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 [3]:
```
visa_df.dtypes
```

Out[3]:
```
case_id                   object
continent                 object
education_of_employee     object
has_job_experience        object
requires_job_training     object
no_of_employees            int64
yr_of_estab                int64
region_of_employment      object
prevailing_wage          float64
unit_of_wage              object
full_time_position        object
case_status               object
dtype: object
```

$prevailing - wage$

In [4]:
```
p_wage=visa_df['prevailing_wage']
p_wage
```

Out[4]:
```
0           592.2029
1         83425.6500
2        122996.8600
3         83434.0300
4        149907.3900
            ...
25475     77092.5700
25476    279174.7900
25477    146298.8500
25478     86154.7700
25479     70876.9100
Name: prevailing_wage, Length: 25480, dtype: float64
```

- count
- max
- min
- mean
- median
- 25p
- 50p
- 75p

In [5]:
```
p_wage.count()
```

Out[5]:
```
25480
```

In [6]:
```
p_wage=visa_df[['prevailing_wage']]
p_wage.count().iloc[0]

p_wage=visa_df['prevailing_wage']
p_wage.count()
```

Out[6]:
```
25480
```

In [7]:

```
p_wage=visa_df['prevailing_wage']
wage_count=p_wage.count()
wage_mean=round(p_wage.mean(),2)
wage_median=round(p_wage.median(),2)
wage_max=round(p_wage.max(),2)
wage_min=round(p_wage.min(),2)
# print(wage_count)
# print(wage_mean)
# print(wage_median)
# print(wage_max)
# print(wage_min)

list1=[wage_count,wage_max,wage_min,wage_mean,wage_median]
index_list=['count','max','min','mean','median']
pd.DataFrame(list1,
             columns=['prevailing_wage'],
             index=index_list)
```

Out[7]:

	prevailing_wage
count	25480.00
max	319210.27
min	2.14
mean	74455.81
median	70308.21

In [8]:

```
# Numerical columns seperaetly
num_cols=visa_df.select_dtypes(exclude='object').columns
dict1={}
for i in num_cols:
    count=visa_df[i].count()
    mean=round(visa_df[i].mean(),2)
    median=round(visa_df[i].median(),2)
    maxx=round(visa_df[i].max(),2)
    minn=round(visa_df[i].min(),2)
    list1=[count,maxx,minn,mean,median]
    dict1[i]=list1
index_list=['count','max','min','mean','median']
numer_df=pd.DataFrame(dict1,index=index_list)
numer_df.to_csv("numer_df.csv")
numer_df
```

Out[8]:

	no_of_employees	yr_of_estab	prevailing_wage
count	25480.00	25480.00	25480.00
max	602069.00	2016.00	319210.27
min	-26.00	1800.00	2.14
mean	5667.04	1979.41	74455.81
median	2109.00	1997.00	70308.21

In [9]:

```
visa_df.describe()
```

Out[9]:

	no_of_employees	yr_of_estab	prevailing_wage
count	25480.000000	25480.000000	25480.000000
mean	5667.043210	1979.409929	74455.814592
std	22877.928848	42.366929	52815.942327
min	-26.000000	1800.000000	2.136700
25%	1022.000000	1976.000000	34015.480000
50%	2109.000000	1997.000000	70308.210000
75%	3504.000000	2005.000000	107735.512500
max	602069.000000	2016.000000	319210.270000

In [10]:

```
p_wage=visa_df['prevailing_wage']
wage_count=p_wage.count()
wage_mean=round(p_wage.mean(),2)
wage_median=round(p_wage.median(),2)
wage_max=round(p_wage.max(),2)
wage_min=round(p_wage.min(),2)
wage_std=round(p_wage.std(),2)

list1=[wage_count,wage_max,wage_min,
       wage_mean,wage_median,wage_std]
index_list=['count','max','min','mean','median','std']
pd.DataFrame(list1,
             columns=['prevailing_wage'],
             index=index_list)
```

Out[10]:

	prevailing_wage
count	25480.00
max	319210.27
min	2.14
mean	74455.81
median	70308.21
std	52815.94

In [11]:

```
# what ever we did  the calculations on above
# by using pandas dataframe way
# the same we can achieve by numpy package also
```

In [12]:

```
# wage_mean=round(p_wage.mean(),2)=== pandas
p_wage=visa_df['prevailing_wage']
np.mean(p_wage)
np.median(p_wage)
np.max(p_wage)
np.min(p_wage)
np.std(p_wage)
```

Out[12]:

```
52814.90589711402
```

percentile-quantile

- In the numpy package we have np.percentile() and np.quantile()
- percentile: if you want to 25p 25
- quantile: q1=25p (0.25) q2=50p q3=75p
- Assume that a student got 120 Marks 95P
- 95% of students has marks below 120

In [13]:
```
np.percentile(p_wage,25)
```

Out[13]:
```
34015.479999999996
```

In [14]:
```
np.quantile(p_wage,0.25)
```

Out[14]:
```
34015.479999999996
```

In [15]:
```
p_wage=visa_df['prevailing_wage']
########### Pandas series #####################
wage_count=p_wage.count()
wage_mean=round(p_wage.mean(),2)
wage_median=round(p_wage.median(),2)
wage_max=round(p_wage.max(),2)
wage_min=round(p_wage.min(),2)
wage_std=round(p_wage.std(),2)
######### Numpy ############################
wage_25p=round(np.percentile(p_wage,25),2)
wage_50p=round(np.percentile(p_wage,50),2)
wage_75p=round(np.percentile(p_wage,75),2)

list1=[wage_count,wage_max,wage_min,
       wage_mean,wage_median,wage_std,
       wage_25p,wage_50p,wage_75p]

index_list=['count','max','min','mean',
            'median','std','25%','50%','75%']
pd.DataFrame(list1,
             columns=['prevailing_wage'],
             index=index_list)
```

Out[15]:

	prevailing_wage
count	25480.00
max	319210.27
min	2.14
mean	74455.81
median	70308.21
std	52815.94
25%	34015.48
50%	70308.21
75%	107735.51

In [16]:

```
# Numerical columns seperaetly
num_cols=visa_df.select_dtypes(exclude='object').columns
dict1={}
for i in num_cols:
    count=visa_df[i].count()
    mean=round(visa_df[i].mean(),2)
    median=round(visa_df[i].median(),2)
    maxx=round(visa_df[i].max(),2)
    minn=round(visa_df[i].min(),2)
    std=round(visa_df[i].std(),2)
    p25=round(np.percentile(visa_df[i],25),2)
    p50=round(np.percentile(visa_df[i],50),2)
    p75=round(np.percentile(visa_df[i],75),2)
    list1=[count,maxx,minn,mean,median,std,p25,p50,p75]
    dict1[i]=list1
index_list=['count','max','min','mean',
            'median','std','25%','50%','75%']
numer_df=pd.DataFrame(dict1,index=index_list)
numer_df.to_csv("numer_df.csv")
numer_df
```

Out[16]:

	no_of_employees	yr_of_estab	prevailing_wage
count	25480.00	25480.00	25480.00
max	602069.00	2016.00	319210.27
min	-26.00	1800.00	2.14
mean	5667.04	1979.41	74455.81
median	2109.00	1997.00	70308.21
std	22877.93	42.37	52815.94
25%	1022.00	1976.00	34015.48
50%	2109.00	1997.00	70308.21
75%	3504.00	2005.00	107735.51

In [17]:

```
#pwage 25p = 34015

#25% of total employees has wages below 34015

#100
#25 members salary  < 34k

50*(25480)/100

#6370 employees has wages less than 34015
#12740 employees has wages less than 70308.21
```

Out[17]:

```
12740.0
```

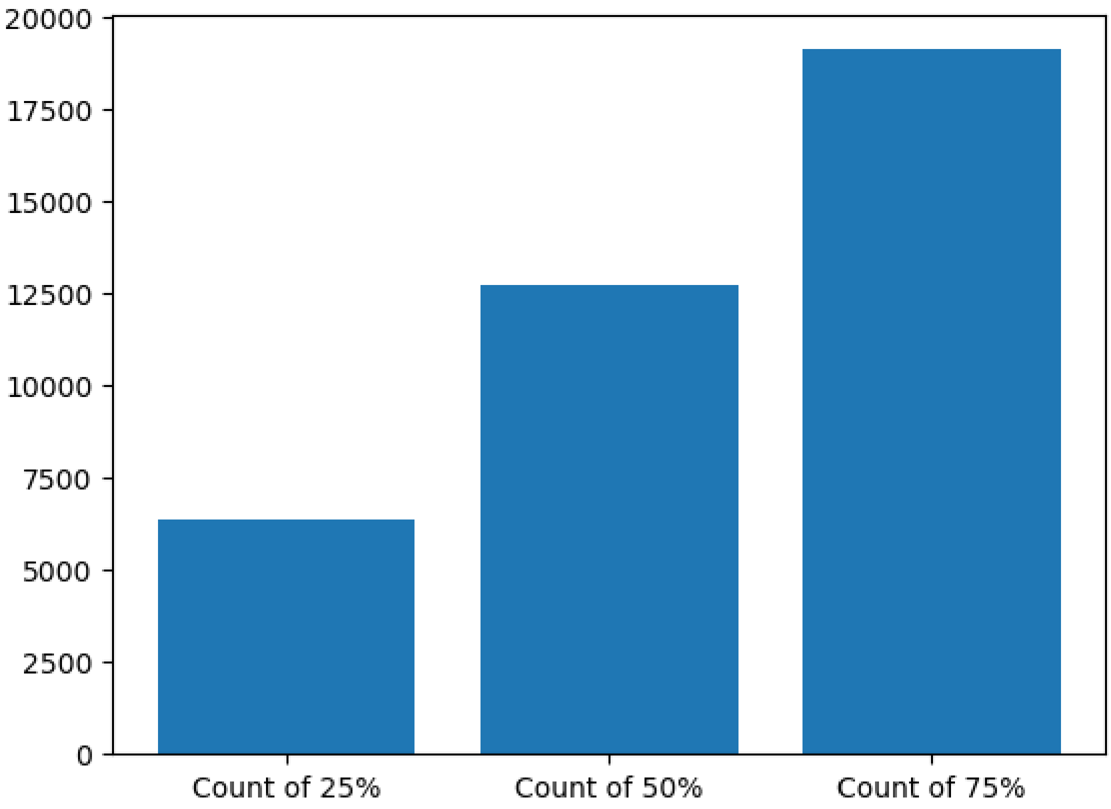
In [18]:

```
p_wage=visa_df['prevailing_wage']
count_25p=len(p_wage[p_wage<np.percentile(p_wage,25)])
count_50p=len(p_wage[p_wage<np.percentile(p_wage,50)])
count_75p=len(p_wage[p_wage<np.percentile(p_wage,75)])

l1=['Count of 25%','Count of 50%','Count of 75%']
l2=[count_25p,count_50p,count_75p]
d1=pd.DataFrame(zip(l1,l2),columns=['Till per','Count'])
plt.bar('Till per','Count',data=d1)
```

Out[18]:

```
<BarContainer object of 3 artists>
```



In [19]:

```
# You want to extract a dataframe
# which has wages less than 34015(25p)

# 100    25 mem    34k
```

In [20]:

```
# step-1:  take the reference column first
# Step-2: apply the condition
#         it will provide True or Flase
# Step-3: Apply the original dataframe on top of that
#         So that it will give only True values

p_wage=visa_df['prevailing_wage']
p_25=np.percentile(p_wage,25)
con=p_wage<p_25
visa_df[con]

visa_df[visa_df['prevailing_wage']<34015]
```

Out[20]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_trainii
0	EZYV01	Asia	High School	N	
7	EZYV08	North America	Bachelor's	Y	
12	EZYV13	Asia	Bachelor's	Y	
16	EZYV17	Europe	Master's	Y	
17	EZYV18	Asia	Master's	Y	
...	...	...	...	...	
25461	EZYV25462	Asia	Master's	Y	
25465	EZYV25466	North America	High School	N	
25466	EZYV25467	Europe	Bachelor's	Y	
25470	EZYV25471	North America	Master's	Y	
25473	EZYV25474	Asia	Bachelor's	Y	

6370 rows × 12 columns

In [21]:

```
p_wage=visa_df['prevailing_wage']
p_50=np.percentile(p_wage,50)
con=p_wage<p_50
visa_df[con]
```

Out[21]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
0	EZYV01	Asia	High School	N	
6	EZYV07	Asia	Bachelor's	N	
7	EZYV08	North America	Bachelor's	Y	
9	EZYV10	Europe	Doctorate	Y	
12	EZYV13	Asia	Bachelor's	Y	
...	...	...	...	...	
25465	EZYV25466	North America	High School	N	
25466	EZYV25467	Europe	Bachelor's	Y	
25470	EZYV25471	North America	Master's	Y	
25473	EZYV25474	Asia	Bachelor's	Y	
25474	EZYV25475	Africa	Doctorate	N	

12740 rows × 12 columns

In [22]:

```
# between 25p  to   50p
# between 34k  to   70k
# >25p and <50p
p_wage=visa_df['prevailing_wage']
p_25=np.percentile(p_wage,25)
p_50=np.percentile(p_wage,50)

# between 25p  to   50p

con1=p_wage>p_25
con2=p_wage<p_50

visa_df[con1&con2]

#visa_df[(visa_df['prevailing_wage']>34015)&(visa_df['prevailing_wage']<700
```

Out[22]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
6	EZYV07	Asia	Bachelor's	N	
9	EZYV10	Europe	Doctorate	Y	
22	EZYV23	Asia	Master's	Y	
28	EZYV29	Asia	Master's	Y	
38	EZYV39	Asia	Bachelor's	Y	
...	...	...	...	...	
25449	EZYV25450	Asia	Bachelor's	Y	
25454	EZYV25455	Asia	Bachelor's	N	
25456	EZYV25457	Asia	Bachelor's	Y	
25459	EZYV25460	Asia	High School	Y	
25474	EZYV25475	Africa	Doctorate	N	

6370 rows × 12 columns

In [24]:

```
# till 50 =12740
# till 25 =6370
# between 25 to 50 = 12740-6370=6370
```

In [25]:

```python
p_wage =visa_df['prevailing_wage']
p_25 = np.percentile(p_wage,25)
p_75 = np.percentile(p_wage,75)
con_1 = p_wage<p_25
con_2 = p_wage>p_75
visa_df[con_1 | con_2]
```

Out[25]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
0	EZYV01	Asia	High School	N	
2	EZYV03	Asia	Bachelor's	N	
4	EZYV05	Africa	Master's	Y	
7	EZYV08	North America	Bachelor's	Y	
12	EZYV13	Asia	Bachelor's	Y	
...	...	...	...	...	
25469	EZYV25470	North America	Master's	Y	
25470	EZYV25471	North America	Master's	Y	
25473	EZYV25474	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	

12740 rows × 12 columns

In [ ]:

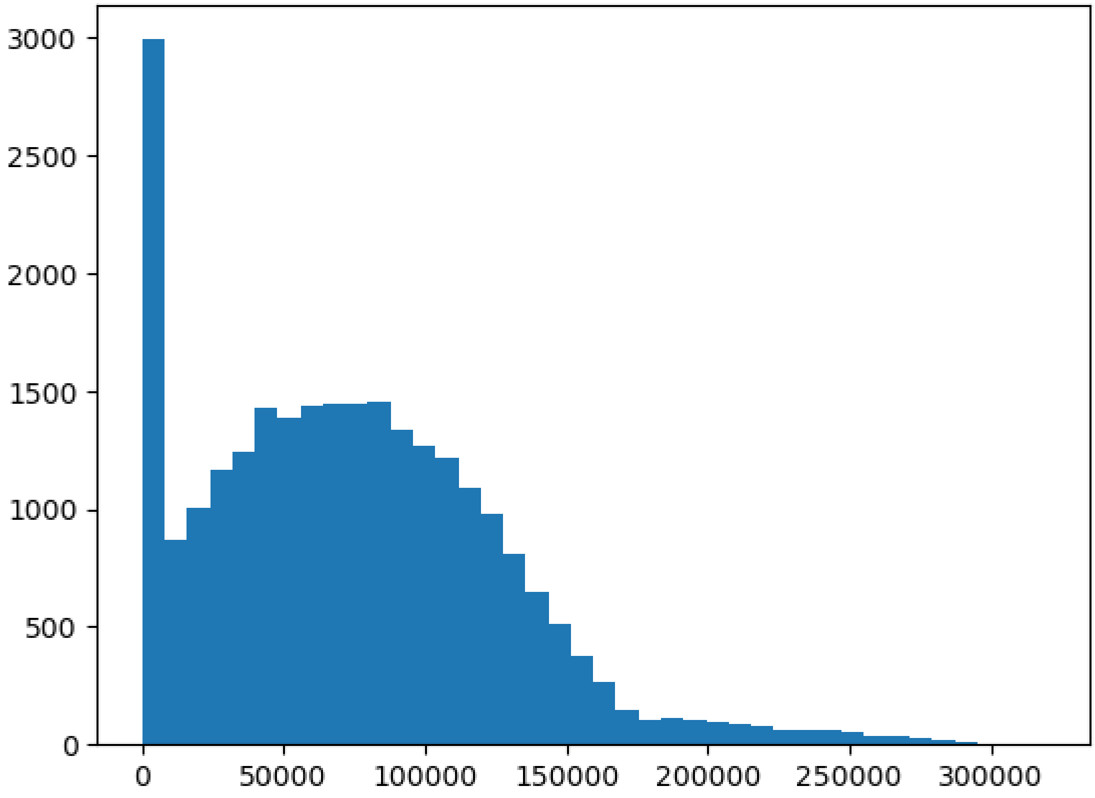
```python
# You are good at writing the conditions
```

Histogram

- From raw data will make class intervals
- Will count the observations in each class intervals
- Freqiency distribution table
- Plot of Frequency distribution table is Hitogram

In [26]:
```
p_wage=visa_df['prevailing_wage']
freq,interval,n=plt.hist(p_wage,bins=40)
freq,interval
```

Out[26]:
```
(array([2992.,  871., 1005., 1170., 1242., 1434., 1385., 1443., 1444.,
        1445., 1457., 1335., 1268., 1217., 1088.,  978.,  807.,  645.,
         509.,  373.,  264.,  144.,  105.,  111.,  107.,   99.,   88.,
          79.,   65.,   64.,   58.,   53.,   33.,   33.,   29.,   19.,
           7.,    3.,    6.,    5.]),
 array([2.13670000e+00, 7.98234003e+03, 1.59625434e+04, 2.39427467e+04,
        3.19229500e+04, 3.99031534e+04, 4.78833567e+04, 5.58635600e+04,
        6.38437634e+04, 7.18239667e+04, 7.98041700e+04, 8.77843734e+04,
        9.57645767e+04, 1.03744780e+05, 1.11724983e+05, 1.19705187e+05,
        1.27685390e+05, 1.35665593e+05, 1.43645797e+05, 1.51626000e+05,
        1.59606203e+05, 1.67586407e+05, 1.75566610e+05, 1.83546813e+05,
        1.91527017e+05, 1.99507220e+05, 2.07487423e+05, 2.15467627e+05,
        2.23447830e+05, 2.31428033e+05, 2.39408237e+05, 2.47388440e+05,
        2.55368643e+05, 2.63348847e+05, 2.71329050e+05, 2.79309253e+05,
        2.87289457e+05, 2.95269660e+05, 3.03249863e+05, 3.11230067e+05,
        3.19210270e+05]))
```



In [27]:
```
2.13670000e+00   #  2.13
7.98234003e+03   #  7982
```

Out[27]:
```
7982.34003
```

In [28]:
```
#2.13    to    7982.34003     (2992)

p_wage=visa_df['prevailing_wage']
con1=p_wage>2.13
con2=p_wage<7982.34003
len(visa_df[con1&con2])
```

Out[28]:
```
2992
```

In [29]:

```
p_wage=visa_df['prevailing_wage']
con1=p_wage>7.98234003e+03
con2=p_wage<1.59625434e+04
len(visa_df[con1&con2])
```

Out[29]:

```
871
```

In [ ]:

```
# # Histogram
# # what do you want represent in graphical way

# p_wage.values

# # raw observations
# # 25480 observations
# # we are dividng into 40 intervals

# l1=sorted(p_wage.values)
# l1.index(2.1367) #0
# l1.index()  #  2991   2992
```

In [30]:

```
freq
```

Out[30]:

```
array([2992.,  871., 1005., 1170., 1242., 1434., 1385., 1443., 1444.,
       1445., 1457., 1335., 1268., 1217., 1088.,  978.,  807.,  645.,
        509.,  373.,  264.,  144.,  105.,  111.,  107.,   99.,   88.,
         79.,   65.,   64.,   58.,   53.,   33.,   33.,   29.,   19.,
          7.,    3.,    6.,    5.])
```

In [31]:

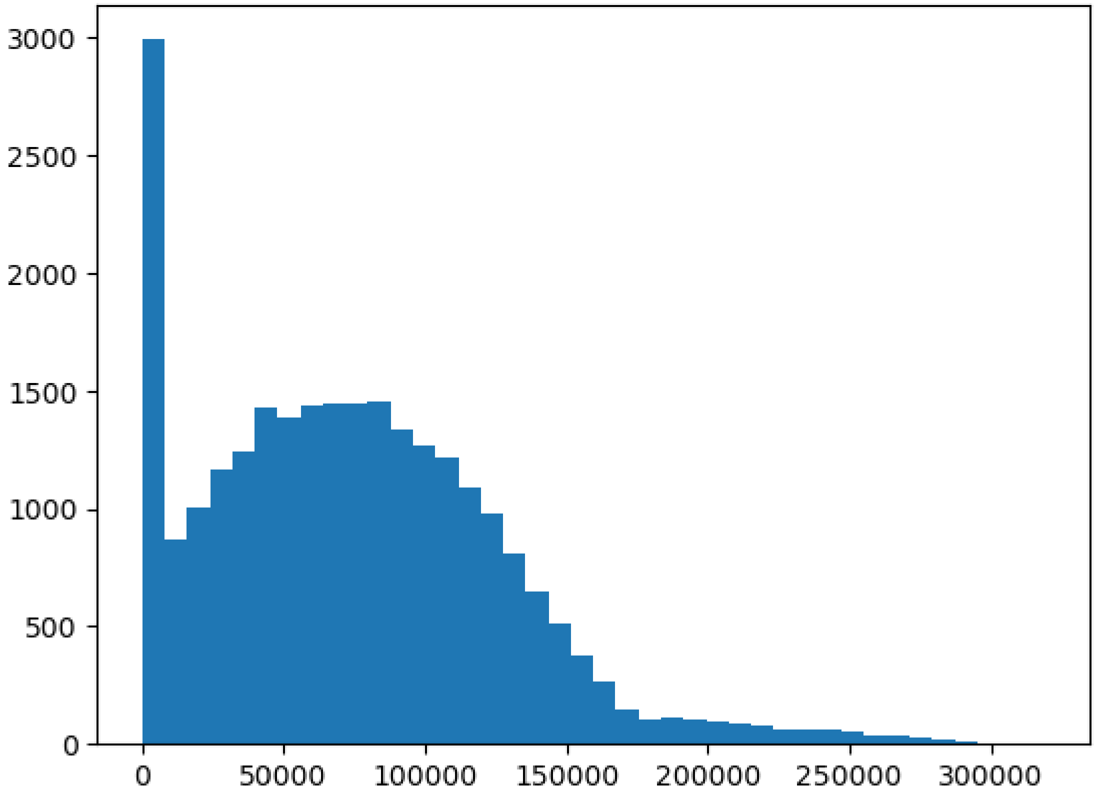
```
interval
```

Out[31]:

```
array([2.13670000e+00, 7.98234003e+03, 1.59625434e+04, 2.39427467e+04,
       3.19229500e+04, 3.99031534e+04, 4.78833567e+04, 5.58635600e+04,
       6.38437634e+04, 7.18239667e+04, 7.98041700e+04, 8.77843734e+04,
       9.57645767e+04, 1.03744780e+05, 1.11724983e+05, 1.19705187e+05,
       1.27685390e+05, 1.35665593e+05, 1.43645797e+05, 1.51626000e+05,
       1.59606203e+05, 1.67586407e+05, 1.75566610e+05, 1.83546813e+05,
       1.91527017e+05, 1.99507220e+05, 2.07487423e+05, 2.15467627e+05,
       2.23447830e+05, 2.31428033e+05, 2.39408237e+05, 2.47388440e+05,
       2.55368643e+05, 2.63348847e+05, 2.71329050e+05, 2.79309253e+05,
       2.87289457e+05, 2.95269660e+05, 3.03249863e+05, 3.11230067e+05,
       3.19210270e+05])
```

```
In [33]: plt.hist(visa_df['prevailing_wage'],bins=40)
```

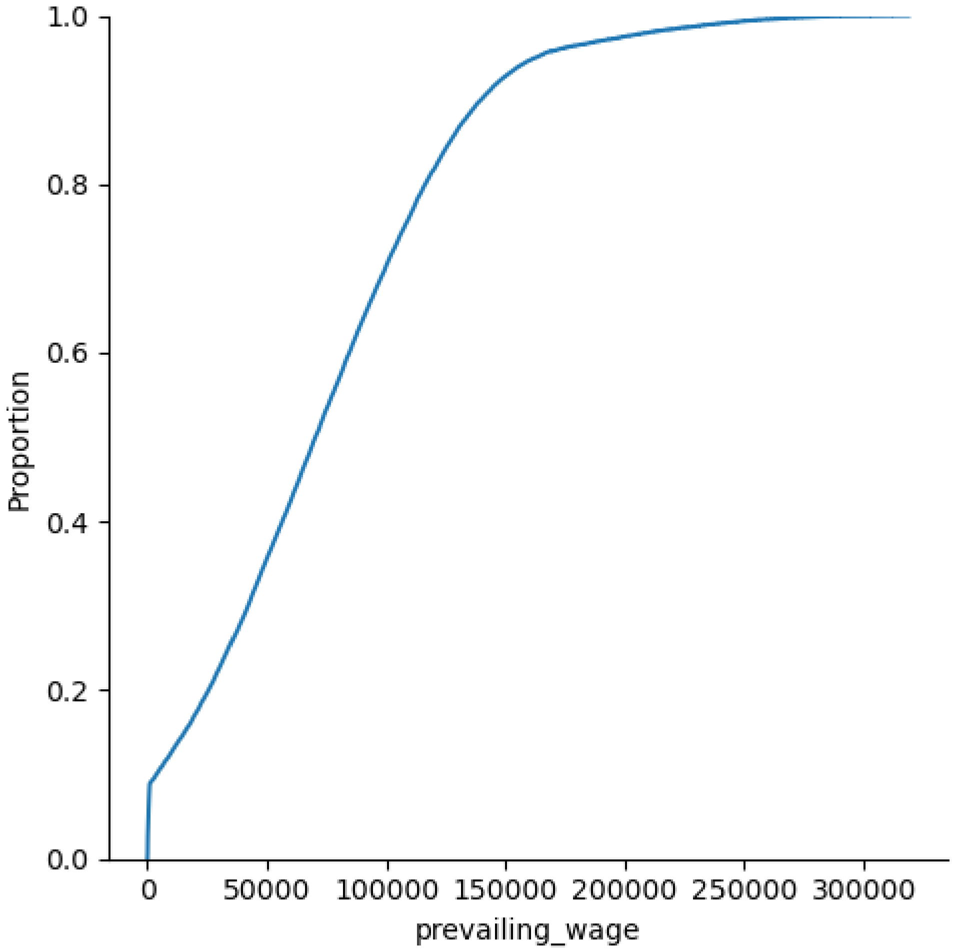
```
Out[33]: (array([2992.,  871., 1005., 1170., 1242., 1434., 1385., 1443., 1444.,
                 1445., 1457., 1335., 1268., 1217., 1088.,  978.,  807.,  645.,
                  509.,  373.,  264.,  144.,  105.,  111.,  107.,   99.,   88.,
                   79.,   65.,   64.,   58.,   53.,   33.,   33.,   29.,   19.,
                    7.,    3.,    6.,    5.]),
          array([2.13670000e+00, 7.98234003e+03, 1.59625434e+04, 2.39427467e+04,
                 3.19229500e+04, 3.99031534e+04, 4.78833567e+04, 5.58635600e+04,
                 6.38437634e+04, 7.18239667e+04, 7.98041700e+04, 8.77843734e+04,
                 9.57645767e+04, 1.03744780e+05, 1.11724983e+05, 1.19705187e+05,
                 1.27685390e+05, 1.35665593e+05, 1.43645797e+05, 1.51626000e+05,
                 1.59606203e+05, 1.67586407e+05, 1.75566610e+05, 1.83546813e+05,
                 1.91527017e+05, 1.99507220e+05, 2.07487423e+05, 2.15467627e+05,
                 2.23447830e+05, 2.31428033e+05, 2.39408237e+05, 2.47388440e+05,
                 2.55368643e+05, 2.63348847e+05, 2.71329050e+05, 2.79309253e+05,
                 2.87289457e+05, 2.95269660e+05, 3.03249863e+05, 3.11230067e+05,
                 3.19210270e+05]),
          <BarContainer object of 40 artists>)
```



```
In [45]: sns.displot(visa_df['prevailing_wage'],kind='ecdf')
         # ecdf= coumulative distribution plot
```

```
C:\Users\omkar\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWa
rning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

```
Out[45]: <seaborn.axisgrid.FacetGrid at 0x20b15188e10>
```



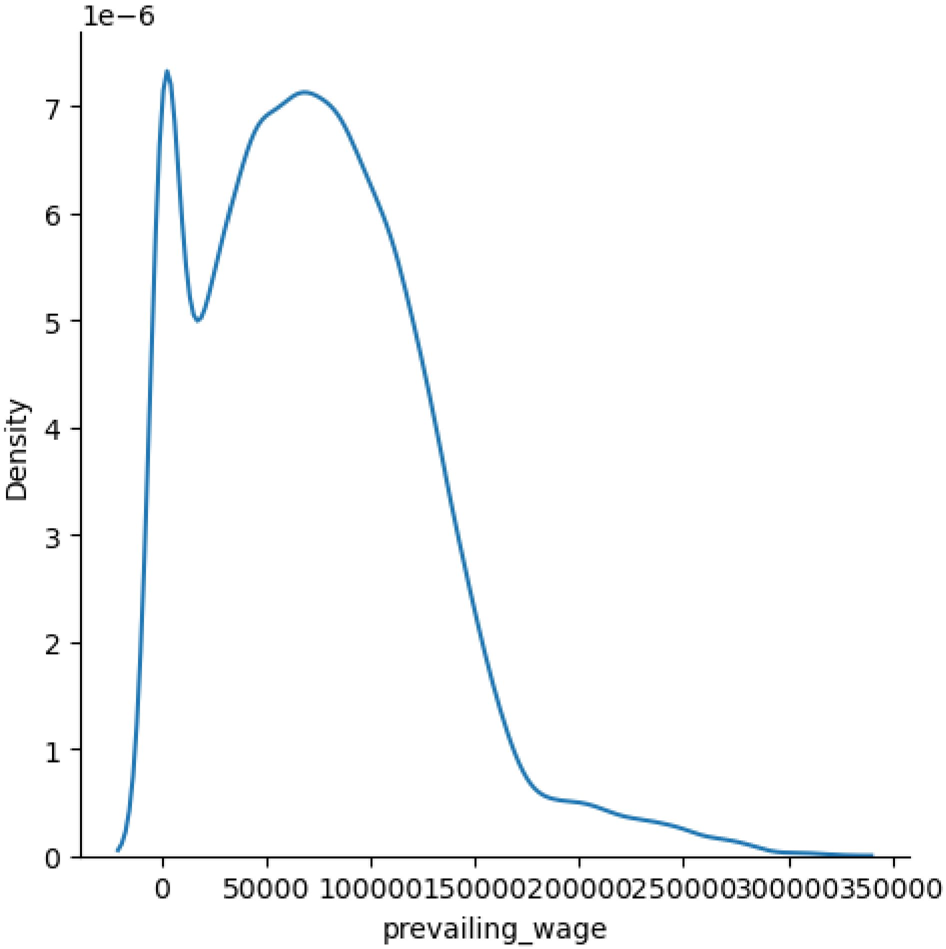
In [43]:

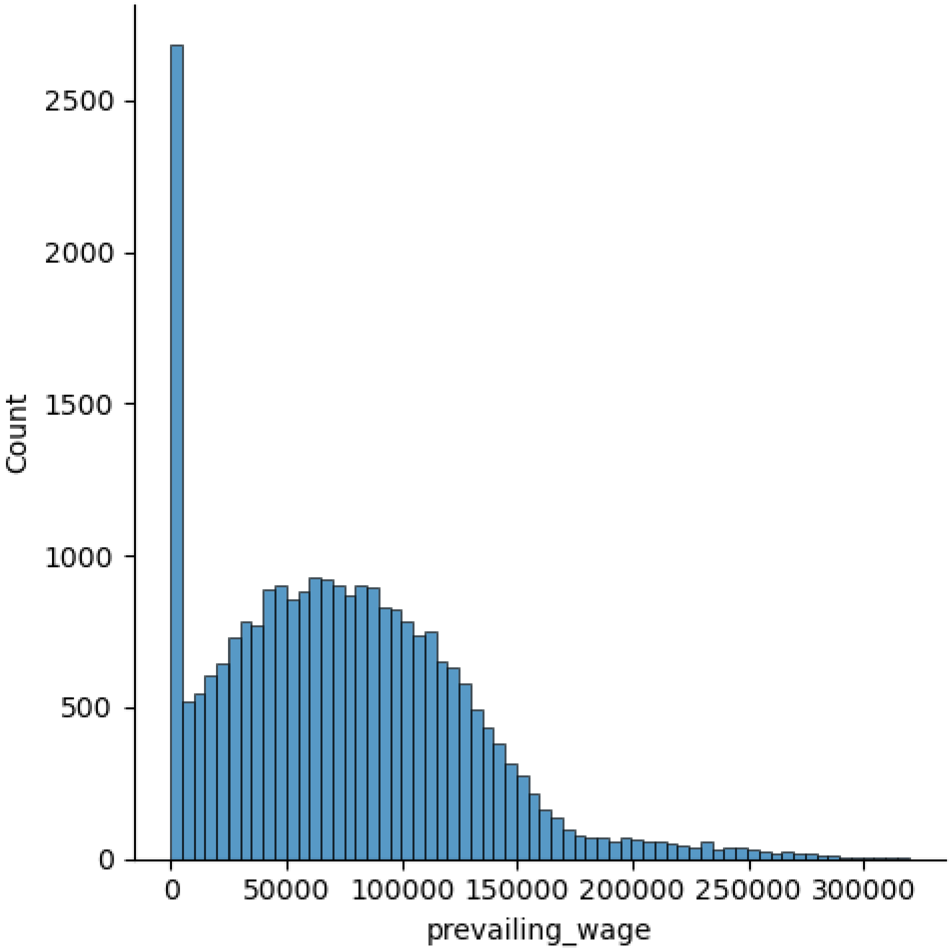
```
sns.displot(visa_df['prevailing_wage'],kind='kde') # kernal density estimat
sns.displot(visa_df['prevailing_wage'])
# ecdf= coumulative distribution plot
```

```
C:\Users\omkar\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWa
rning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
C:\Users\omkar\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWa
rning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[43]:

```
<seaborn.axisgrid.FacetGrid at 0x20b1d361e10>
```

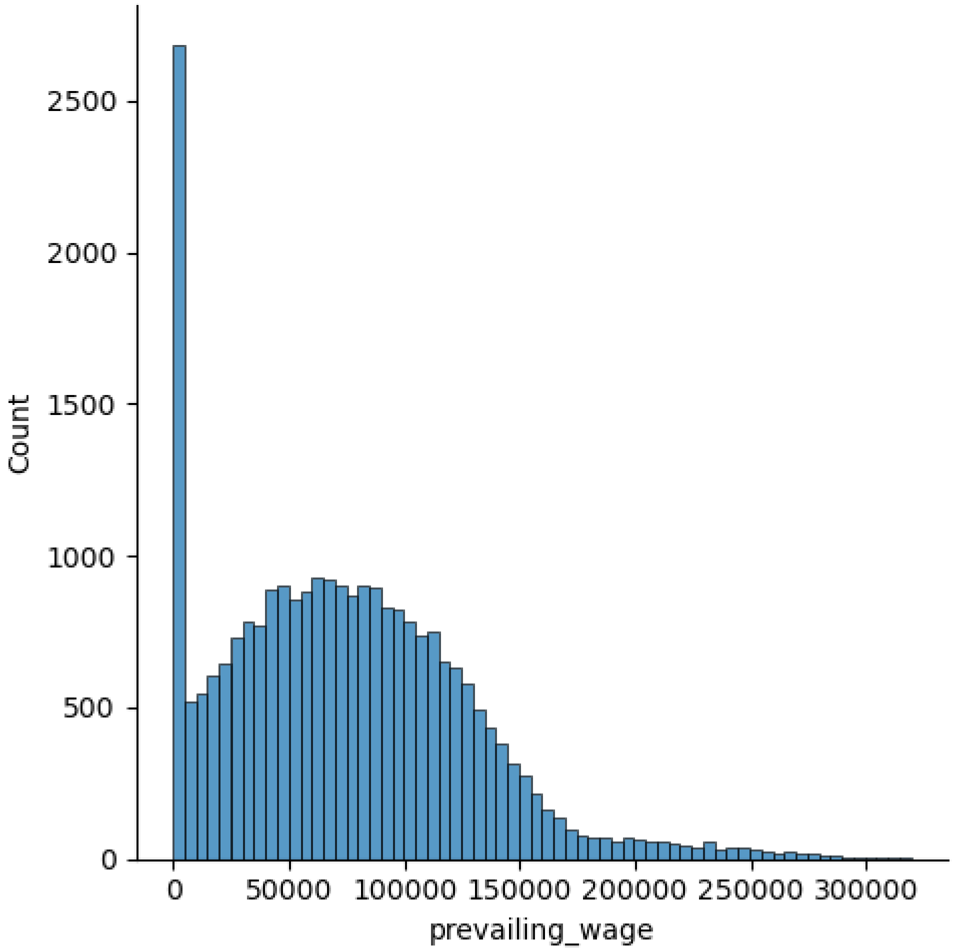


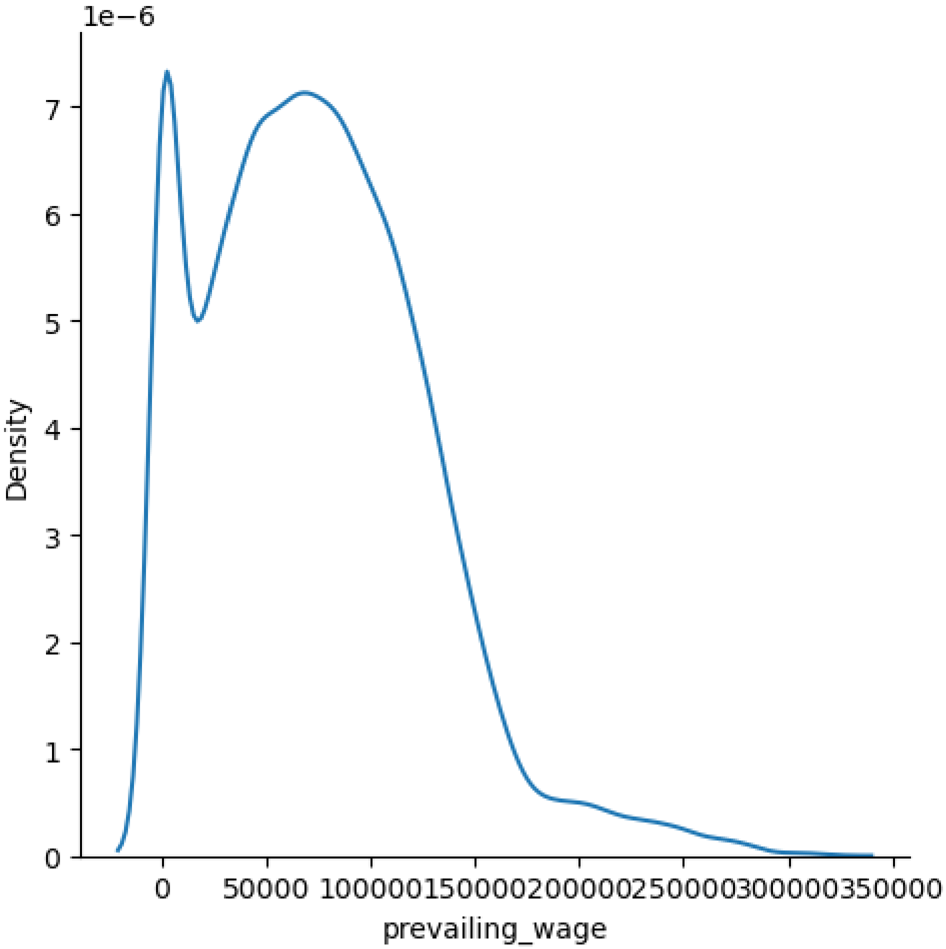

Count
2500
2000
1500
1000
500
0
0
50000
100000
150000
200000
250000
300000
prevailing_wage

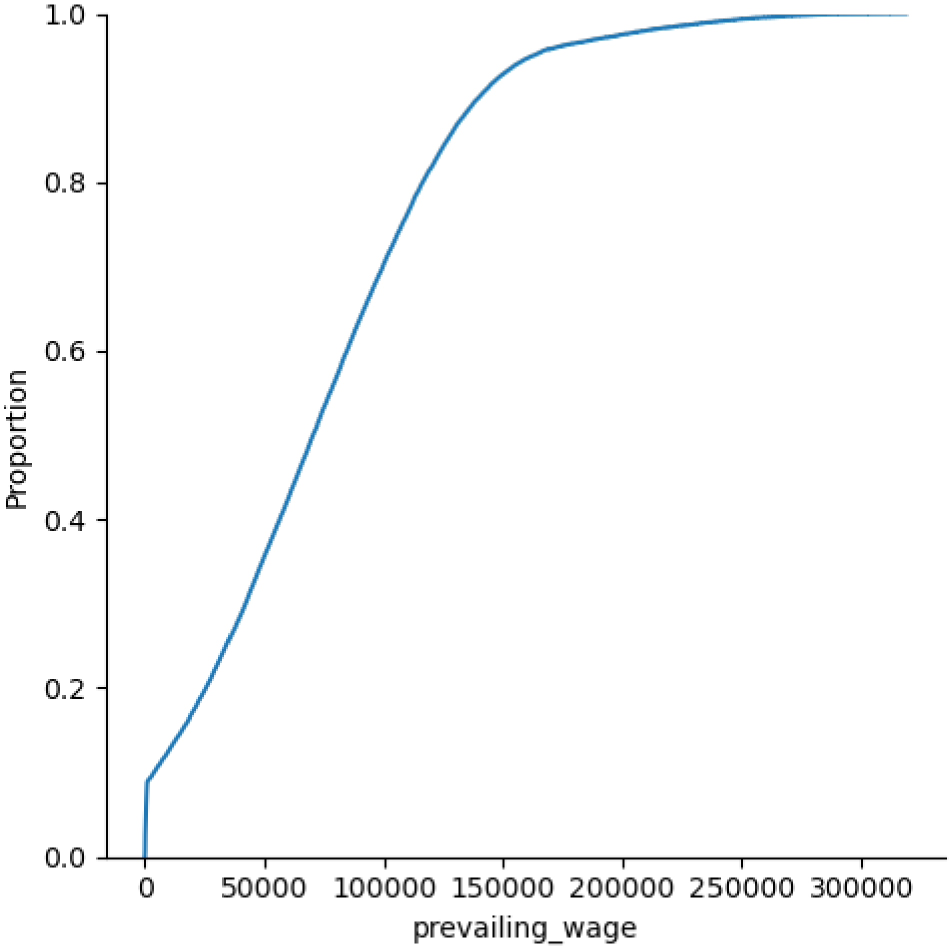
```
In [46]: sns.displot(visa_df['prevailing_wage']) # Histogram
         sns.displot(visa_df['prevailing_wage'],kind='kde')  # kernal density estima
         sns.displot(visa_df['prevailing_wage'],kind='ecdf') # couumulative distribu
```

```
C:\Users\omkar\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWa
rning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
C:\Users\omkar\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWa
rning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
C:\Users\omkar\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWa
rning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

```
Out[46]: <seaborn.axisgrid.FacetGrid at 0x20b1e72f810>
```

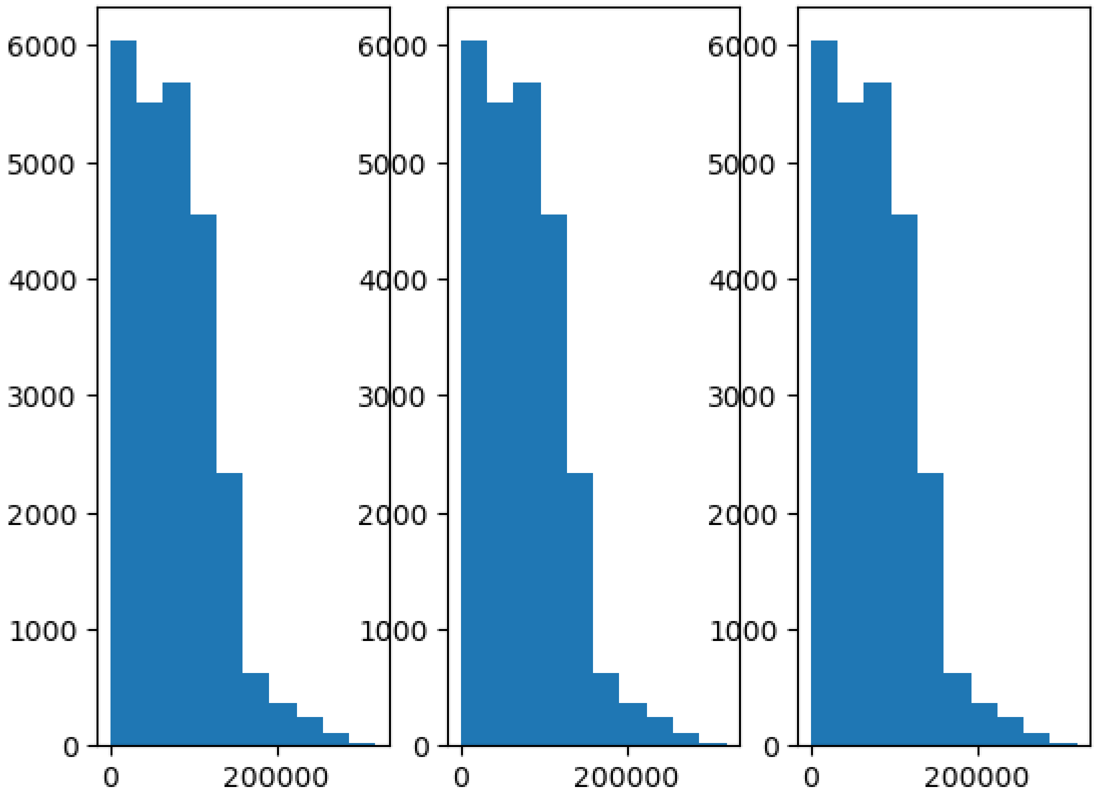



1e−6
Density
0
1
2
3
4
5
6
7
0
50000
100000
150000
200000
250000
300000
350000
prevailing_wage


Proportion
0.0
0.2
0.4
0.6
0.8
1.0
0
50000
100000
150000
200000
250000
300000
prevailing_wage

```
In [52]: plt.subplot(1,3,1)    #  1 row  3 columns
         plt.hist(visa_df['prevailing_wage'])
         plt.subplot(1,3,2)
         plt.hist(visa_df['prevailing_wage'])
         plt.subplot(1,3,3)
         plt.hist(visa_df['prevailing_wage'])
```

```
Out[52]: (array([6038., 5504., 5681., 4551., 2334.,  624.,  373.,  240.,  114.,
                   21.]),
          array([2.13670000e+00, 3.19229500e+04, 6.38437634e+04, 9.57645767e+04,
                 1.27685390e+05, 1.59606203e+05, 1.91527017e+05, 2.23447830e+05,
                 2.55368643e+05, 2.87289457e+05, 3.19210270e+05]),
          <BarContainer object of 10 artists>)
```



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```