US Employment Rate Analysis

Data Preparation

Import Library

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

from datapackage import Package
from statsmodels.tsa.arima.model import ARIMA
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from math import sqrt
from sklearn.metrics import r2_score
from powerbiclient import QuickVisualize, get_dataset_config, Report
from powerbiclient.authentication import DeviceCodeLoginAuthentication
```

Load Data

```
package =
Package('https://datahub.io/core/employment-us/datapackage.json')

print(package.resource_names)

for resource in package.resources:
    if resource.descriptor['datahub']['type'] == 'derived/csv':
        data = pd.read_csv(resource.raw_iter())
        print(data)
        #print(resource.read())
```

```
['validation_report', 'aat1_csv', 'aat1_json', 'employment-us_zip',
'aat1']
    year  population  labor_force  population_percent  employed_total
\
0   1941       99900        55910                56.0           50350

1   1942       98640        56410                57.2           53750

2   1943       94640        55540                58.7           54470

3   1944       93220        54630                58.6           53960

4   1945       94090        53860                57.2           52820
```

```
..   ...         ...          ...                   ...             ...
66  2006      228815       151428                  66.2          144427
67  2007      231867       153124                  66.0          146047
68  2008      233788       154287                  66.0          145362
69  2009      235801       154142                  65.4          139877
70  2010      237830       153889                  64.7          139064

    employed_percent  agrictulture_ratio  nonagriculture_ratio
unemployed  \
0               50.4                9100                 41250
5560
1               54.5                9250                 44500
2660
2               57.6                9080                 45390
1070
3               57.9                8950                 45010
670
4               56.1                8580                 44240
1040
..               ...                 ...                   ...
...
66              63.1                2206                142221
7001
67              63.0                2095                143952
7078
68              62.2                2168                143194
8924
69              59.3                2103                137775
14265
70              58.5                2206                136858
14825

    unemployed_percent  not_in_labor  footnotes
0                  9.9         43990        NaN
1                  4.7         42230        NaN
2                  1.9         39100        NaN
3                  1.2         38590        NaN
4                  1.9         40230        NaN
..                 ...           ...        ...
66                 4.6         77387        1.0
67                 4.6         78743        1.0
68                 5.8         79501        1.0
69                 9.3         81659        1.0
70                 9.6         83941        1.0
```

```
[71 rows x 12 columns]
```

Data Preview

```
data.head()
```

```
   year  population  labor_force  population_percent
employed_total  \
0  1941       99900        55910                56.0           50350

1  1942       98640        56410                57.2           53750

2  1943       94640        55540                58.7           54470

3  1944       93220        54630                58.6           53960

4  1945       94090        53860                57.2           52820


   employed_percent  agrictulture_ratio  nonagriculture_ratio
unemployed  \
0              50.4                9100                 41250
5560
1              54.5                9250                 44500
2660
2              57.6                9080                 45390
1070
3              57.9                8950                 45010
670
4              56.1                8580                 44240
1040

   unemployed_percent  not_in_labor  footnotes
0                 9.9         43990        NaN
1                 4.7         42230        NaN
2                 1.9         39100        NaN
3                 1.2         38590        NaN
4                 1.9         40230        NaN
```

```
data.tail()
```

```
    year  population  labor_force  population_percent  employed_total
\
66  2006      228815       151428                66.2          144427

67  2007      231867       153124                66.0          146047

68  2008      233788       154287                66.0          145362

69  2009      235801       154142                65.4          139877
```

```
70  2010      237830       153889                64.7          139064


    employed_percent  agrictulture_ratio  nonagriculture_ratio  
unemployed  \
66              63.1                2206                142221        
7001   
67              63.0                2095                143952        
7078   
68              62.2                2168                143194        
8924   
69              59.3                2103                137775       
14265   
70              58.5                2206                136858       
14825   

    unemployed_percent  not_in_labor  footnotes  
66                 4.6         77387        1.0  
67                 4.6         78743        1.0  
68                 5.8         79501        1.0  
69                 9.3         81659        1.0  
70                 9.6         83941        1.0  
```

Immediately from the head of the data we can see a misspelling in the column 'agrictulture_ratio' which supposed to be 'agriculture_ratio'.

Data Exploration

Checking Data

Shape of Data

```
data.shape

(71, 12)
```

General Information of Data

Column Name, Count, Data Type

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   year                  71 non-null     int64  
 1   population            71 non-null     int64  
```

```
 2   labor_force          71 non-null     int64
 3   population_percent   71 non-null     float64
 4   employed_total       71 non-null     int64
 5   employed_percent     71 non-null     float64
 6   agricutlture_ratio   71 non-null     int64
 7   nonagriculture_ratio  71 non-null     int64
 8   unemployed           71 non-null     int64
 9   unemployed_percent   71 non-null     float64
 10  not_in_labor         71 non-null     int64
 11  footnotes            21 non-null     float64
dtypes: float64(4), int64(8)
memory usage: 6.8 KB
```

General Statistic Information

```
data.describe()
```

```
              year     population    labor_force
population_percent  \
count    71.000000      71.000000      71.000000            71.000000

mean   1975.098592  156272.521127   98707.492958            62.230986

std      20.486340   44979.642300   33380.727804             3.513793

min    1941.000000   93220.000000   53860.000000            55.800000

25%    1957.500000  112996.000000   67284.000000            59.200000

50%    1975.000000  153153.000000   93774.000000            61.300000

75%    1992.500000  193821.500000  128652.500000            66.000000

max    2010.000000  237830.000000  154287.000000            67.100000


       employed_total  employed_percent  agricutlture_ratio  \
count       71.000000         71.000000           71.000000
mean     93033.577465         58.773239         4456.492958
std      30992.290437          3.140380         2055.547406
min      50350.000000         50.400000         2095.000000
25%      63935.000000         56.400000         3235.000000
50%      86794.000000         57.800000         3440.000000
75%     119526.000000         61.950000         5766.500000
max     146047.000000         64.400000         9250.000000

       nonagriculture_ratio    unemployed  unemployed_percent
not_in_labor  \
count             71.000000     71.000000           71.000000
71.000000
mean           88577.028169   5673.774648            5.509859
```

```
57564.929577
std           32739.727633   2954.030173              1.818803
11795.889143
min           41250.000000    670.000000              1.200000
38590.000000
25%           57818.500000   3131.500000              4.350000
45969.000000
50%           83279.000000   5692.000000              5.500000
59377.000000
75%          116357.000000   7614.000000              6.450000
65169.000000
max          143952.000000  14825.000000              9.900000
83941.000000

       footnotes
count       21.0
mean         1.0
std          0.0
min          1.0
25%          1.0
50%          1.0
75%          1.0
max          1.0
```

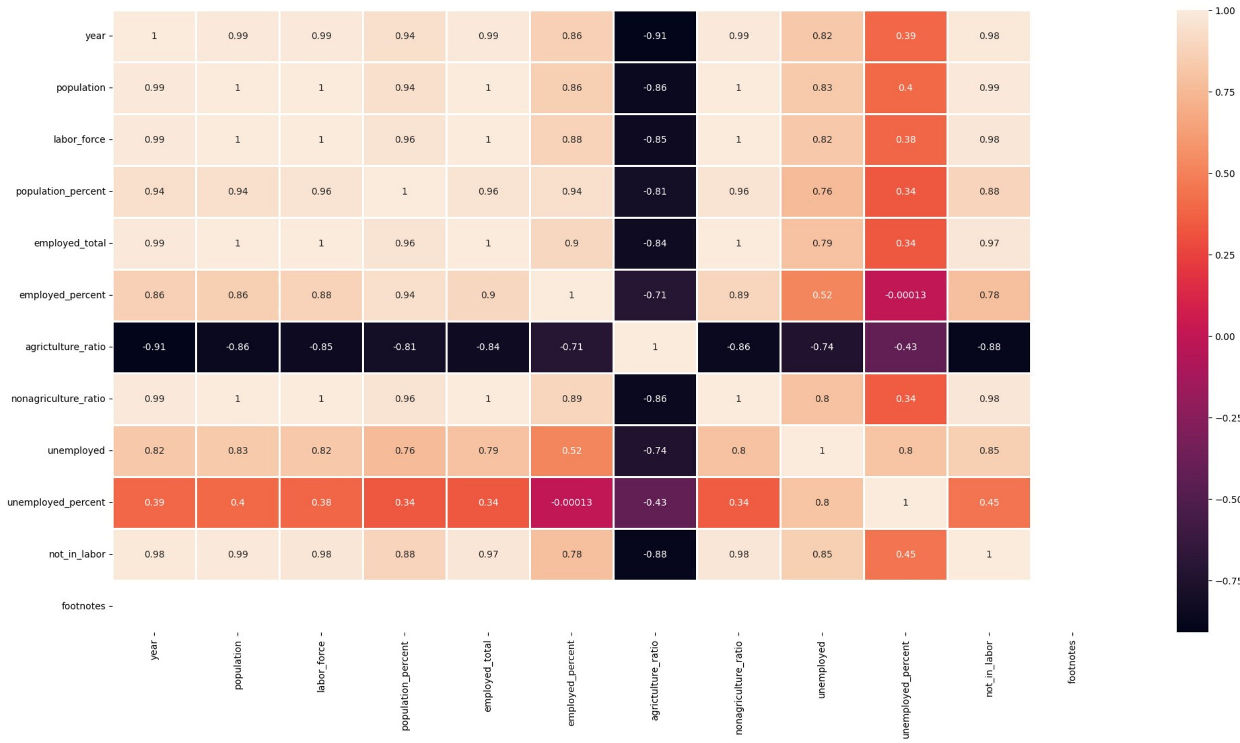
Missing Value

```
data.isnull().sum()
```

```
year                      0
population                0
labor_force               0
population_percent        0
employed_total            0
employed_percent          0
agrictulture_ratio        0
nonagriculture_ratio      0
unemployed                0
unemployed_percent        0
not_in_labor              0
footnotes                50
dtype: int64
```

Variable Coorelation

```
plt.figure(figsize=(24,12))
corr = data.corr()
sns.heatmap(corr, annot = True, linewidths=1)
plt.show()
```

