test

February 13, 2024

[303]:

```
import pandas as pd
from tabulate import tabulate
# reading csv files
```

[304]:

```
colnames =[
"state",
"county",
"community",
"communityname",
"fold",
"population",
"householdsize",
"racepctblack",
"racePctWhite",
"racePctAsian",
"racePctHisp",
"agePct12t21",
"agePct12t29",
"agePct16t24",
"agePct65up",
"numbUrban",
"pctUrban",
"medIncome",
"pctWWage",
"pctWFarmSelf",
"pctWInvInc",
"pctWSocSec",
"pctWPubAsst",
"pctWRetire",
"medFamInc",
"perCapInc",
"whitePerCap",
"blackPerCap",
"indianPerCap",
"AsianPerCap",
"OtherPerCap",
"HispPerCap",
"NumUnderPov",
"PctPopUnderPov",
"PctLess9thGrade",
"PctNotHSGrad",
"PctBSorMore",
"PctUnemployed",
"PctEmploy",
"PctEmplManu",
"PctEmplProfServ",
"PctOccupManu",
"PctOccupMgmtProf",
"MalePctDivorce",
"MalePctNevMarr",
"FemalePctDiv",
"TotalPctDiv",
"PersPerFam",
"PctFam2Par",
"PctKids2Par",
"PctYoungKids2Par",
"PctTeen2Par",
"PctWorkMomYoungKids",
"PctWorkMom",
"NumIlleg",
"PctIlleg",
"NumImmig",
"PctImmigRecent",
"PctImmigRec5",
"PctImmigRec8",
"PctImmigRec10",
"PctRecentImmig",
"PctRecImmig5",
"PctRecImmig8",
"PctRecImmig10",
"PctSpeakEnglOnly",
"PctNotSpeakEnglWell",
"PctLargHouseFam",
"PctLargHouseOccup",
"PersPerOccupHous",
"PersPerOwnOccHous",
"PersPerRentOccHous",
"PctPersOwnOccup",
"PctPersDenseHous",
"PctHousLess3BR",
"MedNumBR",
"HousVacant",
"PctHousOccup",
"PctHousOwnOcc",
"PctVacantBoarded",
"PctVacMore6Mos",
"MedYrHousBuilt",
"PctHousNoPhone",
"PctWOFullPlumb",
"OwnOccLowQuart",
"OwnOccMedVal",
"OwnOccHiQuart",
"RentLowQ",
"RentMedian",
"RentHighQ",
"MedRent",
"MedRentPctHousInc",
"MedOwnCostPctInc",
"MedOwnCostPctIncNoMtg",
"NumInShelters",
"NumStreet",
"PctForeignBorn",
"PctBornSameState",
"PctSameHouse85",
"PctSameCity85",
"PctSameState85",
"LemasSwornFT",
"LemasSwFTPerPop",
"LemasSwFTFieldOps",
"LemasSwFTFieldPerPop",
"LemasTotalReq",
"LemasTotReqPerPop",
"PolicReqPerOffic",
"PolicPerPop",
"RacialMatchCommPol",
"PctPolicWhite",
"PctPolicBlack",
"PctPolicHisp",
"PctPolicAsian",
"PctPolicMinor",
"OfficAssgnDrugUnits",
"NumKindsDrugsSeiz",
"PolicAveOTWorked",
"LandArea",
"PopDens",
"PctUsePubTrans",
"PolicCars",
"PolicOperBudg",
"LemasPctPolicOnPatr",
"LemasGangUnitDeploy",
"LemasPctOfficDrugUn",
"PolicBudgPerPop",
"ViolentCrimesPerPop",
]
```

columns we should keep

[“communityname”, “population”, “householdsize”, “agePct12t21”, “agePct12t29”, “agePct16t24”, “agePct65up”, “medIncome”, “pctWWage”, “pctWFarmSelf”, “pctWInvInc”, “pctWSocSec”, “pctWPubAsst”, “pctWRetire”, “medFamInc”, “perCapInc”, “NumUnderPov”, “PctPopUnderPov”, “PctLess9thGrade”, “PctNotHSGrad”, “PctBSorMore”, “PctUnemployed”, “PctEmploy”, “PctEmplManu”, “PctEmplProfServ”, “PctOccupManu”, “PctOccupMgmtProf”, “MalePctDivorce”, “MalePctNevMarr”, “FemalePctDiv”, “TotalPctDiv”, “PersPerFam”, “PctFam2Par”, “PctKids2Par”, “PctYoungKids2Par”, “PctTeen2Par”, “PctWorkMomYoungKids”, “PctWorkMom”, “NumIlleg”, “PctIlleg”, “NumImmig”, “PctImmigRecent”, “PctImmigRec5”, “PctImmigRec8”, “PctImmigRec10”, “PctRecentImmig”, “PctRecImmig5”, “PctRecImmig8”, “PctRecImmig10”, “PctSpeakEnglOnly”, “PctNotSpeakEnglWell”, “PctLargHouseFam”, “PctLargHouseOccup”, “PersPerOccupHous”, “PersPerOwnOccHous”, “PersPerRentOccHous”, “PctPersOwnOccup”, “PctPersDenseHous”, “PctHousLess3BR”, “MedNumBR”, “HousVacant”, “PctHousOccup”, “PctHousOwnOcc”, “PctVacantBoarded”, “PctVacMore6Mos”, “MedYrHousBuilt”, “PctHousNoPhone”, “PctWOFullPlumb”, “OwnOccLowQuart”, “OwnOccMedVal”, “OwnOccHiQuart”, “RentLowQ”, “RentMedian”, “RentHighQ”, “MedRent”, “MedRentPctHousInc”, “MedOwnCostPctInc”, “MedOwnCostPctIncNoMtg”, “NumInShelters”, “PctForeignBorn”, “PctBornSameState”, “LemasSwornFT”, “LemasSwFTPerPop”, “LemasSwFTFieldOps”, “LemasSwFTFieldPerPop”, “LemasTotalReq”, “LemasTotReqPerPop”, “PolicReqPerOffic”, “PolicPerPop”, “RacialMatchCommPol”, “OfficAssgnDrugUnits”, “NumKindsDrugsSeiz”, “PolicAveOTWorked”, “PopDens”, “PolicCars”, “PolicOperBudg”, “LemasPctPolicOnPatr”, “LemasGangUnitDeploy”, “LemasPctOfficDrugUn”, “PolicBudgPerPop”, “ViolentCrimesPerPop”,]

[305]:
```
data =  pd.read_csv('communities.data', names=colnames, header=None)
print(len(data))
data.replace("?", pd.NA, inplace=True)
data.dropna(axis=1, inplace=True)
```

```
1994
```

[306]:
```
print(data.columns.tolist())
```

```
['state', 'communityname', 'fold', 'population', 'householdsize',
'racepctblack', 'racePctWhite', 'racePctAsian', 'racePctHisp', 'agePct12t21',
'agePct12t29', 'agePct16t24', 'agePct65up', 'numbUrban', 'pctUrban',
'medIncome', 'pctWWage', 'pctWFarmSelf', 'pctWInvInc', 'pctWSocSec',
'pctWPubAsst', 'pctWRetire', 'medFamInc', 'perCapInc', 'whitePerCap',
'blackPerCap', 'indianPerCap', 'AsianPerCap', 'HispPerCap', 'NumUnderPov',
'PctPopUnderPov', 'PctLess9thGrade', 'PctNotHSGrad', 'PctBSorMore',
'PctUnemployed', 'PctEmploy', 'PctEmplManu', 'PctEmplProfServ', 'PctOccupManu',
'PctOccupMgmtProf', 'MalePctDivorce', 'MalePctNevMarr', 'FemalePctDiv',
'TotalPctDiv', 'PersPerFam', 'PctFam2Par', 'PctKids2Par', 'PctYoungKids2Par',
'PctTeen2Par', 'PctWorkMomYoungKids', 'PctWorkMom', 'NumIlleg', 'PctIlleg',
'NumImmig', 'PctImmigRecent', 'PctImmigRec5', 'PctImmigRec8', 'PctImmigRec10',
'PctRecentImmig', 'PctRecImmig5', 'PctRecImmig8', 'PctRecImmig10',
'PctSpeakEnglOnly', 'PctNotSpeakEnglWell', 'PctLargHouseFam',
'PctLargHouseOccup', 'PersPerOccupHous', 'PersPerOwnOccHous',
'PersPerRentOccHous', 'PctPersOwnOccup', 'PctPersDenseHous', 'PctHousLess3BR',
'MedNumBR', 'HousVacant', 'PctHousOccup', 'PctHousOwnOcc', 'PctVacantBoarded',
'PctVacMore6Mos', 'MedYrHousBuilt', 'PctHousNoPhone', 'PctWOFullPlumb',
'OwnOccLowQuart', 'OwnOccMedVal', 'OwnOccHiQuart', 'RentLowQ', 'RentMedian',
'RentHighQ', 'MedRent', 'MedRentPctHousInc', 'MedOwnCostPctInc',
'MedOwnCostPctIncNoMtg', 'NumInShelters', 'NumStreet', 'PctForeignBorn',
'PctBornSameState', 'PctSameHouse85', 'PctSameCity85', 'PctSameState85',
'LandArea', 'PopDens', 'PctUsePubTrans', 'LemasPctOfficDrugUn',
'ViolentCrimesPerPop']
```

[307]:

```
columns_of_interest = ["medIncome",
                       "ViolentCrimesPerPop",
                       "population",
                       "pctWPubAsst",
                       "pctWRetire",
                       "householdsize",
                       "PctUnemployed",
                       "PctWorkMomYoungKids",
                       "PctNotSpeakEnglWell",
                       "HousVacant",
                       "PctVacMore6Mos",
                       "PctWOFullPlumb",
                       "MedRent",
                       "NumStreet",
                       "PopDens",
                       ]

data = data[columns_of_interest]


data_sort = data.sort_values('ViolentCrimesPerPop')
least_crime_100 = data_sort.head(10)
print(tabulate(data_sort.head(10), headers='keys', tablefmt='pretty'))
print(tabulate(data_sort.tail(10), headers='keys', tablefmt='pretty'))
```

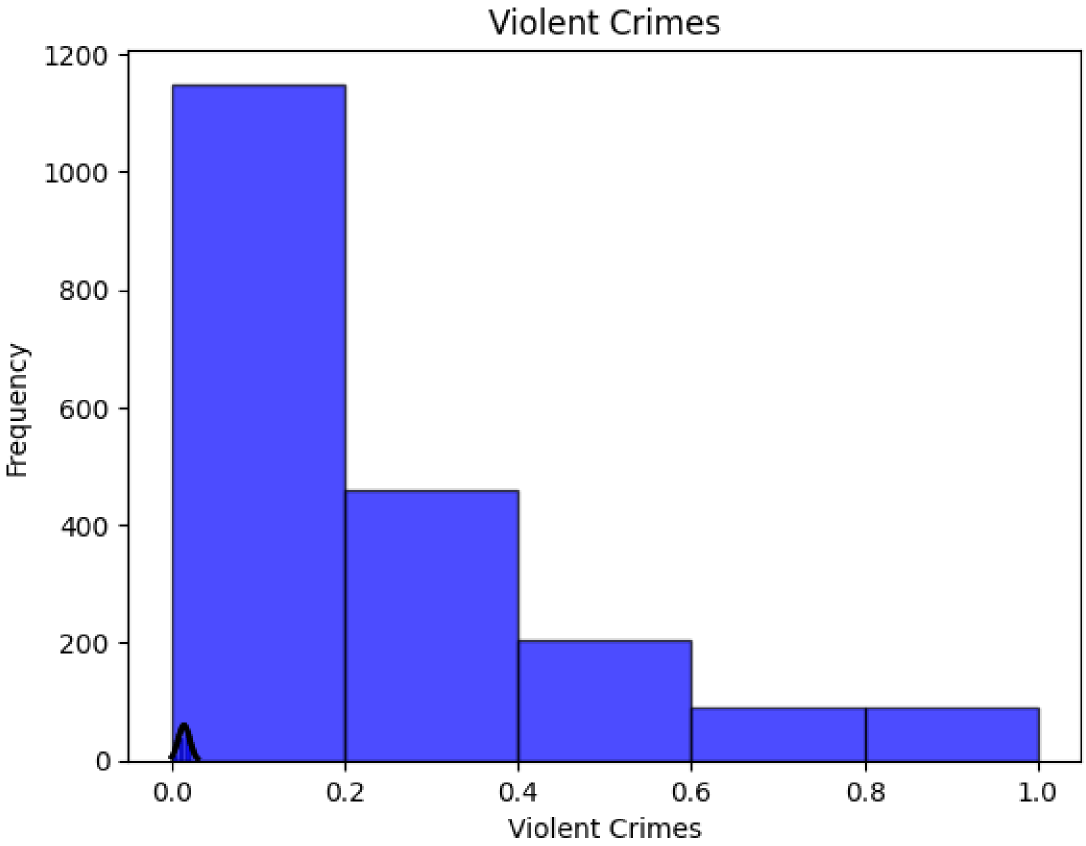
```
+------+-----------+---------------------+------------+-------------+-----------
-+---------------+---------------+---------------------+---------------------+--
----------+----------------+----------------+---------+-----------+---------+
|      | medIncome | ViolentCrimesPerPop | population | pctWPubAsst | pctWRetire
| householdsize | PctUnemployed | PctWorkMomYoungKids | PctNotSpeakEnglWell |
HousVacant | PctVacMore6Mos | PctWOFullPlumb | MedRent | NumStreet | PopDens |
+------+-----------+---------------------+------------+-------------+-----------
-+---------------+---------------+---------------------+---------------------+--
----------+----------------+----------------+---------+-----------+---------+
|  83  |    0.4    |         0.0         |    0.01    |     0.2     |    0.8
|     0.52      |     0.23      |         0.4         |        0.03         |
0.0     |      0.68      |      0.09      |  0.25   |    0.0    |  0.05   |
| 1230 |   0.19    |         0.0         |    0.01    |     0.5     |    0.74
|     0.67      |     0.61      |        0.79         |        0.12         |
0.01    |      0.65      |      0.34      |  0.16   |    0.0    |  0.22   |
| 1462 |   0.69    |         0.0         |    0.01    |    0.03     |    0.4
|     0.65      |     0.23      |        0.19         |        0.05         |
0.02    |      0.28      |      0.08      |  0.62   |    0.0    |  0.07   |
| 773  |    0.2    |         0.0         |    0.01    |    0.36     |    0.65
|     0.41      |     0.54      |         0.2         |        0.04         |
0.02    |      0.57      |      0.14      |  0.24   |   0.03    |  0.21   |
| 1656 |   0.81    |         0.0         |    0.01    |    0.08     |    0.39
|     0.63      |     0.12      |        0.47         |        0.04         |
0.01    |      0.54      |      0.26      |  0.43   |    0.0    |  0.03   |
| 519  |   0.21    |         0.0         |    0.0     |    0.37     |    0.2
|     0.31      |     0.04      |        0.81         |        0.01         |
0.01    |      0.6       |      0.33      |  0.09   |    0.0    |  0.09   |
| 529  |   0.33    |         0.0         |    0.0     |    0.28     |    0.53
|     0.37      |      0.3      |        0.74         |        0.23         |
0.01    |      0.65      |      0.66      |  0.37   |    0.0    |  0.15   |
| 1174 |   0.87    |         0.0         |    0.02    |    0.07     |    0.53
|     0.51      |      0.1      |        0.33         |        0.02         |
0.01    |      0.61      |      0.09      |   0.7   |    0.0    |  0.05   |
| 342  |   0.59    |         0.0         |    0.0     |    0.06     |    0.13
|      1.0      |     0.03      |        0.43         |        0.01         |
0.01    |      0.52      |      0.31      |  0.51   |    0.0    |  0.04   |
| 426  |   0.48    |         0.0         |    0.01    |    0.07     |    0.34
|     0.54      |     0.25      |        0.33         |        0.01         |
0.01    |      0.24      |      0.23      |  0.49   |    0.0    |  0.05   |
+------+-----------+---------------------+------------+-------------+-----------
-+---------------+---------------+---------------------+---------------------+--
----------+----------------+----------------+---------+-----------+---------+
+------+-----------+---------------------+------------+-------------+-----------
-+---------------+---------------+---------------------+---------------------+--
----------+----------------+----------------+---------+-----------+---------+
|      | medIncome | ViolentCrimesPerPop | population | pctWPubAsst | pctWRetire
| householdsize | PctUnemployed | PctWorkMomYoungKids | PctNotSpeakEnglWell |
HousVacant | PctVacMore6Mos | PctWOFullPlumb | MedRent | NumStreet | PopDens |
+------+-----------+---------------------+------------+-------------+-----------
-+---------------+---------------+---------------------+---------------------+--
----------+----------------+----------------+---------+-----------+---------+
| 1025 |   0.12    |         1.0         |    0.02    |    0.54     |    0.46
|      0.4      |      0.5      |        0.71         |        0.03         |
0.04    |      0.67      |      0.32      |  0.13   |   0.01    |  0.14   |
| 1044 |   0.32    |         1.0         |    0.96    |    0.43     |    0.53
|     0.32      |     0.44      |        0.63         |        0.16         |
1.0     |      0.55      |      0.47      |  0.35   |    0.3    |  0.83   |
| 1041 |    0.3    |         1.0          |    0.1     |    0.25     |    0.52
|     0.23       |      0.39       |        0.54          |         0.26          |
0.56     |       0.33       |      0.16       |  0.43   |  0.15     |   0.3    |
| 828  |   0.18    |         1.0          |    0.02    |    0.56     |    0.57
|     0.17       |      0.58       |        0.42          |         0.04          |
0.08     |       0.59       |      0.56       |  0.25   |   0.0     |  0.77    |
| 1957 |   0.25    |         1.0          |    0.27    |    0.26     |    0.42
|     0.32       |      0.31       |        0.71          |         0.03          |
0.45     |       0.45       |      0.29       |  0.28   |   0.0     |  0.14    |
| 1134 |   0.09    |         1.0          |    0.56    |    0.89     |    0.22
|     0.49       |      0.75       |        0.43          |          1.0          |
0.76     |       0.38       |      0.83       |  0.26   |  0.61     |  0.84    |
| 149  |   0.23    |         1.0          |    0.02    |    0.41     |    0.38
|     0.47       |      0.31       |        0.33          |         0.13          |
0.05     |       0.36       |      0.42       |  0.26   |  0.01     |  0.09    |
| 146  |   0.17    |         1.0          |    0.34    |    0.47     |    0.45
|     0.43       |      0.62       |        0.56          |         0.05          |
0.73     |       0.7        |      0.29       |  0.2    |  0.02     |  0.25    |
| 1154 |   0.15    |         1.0          |    0.05    |    0.97     |    0.52
|     0.56       |      0.86       |         0.6          |         0.06          |
0.1      |       0.77       |      0.41       |  0.25   |   0.0     |  0.72    |
| 909  |   0.25    |         1.0          |    0.68    |    0.37     |    0.42
|     0.33       |      0.46       |        0.63          |         0.06          |
1.0      |       0.45       |      0.29       |  0.26   |  0.22     |  0.12    |
+------+-----------+----------------------+------------+-------------+-----------
-+----------------+-----------------+----------------------+----------------------+--
----------+----------------+----------------+---------+-----------+---------+
```

[308]:
```python
safest_100 = data_sort.head(100)
unsafest_100 = data_sort.tail(100)



# look for regression model linear regression model
# statistical models python
```

[309]:
```python
import matplotlib.pyplot as plt
from scipy.stats import norm
import numpy as np
violent_crimes = safest_100["ViolentCrimesPerPop"]
print(violent_crimes.head())
print(violent_crimes.tail())
data_violent = data["ViolentCrimesPerPop"]
print(data_violent.head())
print(data_violent.tail())

mu, std = norm.fit(violent_crimes)
```

```
x = np.linspace(0, .03, 100)
p = norm.pdf(x, mu, std)
p_data = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2)
plt.plot(x, p_data, 'k', linewidth=2)
plt.hist(violent_crimes, bins=5,  alpha=0.7, color='blue', edgecolor='black')
plt.hist(data_violent, bins=5,  alpha=0.7, color='blue', edgecolor='black')
plt.xlabel('Violent Crimes')
plt.ylabel('Frequency')
plt.title('Violent Crimes')

# Show the plot
plt.show()
```

```
83      0.0
1230    0.0
1462    0.0
773     0.0
1656    0.0
Name: ViolentCrimesPerPop, dtype: float64
796     0.02
800     0.02
1761    0.02
194     0.02
1708    0.02
Name: ViolentCrimesPerPop, dtype: float64
0    0.20
1    0.67
2    0.43
3    0.12
4    0.03
Name: ViolentCrimesPerPop, dtype: float64
1989    0.09
1990    0.45
1991    0.23
1992    0.19
1993    0.48
Name: ViolentCrimesPerPop, dtype: float64
```



[339]:

```
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
```

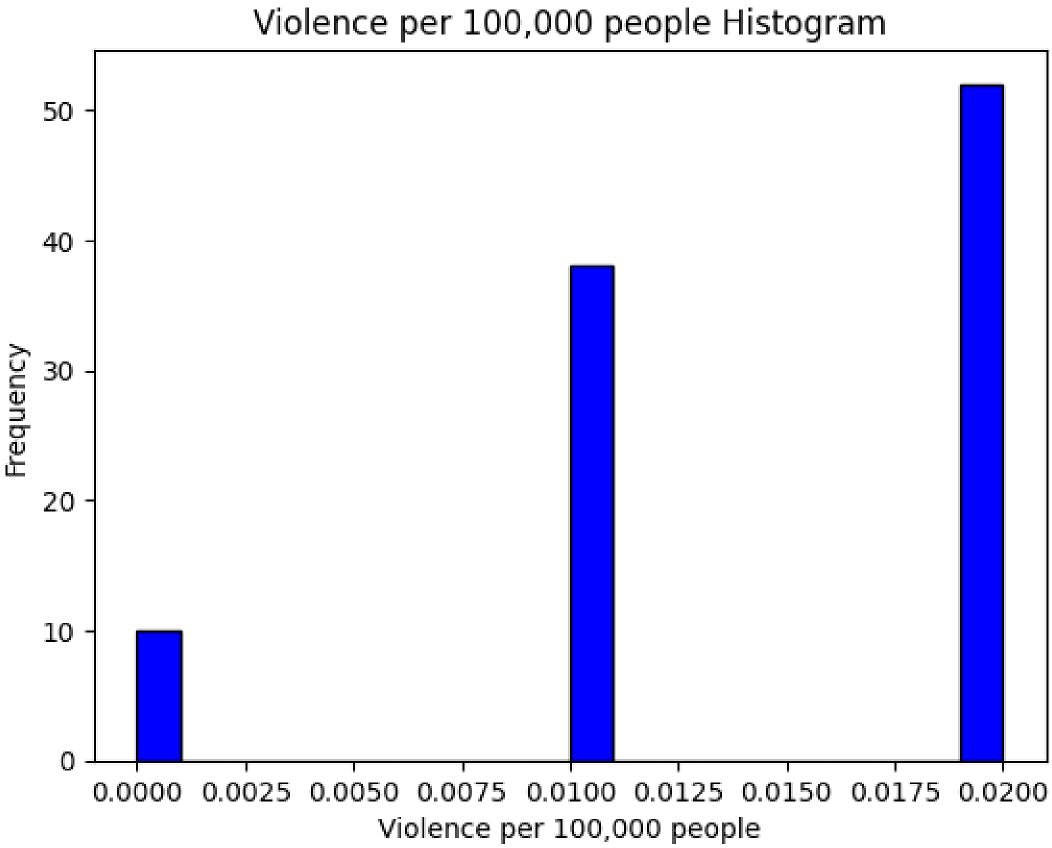
[ ]:

```
#histogram

def histogram(item, str):
  plt.hist(item, bins=20, color='blue', edgecolor='black')
  plt.title(f'{str} Histogram')
  plt.xlabel(str)
  plt.ylabel('Frequency')
  plt.show()
```

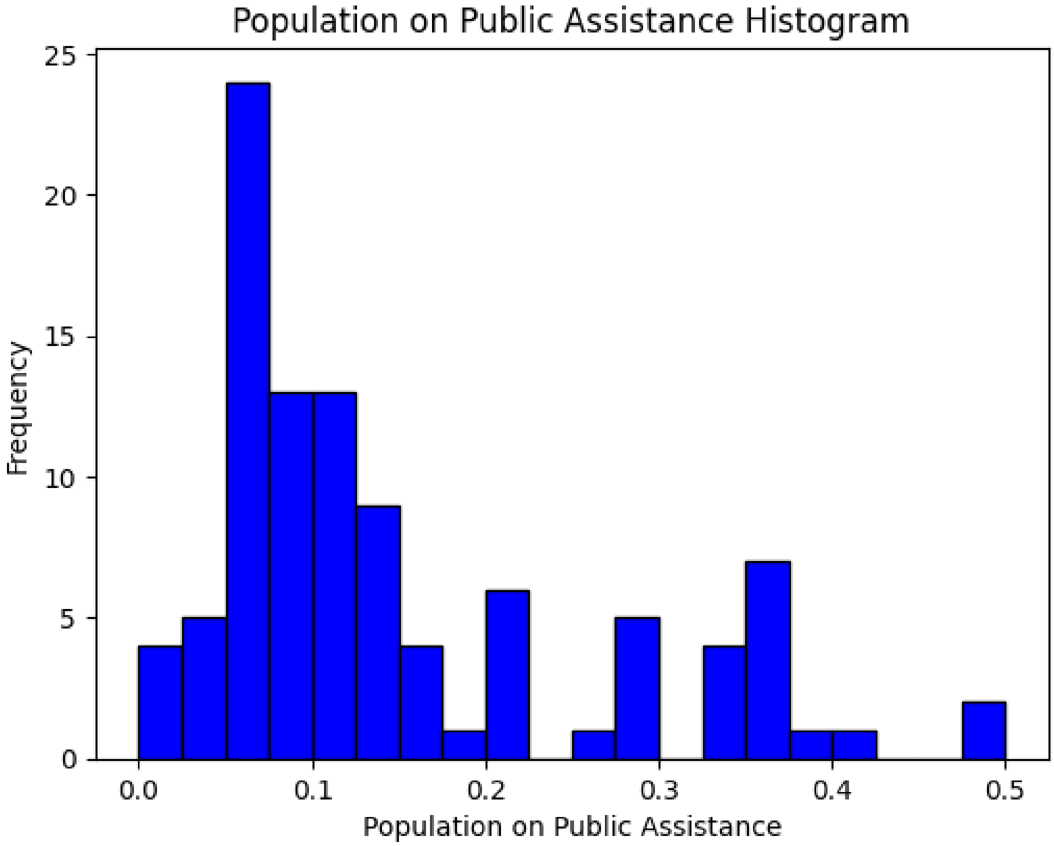
[369]:

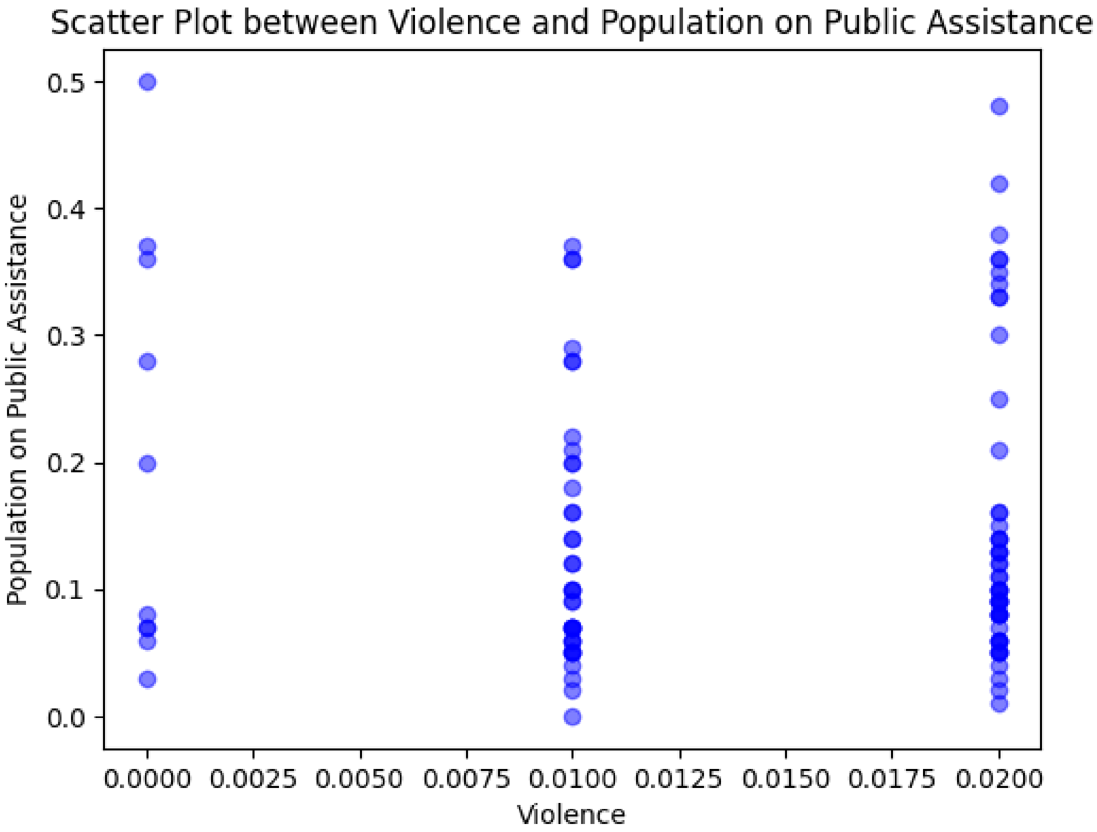
```
#scatter plot

def scatter(x_axis, y_axis, x_str, y_str):
  len(x_axis)
  len(y_axis)
  plt.scatter(x_axis, y_axis, alpha=0.5, color='blue')
  plt.title(f'Scatter Plot between {x_str} and {y_str}')
  plt.xlabel(x_str)
  plt.ylabel(y_str)
  plt.show()
```

[385]:
```
violence  = safest_100["ViolentCrimesPerPop"]
histogram(violence, "Violence per 100,000 people")
```



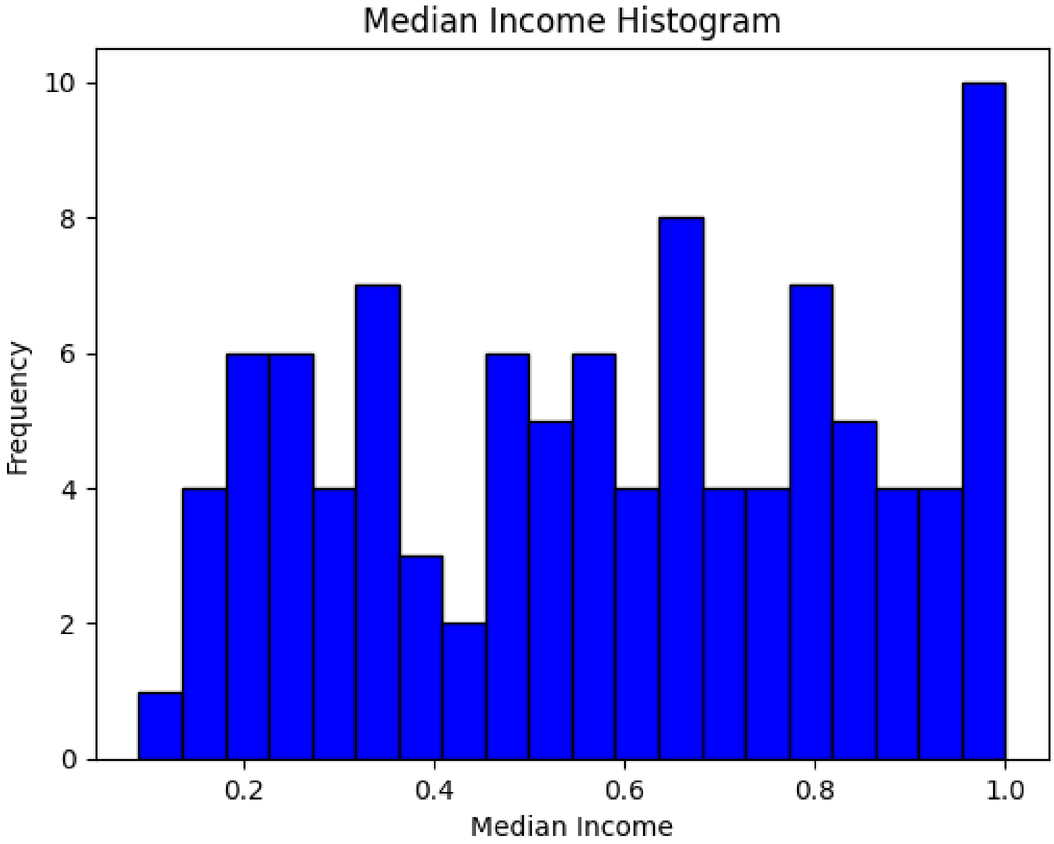
[370]:
```
pctWPubAsst = safest_100["pctWPubAsst"]
histogram(pctWPubAsst, "Population on Public Assistance")
scatter(violence, pctWPubAsst, "Violence", "Population on Public Assistance")

#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, pctWPubAsst)
print(pval_no_chage)
```

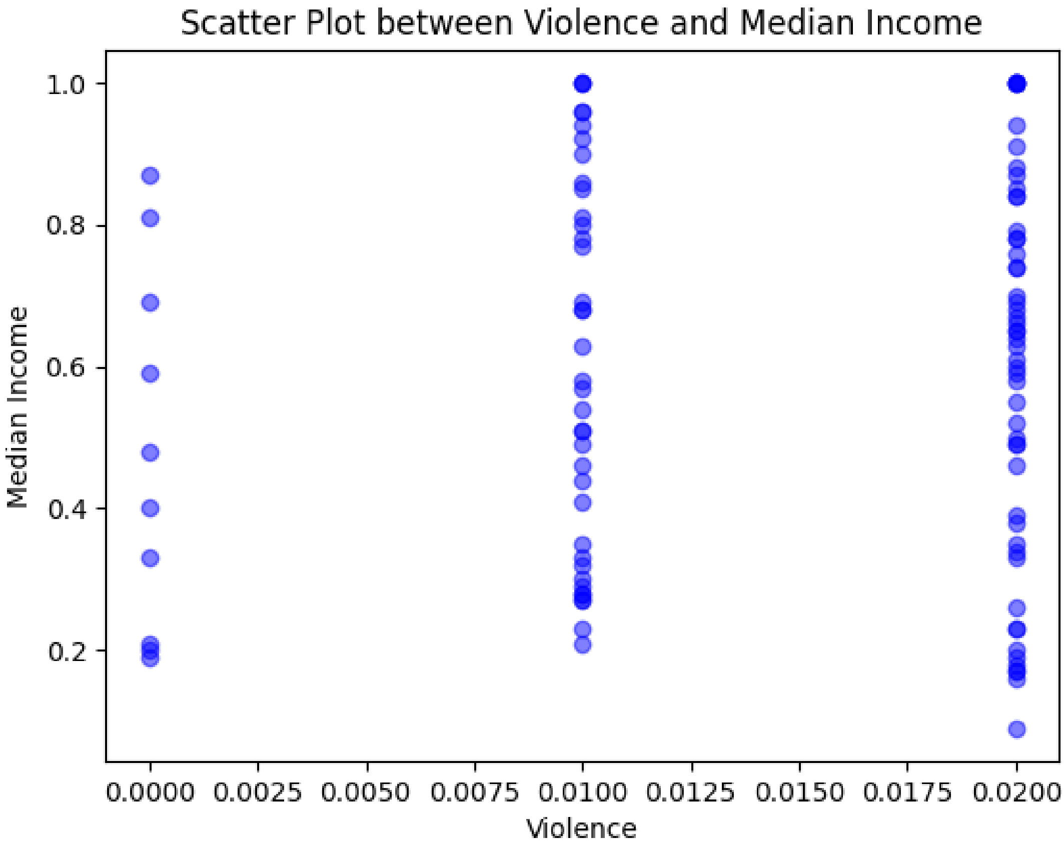

Population on Public Assistance Histogram
Frequency
25
20
15
10
5
0
0.0
0.1
0.2
0.3
0.4
0.5
Population on Public Assistance



```
1.5664417496455568e-23
```

[371]:
```
medIncome = safest_100["medIncome"]
histogram(medIncome, "Median Income")
scatter(violence, medIncome, "Violence", "Median Income")
#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, medIncome)
print(pval_no_chage)
```

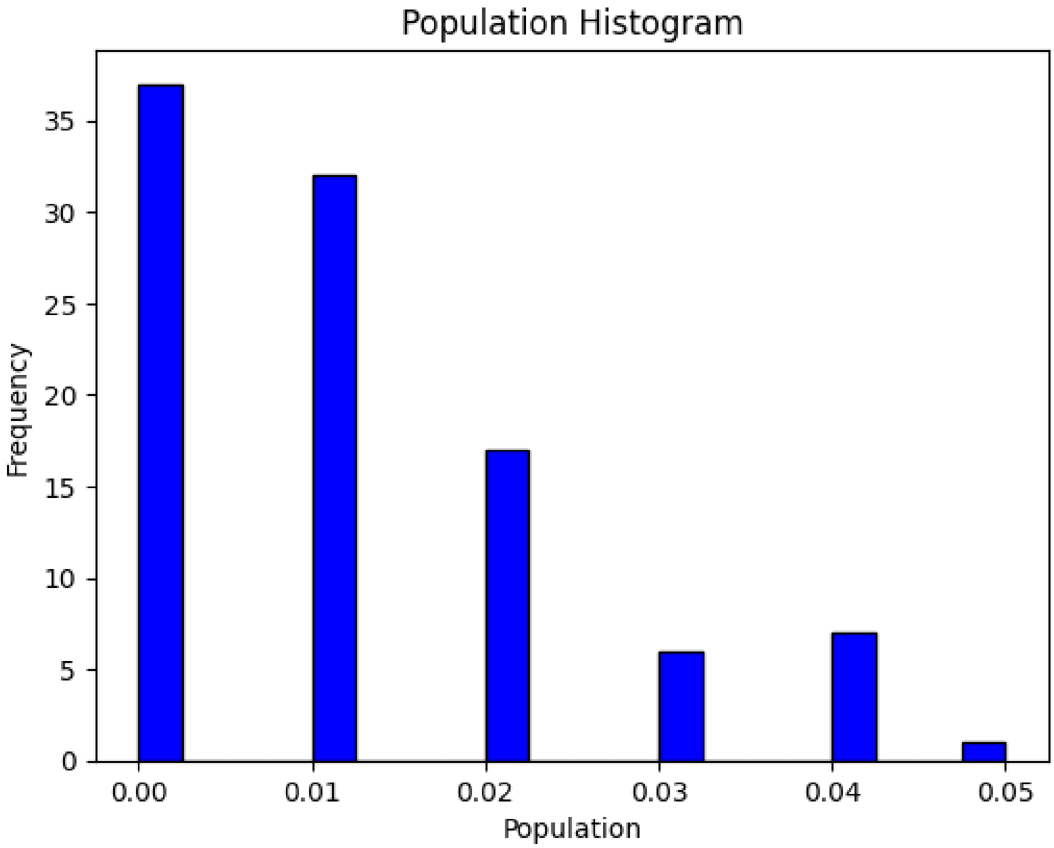

Median Income Histogram
10
8
6
4
2
0
Frequency
0.2
0.4
0.6
0.8
1.0
Median Income

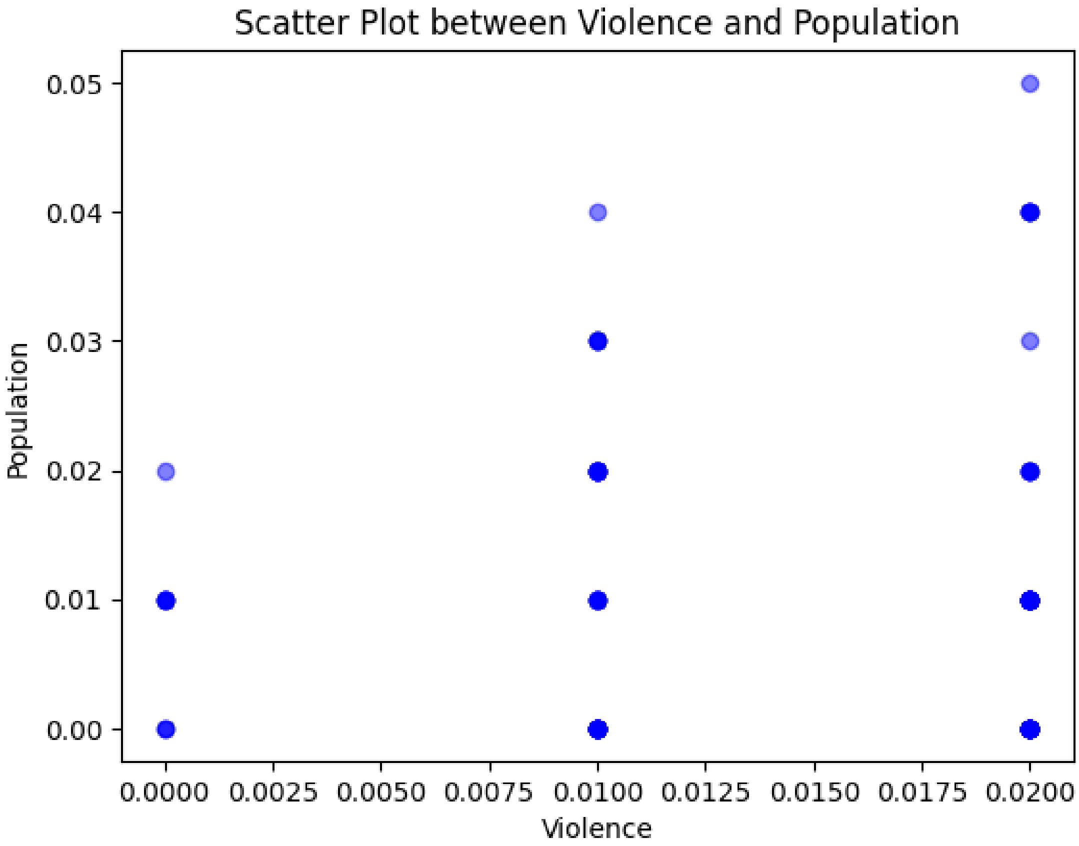


```
7.026390500712107e-54
```

[372]:

```
population = safest_100["population"]
histogram(population, "Population")
scatter(violence, population, "Violence", "Population")
#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, population)
print(pval_no_chage)
```

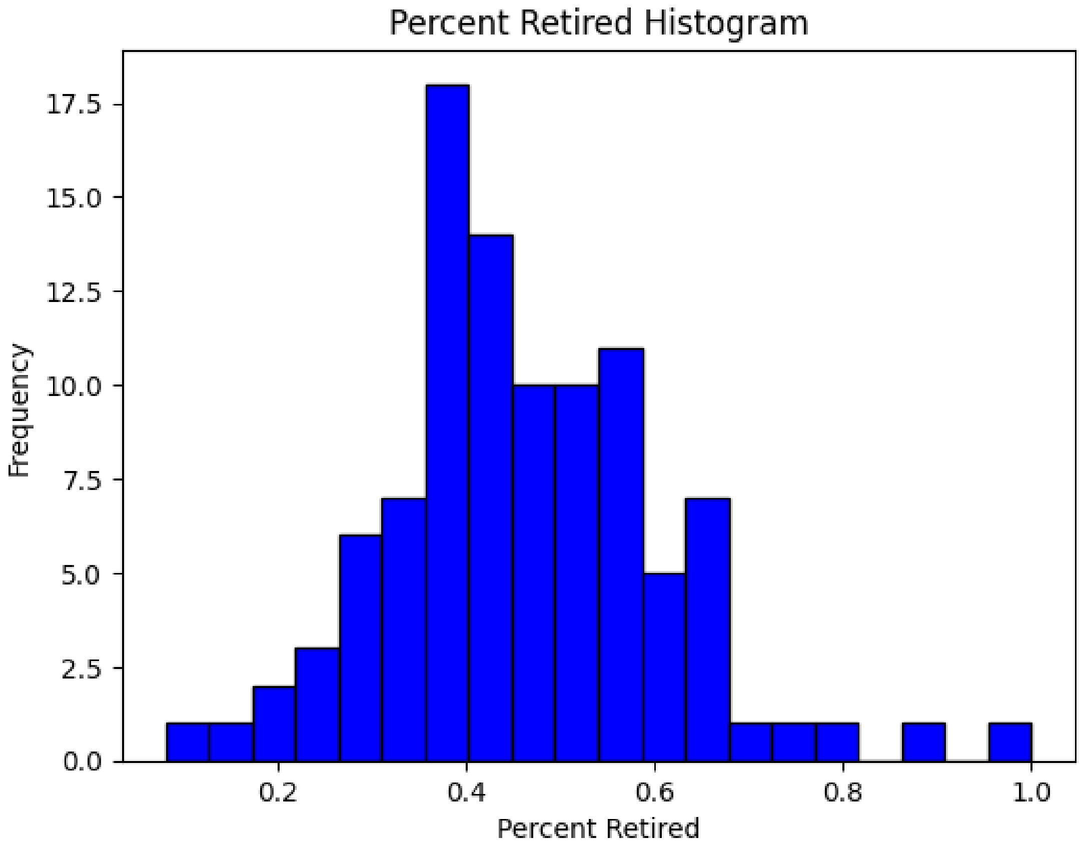

Population Histogram
Frequency
0
5
10
15
20
25
30
35
0.00
0.01
0.02
0.03
0.04
0.05
Population

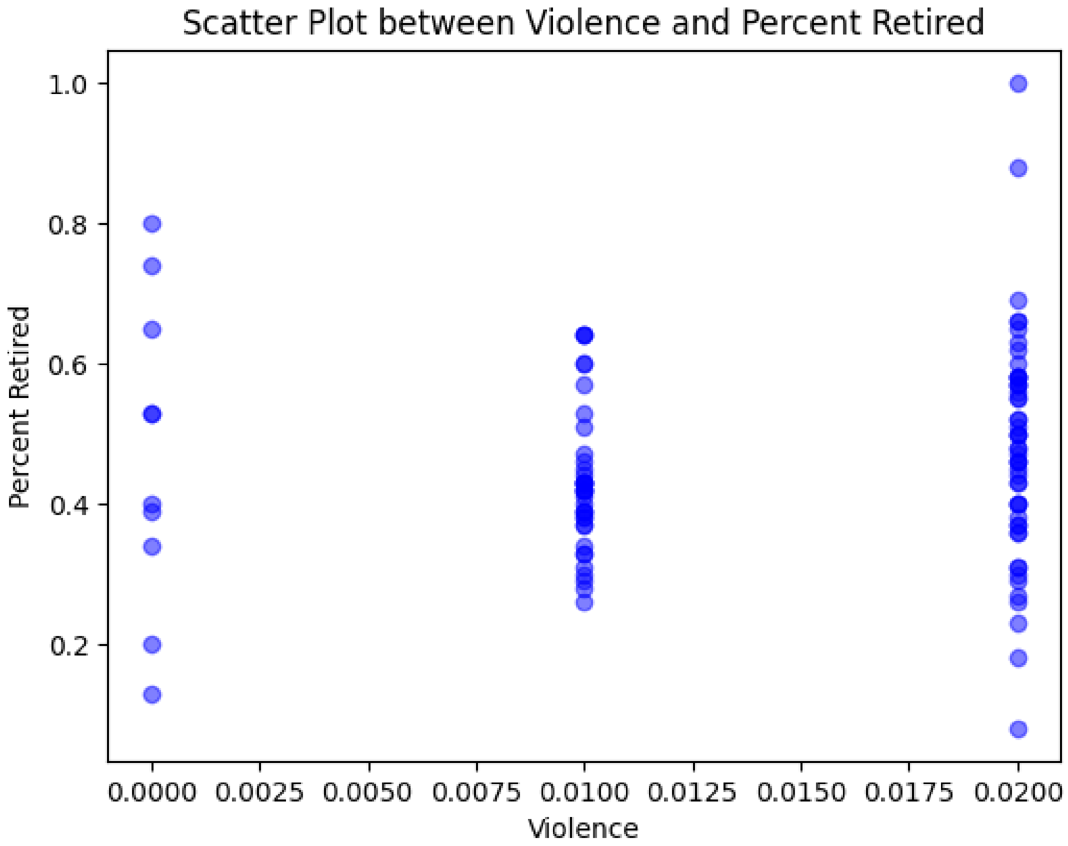


```
0.07899206419562632
```

[374]:

```
pctWRetire = safest_100["pctWRetire"]
histogram(pctWRetire, "Percent Retired")
scatter(violence, pctWRetire, "Violence", "Percent Retired")
#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, pctWRetire)
print(pval_no_chage)
```

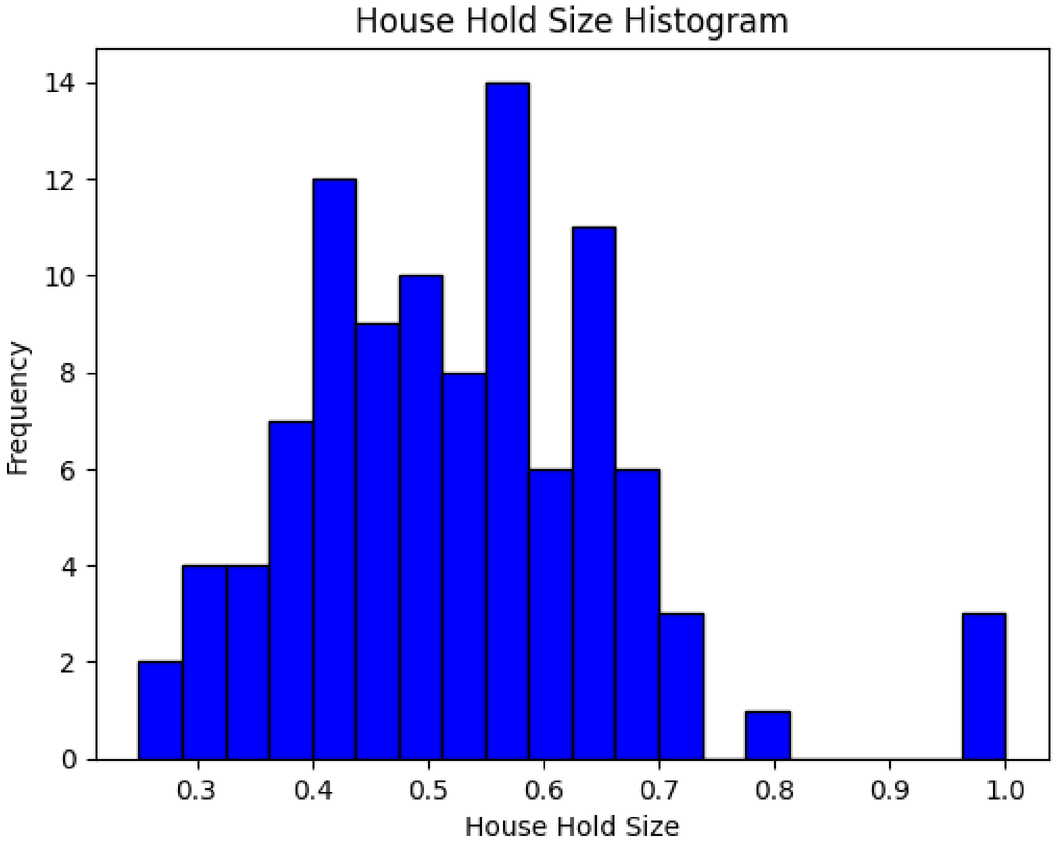

Percent Retired Histogram
Frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0.2
0.4
0.6
0.8
1.0
Percent Retired

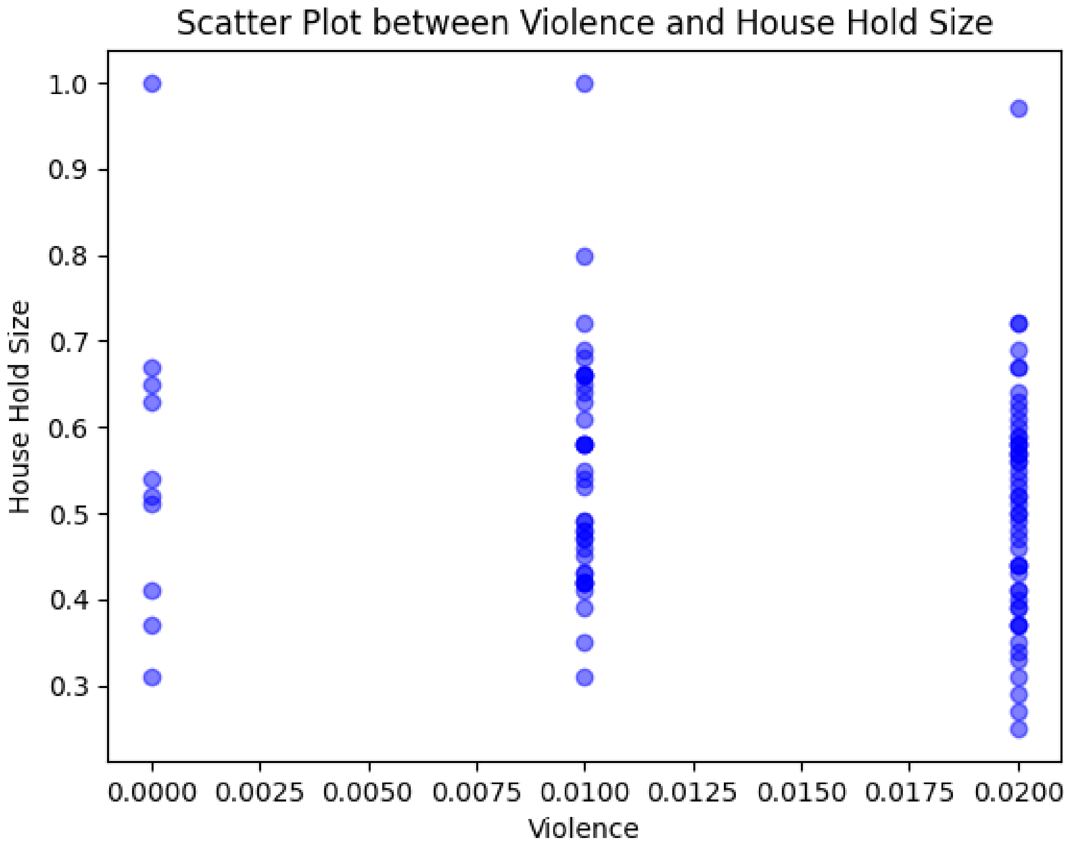


```
4.076304180344116e-75
```

[375]:

```
householdsize = safest_100["householdsize"]
histogram(householdsize, "House Hold Size")
scatter(violence, householdsize, "Violence", "House Hold Size")
#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, householdsize)
print(pval_no_chage)
```

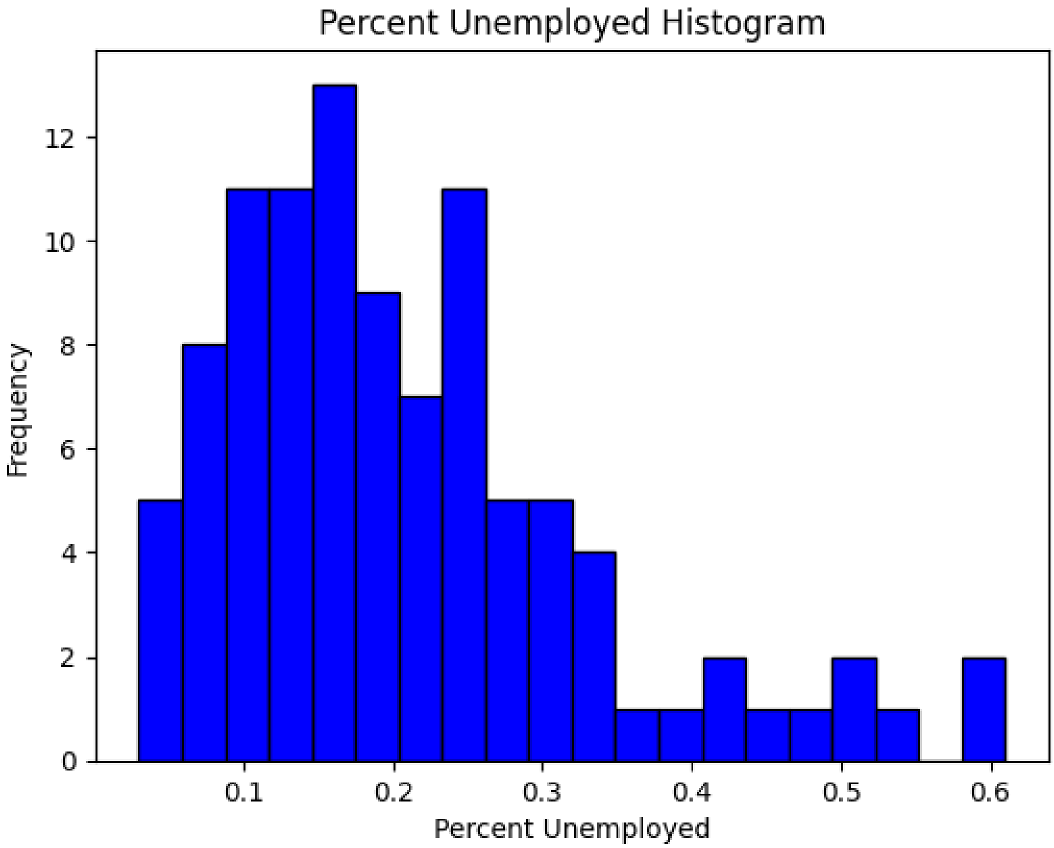

House Hold Size Histogram
Frequency
0
2
4
6
8
10
12
14
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
House Hold Size

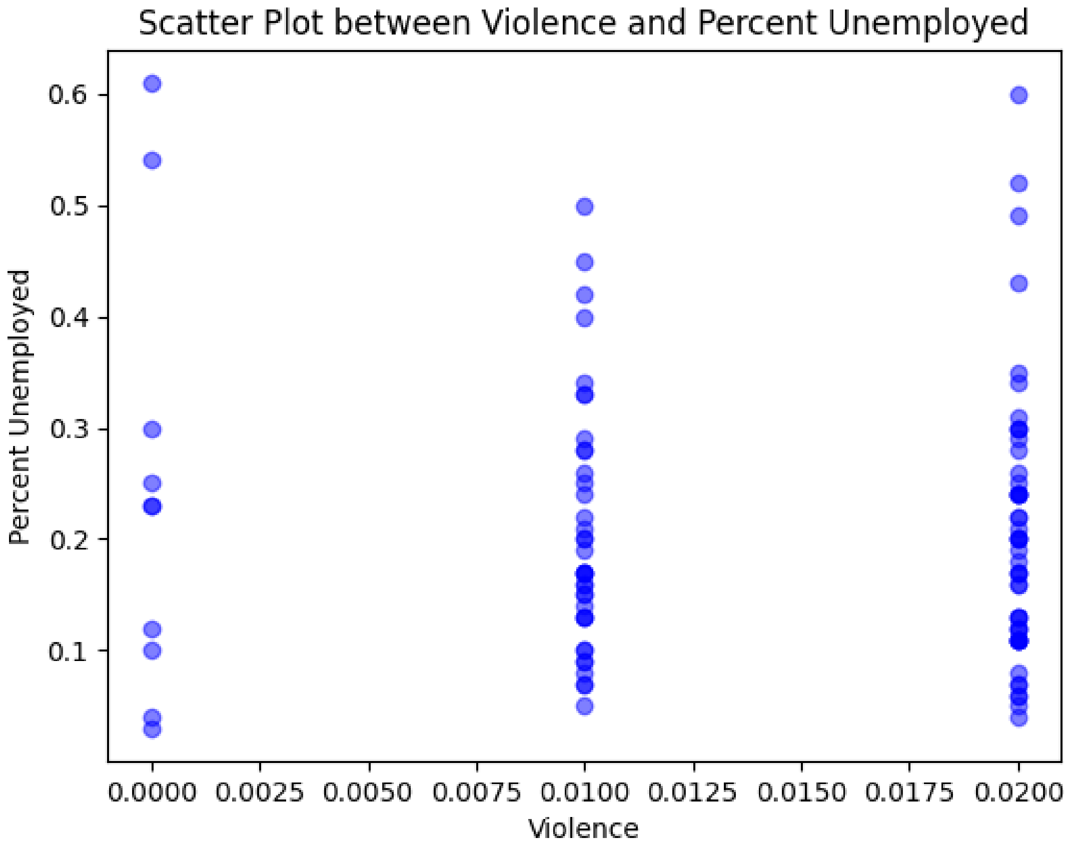


```
1.7212874284677357e-88
```

[376]:

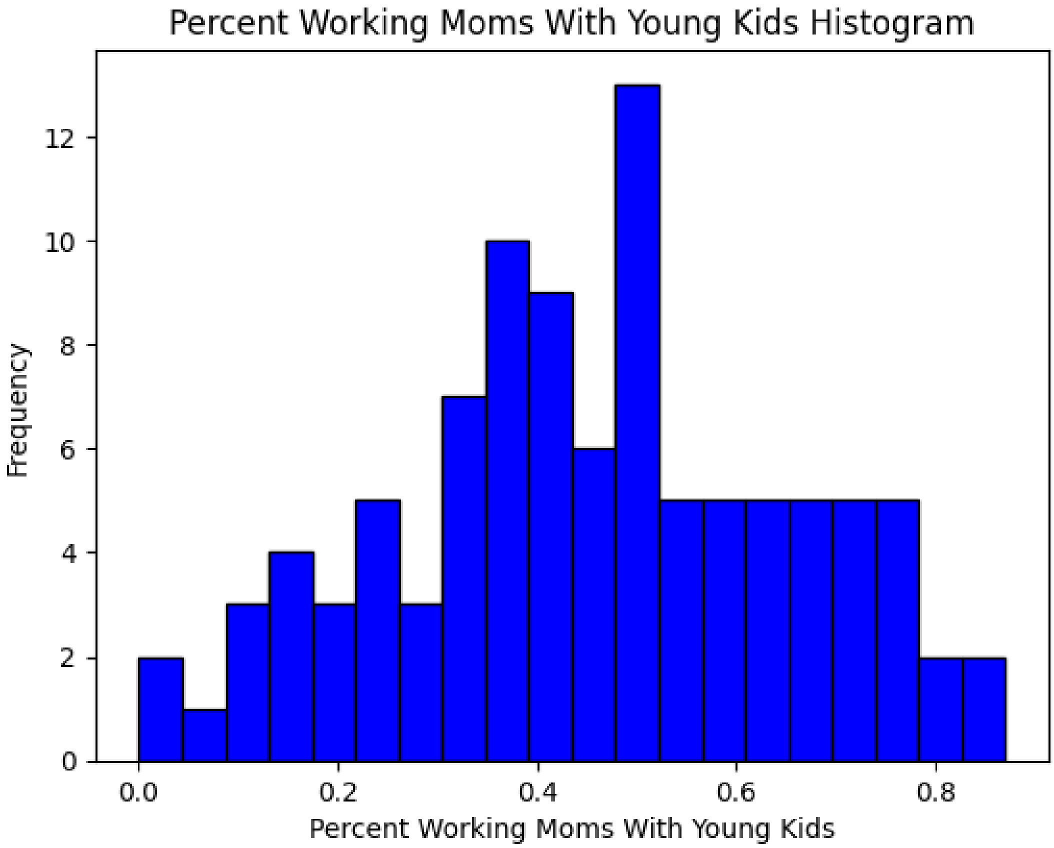
```
PctUnemployed = safest_100["PctUnemployed"]
histogram(PctUnemployed, "Percent Unemployed")
scatter(violence, PctUnemployed, "Violence", "Percent Unemployed")
#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, PctUnemployed)
print(pval_no_chage)
```

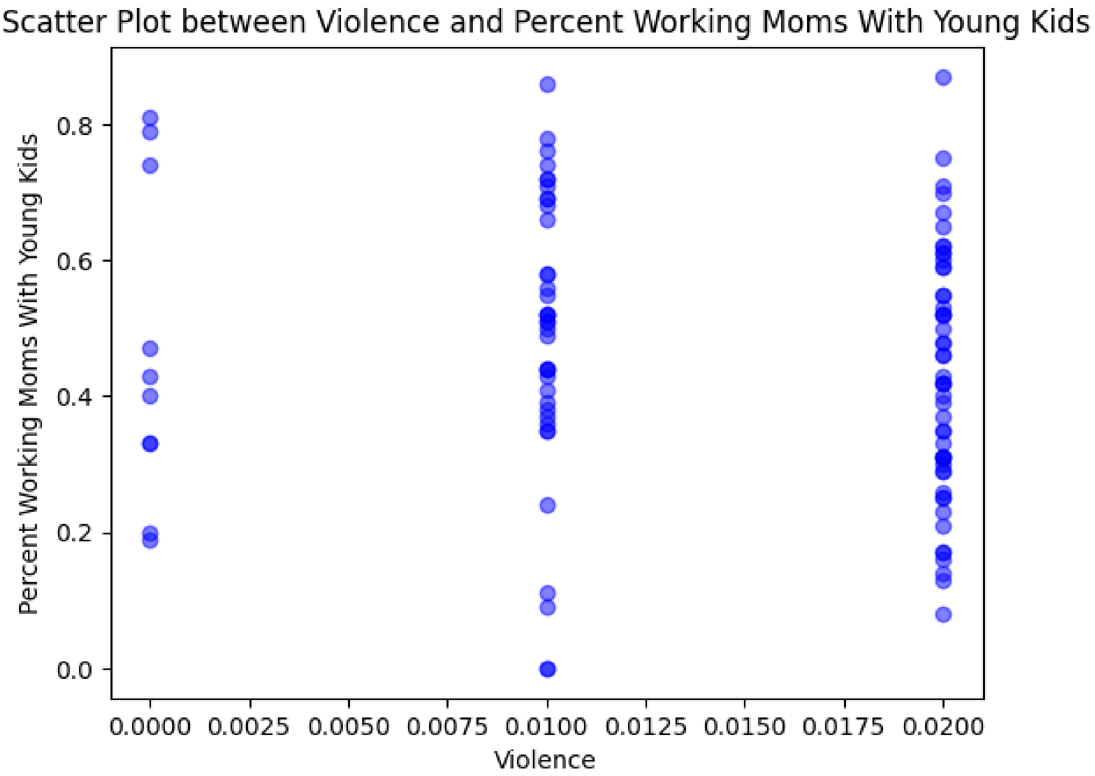

Percent Unemployed Histogram
Frequency
0
2
4
6
8
10
12
0.1
0.2
0.3
0.4
0.5
0.6
Percent Unemployed



```
3.033882402817256e-36
```

[377]:
```python
PctWorkMomYoungKids = safest_100["PctWorkMomYoungKids"]
histogram(PctWorkMomYoungKids, "Percent Working Moms With Young Kids")
scatter(violence, PctWorkMomYoungKids, "Violence", "Percent Working Moms With␣
 ↪Young Kids")

#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, PctWorkMomYoungKids)
print(pval_no_chage)
```

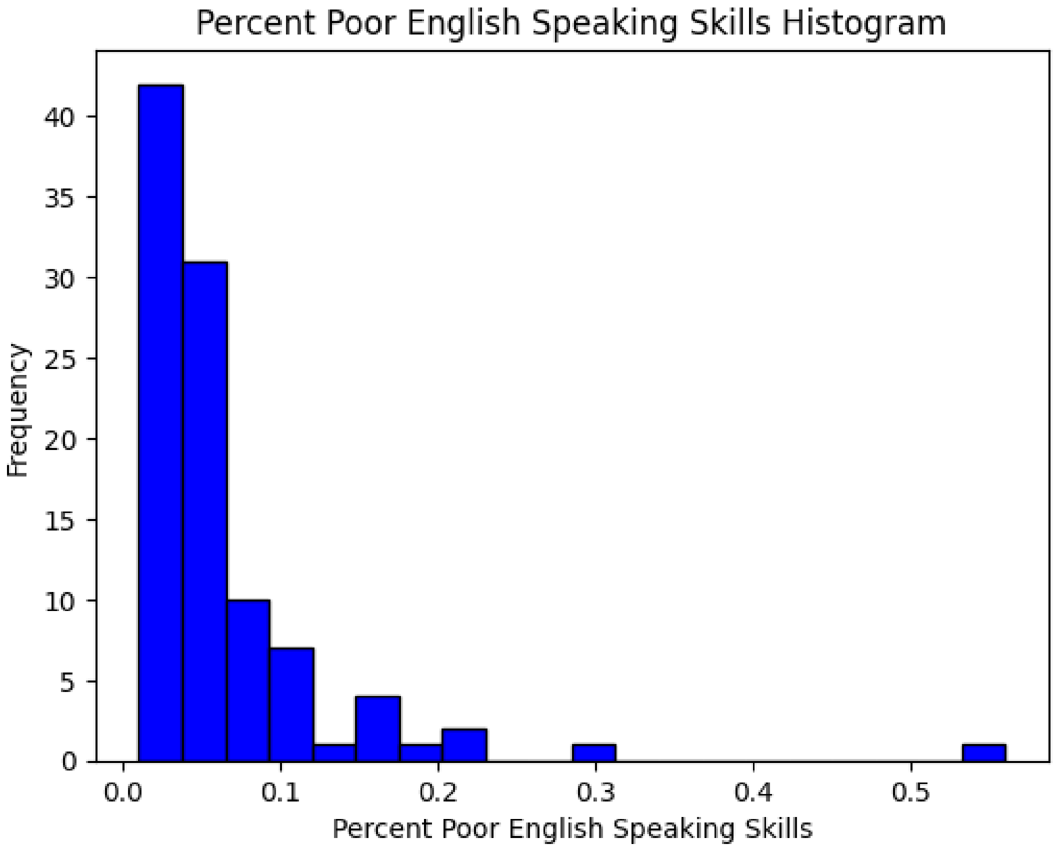

Percent Working Moms With Young Kids Histogram
Frequency
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
Percent Working Moms With Young Kids

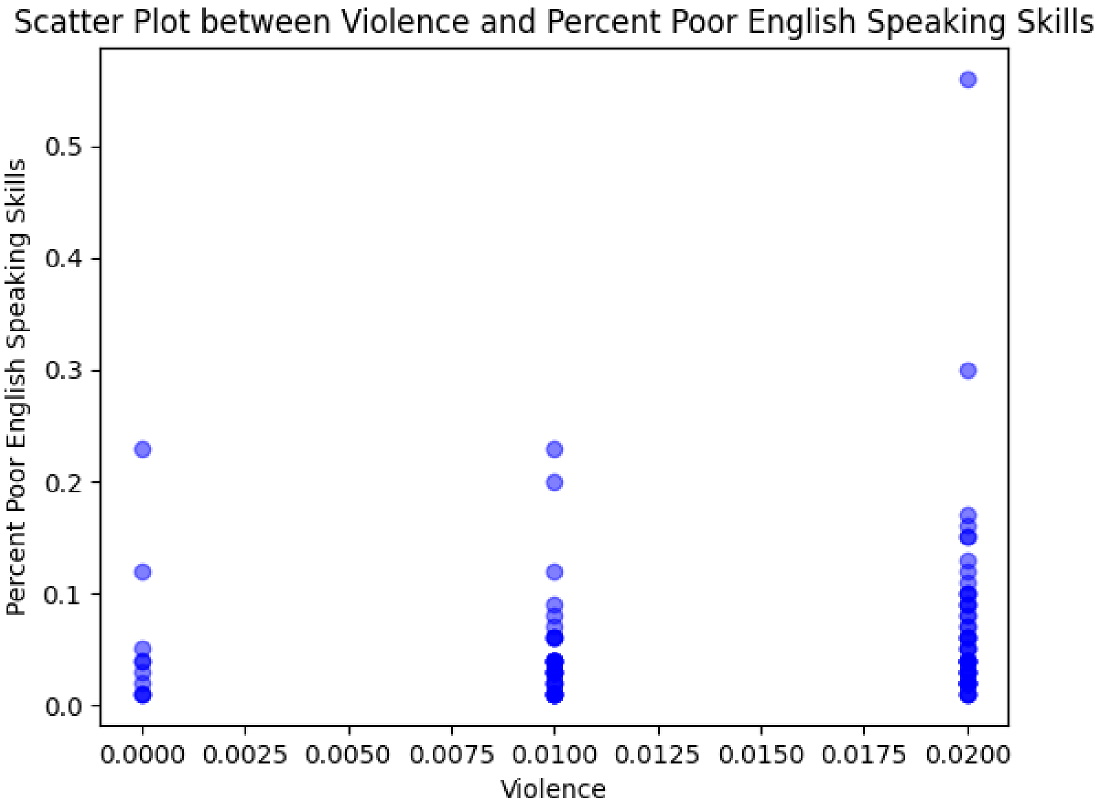


```
9.864651402423239e-56
```

[378]:

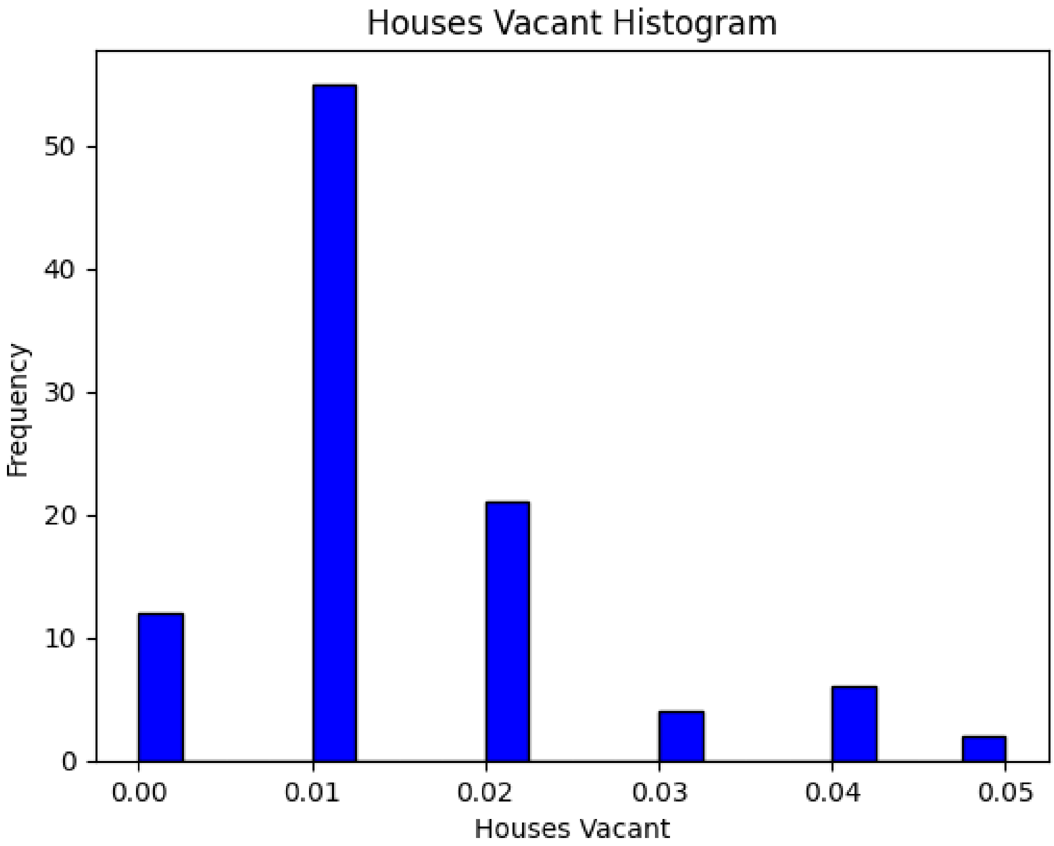
```
PctNotSpeakEnglWell = safest_100["PctNotSpeakEnglWell"]
histogram(PctNotSpeakEnglWell, "Percent Poor English Speaking Skills")
scatter(violence, PctNotSpeakEnglWell, "Violence", "Percent Poor English␣
 ↪Speaking Skills")

#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, PctNotSpeakEnglWell)
print(pval_no_chage)
```

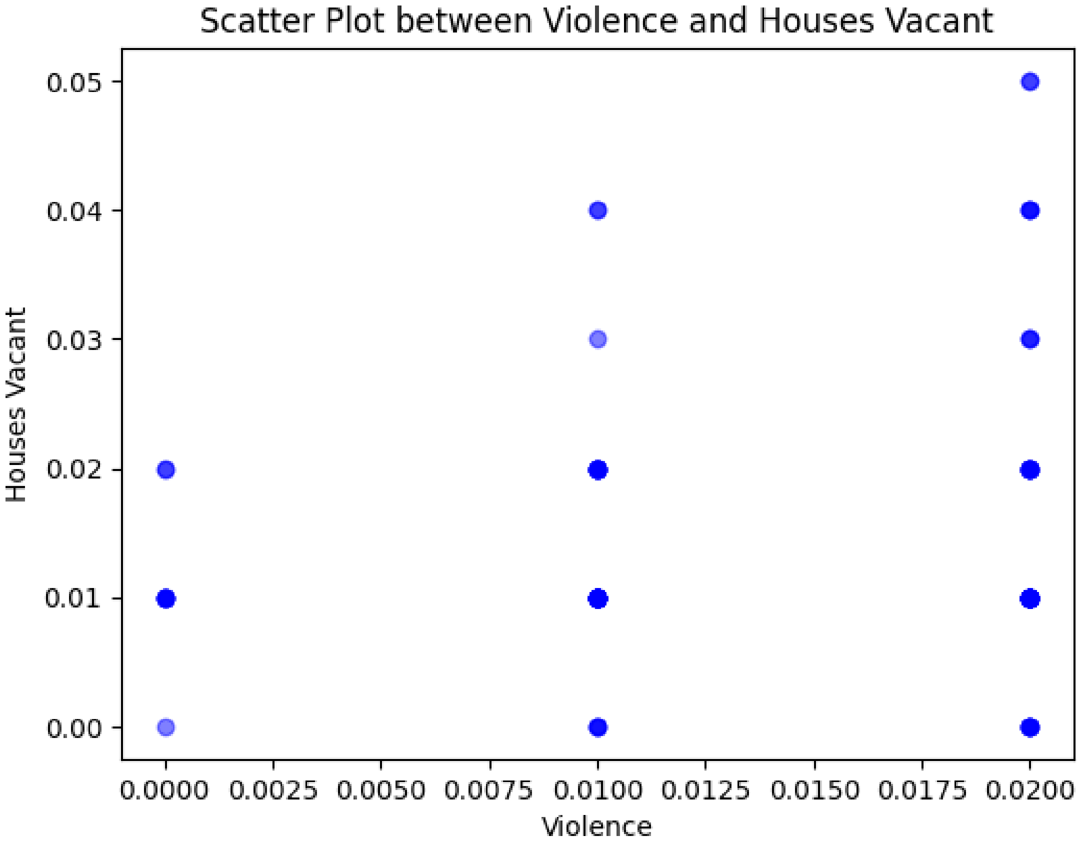

Percent Poor English Speaking Skills Histogram
Frequency
0
5
10
15
20
25
30
35
40
0.0
0.1
0.2
0.3
0.4
0.5
Percent Poor English Speaking Skills



```
1.0445015806239234e-09
```

[379]:
```
HousVacant = safest_100["HousVacant"]
histogram(HousVacant, "Houses Vacant")
scatter(violence, HousVacant, "Violence", "Houses Vacant")

#do not reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, HousVacant)
print(pval_no_chage)
```

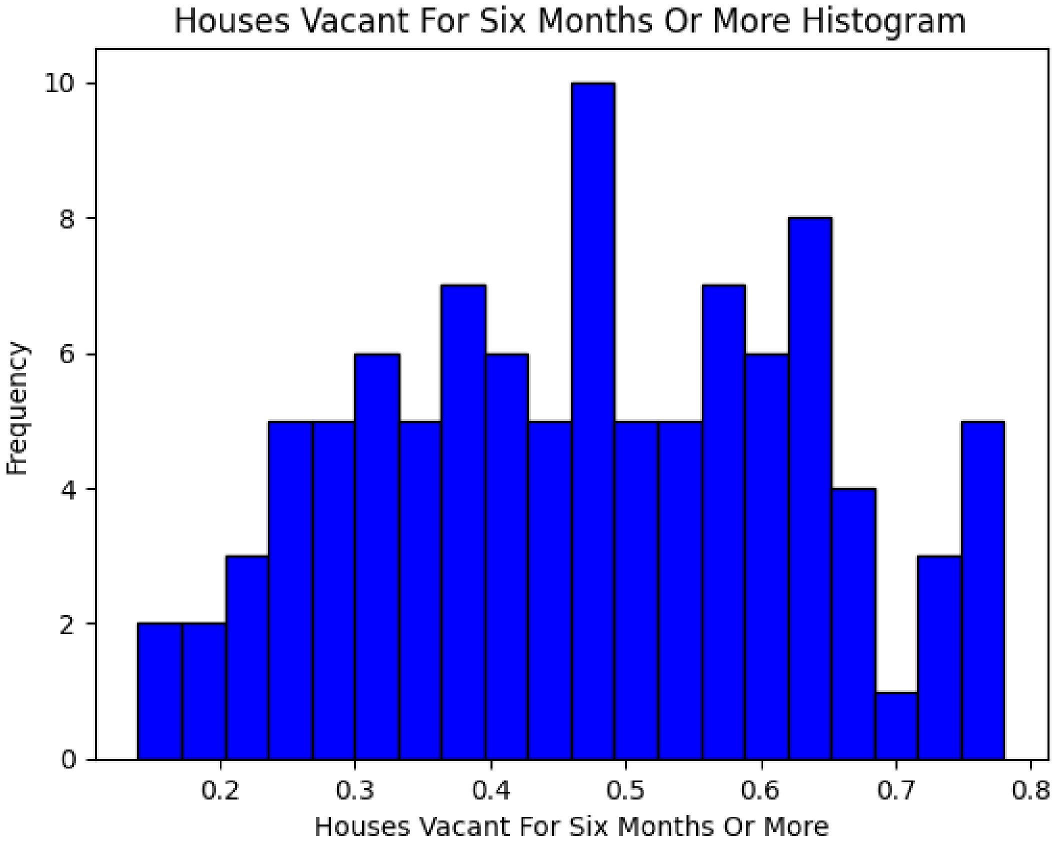

Houses Vacant Histogram
Frequency
0
10
20
30
40
50
0.00
0.01
0.02
0.03
0.04
0.05
Houses Vacant

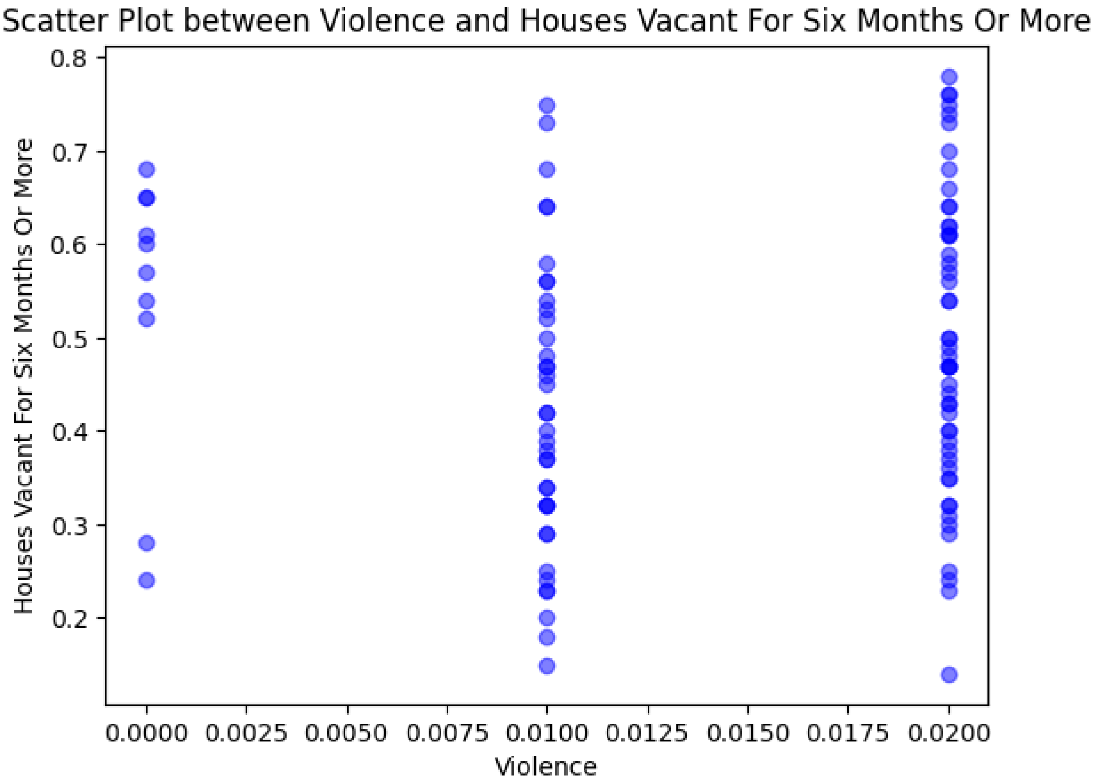


```
0.937551174688849
```

[380]:

```
PctVacMore6Mos = safest_100["PctVacMore6Mos"]
histogram(PctVacMore6Mos, "Houses Vacant For Six Months Or More")
scatter(violence, PctVacMore6Mos, "Violence", "Houses Vacant For Six Months Or␣
↪More")

#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, PctVacMore6Mos)
print(pval_no_chage)
```

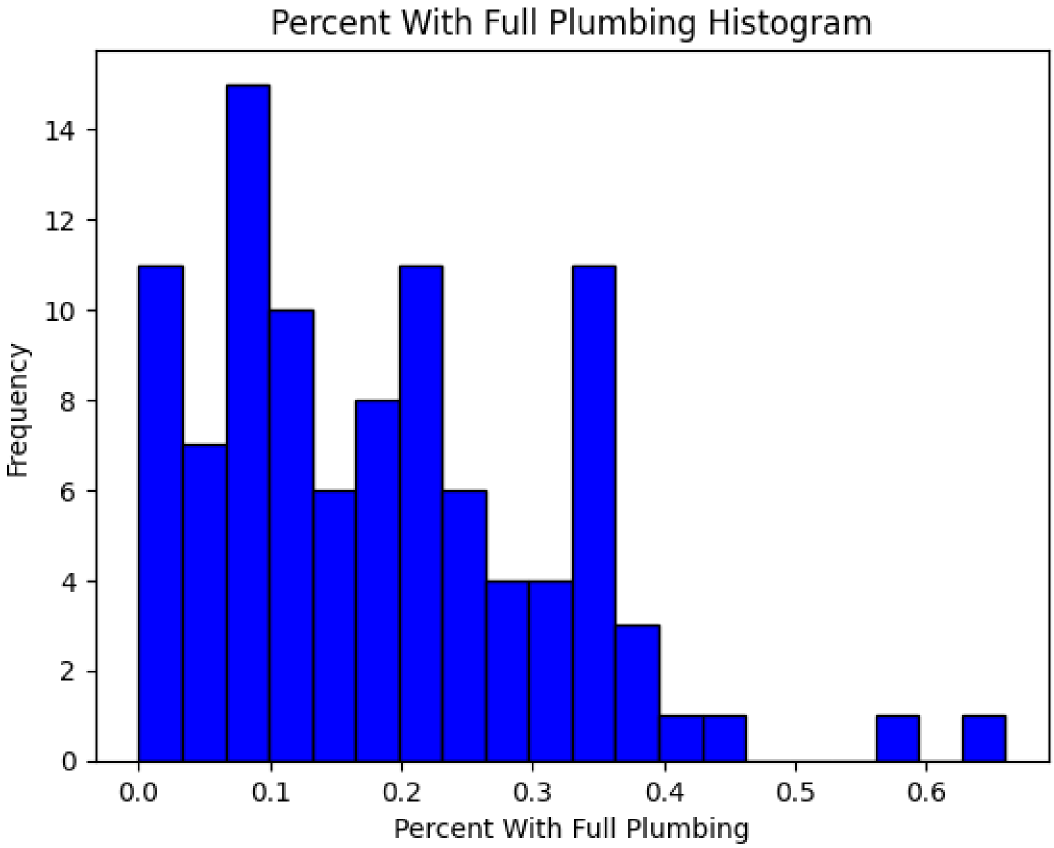

Houses Vacant For Six Months Or More Histogram
Frequency
0
2
4
6
8
10
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Houses Vacant For Six Months Or More

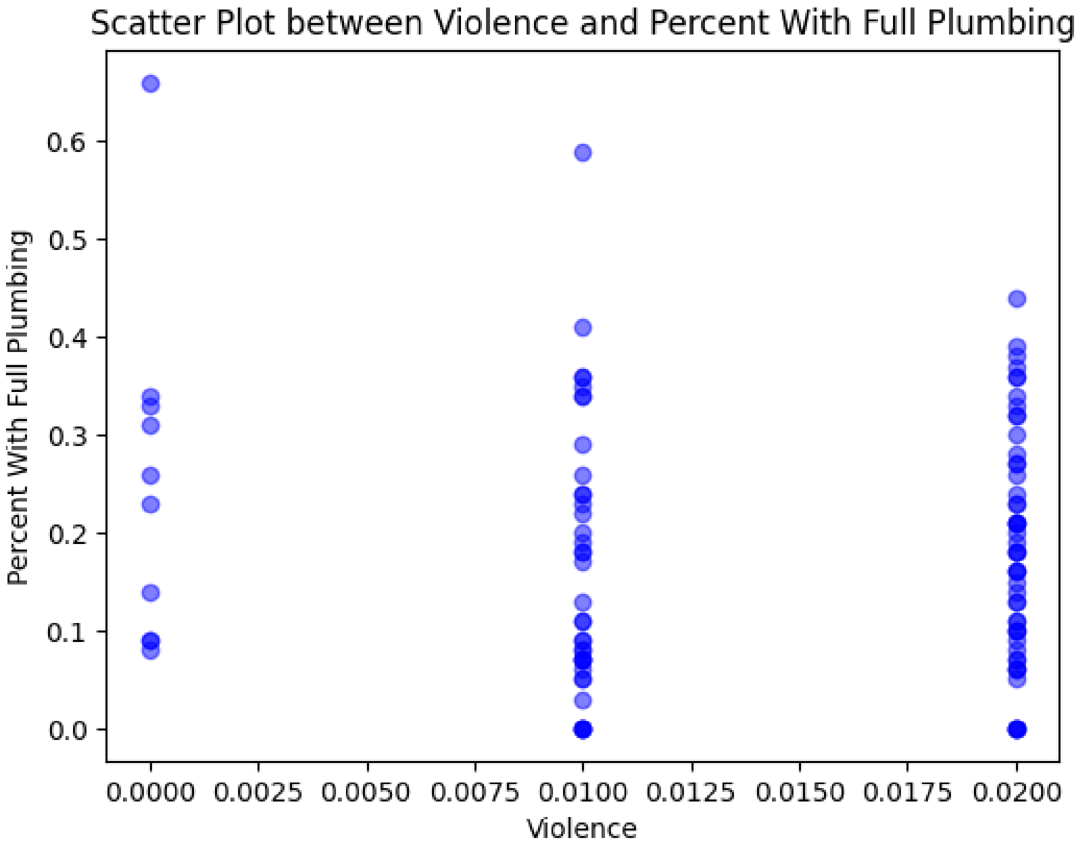


```
7.129755738720952e-72
```

[381]:

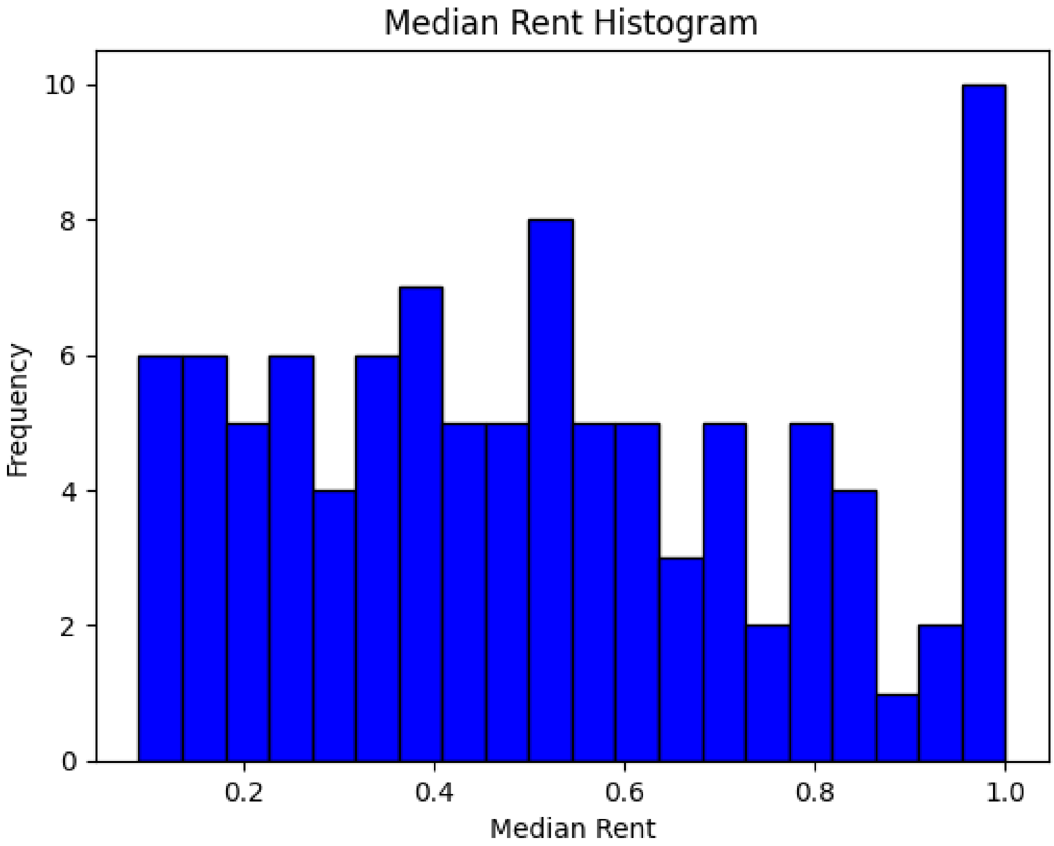
```python
PctWOFullPlumb = safest_100["PctWOFullPlumb"]
histogram(PctWOFullPlumb, "Percent With Full Plumbing")
scatter(violence, PctWOFullPlumb, "Violence", "Percent With Full Plumbing")

#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, PctWOFullPlumb)
print(pval_no_chage)
```

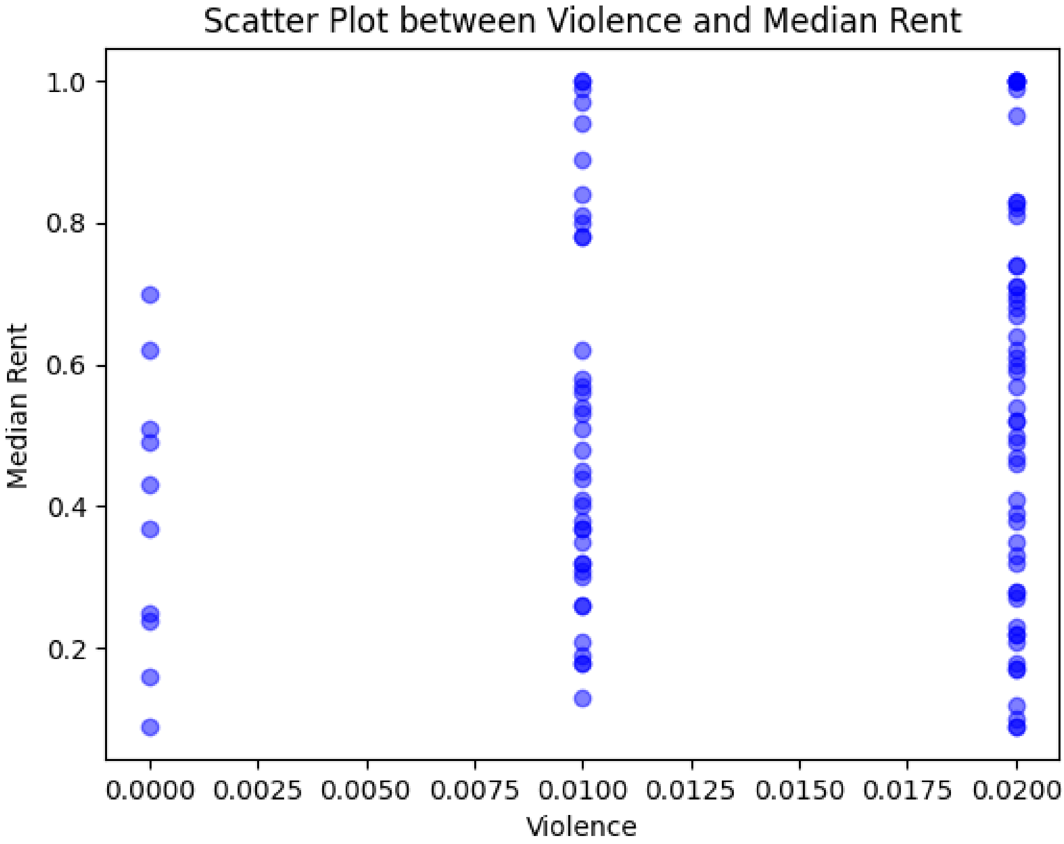

Percent With Full Plumbing Histogram
Frequency
0
2
4
6
8
10
12
14
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Percent With Full Plumbing



```
1.777957110883287e-27
```

```
[382]: MedRent = safest_100["MedRent"]
histogram(MedRent, "Median Rent")
scatter(violence, MedRent, "Violence", "Median Rent")

#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, MedRent)
print(pval_no_chage)
```

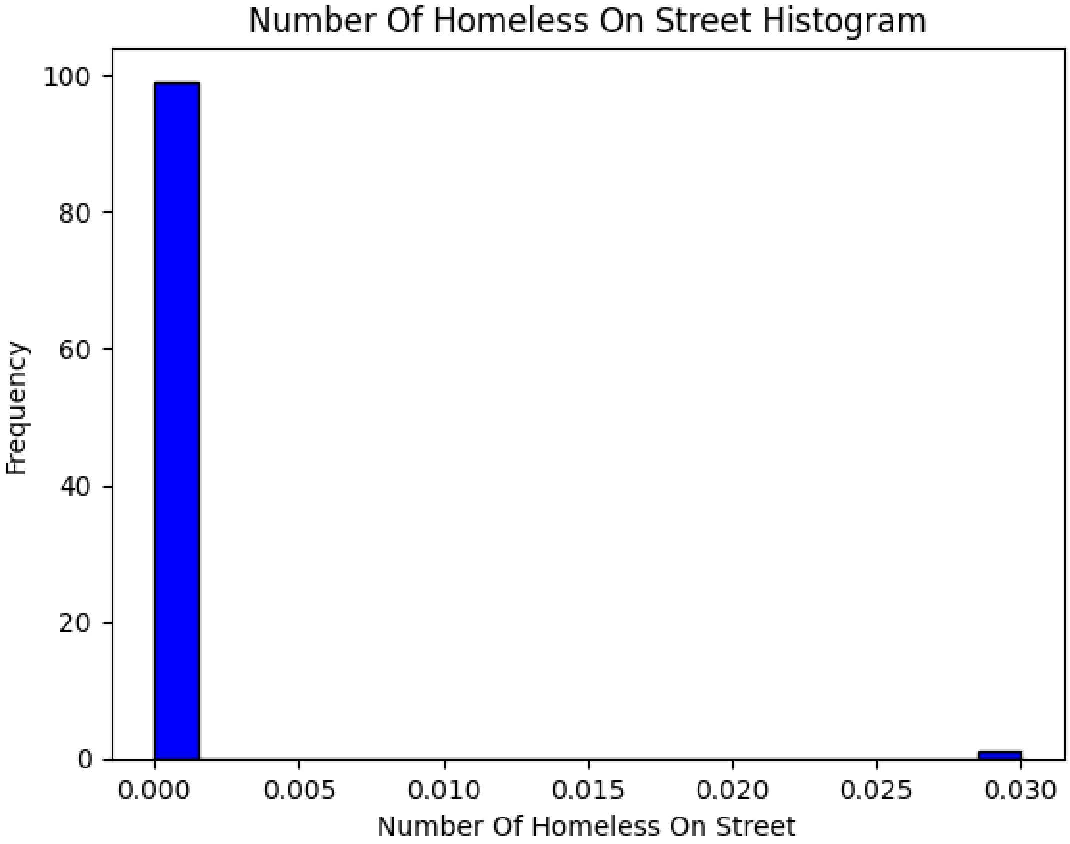

Median Rent Histogram
Frequency
0
2
4
6
8
10
0.2
0.4
0.6
0.8
1.0
Median Rent

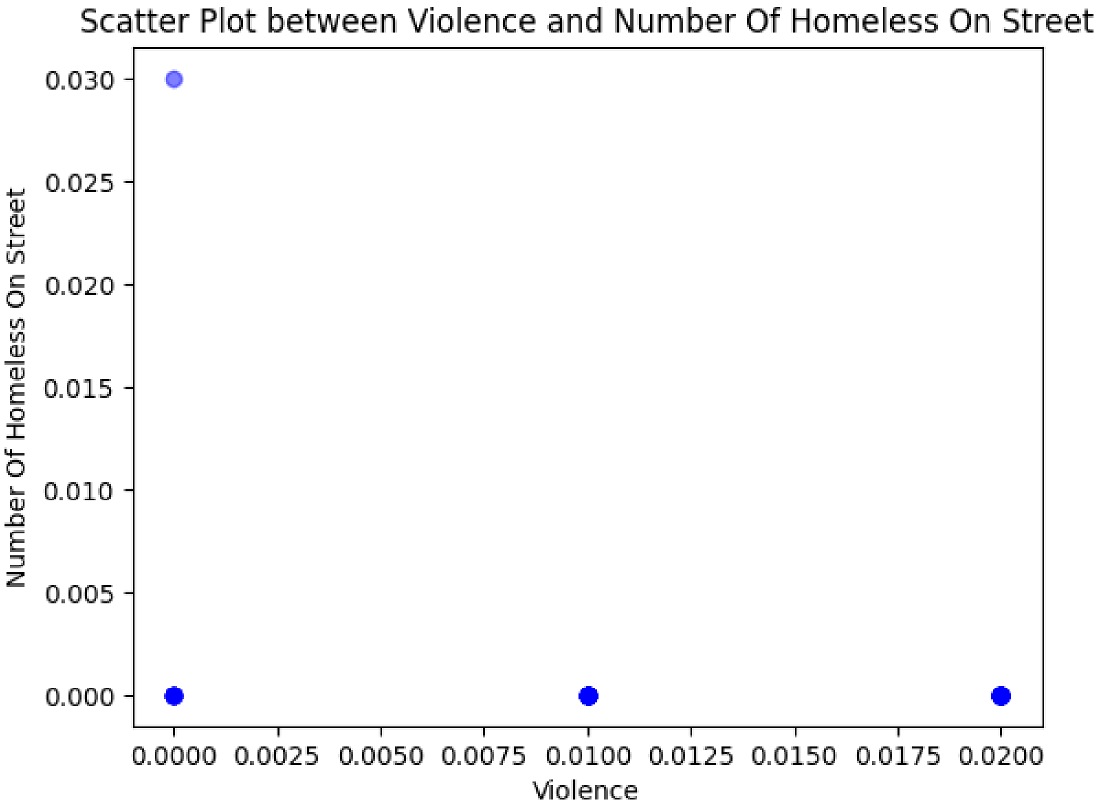


```
1.9518252884091816e-45
```

[383]:

```
NumStreet = safest_100["NumStreet"]
histogram(NumStreet, "Number Of Homeless On Street")
scatter(violence, NumStreet, "Violence", "Number Of Homeless On Street")

#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, NumStreet)
print(pval_no_chage)
```

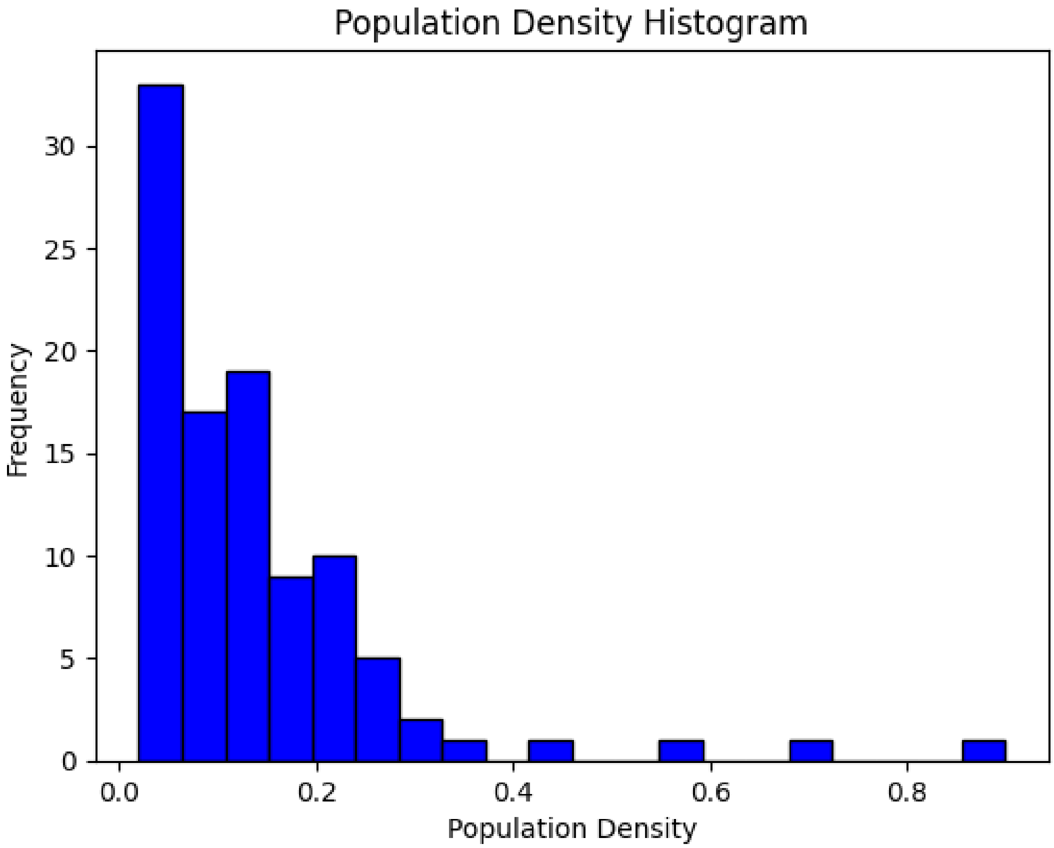

Number Of Homeless On Street Histogram
Frequency
100
80
60
40
20
0
0.000
0.005
0.010
0.015
0.020
0.025
0.030
Number Of Homeless On Street

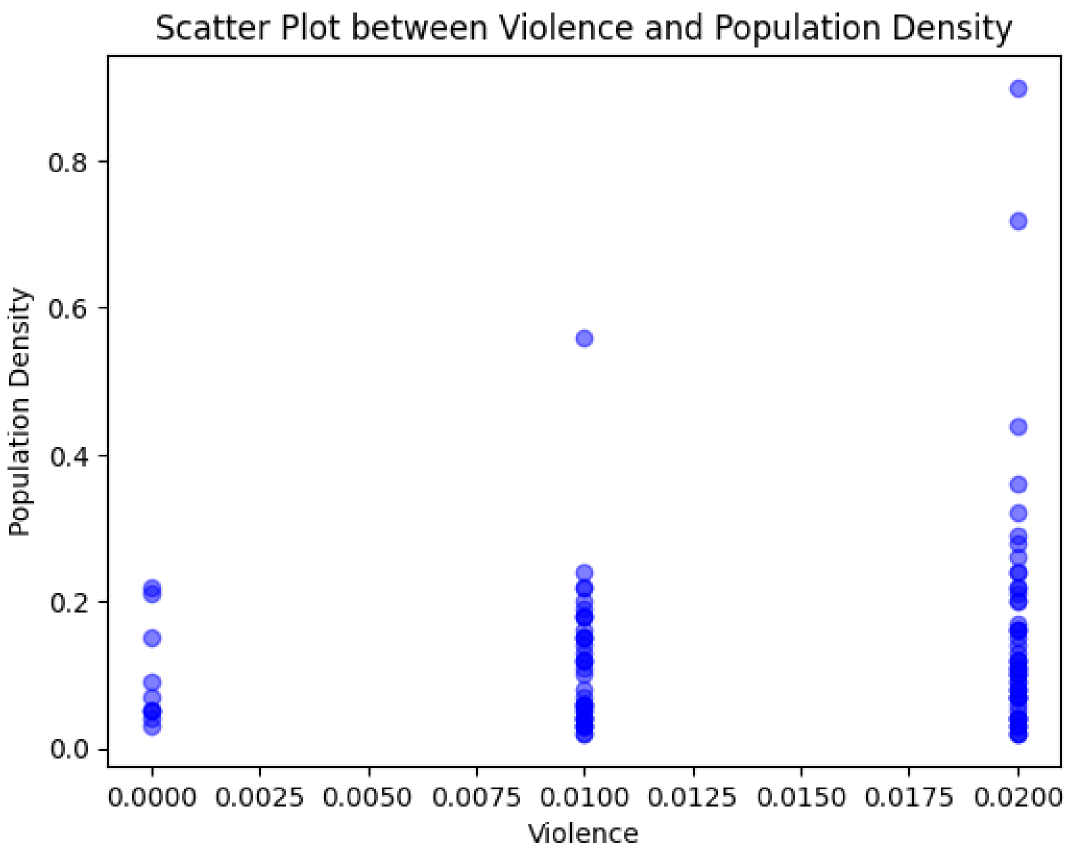


```
2.3541337503634422e-46
```

[384]:

```
PopDens = safest_100["PopDens"]
histogram(PopDens, "Population Density")
scatter(violence, PopDens, "Violence", "Population Density")

#reject null hypothesis
stat, pval_no_chage = stats.ttest_ind(violence, PopDens)
print(pval_no_chage)
```


Population Density Histogram
Frequency
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
Population Density



```
1.4273773237361945e-16
```

[ ]: