Import Packages

In [98]:

```
import pandas as pd
pd.options.display.max_columns = 100
import matplotlib.pyplot as plot
```

In [99]:

```
df = pd.read_csv("C:/Users/bgbai/OneDrive/Desktop/WomenOccupation2.csv")
```

In [100]:

```
df.head()
```

Out[100]:

	Career Category	Total\nemployed	Women	White	Black or\nAfrican\nAmerican	Asian	Hispanic\nor Latino
0	Management professional and related occupations	63644.0	51.7	78.7	9.7	8.6	10.4
1	Management business and financial operations o...	27143.0	44.6	81.7	8.8	6.7	10.9
2	Management occupations	18564.0	40.4	83.4	8.0	5.8	10.7
3	Chief executives	1669.0	29.3	88.0	4.3	5.4	7.4
4	General and operations managers	1057.0	30.5	84.4	7.1	4.5	12.4

In [202]:

```
df.tail()
```

Out[202]:

	Career Category	Total\nemployed	Women	White	Black or\nAfrican\nAmerican	Asian	Hispanic\nor Latino
591	Laborers and freight stock and material movers...	1995.0	4	4	18.6	4.0	28.0
593	Packers and packagers hand	640.0	4	4	25.9	8.0	32.5
594	Stockers and order fillers	1570.0	4	4	19.9	3.9	22.3
596	Refuse and recyclable material collectors	98.0	4	4	27.9	1.3	31.4
597	Other material moving workers	62.0	4	4	18.6	0.0	14.3

In [164]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 386 entries, 0 to 597
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Career Category              386 non-null    object 
 1   Total
employed               386 non-null    float64
 2   Women                        386 non-null    int64  
 3   White                        386 non-null    int64  
 4   Black or
African
American  386 non-null    float64
 5   Asian                        386 non-null    float64
 6   Hispanic
or Latino         386 non-null    float64
dtypes: float64(4), int64(2), object(1)
memory usage: 22.6+ KB
```

In [165]:

```
df.describe()
```

Out[165]:

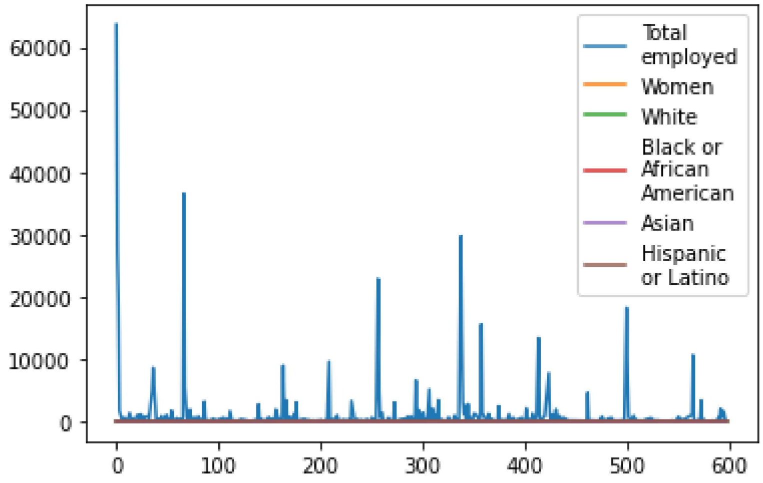
	Total\nemployed	Women	White	Black or\nAfrican\nAmerican	Asian	Hispanic\nor Latino
count	386.000000	386.0	386.0	386.000000	386.000000	386.000000
mean	1301.463731	4.0	4.0	11.938601	6.622798	16.574611
std	4841.220141	0.0	0.0	7.137211	6.971452	9.974093
min	50.000000	4.0	4.0	0.000000	0.000000	2.000000
25%	99.000000	4.0	4.0	7.000000	2.800000	9.925000
50%	195.500000	4.0	4.0	10.500000	4.700000	14.050000
75%	595.500000	4.0	4.0	15.950000	7.800000	21.175000
max	63644.000000	4.0	4.0	38.400000	76.700000	73.100000

In [166]:
```python
from skimpy import skim
skim(df)
```

```
╭──────────────────────────────────────────── skimpy summary ───────────────
│          Data Summary                Data Types
│ ┏━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━┓ ┏━━━━━━━━━━━━━┳━━━━━━━┓
│ ┃ dataframe           ┃ Values ┃ ┃ Column Type ┃ Count ┃
│ ┡━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━┩ ┡━━━━━━━━━━━━━╇━━━━━━━┩
│ │ Number of rows      │ 386    │ │ float64     │ 4     │
│ │ Number of columns   │ 7      │ │ int64       │ 2     │
│ └─────────────────────┴────────┘ │ object      │ 1     │
│                                  └─────────────┴───────┘
│                                          number
│ ┏━━━━━━━━━━┳━━━━━━━━┳━━━━━━━━━┳━━━━━━━━┳━━━━━━━━┳━━━━━━━┳━━━━━━━┳━━━━━━━━┳━━━━━━━━
│ ┃          ┃ missin ┃ complet ┃ mean   ┃ sd     ┃ p0    ┃ p25   ┃ p75    ┃ p100
│ ┃          ┃ g      ┃ e rate  ┃        ┃        ┃       ┃       ┃        ┃
│ ┡━━━━━━━━━━╇━━━━━━━━╇━━━━━━━━━╇━━━━━━━━╇━━━━━━━━╇━━━━━━━╇━━━━━━━╇━━━━━━━━╇━━━━━━━━
│ │ Total    │ 0.0    │ 1.0     │ 1300.0 │ 4800.0 │ 50.0  │ 99.0  │ 600.0  │ 64000.
│ │ employe  │        │         │        │        │       │       │        │ 0
│ │ d        │        │         │        │        │       │       │        │
│ │ Women    │ 0.0    │ 1.0     │ 4.0    │ 0.0    │ 4.0   │ 4.0   │ 4.0    │ 4.0
│ │ White    │ 0.0    │ 1.0     │ 4.0    │ 0.0    │ 4.0   │ 4.0   │ 4.0    │ 4.0
│ │ Black    │ 0.0    │ 1.0     │ 12.0   │ 7.1    │ 0.0   │ 7.0   │ 16.0   │ 38.0
│ │ or       │        │         │        │        │       │       │        │
│ │ African  │        │         │        │        │       │       │        │
│ │ Ame      │        │         │        │        │       │       │        │
│ │ Asian    │ 0.0    │ 1.0     │ 6.6    │ 7.0    │ 0.0   │ 2.8   │ 7.8    │ 77.0
│ │ Hispani  │ 0.0    │ 1.0     │ 17.0   │ 10.0   │ 2.0   │ 9.9   │ 21.0   │ 73.0
│ │ c        │        │         │        │        │       │       │        │
│ │ or       │        │         │        │        │       │       │        │
│ │ Latino   │        │         │        │        │       │       │        │
│ └──────────┴────────┴─────────┴────────┴────────┴───────┴───────┴────────┴────────
╰──────────────────────────────────────────── End ──────────────────────────
```

In [167]:
```python
df.plot()
```

Out[167]:
```
<AxesSubplot:>
```



In [168]:
```
df.columns
```

Out[168]:
```
Index(['Career Category', 'Total\nemployed', 'Women', 'White',
       'Black or\nAfrican\nAmerican', 'Asian', 'Hispanic\nor Latino'],
      dtype='object')
```

In [169]:
```
df.Women.value_counts()
```

Out[169]:
```
4    386
Name: Women, dtype: int64
```

In [170]:
```
def recode_Women(series):
    if series == '0':
        return 0
    elif series == '1':
        return 1
    elif series == '2':
        return 2
    elif series == '3':
        return 3
    else:
        return 4
```

In [171]:
```
df['Women'] = df.Women.apply(recode_Women)
```

In [172]:
```
df.White.value_counts()
```

Out[172]:
```
4    386
Name: White, dtype: int64
```

In [173]:
```
def recode_White(series):
    if series == '0':
        return 0
    elif series == '1':
        return 1
    elif series == '2':
        return 2
    elif series == '3':
        return 3
    else:
        return 4
```

In [174]:
```
df['White'] = df.White.apply(recode_White)
```

Select X and Y, train- test split data

In [175]:
```
df.columns
```

Out[175]:
```
Index(['Career Category', 'Total\nemployed', 'Women', 'White',
       'Black or\nAfrican\nAmerican', 'Asian', 'Hispanic\nor Latino'],
      dtype='object')
```

In [176]:
```
df.isnull().sum()
```

Out[176]:
```
Career Category                 0
Total\nemployed                 0
Women                           0
White                           0
Black or\nAfrican\nAmerican     0
Asian                           0
Hispanic\nor Latino             0
dtype: int64
```

In [177]:
```
df.shape
```

Out[177]:
```
(386, 7)
```

In [178]:
```
df.dropna(inplace=True)
```

In [179]:
```
X = df[['Total\nemployed', 'Women', 'White', 'Black or\nAfrican\nAmerican', 'Hisp
```

In [180]:
```
X.head()
```

Out[180]:

	Total\nemployed	Women	White	Black or\nAfrican\nAmerican	Hispanic\nor Latino	Asian
0	63644.0	4	4	9.7	10.4	8.6
1	27143.0	4	4	8.8	10.9	6.7
2	18564.0	4	4	8.0	10.7	5.8
3	1669.0	4	4	4.3	7.4	5.4
4	1057.0	4	4	7.1	12.4	4.5

In [181]:
```
y = df['Career Category']
```

In [182]:
```
y.head()
```

Out[182]:
```
0      Management professional and related occupations
1    Management business and financial operations o...
2                               Management occupations
3                                     Chief executives
4                         General and operations managers
Name: Career Category, dtype: object
```

In [183]:
```
from sklearn.model_selection import train_test_split
```

In [184]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

##Predict and Model Metrics

```
In [185]: from sklearn.tree import DecisionTreeClassifier
```

```
In [186]: clf = DecisionTreeClassifier(random_state=0)
```

```
In [187]: clf.fit(X_train, y_train)

Out[187]: DecisionTreeClassifier(random_state=0)
```

```
In [188]: prediction = clf.predict(X_test)
```

```
In [189]: from sklearn.metrics import classification_report
```

```
In [190]: classification_report(y_test, prediction)
```

```
C:\Users\bgbai\anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1221: UndefinedMetricWarning: Precision and F-score are ill-defined and bei
ng set to 0.0 in labels with no predicted samples. Use `zero_division` parame
ter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\bgbai\anaconda3\lib\site-packages\sklearn\metrics\_classification.p
y:1221: UndefinedMetricWarning: Recall and F-score are ill-defined and being
set to 0.0 in labels with no true samples. Use `zero_division` parameter to c
ontrol this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

Histograms

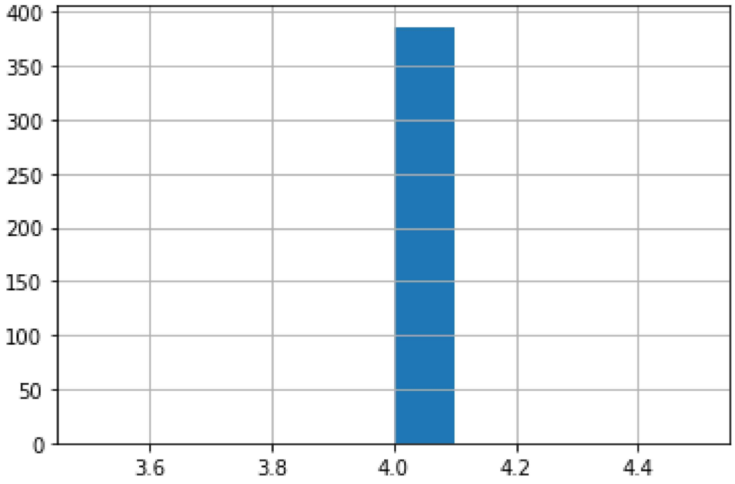
```
In [191]: import pandas as pd
          import seaborn as sns
```

```
In [192]: sns.get_dataset_names()
```

```
Out[192]: ['anagrams',
           'anscombe',
           'attention',
           'brain_networks',
           'car_crashes',
           'diamonds',
           'dots',
           'exercise',
           'flights',
           'fmri',
           'gammas',
           'geyser',
           'iris',
           'mpg',
           'penguins',
           'planets',
           'taxis',
           'tips',
           'titanic']
```

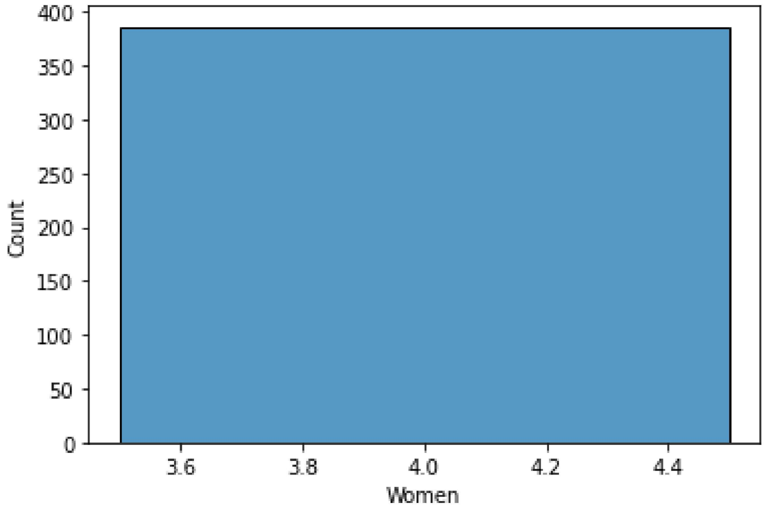
```
In [193]: df['Women'].hist()
```

```
Out[193]: <AxesSubplot:>
```



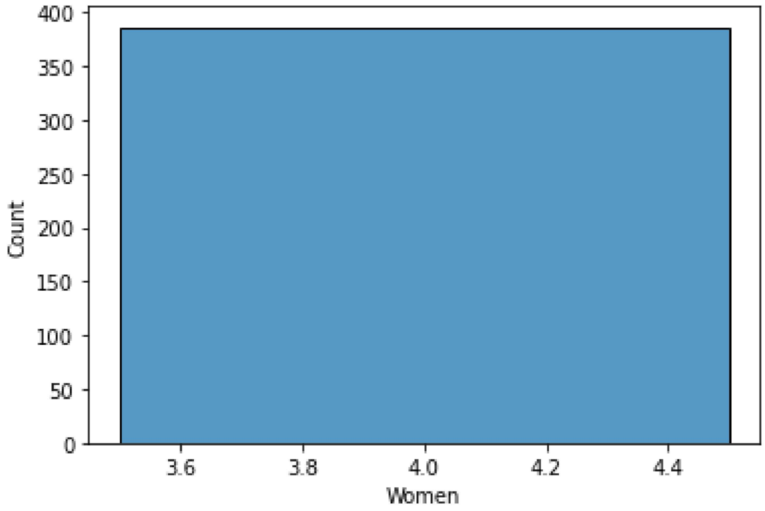
In [194]:
```
sns.histplot(df['Women'])
```

Out[194]:
```
<AxesSubplot:xlabel='Women', ylabel='Count'>
```



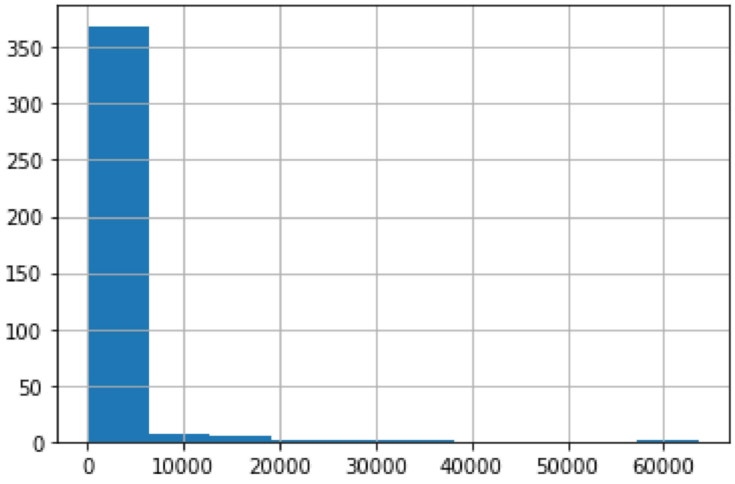
In [195]:
```
sns.histplot(df['Women'])
```

Out[195]:
```
<AxesSubplot:xlabel='Women', ylabel='Count'>
```



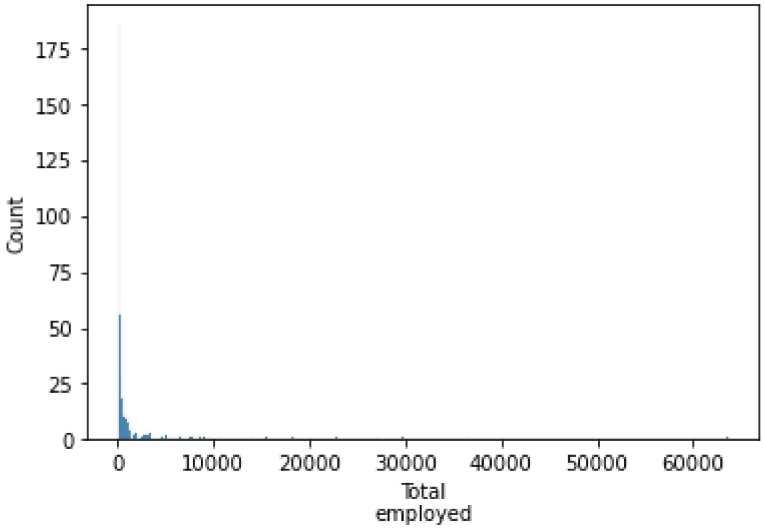
```
In [196]: df['Total\nemployed'].hist()
```

```
Out[196]: <AxesSubplot:>
```



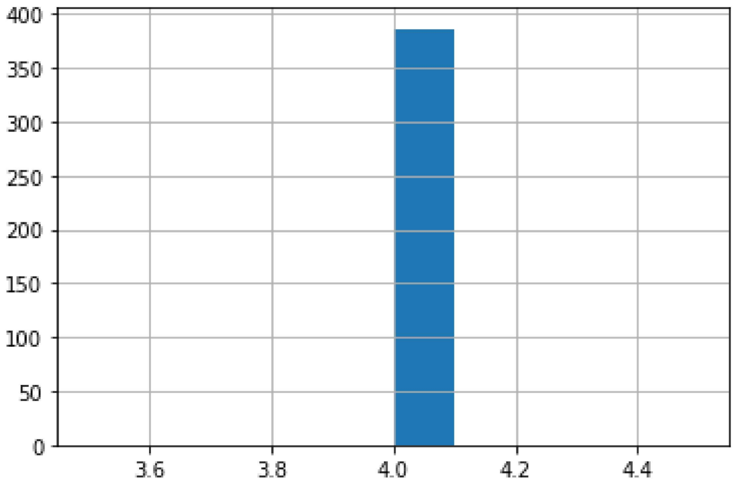
```
In [197]: sns.histplot(df['Total\nemployed'])
```

```
Out[197]: <AxesSubplot:xlabel='Total\nemployed', ylabel='Count'>
```



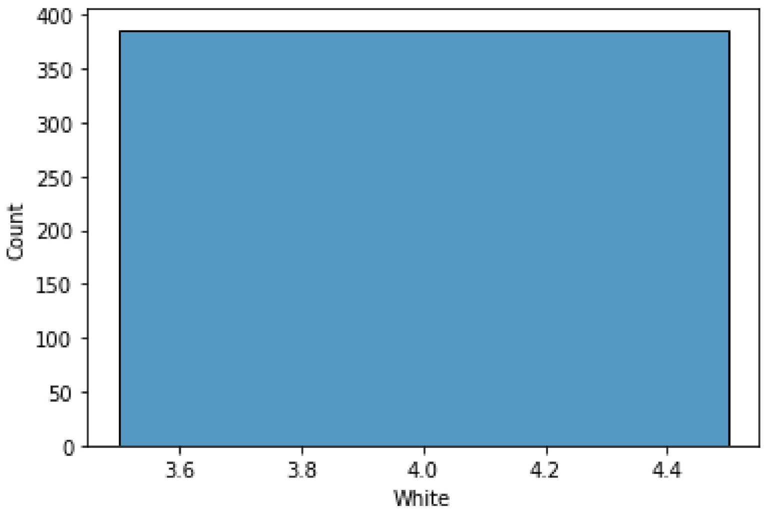
```
In [198]: df['White'].hist()
```

```
Out[198]: <AxesSubplot:>
```



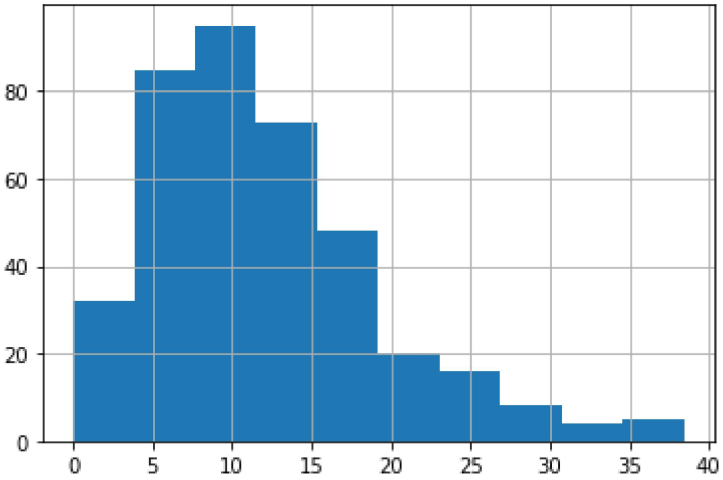
```
In [199]: sns.histplot(df['White'])
```

```
Out[199]: <AxesSubplot:xlabel='White', ylabel='Count'>
```



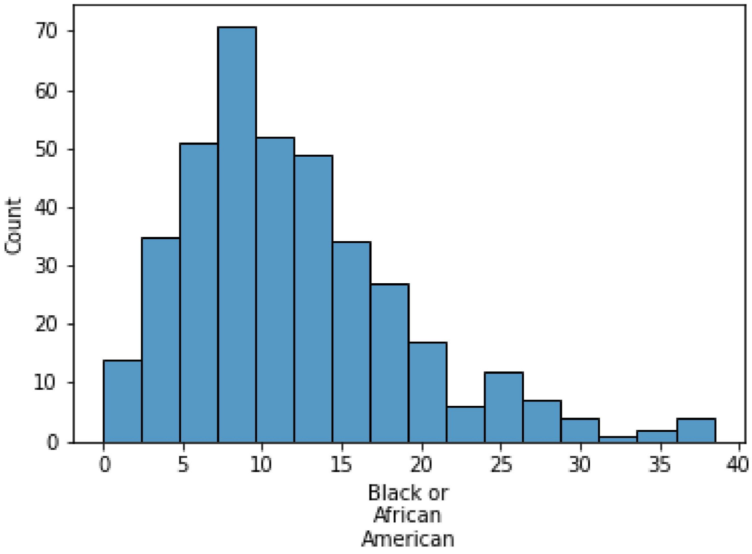
```
In [200]: df['Black or\nAfrican\nAmerican'].hist()

Out[200]: <AxesSubplot:>
```



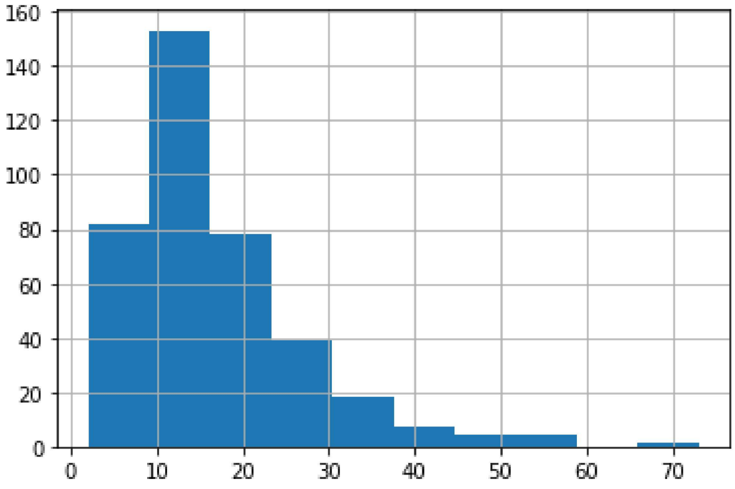
```
In [152]: sns.histplot(df['Black or\nAfrican\nAmerican'])

Out[152]: <AxesSubplot:xlabel='Black or\nAfrican\nAmerican', ylabel='Count'>
```



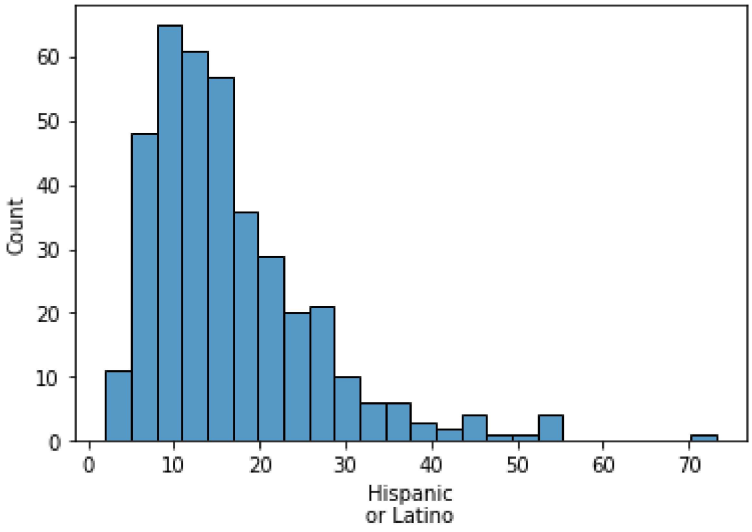
```
In [153]: df['Hispanic\nor Latino'].hist()

Out[153]: <AxesSubplot:>
```



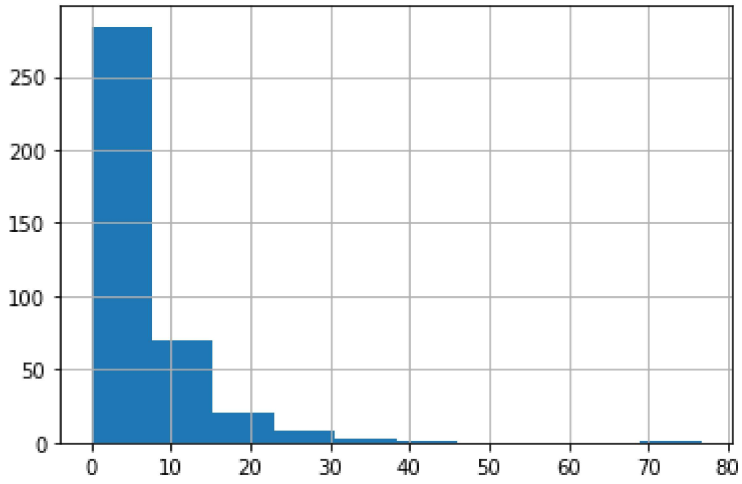
```
In [154]: sns.histplot(df['Hispanic\nor Latino'])

Out[154]: <AxesSubplot:xlabel='Hispanic\nor Latino', ylabel='Count'>
```



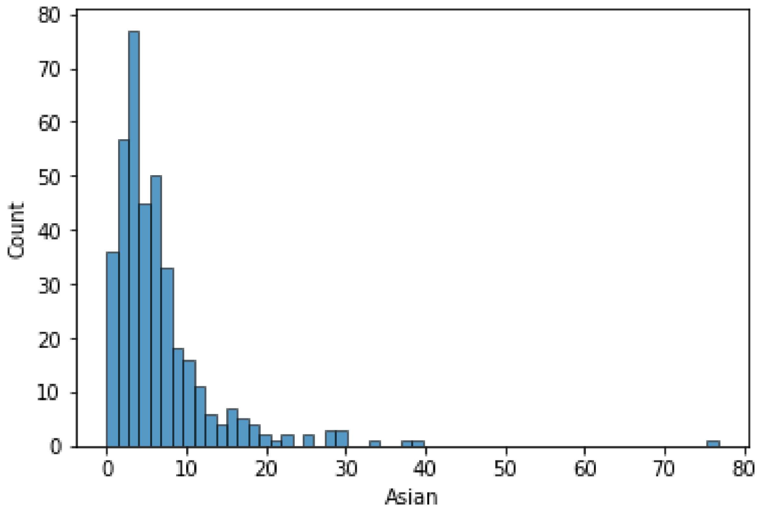
```
In [155]: df['Asian'].hist()
```

```
Out[155]: <AxesSubplot:>
```



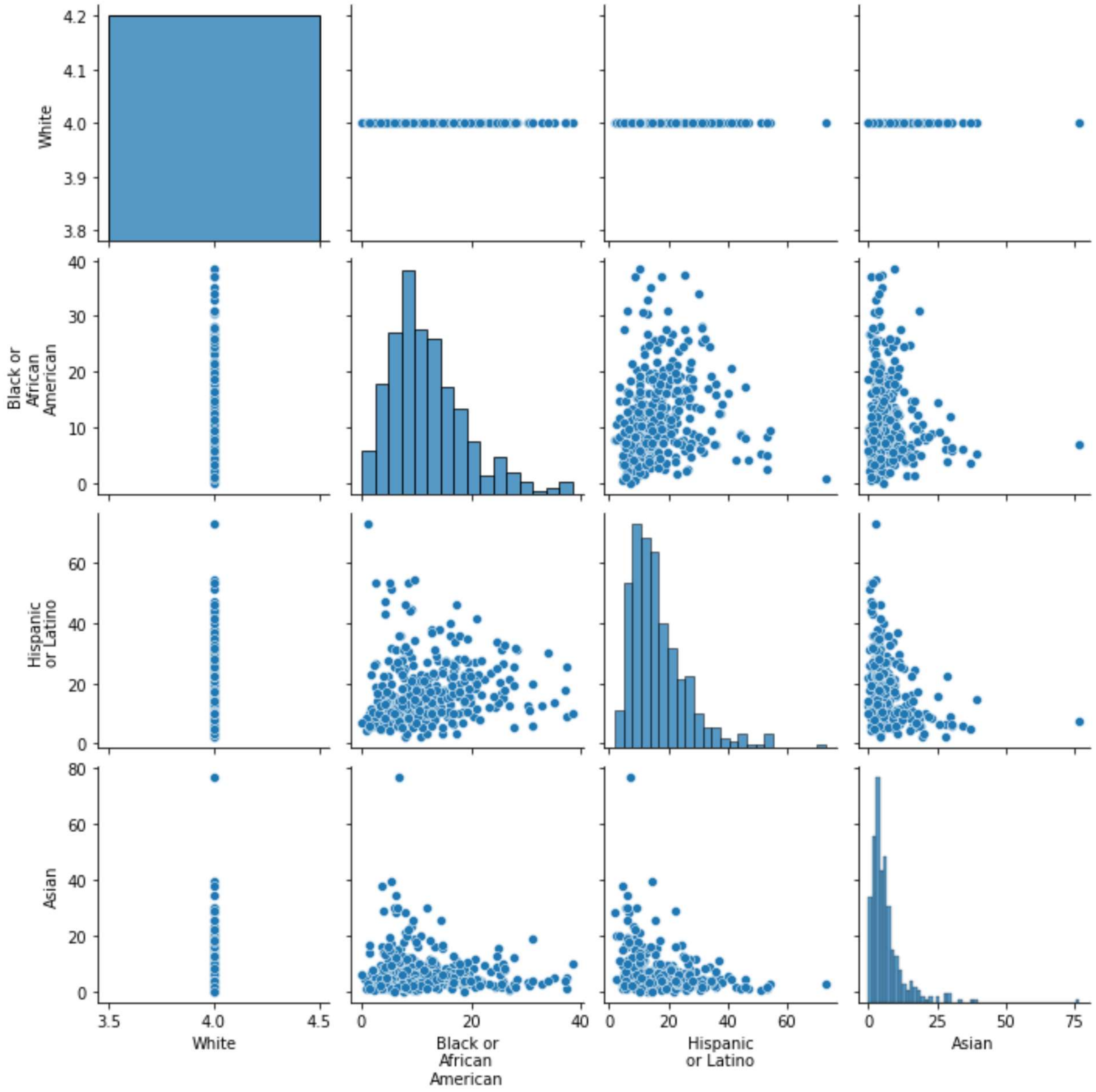
```
In [156]: sns.histplot(df['Asian'])
```

```
Out[156]: <AxesSubplot:xlabel='Asian', ylabel='Count'>
```



```
In [158]: continuous_vars = ['White', 'Black or\nAfrican\nAmerican', 'Hispanic\nor Latino',
In [159]: ## Select which columns to use and plot hists and scatters
          sns.pairplot(df[continuous_vars])
```

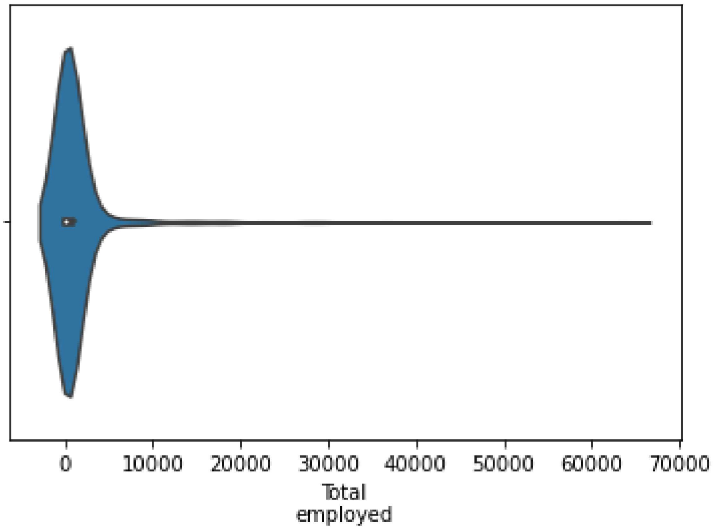
```
Out[159]: <seaborn.axisgrid.PairGrid at 0xdc98fa0>
```



```
In [161]: ##Violin Plots
In [162]: sns.violinplot(df['Total\nemployed'])
```

```
C:\Users\bgbai\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[162]: <AxesSubplot:xlabel='Total\nemployed'>
```



##Barplot

```
In [207]: df.groupby('White')['0'].mean().plot(kind='bar')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-207-8c2a591e5f91> in <module>
----> 1 df.groupby('White')['0'].mean().plot(kind='bar')

~\anaconda3\lib\site-packages\pandas\core\groupby\generic.py in __getitem__(sel
f, key)
   1536                 stacklevel=2,
   1537             )
-> 1538         return super().__getitem__(key)
   1539 
   1540     def _gotitem(self, key, ndim: int, subset=None):

~\anaconda3\lib\site-packages\pandas\core\base.py in __getitem__(self, key)
    230         else:
    231             if key not in self.obj:
--> 232                 raise KeyError(f"Column not found: {key}")
    233             subset = self.obj[key]
    234             ndim = subset.ndim

KeyError: 'Column not found: 0'
```

In [ ]: