```
import csv
import pandas as pd
import geopandas
import numpy as np
from shapely.geometry import Point
import missingno as msn
import seaborn as sns
import matplotlib.pyplot as plt
import math
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import plotly.express as px
import matplotlib.dates as mdates
import plotly.graph_objects as go
from numpy import linalg as LA
import plotly as ply
from sklearn import datasets
from sklearn.decomposition import PCA
from matplotlib import rc
from mpl_toolkits.mplot3d import Axes3D



#Chicago & DC & Philadelphia
df_index =
[2017,2016,2015,2014,2013,2012,2011,2010,2017,2016,2015,2014,2013,20
12,2011,2010,2017,2016,2015,2014,2013,2012,2011,2010]
Total_population =
[6821,7308,7673,7622,7536,7011,7279,6970,3528,3540,3548,3313,3220,30
81,3029,3061,2683,2563,2331,2285,2215,2009,1897,1919]
#TotalP_phi=[1919,1897,2009,2215,2285,2331,2563,2683]
Total_Chinese =
[87.89,85.53,84.23,83.64,82.59,80.00,80.86,81.42,6.32,6.47,6.79,9.78
,6.89,8.43,8.45,11.33,0.53,0.45,0.42,0.55,0.48,0.54,0.71,0.76]
#TotalC_phi=[0.76,0.71,0.54,0.48,0.42,0.45,0.53]
Total_White =[5.56, 5.04, 3.30, 4.70, 6.02, 6.42, 7.93, 10.65,
41.52, 32.03, 32.27, 29.64, 30.09, 26.81, 30.04,
22.54,0.26,0.32,0.4,0.34,0.30,0.35,0.21,0.19]
#TotalW_phi=[0.19,0.21,0.35,0.30,0.34,0.4,0.32,0.26]
Total_Black = [2.26, 3.46, 4.50, 3.96, 3.09, 4.21, 3.37, 3.54, 45.5,
49.18, 48.79, 48.81, 52.24, 54.53, 53.78,
60.63,0.05,0.06,0.06,0.06,0.04,0.02,0.01,0.05] #TotalB_Phi =
[0.05,0.01,0.02,0.04,0.06,0.06,0.06,0.05]
Education_per =
[23,22,21,21,23,22,26,21,55,49.8,48.8,46.6,49.5,41.5,40,37,
36.8,51.8,54.3,49.5,47.6,46.8,31.6,33.3]
Medium_Income =
[26905,31186,27153,28478,28015,26635,34761,32444,67409,59980,63015,7
5125,61875,61417,57943,42244,30746,50455,59891,46375,51585,49928,374
83,43218]
Poverty_per =
[37.3,35.1,37,36,37.2,38,29.7,29.7,18.9,23.2,25.7,22.2,21.8,20.3,13.
5,13.7,37.6,28.4,26.1,19.9,17.5,12.1,8.3,7.2]
```

```
Chi_H_MI = [52497,50434,48522,47831,47270,47408,47371,46877]
T_MI = [26905,31186,27153,28478,28015,26635,34761,32444]
Gross_rent =
[822,778,769,783,705,684,712,644,1211,971,989,960,964,993,894,787,11
22,1192,1263,1150,1038,905,796,802]
City =
['Chicago','Chicago','Chicago','Chicago','Chicago','Chicago','Chicag
o','Chicago','DC','DC','DC','DC','DC','DC','DC','DC','Philadelphia',
'Philadelphia','Philadelphia','Philadelphia','Philadelphia','Philade
lphia','Philadelphia','Philadelphia']

data =
{'df_index':df_index,'Total_Chinese':Total_Chinese,'Total_White':Tot
al_White,'Total_Black':Total_Black,'Education_per':Education_per,
'Medium_Income':Medium_Income,'Poverty_per':Poverty_per,'Gross_rent'
:Gross_rent,'City':City}

data = pd.DataFrame(data)
data.to_csv(r'/Users/shiyipeng/Desktop/project/dt.csv')

features =
['Total_Chinese','Total_White','Total_Black','Education_per','Medium
_Income','Poverty_per','Gross_rent']
X = data.loc[:,features].values
y = data.loc[:,['City']].values

from sklearn.preprocessing import StandardScaler
import random
from sklearn.decomposition import PCA

X = StandardScaler().fit_transform(X)
random.seed(100)

#2-D analysis
pca = PCA(n_components=2)
X_1 = pca.fit_transform(X)
print(pca.explained_variance_ratio_)
pd.DataFrame(abs(pca.components_)).to_csv(r'/Users/shiyipeng/
Desktop/project/Eigen.csv')
print(pd.DataFrame(abs(pca.components_)))
final_Df = pd.DataFrame(data=X_1,columns=['pc1','pc2'])
final_Df=pd.concat([final_Df,data[['City']]],axis=1)

fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('2 component PCA', fontsize = 20)
targets = ['Chicago', 'DC','Philadelphia']
colors = ['r', 'g', 'b']
for target, color in zip(targets,colors):
    indicesToKeep = final_Df['City'] == target
    ax.scatter(final_Df.loc[indicesToKeep, 'pc1']
               , final_Df.loc[indicesToKeep, 'pc2']
```

```
               , c = color
               , s = 50,label=data['df_index'])
ax.legend(targets)
ax.grid()


#3-D analysis

sns.set_style("white")
pca_2 = PCA(n_components=3)
X_2 = pca_2.fit_transform(X)
pd.DataFrame(abs(pca_2.components_)).to_csv(r'/Users/shiyipeng/
Desktop/project/3D_eigen.csv')
result=pd.DataFrame(data=X_2,columns =['pc1','pc2','pc3'])
result = pd.concat([result,data['City']],axis=1)

fig2 = plt.figure()
ax = fig2.add_subplot(111,projection='3d')
colors = ['r','g','b']
targets = ['Chicago', 'DC','Philadelphia']
for city,color in zip(targets,colors):
    indicesToKeep = result['City'] == city

ax.scatter(result.loc[indicesToKeep,'pc1'],result.loc[indicesToKeep,
'pc2'],result.loc[indicesToKeep,'pc3'],c=color,cmap="Set2_r",s=60)
xAxisLine = ((min(result['pc1']), max(result['pc1'])), (0, 0),
(0,0))
ax.plot(xAxisLine[0], xAxisLine[1], xAxisLine[2], 'r')
yAxisLine = ((0, 0), (min(result['pc2']), max(result['pc2'])),
(0,0))
ax.plot(yAxisLine[0], yAxisLine[1], yAxisLine[2], 'r')
zAxisLine = ((0, 0), (0,0), (min(result['pc3']),
max(result['pc3'])))
ax.plot(zAxisLine[0], zAxisLine[1], zAxisLine[2], 'r')

ax.set_xlabel("PC1")
ax.set_ylabel("PC2")
ax.set_zlabel("PC3")
ax.set_title("PCA on the Chinatown dataset")
plt.show()

#Chicago Chinatown MHI vs. Metropolitan MHI

ax1 = plt.subplot(111)
green_bar = [int(i/j*100 )for i,j in zip(T_MI,Chi_H_MI)]
orange_bar = [int((1-i/j)*100) for i,j in zip(T_MI,Chi_H_MI)]
barWidth=0.85
names = [2010,2011,2012,2013,2014,2015,2016,2017]
ax1 =
plt.bar(names,green_bar,color="#b5ffb9",edgecolor='white',width=barW
idth)
ax1 = plt.bar(names,orange_bar,
bottom=green_bar,color="#f9bc86",edgecolor='white',width=barWidth)
plt.xticks(names)
```

```
plt.title('Chicago Chinatown MHI vs. Metro MHI')
plt.show()


#DC chinatown income vs metropolitan income

ax2 = plt.subplot(111)
DC_IN = [58526,61835,64267,65830,69235,70848,72935,77649]
DC_CHI_IN = [42244,57943,61417,61875,75125,63015,59980,67409]
green_bar_D = [int(i/j*100 )for i,j in zip(DC_CHI_IN,DC_IN)]

barWidth=0.85
names = [2010,2011,2012,2013,2014,2015,2016,2017]
ax2 =
plt.bar(names,green_bar_D,color="#b5ffb9",edgecolor='white',width=ba
rWidth)
plt.ylim(0,100)
plt.xticks(names)
plt.title('DC MHI vs. Metropolitan MHI')
plt.show()

#rent comparison

ChicagoCT_rent = [822,778,769,783,705,684,712,644]
rent_CT= ChicagoCT_rent[::-1]
City_rent = [1029,987,965,963,949,935,916,885]
City_rent = City_rent[::-1]
year_rent = [2017,2016,2015,2014,2013,2012,2011,2010]
year_rent = year_rent[::-1]
ax3 = plt.subplot(111)
plt.title('CT rent vs. Metropolitan rent')
ax3.plot(year_rent,rent_CT,label="Chinatown")
ax3.plot(year_rent,City_rent,label="metro")

legend = ax3.legend(loc='upper left', shadow=True, fontsize='large')
plt.show()
```