CorrTest_household_personalIncome

June 16, 2021

1 Testing correlation between personal and household income

[1]:

```
import pandas as pd
from pandas import read_csv
import numpy as np
from sklearn.linear_model import LinearRegression
from numpy import cov
from scipy.stats import pearsonr
from scipy.stats import spearmanr
import matplotlib.pyplot as plt
import seaborn as sn


pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

[75]:

```
# Group D
groupD=pd.read_csv("CompleteSet_GroupD.csv")
groupD=groupD.drop("Unnamed: 0",axis=1)
#Strip all leading whitespace in Area column
groupD['Area'] = groupD['Area'].apply(lambda x: x.strip())

#Remove total NZ row
groupD = groupD.loc[(groupD['Area'] != "Total - New Zealand by Regional Council/
↪SA2")]
#Remove total regions
groupD = groupD.loc[(groupD['ParentArea'] != "NewZealand")]

#Keep only 2013 and 2018
groupD = groupD.loc[(groupD['Year'] == 2013) | (groupD['Year']==2018)]


groupD['TotInd_TotPeople']=groupD['totStated_TotInd']+groupD['notStated_TotInd']

groupD['perc_less50k_TotInd']=groupD[["less5k_TotInd",␣
↪"bet5k10k_TotInd","bet10k20k_TotInd","bet20k30k_TotInd",'bet30k50k_TotInd']].
↪sum(axis=1)/groupD['TotInd_TotPeople']
```

```
groupD = groupD.drop(['less5k_TotInd', 'less5k_Wholesale', 'less5k_Retail',
                      'less5k_TransPostWare', 'bet5k10k_TotInd',␣
 ↪'bet5k10k_Wholesale',
                      'bet5k10k_Retail', 'bet5k10k_TransPostWare',␣
 ↪'bet10k20k_TotInd',
                      'bet10k20k_Wholesale', 'bet10k20k_Retail',␣
 ↪'bet10k20k_TransPostWare',
                      'bet20k30k_TotInd', 'bet20k30k_Wholesale',␣
 ↪'bet20k30k_Retail',
                      'bet20k30k_TransPostWare', 'bet30k50k_TotInd',␣
 ↪'bet30k50k_Wholesale',
                      'bet30k50k_Retail', 'bet30k50k_TransPostWare',␣
 ↪'bet50k70k_TotInd',
                      'bet50k70k_Wholesale', 'bet50k70k_Retail',␣
 ↪'bet50k70k_TransPostWare',
                      'greater70k_TotInd', 'greater70k_Wholesale',␣
 ↪'greater70k_Retail',
                      'greater70k_TransPostWare', 'totStated_TotInd',␣
 ↪'totStated_Wholesale',
                      'totStated_Retail', 'totStated_TransPostWare',␣
 ↪'notStated_TotInd',
                      'notStated_Wholesale', 'notStated_Retail',␣
 ↪'notStated_TransPostWare'], axis=1)

print(groupD.shape)
```

```
(4506, 5)
```

[20]:

```
# Group G
groupG=pd.read_csv("CompleteSet_GroupG.csv")
groupG=groupG.drop("Unnamed: 0",axis=1)
#Strip all leading whitespace in Area column
groupG['Area'] = groupG['Area'].apply(lambda x: x.strip())


#Remove total NZ row
groupG = groupG.loc[(groupG['Area'] != "Total - New Zealand by Regional Council/
 ↪SA2")]
#Remove total regions
groupG = groupG.loc[(groupG['ParentArea'] != "NewZealand")]


groupG = groupG.drop(['TotHousehold', 'less20k', 'bet20k_30k', 'bet30k_50k',
       'bet50k_70k', 'bet70k_100k', 'bet100k_150k', 'greater150k', 'totStated',
              'totNotStated',], axis=1)

print(groupG.shape)
```

```
(4506, 4)
```

[76]:
```
comboFrame= pd.merge(groupD, groupG, how="outer", on=["Area",␣
 ↪"ParentArea","Year"])
comboFrame=comboFrame.fillna(0)
```

[77]:
```
comboFrame.columns
```

[77]:
```
Index(['Area', 'ParentArea', 'Year', 'TotInd_TotPeople', 'perc_less50k_TotInd',
       'MedInc'],
      dtype='object')
```

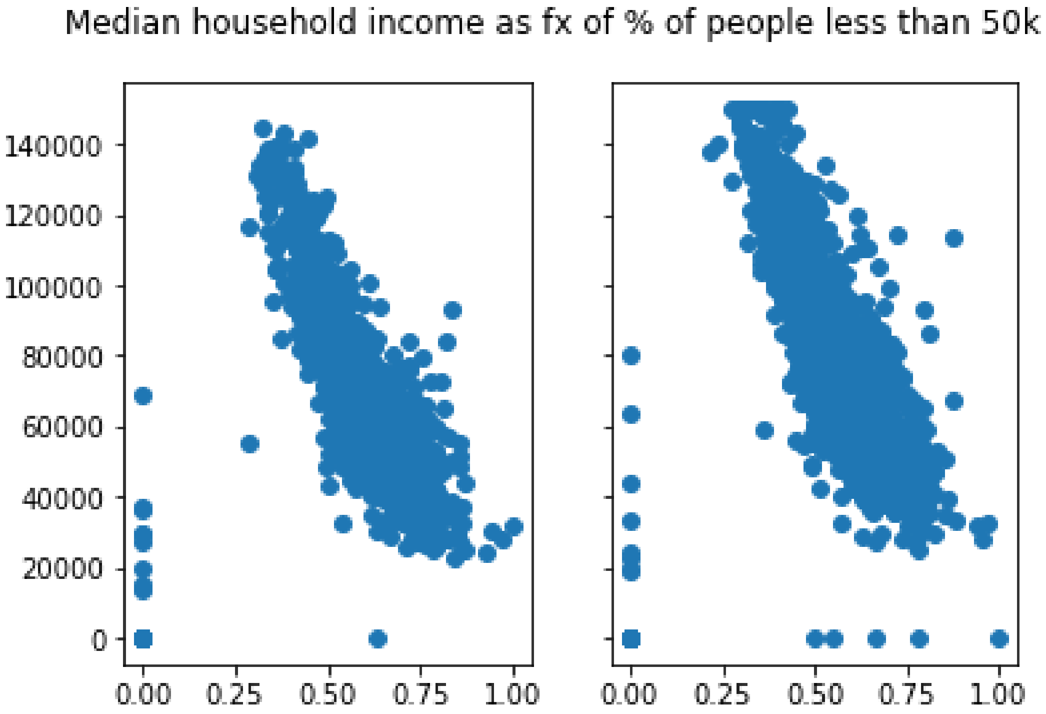
1.0.1 Scatterplot

[78]:
```
medInc_2013 = np.array(comboFrame.loc[(comboFrame['Year'] == 2013)].MedInc.
 ↪tolist())
percLess50k_2013 = np.array(comboFrame.loc[(comboFrame['Year'] == 2013)].
 ↪perc_less50k_TotInd.tolist())


medInc_2018 = np.array(comboFrame.loc[(comboFrame['Year'] == 2018)].MedInc.
 ↪tolist())
percLess50k_2018 = np.array(comboFrame.loc[(comboFrame['Year'] == 2018)].
 ↪perc_less50k_TotInd.tolist())
```

[79]:
```
fig, (ax1, ax2) = plt.subplots(1, 2,sharex=True,sharey=True)
fig.suptitle('Median household income as fx of % of people less than 50k ')
ax1.scatter(percLess50k_2013, medInc_2013)
ax2.scatter(percLess50k_2018, medInc_2018)
```

[79]:
```
<matplotlib.collections.PathCollection at 0x119760520>
```



1.0.2 Correlation tests

[80]:

```
# Covariance

print("2013")
covariance = np.cov([percLess50k_2013], [medInc_2013])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(percLess50k_2013, medInc_2013)
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(percLess50k_2013, medInc_2013)
print('Spearmans correlation: %.3f' % corrS)

print("2018")
covariance = np.cov([percLess50k_2018], [medInc_2018])
print(covariance)

# Pearson's correlation
corrP, _ = pearsonr(percLess50k_2018, medInc_2018)
```

```
print('Pearsons correlation: %.3f' % corrP)

# Spearman's correlation
corrS, _ = spearmanr(percLess50k_2018, medInc_2018)
print('Spearmans correlation: %.3f' % corrS)
```

```
2013
[[2.87525099e-02 1.61383410e+02]
 [1.61383410e+02 6.43667861e+08]]
Pearsons correlation: 0.038
Spearmans correlation: -0.570
2018
[[ 2.67108039e-02 -1.54807622e+02]
 [-1.54807622e+02  9.31815329e+08]]
Pearsons correlation: -0.031
Spearmans correlation: -0.565
```

1.0.3 Linear regression

regressor - # Facilities ; predictor - employee count

[81]:

```
percLess50k_2013 = percLess50k_2013.reshape((-1, 1))
percLess50k_2018 = percLess50k_2018.reshape((-1, 1))
```

[82]:

```
model_2013 = LinearRegression().fit(percLess50k_2013, medInc_2013)
model_2018 = LinearRegression().fit(percLess50k_2018, medInc_2018)
```

[83]:

```
print("2013")
r_sq = model_2013.score(percLess50k_2013, medInc_2013)
print('coefficient of determination:', r_sq)
print('intercept:', model_2013.intercept_)
print('slope:', model_2013.coef_)


print("2018")
r_sq = model_2018.score(percLess50k_2018, medInc_2018)
print('coefficient of determination:', r_sq)
print('intercept:', model_2018.intercept_)
print('slope:', model_2018.coef_)
```

```
2013
coefficient of determination: 0.0014072789383139384
intercept: 59192.080801861244
slope: [5612.84598349]
2018
coefficient of determination: 0.0009628704794407694
```

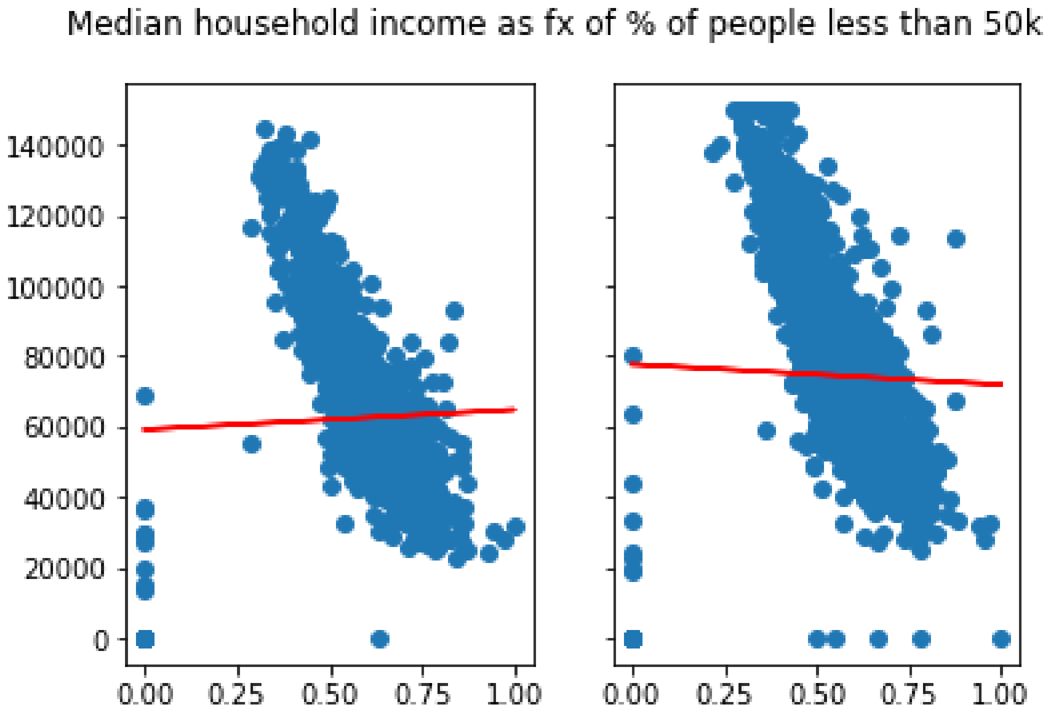
```
intercept: 77761.26683718424
slope: [-5795.69312162]
```

[84]:

```
fig, (ax1, ax2) = plt.subplots(1, 2,sharex=True,sharey=True)
fig.suptitle('Median household income as fx of % of people less than 50k ')
ax1.scatter(percLess50k_2013, medInc_2013)
ax1.plot(percLess50k_2013,model_2013.coef_*percLess50k_2013+model_2013.
 ↪intercept_,'r')
ax2.scatter(percLess50k_2018, medInc_2018)
ax2.plot(percLess50k_2018,model_2018.coef_*percLess50k_2018+model_2018.
 ↪intercept_,'r')
```

[84]:

```
[<matplotlib.lines.Line2D at 0x119b92a90>]
```



[ ]: