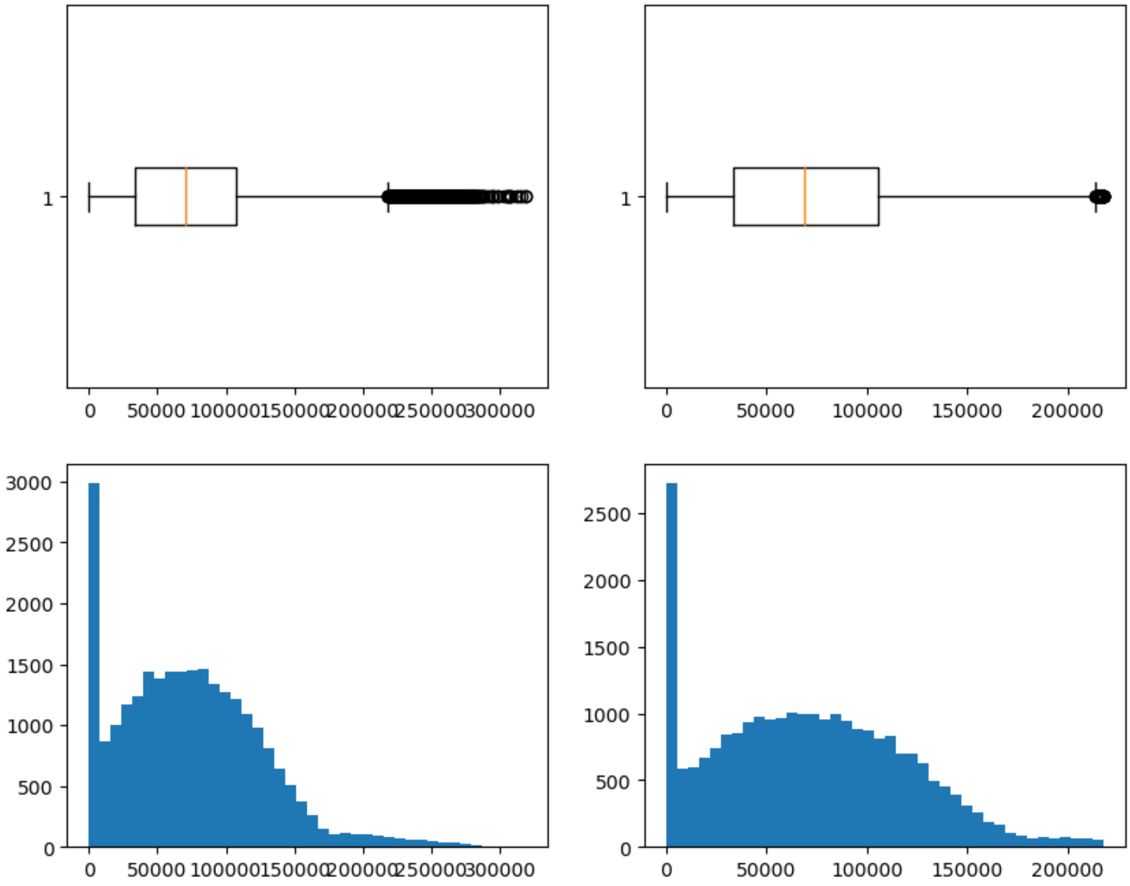
```
len(non_outliers_df)+len(outliers_df)
```

Out[23]:

```
25480
```

In [39]:

```
# Will compare
# Total data (25480) vs Non outliers data (25053)
plt.figure(figsize=(10,8))
plt.subplot(2,2,1)
plt.boxplot(visa_df['prevailing_wage'],vert=False)  # 25480
plt.subplot(2,2,2)
plt.boxplot(non_outliers_df['prevailing_wage'],vert=False) # 25053
plt.subplot(2,2,3)
plt.hist(visa_df['prevailing_wage'],bins=40)
plt.subplot(2,2,4)
plt.hist(non_outliers_df['prevailing_wage'],bins=40)
plt.show()
```



How to deal outliers

- Drop the outliers based some percentage
    - if you have very huge data
    - and the outliers percentage is <2 , then drop the outliers
    - Drop the outliers means , we are removing some rows all the columns
    - In the above examples total count=25480, outliers are =427 , 427*100/25480 = 1.6
    - After removing 427 observations, we have 25053 observation (98% of data)
- Impute (Fill) the outliers with Median value
    - We alreday know that outliers doesnt affect Median value
    - So if you dont want loss the data, and you want fill the outliers then use Median
- Impute (Fill) with UB and LB values (Capping)
    - Fill the outliers with UB value, which are having >UB
    - Fill the outliers with LB value, which are having <LB

In [ ]:

```
# Fill the outliers
# Missing values
# Bi variate multivariate
# Cate to num
# standard
# Transformation
# Feature selection
# PCA
```