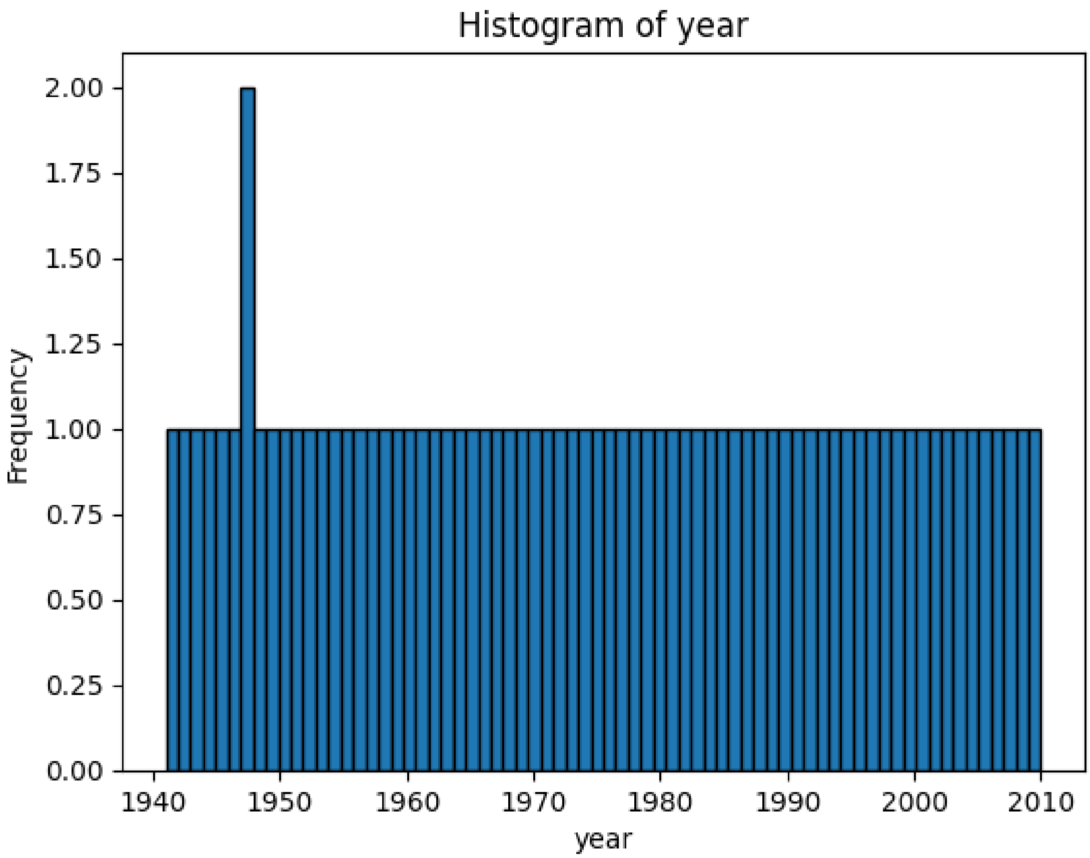


Based on the correlation map above, we can notice that the year variable has high correlation with almost all the data except for the unemployed_percent with correlation as low as 0.39 and agriculture_ratio which shows little to no correlation even showing a negative correlation with almost all the variables within the data. This shows us that population, employment rate (based on employed_total and employed_percent) has been increasing throughout the year, but we can't say the same on the unemployed_percent which shows a weak correlation.

Univariate Analysis

Year Distribution

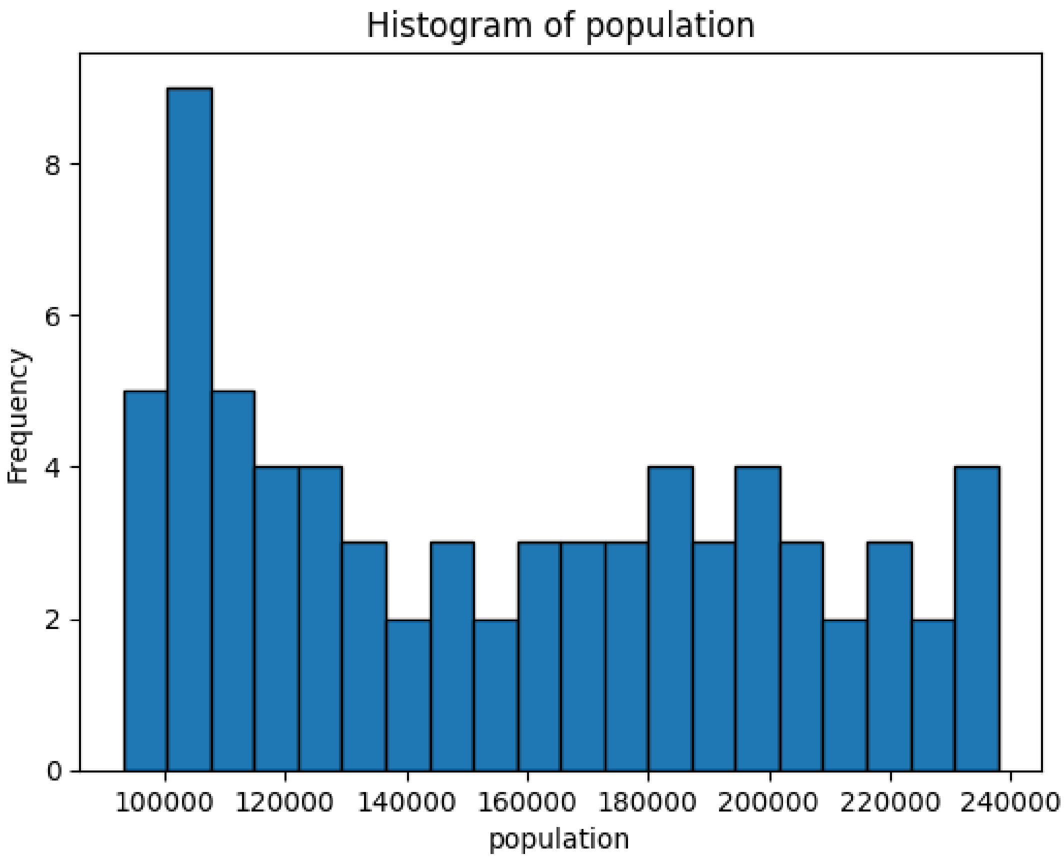
```
column_to_analyze = 'year'
plt.hist(data[column_to_analyze], bins=70, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



Here we can see that the year there are 2 records with the year 1947.

Population Distribution

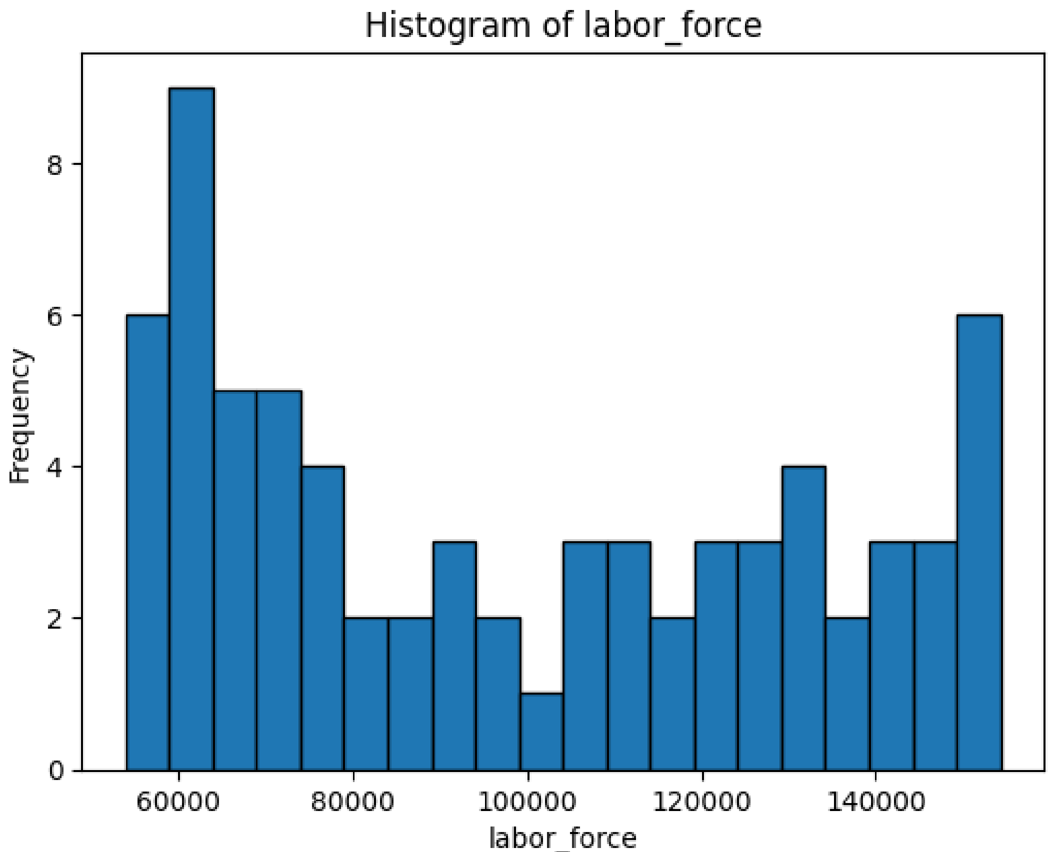
```
column_to_analyze = 'population'
plt.hist(data[column_to_analyze], bins=20, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



The chart for population shows a right skewed histogram where there is a abundance of record with population of around 110000

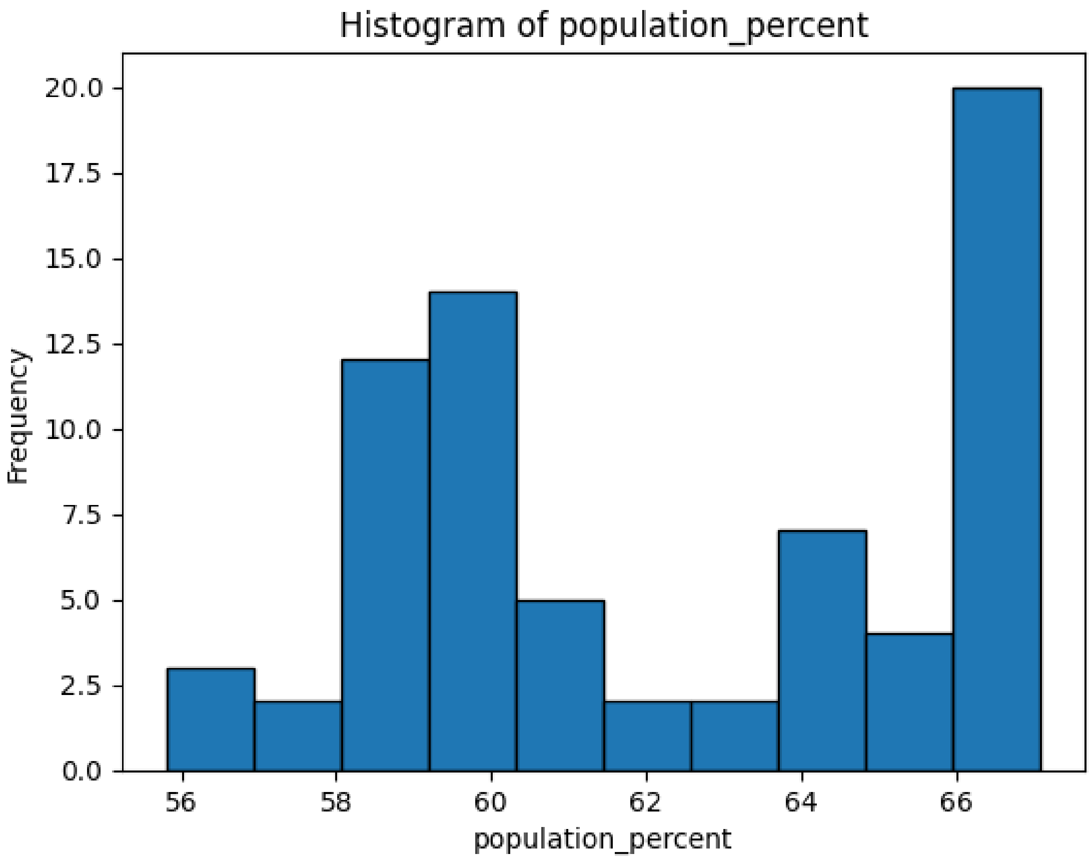
Labor Force Distribution

```
column_to_analyze = 'labor_force'
plt.hist(data[column_to_analyze], bins=20, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



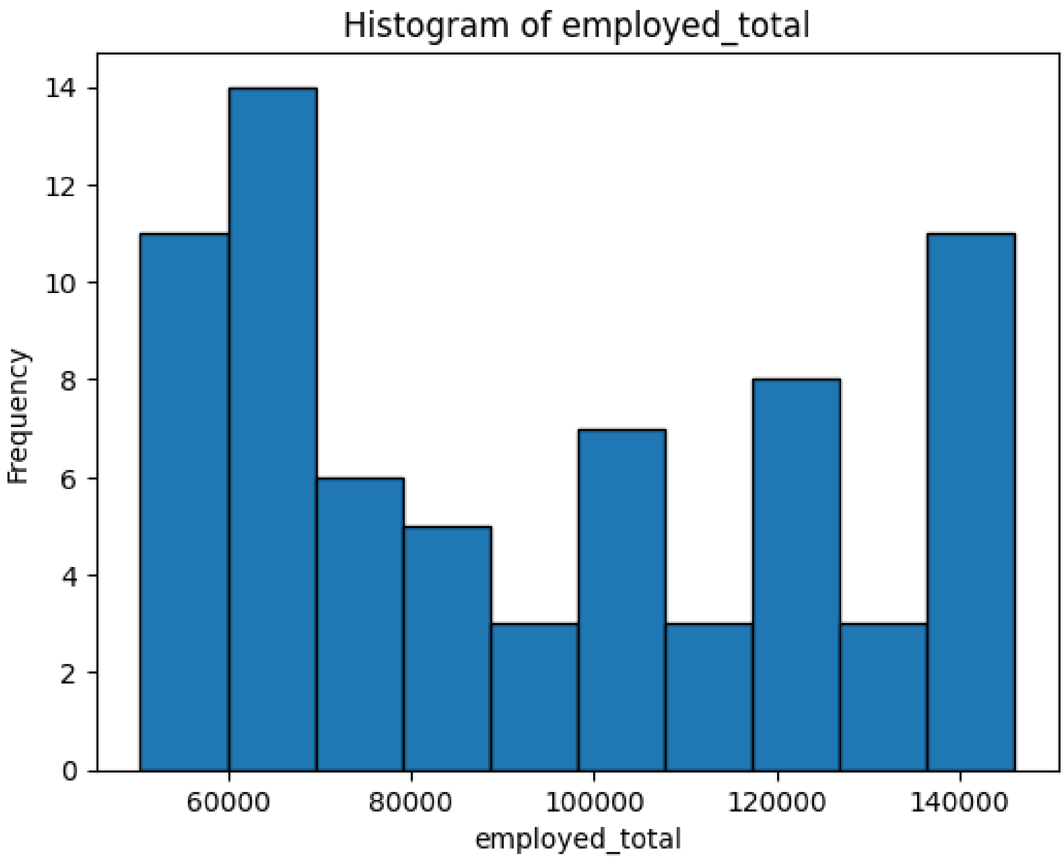
Population Percentage Distribution

```
column_to_analyze = 'population_percent'
plt.hist(data[column_to_analyze], bins=10, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



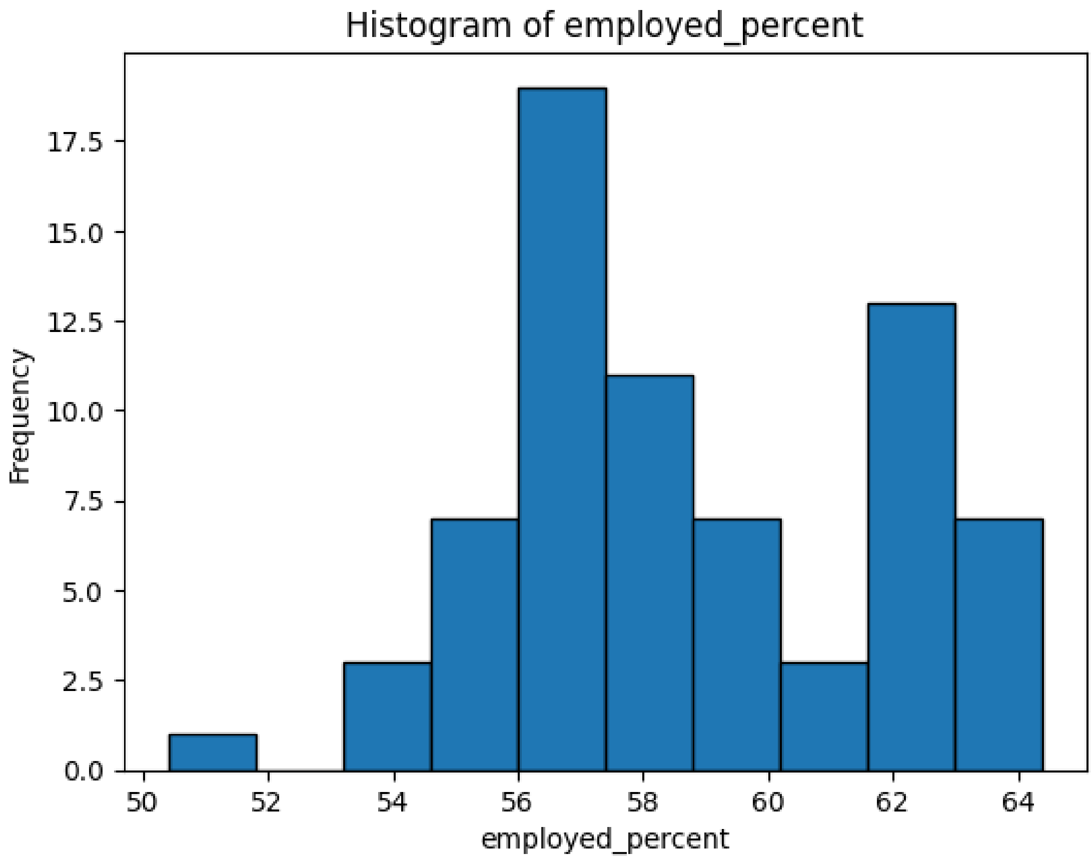
Total Number of Employed Distribution

```
column_to_analyze = 'employed_total'
plt.hist(data[column_to_analyze], bins=10, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



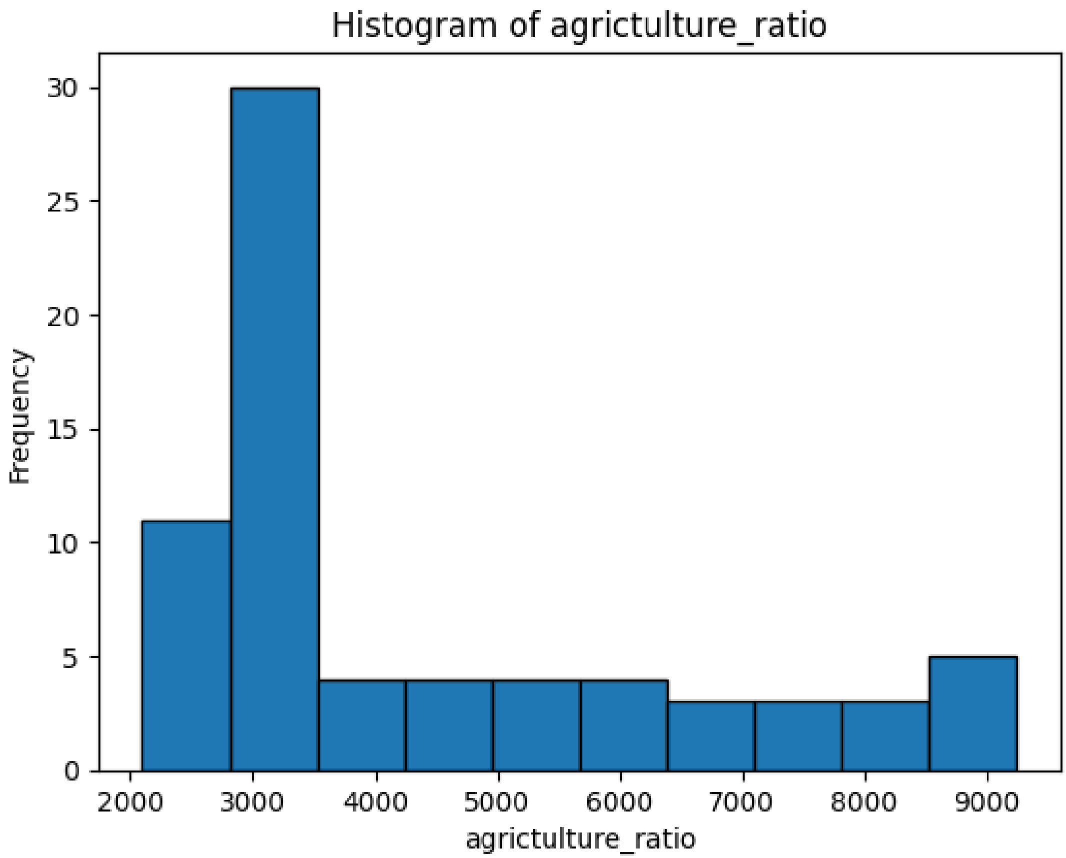
Percentage of Employed Distribution

```
column_to_analyze = 'employed_percent'
plt.hist(data[column_to_analyze], bins=10, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



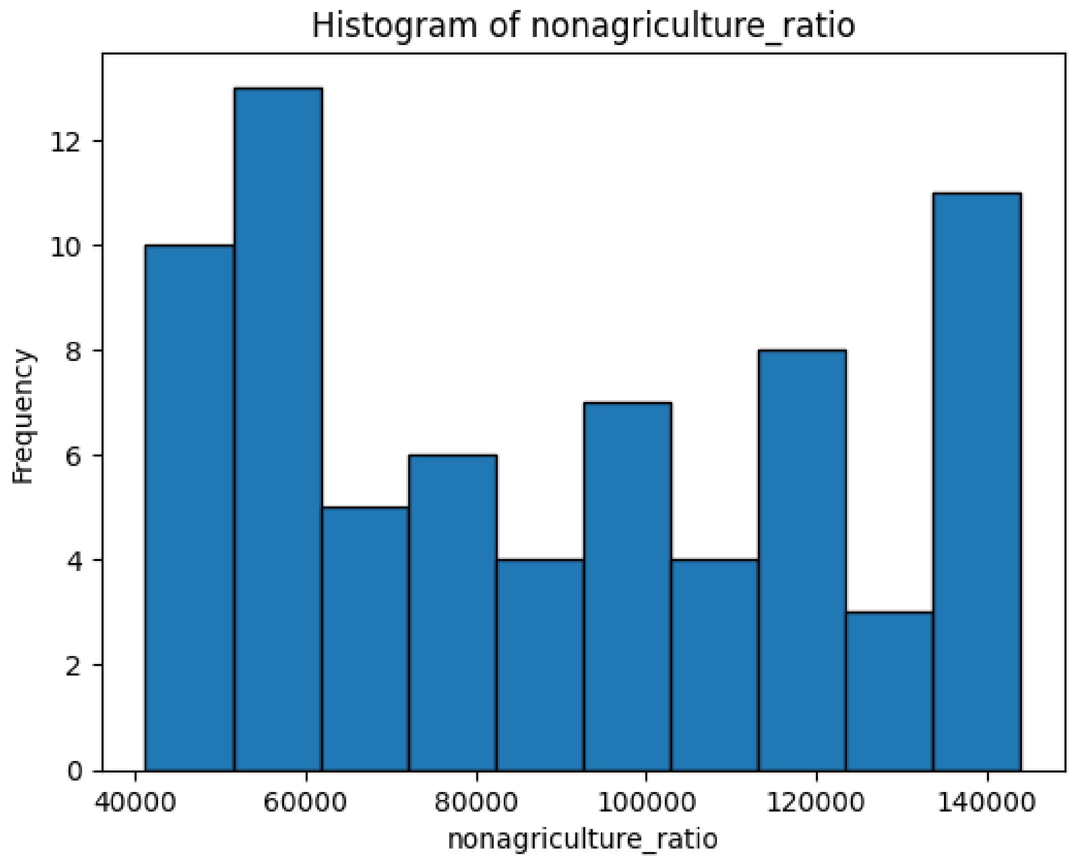
Agriculture Ratio Distribution

```
column_to_analyze = 'agrictulture_ratio'
plt.hist(data[column_to_analyze], bins=10, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



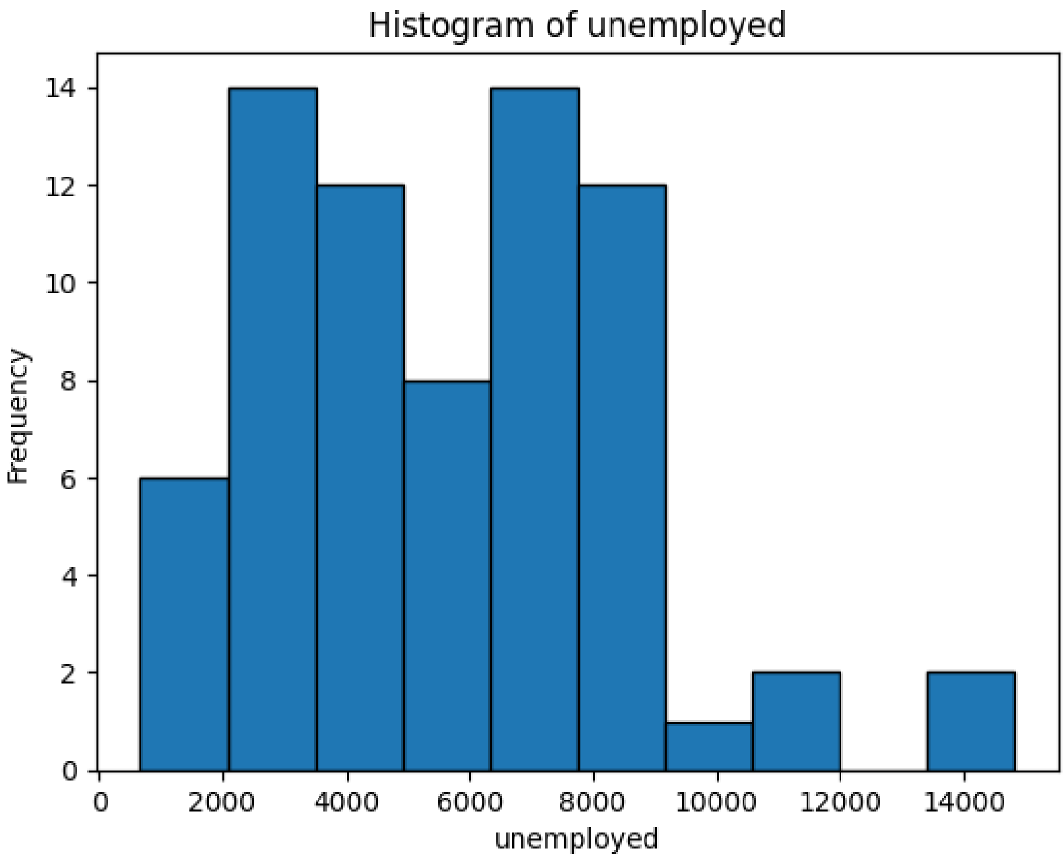
Nonagriculture Ratio Distribution

```
column_to_analyze = 'nonagriculture_ratio'
plt.hist(data[column_to_analyze], bins=10, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



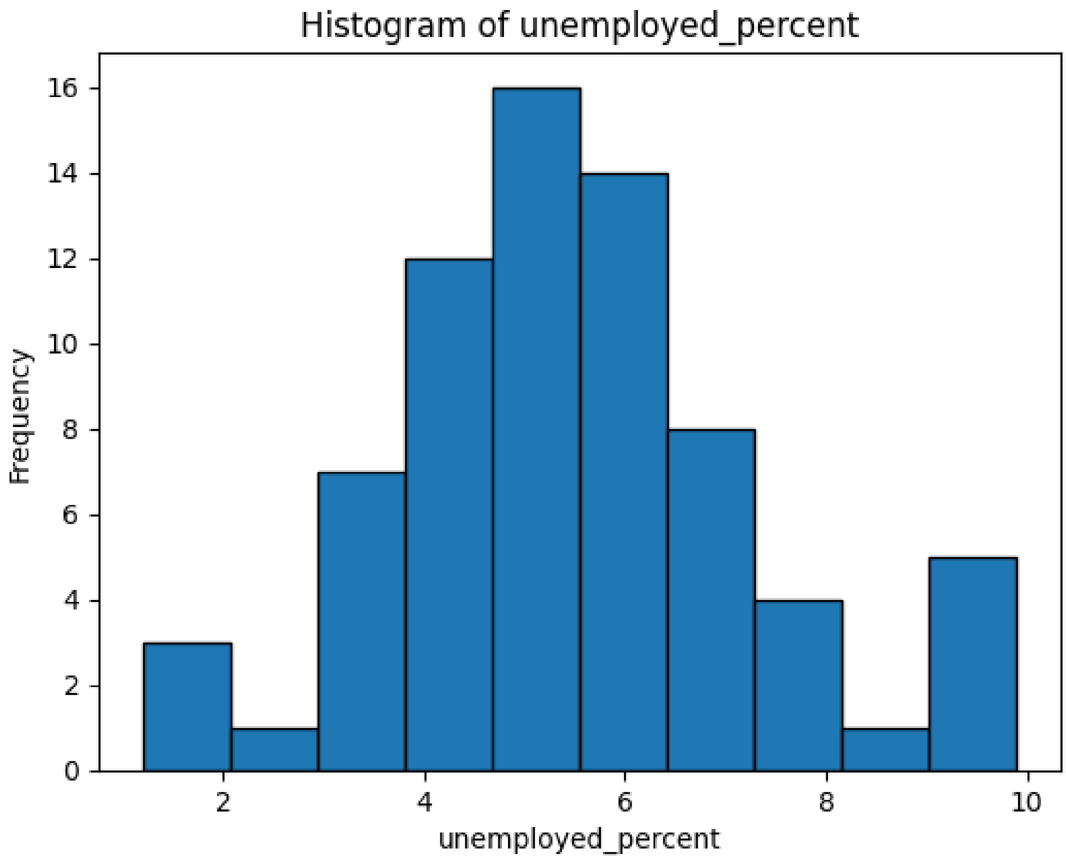
Total Number of Unemployed Distribution

```
column_to_analyze = 'unemployed'
plt.hist(data[column_to_analyze], bins=10, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



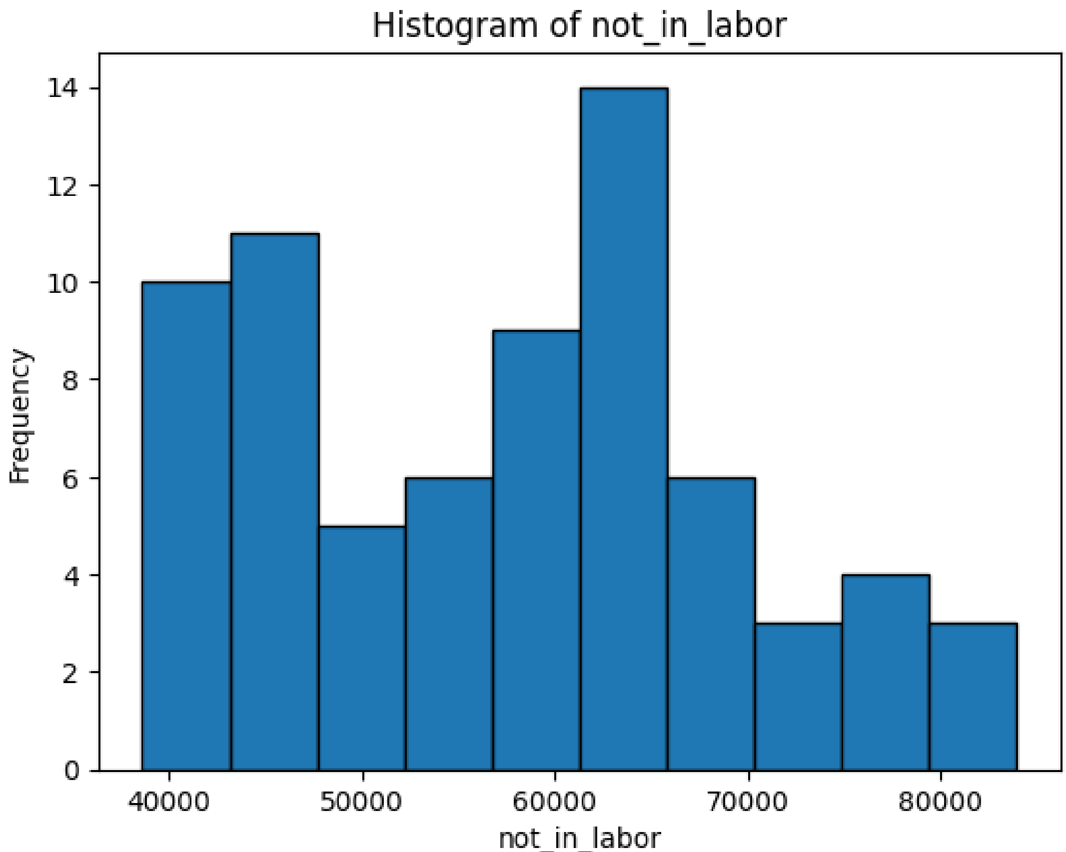
Percentage of Unemployed Distribution

```
column_to_analyze = 'unemployed_percent'
plt.hist(data[column_to_analyze], bins=10, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



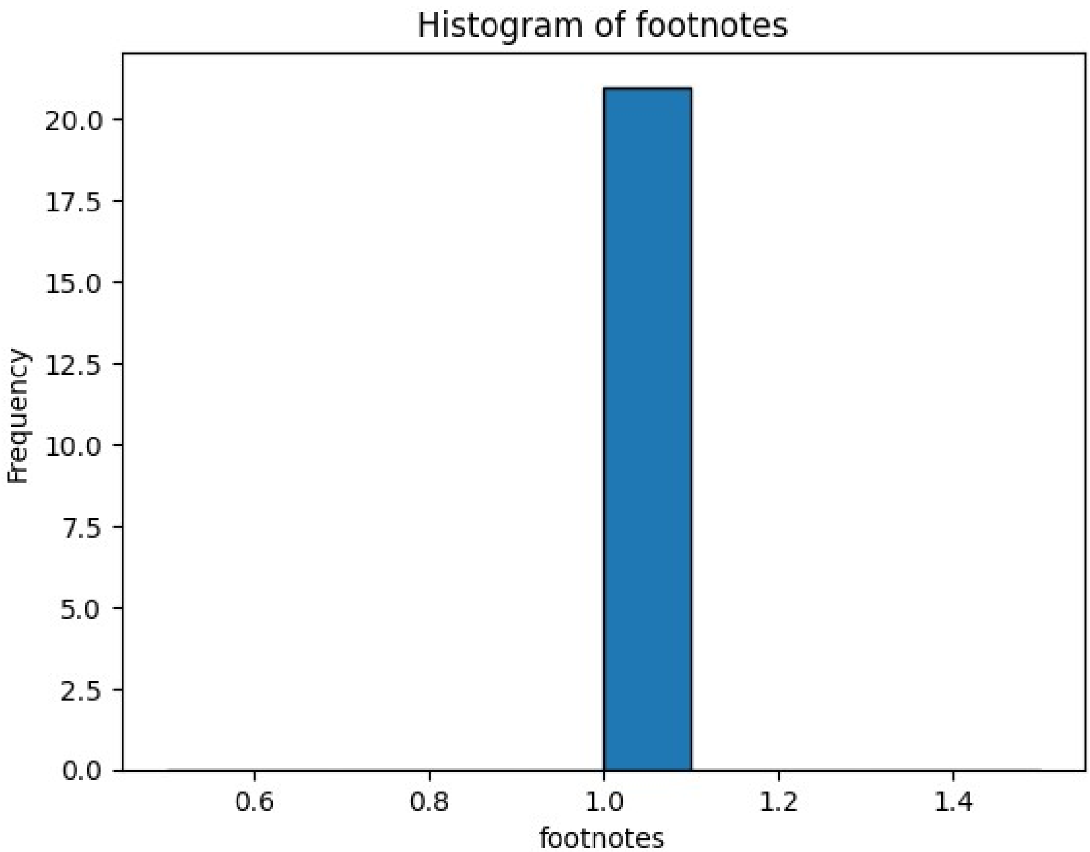
Number of Not in Labor Distribution

```
column_to_analyze = 'not_in_labor'
plt.hist(data[column_to_analyze], bins=10, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



Footnotes Distribution

```
column_to_analyze = 'footnotes'
plt.hist(data[column_to_analyze], bins=10, edgecolor='black')  #
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



Data Preprocessing

Column `agrictulture_ratio`

Fixing the typo agrictulture_ratio.

```
data.rename(columns={'agrictulture_ratio': 'agriculture_ratio'},
inplace=True)
data.head()

   year  population  labor_force  population_percent
employed_total  \
0  1941       99900        55910                56.0            50350

1  1942       98640        56410                57.2            53750

2  1943       94640        55540                58.7            54470

3  1944       93220        54630                58.6            53960

4  1945       94090        53860                57.2            52820
```

```
   employed_percent  agriculture_ratio  nonagriculture_ratio
unemployed  \
0              50.4               9100                 41250
5560
1              54.5               9250                 44500
2660
2              57.6               9080                 45390
1070
3              57.9               8950                 45010
670
4              56.1               8580                 44240
1040

   unemployed_percent  not_in_labor  footnotes
0                 9.9         43990        NaN
1                 4.7         42230        NaN
2                 1.9         39100        NaN
3                 1.2         38590        NaN
4                 1.9         40230        NaN
```

Column `year`

Fixing the duplicated year (1947) by finding the average of both data.

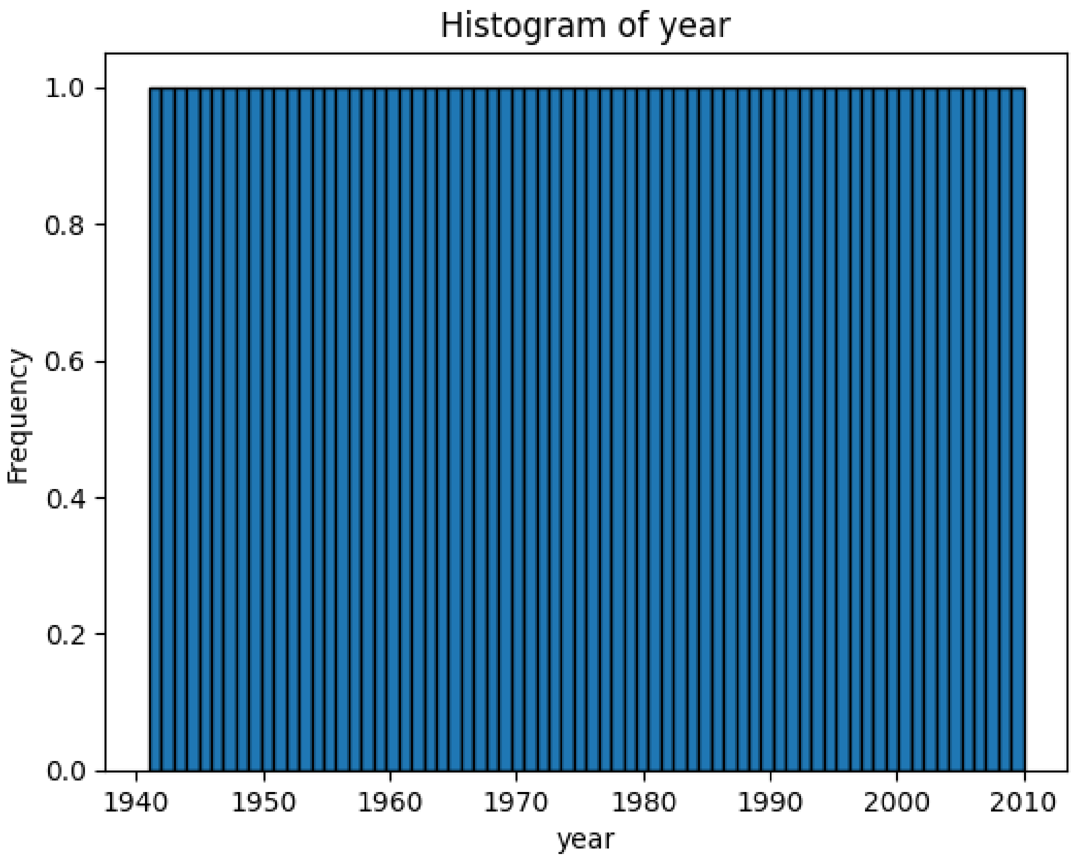
```
# Group by 'year' and calculate the mean for each group
data = data.groupby('year').mean().reset_index()

# Display the resulting DataFrame
print(data)

    year  population  labor_force  population_percent  employed_total
\
0   1941     99900.0      55910.0                56.0         50350.0

1   1942     98640.0      56410.0                57.2         53750.0

2   1943     94640.0      55540.0                58.7         54470.0

3   1944     93220.0      54630.0                58.6         53960.0

4   1945     94090.0      53860.0                57.2         52820.0

..   ...         ...          ...                 ...             ...

65  2006    228815.0     151428.0                66.2        144427.0

66  2007    231867.0     153124.0                66.0        146047.0

67  2008    233788.0     154287.0                66.0        145362.0
```

```
68  2009    235801.0     154142.0                  65.4        139877.0

69  2010    237830.0     153889.0                  64.7        139064.0


    employed_percent  agriculture_ratio  nonagriculture_ratio
unemployed  \
0               50.4             9100.0               41250.0
5560.0
1               54.5             9250.0               44500.0
2660.0
2               57.6             9080.0               45390.0
1070.0
3               57.9             8950.0               45010.0
670.0
4               56.1             8580.0               44240.0
1040.0
..               ...                ...                   ...
...
65              63.1             2206.0              142221.0
7001.0
66              63.0             2095.0              143952.0
7078.0
67              62.2             2168.0              143194.0
8924.0
68              59.3             2103.0              137775.0
14265.0
69              58.5             2206.0              136858.0
14825.0

    unemployed_percent  not_in_labor  footnotes
0                  9.9       43990.0        NaN
1                  4.7       42230.0        NaN
2                  1.9       39100.0        NaN
3                  1.2       38590.0        NaN
4                  1.9       40230.0        NaN
..                 ...           ...        ...
65                 4.6       77387.0        1.0
66                 4.6       78743.0        1.0
67                 5.8       79501.0        1.0
68                 9.3       81659.0        1.0
69                 9.6       83941.0        1.0

[70 rows x 12 columns]
```

Recheck the histogram distribution of column `year`

```
column_to_analyze = 'year'
plt.hist(data[column_to_analyze], bins=70, edgecolor='black')  #
```

```
Adjust the number of bins as needed
plt.xlabel('{}'.format(column_to_analyze))
plt.ylabel('Frequency')
plt.title('Histogram of {}'.format(column_to_analyze))
plt.show()
```



Based on the graph we can see that the duplicated data is now fixed.

Data Analysis

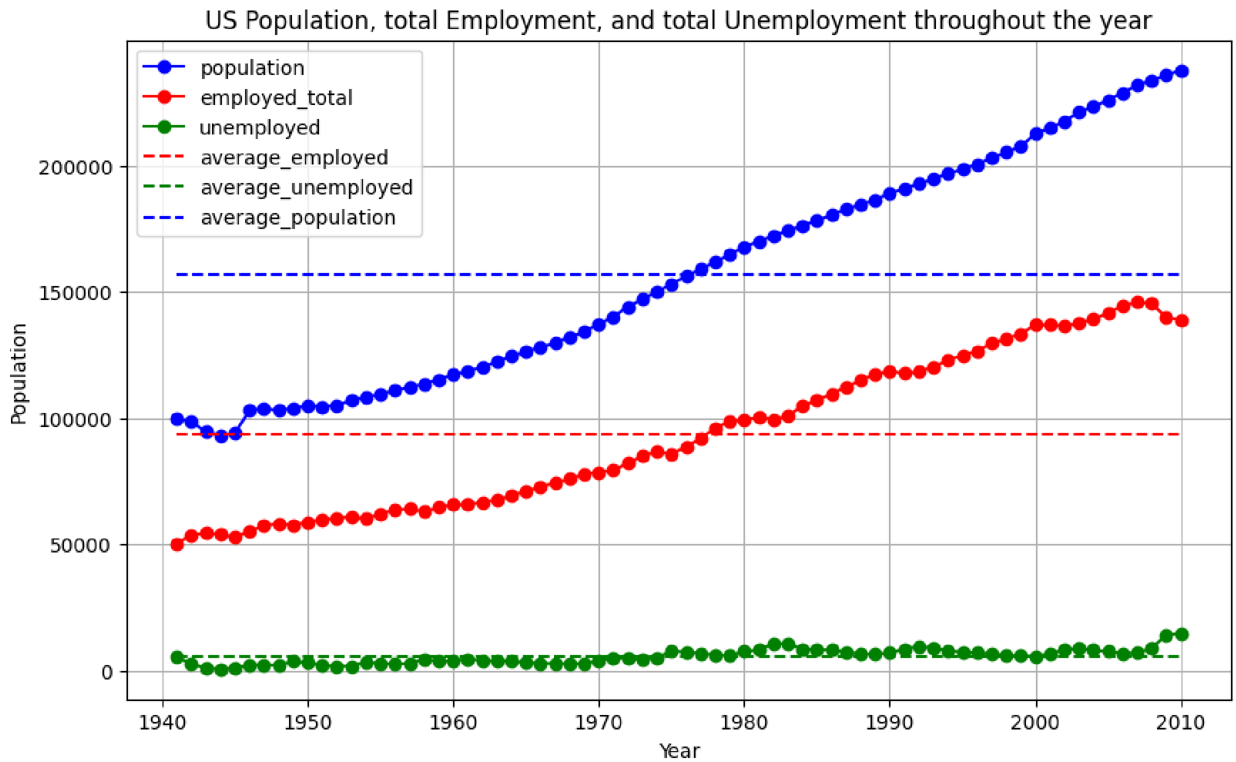
1. US Population, total Employment, and total Unemployment throughout the year.

```
plt.figure(figsize=(10,6))
data['average_employed'] = data['employed_total'].mean()
data['average_unemployed'] = data['unemployed'].mean()
data['average_population'] = data['population'].mean()

plt.plot(data['year'], data['population'], 'b-', label='population',
marker = 'o')
plt.plot(data['year'], data['employed_total'], 'r-',
label='employed_total', marker = 'o')
```

```
plt.plot(data['year'], data['unemployed'], 'g-', label='unemployed',
marker = 'o')
plt.plot(data['year'], data['average_employed'], 'r-',
label='average_employed', linestyle = '--')
plt.plot(data['year'], data['average_unemployed'], 'g-',
label='average_unemployed',linestyle = '--')
plt.plot(data['year'], data['average_population'], 'b-',
label='average_population',linestyle = '--')

plt.xlabel('Year')
plt.ylabel('Population')
plt.legend(loc='upper left')
plt.grid(True)
plt.title('US Population, total Employment, and total Unemployment
throughout the year')
```

```
C:\Users\emili\AppData\Local\Temp\ipykernel_14212\1694127570.py:9:
UserWarning: linestyle is redundantly defined by the 'linestyle'
keyword argument and the fmt string "r-" (-> linestyle='-'). The
keyword argument will take precedence.
  plt.plot(data['year'], data['average_employed'], 'r-',
label='average_employed', linestyle = '--')
C:\Users\emili\AppData\Local\Temp\ipykernel_14212\1694127570.py:10:
UserWarning: linestyle is redundantly defined by the 'linestyle'
keyword argument and the fmt string "g-" (-> linestyle='-'). The
keyword argument will take precedence.
  plt.plot(data['year'], data['average_unemployed'], 'g-',
label='average_unemployed',linestyle = '--')
C:\Users\emili\AppData\Local\Temp\ipykernel_14212\1694127570.py:11:
UserWarning: linestyle is redundantly defined by the 'linestyle'
keyword argument and the fmt string "b-" (-> linestyle='-'). The
keyword argument will take precedence.
  plt.plot(data['year'], data['average_population'], 'b-',
label='average_population',linestyle = '--')
```

```
Text(0.5, 1.0, 'US Population, total Employment, and total
Unemployment throughout the year')
```

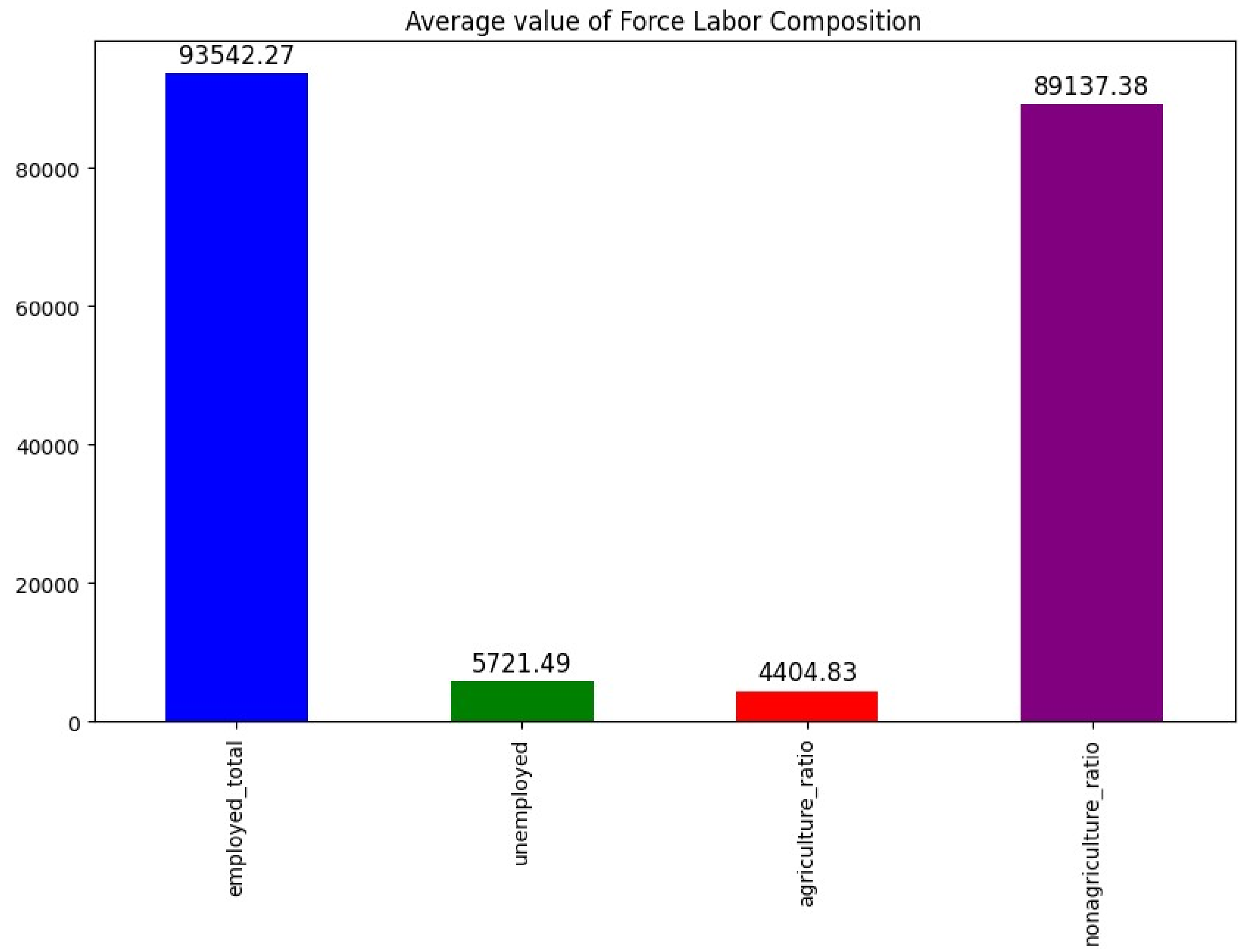


As we can see from the plot above, the total employment is steadily increasing throughout the year, following the increase of the population. However we can also see that the increase in population does not exactly match the increased in employment as we can see a drop of employed in the year 2010. Meanwhile the level of unemployed seems to stay low throughout the year although it shows a slight increase starting from 1970 with the largest jump in 2010.

2. Force Labor Composition in the US

2.1 Average Force Labor composition

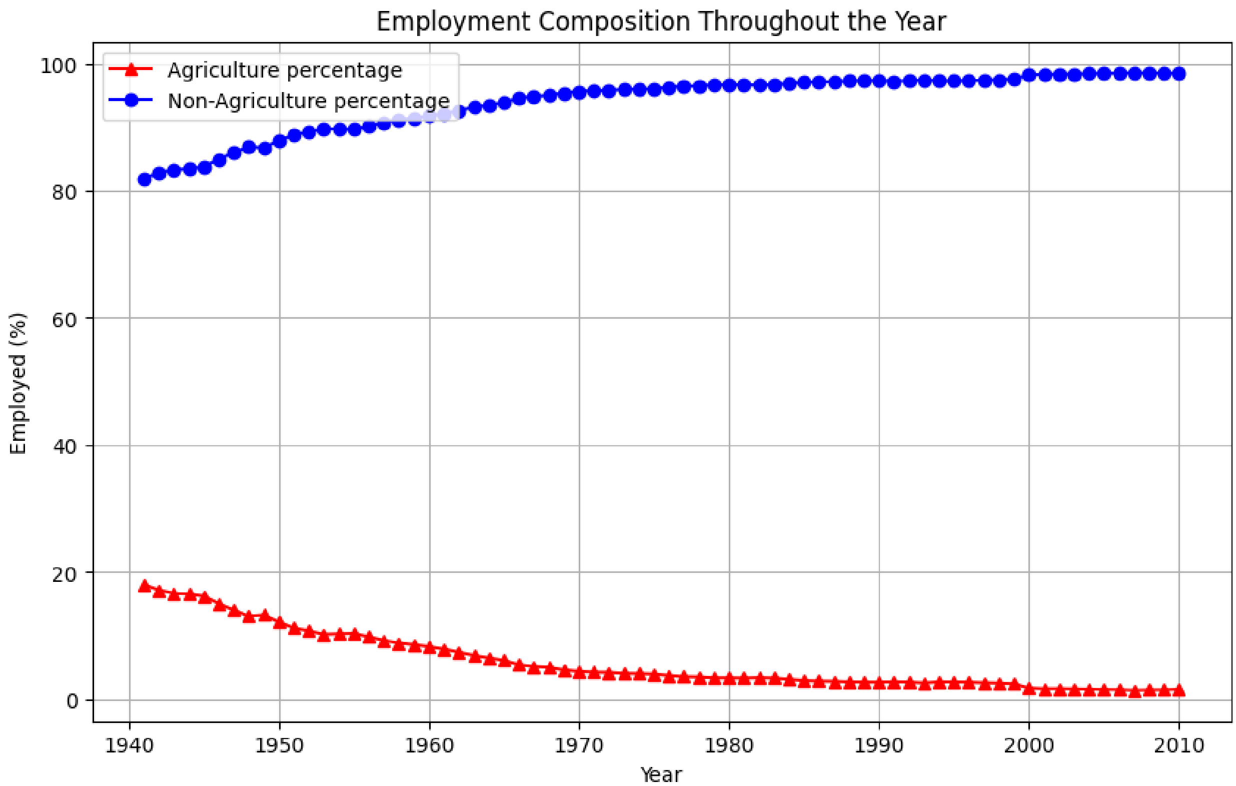
```
average_data = data.mean()

ax = average_data[['employed_total', 'unemployed',
'agriculture_ratio', 'nonagriculture_ratio']].plot(kind='bar',
figsize=(10, 6), color = ['blue', 'green', 'red', 'purple'])

for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() /
2., p.get_height()),
                ha='center', va='center', fontsize=12, color='black',
xytext=(0, 9),
                textcoords='offset points')
plt.title('Average value of Force Labor Composition')
plt.show()
```



The plot above shows the average value of the composition of the labor force which includes composition of the employed_total (agriculture and non agricultute)

2.2 Employment Composition throughout the year

```
data['agriculture_ratio_rate'] = (data['agriculture_ratio'] /
data['employed_total']) * 100
data['non_agriculture_ratio_rate'] = (data['nonagriculture_ratio'] /
data['employed_total']) * 100
plt.figure(figsize=(10,6))
plt.plot(data['year'], data['agriculture_ratio_rate'], 'r-',label =
'Agriculture percentage',marker = '^')
plt.plot(data['year'], data['non_agriculture_ratio_rate'], 'b-',label
= 'Non-Agriculture percentage',marker = 'o')

plt.title('Employment Composition Throughout the Year')
plt.xlabel('Year')
plt.ylabel('Employed (%)')
plt.legend(loc='upper left')
plt.grid(True)

plt.show()
```



As seen from the plot above, we can see the decline of interest in the agriculture field where it started from roughly 18% of the employed to just around 1 or 2%. Meanwhile the interest in non-agriculture field has been increasing in contrast.

3. Footnotes Analysis

```
footnotes_df = data[data['footnotes']==1]
footnotes_df.describe()
```

```
              year     population    labor_force  
population_percent  \
count    21.000000      21.000000      21.000000                 21.000000

mean   1991.142857  191978.190476  124891.523810                 64.514286

std      17.788439   42436.257856   31280.608731                  2.947590

min    1953.000000  107056.000000   63015.000000                 58.800000

25%    1978.000000  161910.000000  102250.000000                 63.200000

50%    1998.000000  205220.000000  137673.000000                 66.000000

75%    2005.000000  226082.000000  149320.000000                 66.500000

max    2010.000000  237830.000000  154287.000000                 67.100000
```

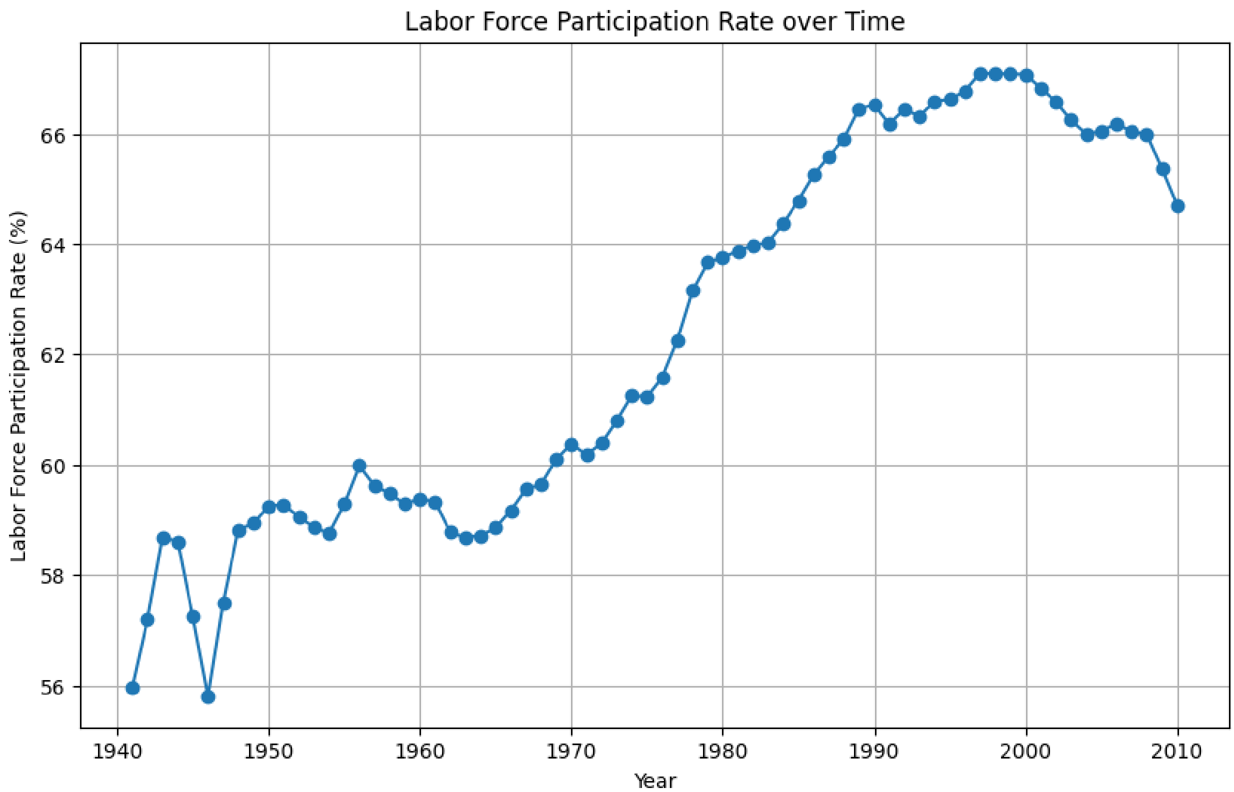
```
       employed_total  employed_percent  agriculture_ratio  \
count       21.000000         21.000000          21.000000   
mean    117774.714286         60.895238        3181.047619   
std      29160.380342          2.907830        1153.559512   
min      61179.000000         55.500000        2095.000000   
25%      96048.000000         58.500000        2206.000000   
50%     131463.000000         62.300000        3223.000000   
75%     139252.000000         63.000000        3409.000000   
max     146047.000000         64.400000        6260.000000   

       nonagriculture_ratio    unemployed  unemployed_percent  
not_in_labor  \
count             21.000000     21.000000           21.000000  
21.000000   
mean          114593.619048   7116.857143            5.585714  
67086.523810   
std            30200.645097   3054.797167            1.559258  
11465.462623   
min            54919.000000   1834.000000            2.900000  
44041.000000   
25%            92661.000000   5692.000000            4.600000  
59659.000000   
50%           128085.000000   7001.000000            5.500000  
67547.000000   
75%           137020.000000   8149.000000            6.000000  
76762.000000   
max           143952.000000  14825.000000            9.600000  
83941.000000   

       footnotes  average_employed  average_unemployed  
average_population  \
count       21.0      2.100000e+01        2.100000e+01  
2.100000e+01   
mean         1.0      9.354227e+04        5.721493e+03  
1.570204e+05   
std          0.0      2.982255e-11        9.319547e-13  
5.964510e-11   
min          1.0      9.354227e+04        5.721493e+03  
1.570204e+05   
25%          1.0      9.354227e+04        5.721493e+03  
1.570204e+05   
50%          1.0      9.354227e+04        5.721493e+03  
1.570204e+05   
75%          1.0      9.354227e+04        5.721493e+03  
1.570204e+05   
max          1.0      9.354227e+04        5.721493e+03  
1.570204e+05   

       agriculture_ratio_rate  non_agriculture_ratio_rate  
```

```
count                    21.000000                     21.000000
mean                      3.236027                     96.763900
std                       2.458280                      2.458467
min                       1.434470                     89.767731
25%                       1.586320                     96.473638
50%                       2.569544                     97.430456
75%                       3.526362                     98.413680
max                      10.232269                     98.565530
```

As seen from the table above, there is no correlation or anything that is unique from the data which includes footnotes. Footnotes itself should essentially means additional notes or understanding on the data, but the data above doesn't really tell us anything.

4. Labor Force Participant Rate

```
data['labor_force_participation_rate'] = (data['labor_force'] / 
data['population']) * 100

plt.figure(figsize=(10,6))
plt.plot(data['year'], data['labor_force_participation_rate'], marker 
= 'o', linestyle = '-')

plt.title('Labor Force Participation Rate over Time')
plt.xlabel('Year')
plt.ylabel('Labor Force Participation Rate (%)')
plt.grid(True)

plt.show()
```



As seen from the chart above, the percentage of the labor force seems to be increasing per population throughout the year with the all time high at around 67% of the population.

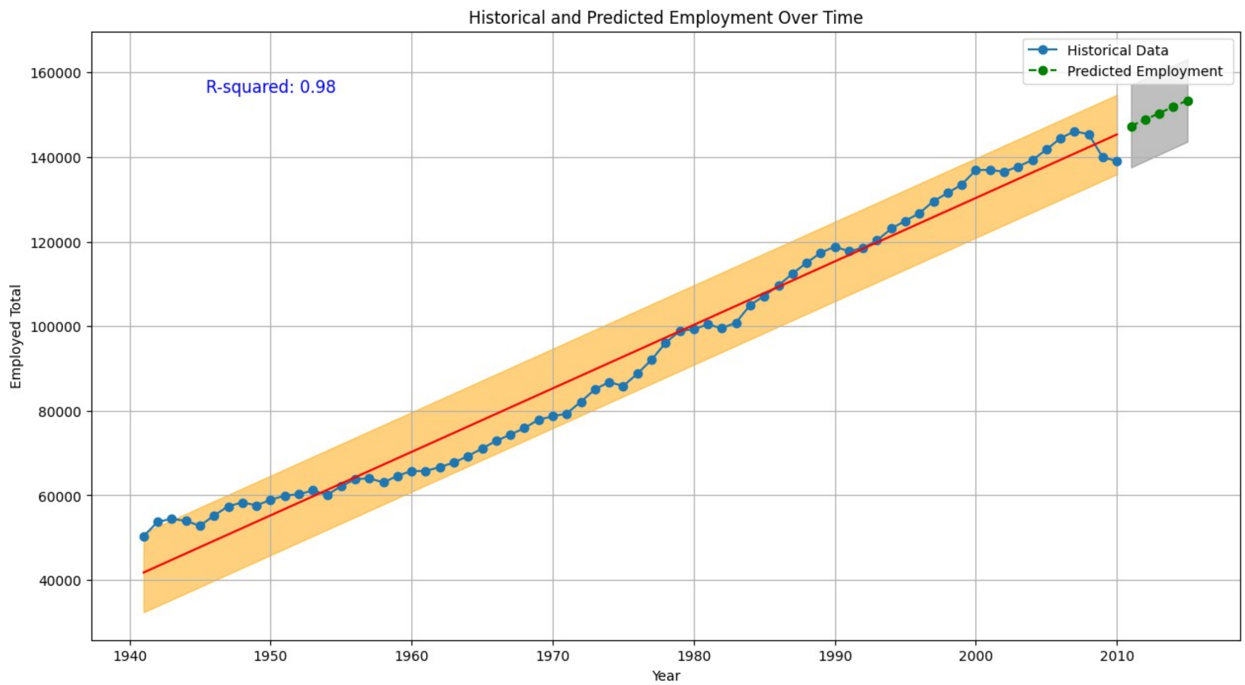
5. Predictive Model for Employed and Unemployed

5.1 Employed Pattern Prediction

```
x = data[['year']]
y = data['employed_total']

# split data
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=42)

# set the train model
model = LinearRegression()
model.fit(x_train, y_train)

# predict employment
future_years = np.array([max(data['year']) + i for i in range(1,
6)]).reshape(-1, 1)
future_predictions = model.predict(future_years)

# interval
mse = mean_squared_error(y_test, model.predict(x_test))
confidence_interval = 1.96 * sqrt(mse)
```

```
lower_bound = future_predictions - confidence_interval
upper_bound = future_predictions + confidence_interval

# linear regression
model_all_data = LinearRegression()
model_all_data.fit(x, y)

# interval
mse_all_data = mean_squared_error(y, model_all_data.predict(x))
confidence_interval_all_data = 1.96 * sqrt(mse_all_data)

lower_bound_all_data = model_all_data.predict(x) -
confidence_interval_all_data
upper_bound_all_data = model_all_data.predict(x) +
confidence_interval_all_data

r_squared = r2_score(y, model_all_data.predict(x))
plt.figure(figsize=(15, 8))
plt.plot(data['year'], data['employed_total'], marker='o',
linestyle='-', label='Historical Data')
plt.plot(future_years, future_predictions, marker='o', linestyle='--',
color='g', label='Predicted Employment ')
plt.fill_between(future_years.flatten(), lower_bound, upper_bound,
color='gray', alpha=0.5)
plt.plot(x, model_all_data.predict(x), color='r')
plt.fill_between(x['year'], lower_bound_all_data,
upper_bound_all_data, color='orange', alpha=0.5)
plt.title('Historical and Predicted Employment Over Time')
plt.xlabel('Year')
plt.ylabel('Employed Total')
plt.grid(True)
plt.legend()

# Display the R-squared as text on the plot
plt.text(0.1, 0.9, f'R-squared: {r_squared:.2f}',
transform=plt.gca().transAxes, fontsize=12, color='blue')
plt.show()
```

```
c:\Users\emili\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\base.py:465: UserWarning: X does not have valid
feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```



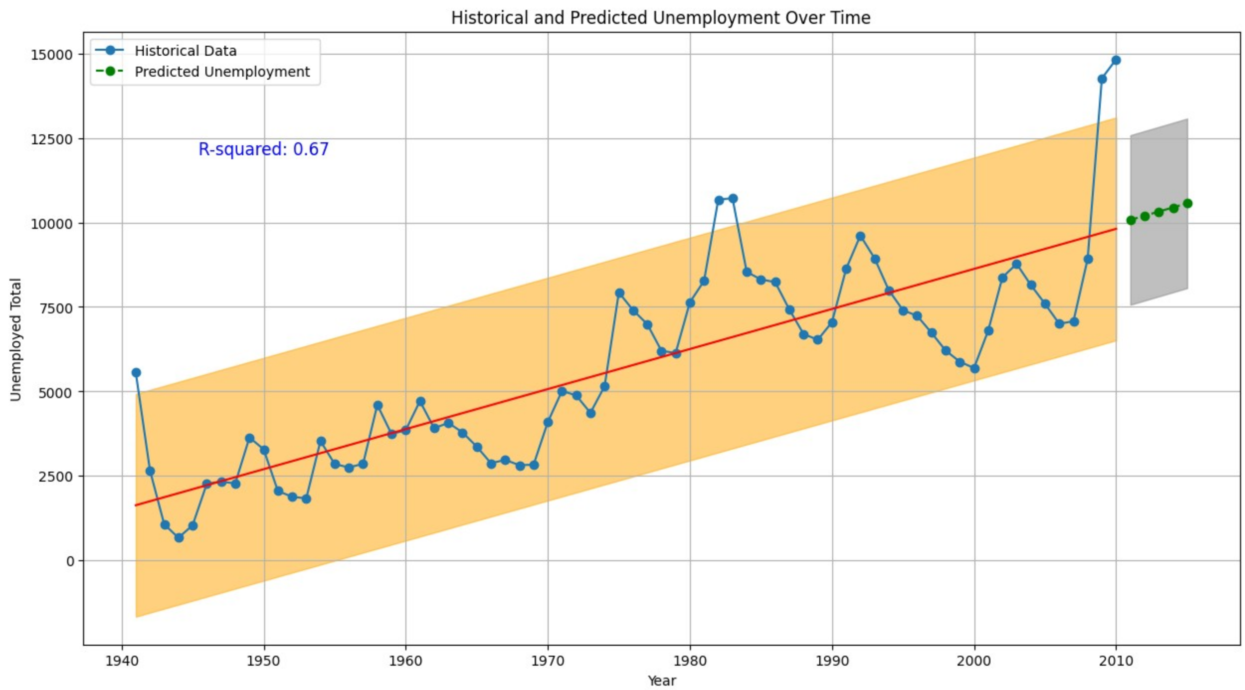
With an accuracy of 98%, there is a chance of an increase in employment rate

5.2 Unemployed Pattern Prediction

```
x = data[['year']]
y = data['unemployed']

# split data
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=42)

# set the train model
model = LinearRegression()
model.fit(x_train, y_train)

# predict employment
future_years = np.array([max(data['year']) + i for i in range(1,
6)]).reshape(-1, 1)
future_predictions = model.predict(future_years)

# interval
mse = mean_squared_error(y_test, model.predict(x_test))
confidence_interval = 1.96 * sqrt(mse)

lower_bound = future_predictions - confidence_interval
upper_bound = future_predictions + confidence_interval

# linear regression
model_all_data = LinearRegression()
model_all_data.fit(x, y)
```

```python
# interval
mse_all_data = mean_squared_error(y, model_all_data.predict(x))
confidence_interval_all_data = 1.96 * sqrt(mse_all_data)

lower_bound_all_data = model_all_data.predict(x) -
confidence_interval_all_data
upper_bound_all_data = model_all_data.predict(x) +
confidence_interval_all_data

r_squared = r2_score(y, model_all_data.predict(x))
plt.figure(figsize=(15, 8))
plt.plot(data['year'], data['unemployed'], marker='o', linestyle='-',
label='Historical Data')
plt.plot(future_years, future_predictions, marker='o', linestyle='--',
color='g', label='Predicted Unemployment ')
plt.fill_between(future_years.flatten(), lower_bound, upper_bound,
color='gray', alpha=0.5)
plt.plot(x, model_all_data.predict(x), color='r')
plt.fill_between(x['year'], lower_bound_all_data,
upper_bound_all_data, color='orange', alpha=0.5)
plt.title('Historical and Predicted Unemployment Over Time')
plt.xlabel('Year')
plt.ylabel('Unemployed Total')
plt.grid(True)
plt.legend()

# Display the R-squared as text on the plot
plt.text(0.1, 0.8, f'R-squared: {r_squared:.2f}',
transform=plt.gca().transAxes, fontsize=12, color='blue')
plt.show()
```

```
c:\Users\emili\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\base.py:465: UserWarning: X does not have valid
feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```



From the prediction of unemployment trend over time graph above, it can be seen that there is an orange area, which indicates the linear regression pattern and provide upper and lower boundary. There was periods of time that are beyond the given upper boundary which indicates the recession periods, which shows that during recession, the number of unemployment are increasing beyond and breaching the upper boundary of the normal area.

6. Pre and Post Recession Analysis

6.1 Employment Pattern Analysis

```
recession1_start_year = 1981
recession1_end_year = 1982
recession2_start_year = 2008
recession2_end_year = 2009

pre_1981_mask = (data['year'] <= recession1_start_year)
between_1981_1982_mask = ((data['year'] >= recession1_start_year) &
(data['year'] <= recession1_end_year))
between_1982_2008_mask = ((data['year'] >= recession1_end_year) &
(data['year'] <= recession2_start_year))
between_2008_2009_mask = ((data['year'] >= recession2_start_year) &
(data['year'] <= recession2_end_year))
post_2009_mask = (data['year'] >= recession2_end_year)

plt.figure(figsize=(12, 6))

plt.plot(data[pre_1981_mask]['year'], data[pre_1981_mask]
['employed_percent'], color='black')
plt.plot(data[between_1981_1982_mask]['year'],
```

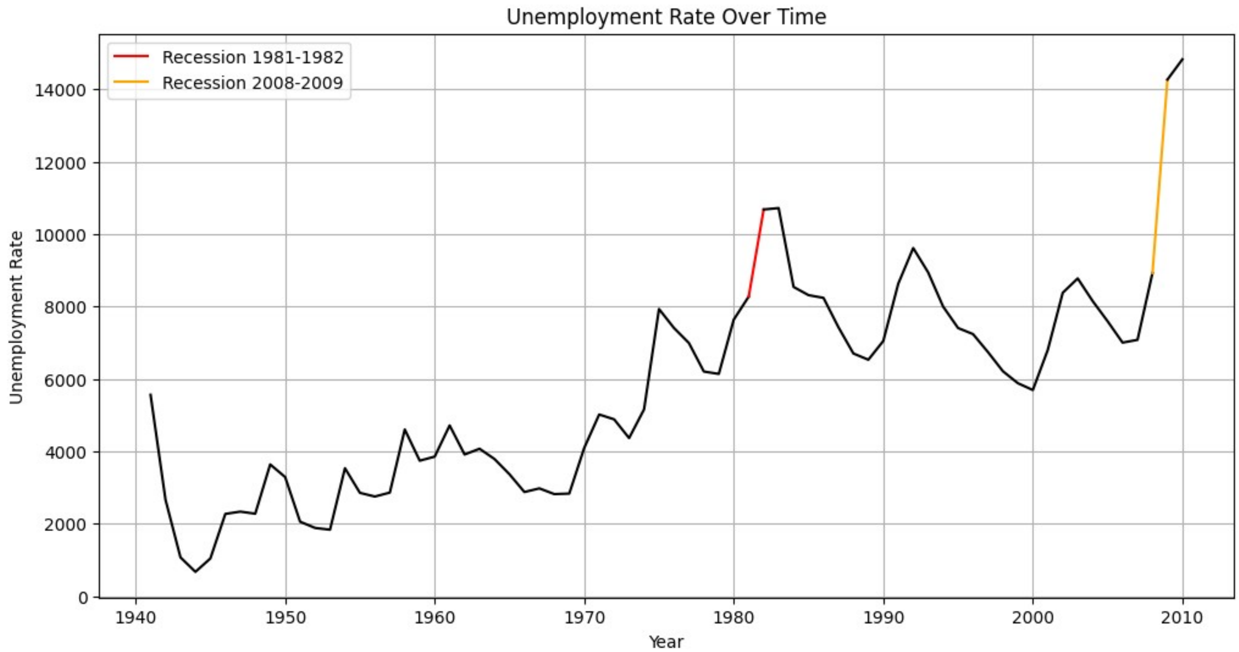
```
data[between_1981_1982_mask]['employed_percent'], color='red',
label='Recession 1981-1982')
plt.plot(data[between_1982_2008_mask]['year'],
data[between_1982_2008_mask]['employed_percent'], color='black')
plt.plot(data[between_2008_2009_mask]['year'],
data[between_2008_2009_mask]['employed_percent'], color='orange',
label='Recession 2008-2009')
plt.plot(data[post_2009_mask]['year'], data[post_2009_mask]
['employed_percent'], color='black')

plt.title('Employment Rate Over Time')
plt.xlabel('Year')
plt.ylabel('Employment Rate')
plt.grid(True)
plt.legend()
plt.show()
```

From the Employment Rate Over Time graph above, it can be seen in green and red line, that there were significant drops during recession periods, which occur during 1981-1982 and 2008-2009. The pattern indicates that both drops are approaching the `Employment Rate` value of 58. However the bounce back after the 1981-1982 recession can be seen in year 1983.

6.2 Unemployment Pattern Analysis

```
recession1_start_year = 1981
recession1_end_year = 1982
recession2_start_year = 2008
recession2_end_year = 2009
```

```
pre_1981_mask = (data['year'] <= recession1_start_year)
between_1981_1982_mask = ((data['year'] >= recession1_start_year) &
(data['year'] <= recession1_end_year))
between_1982_2008_mask = ((data['year'] >= recession1_end_year) &
(data['year'] <= recession2_start_year))
between_2008_2009_mask = ((data['year'] >= recession2_start_year) &
(data['year'] <= recession2_end_year))
post_2009_mask = (data['year'] >= recession2_end_year)

plt.figure(figsize=(12, 6))

plt.plot(data[pre_1981_mask]['year'], data[pre_1981_mask]
['unemployed'], color='black')
plt.plot(data[between_1981_1982_mask]['year'],
data[between_1981_1982_mask]['unemployed'], color='red',
label='Recession 1981-1982')
plt.plot(data[between_1982_2008_mask]['year'],
data[between_1982_2008_mask]['unemployed'], color='black')
plt.plot(data[between_2008_2009_mask]['year'],
data[between_2008_2009_mask]['unemployed'], color='orange',
label='Recession 2008-2009')
plt.plot(data[post_2009_mask]['year'], data[post_2009_mask]
['unemployed'], color='black')

plt.title('Unemployment Rate Over Time')
plt.xlabel('Year')
plt.ylabel('Unemployment Rate')
plt.grid(True)
plt.legend()
plt.show()
```



The Unemployment Rate Over Time graph above indicate that the during the both recession, there were a significant increase in the unemployment rate, which can be seen in red and yellow line. However, after the recession 1981-1982, there was a decrease amount of unemployment rate and it showed in a ranging type of graph. This period is keep ranging until the Global Recession 2008-2009 period.

From both of the line graphs above, it can be seen during the recession period, there were an increase in Unemployment in the US. However from the given graph also, the comparison between both picture was unable to be scaled in a same size. The ratio for unemployement rate was too huge if was compared directly with the employment rate. This showing there was huge number of lay-off activities and a small number of people hired.

Data Export

Export to CSV

```
data.to_csv('US_Employment_Rate.csv')
```

This part of the code is used to export the data to a csv file to be present in a form of a dashboard through Power BI.