```
from numpy.lib.function_base import corrcoef
import pandas as pd
import numpy as np
from numpy import mean
from numpy import std
from numpy import cov
import csv
import random
import matplotlib.pyplot as plt
from skimage import io
!pip install geopandas
import geopandas as gpd
from shapely.geometry import point
#import urllib


#Import COVID deaths data from
https://data.cdc.gov/NCHS/Provisional-COVID-19-Deaths-by-Place-of-Death-an
d-/uggs-hy5q
#Direct link to csv file:
https://data.cdc.gov/api/views/uggs-hy5q/rows.csv
df =
pd.read_csv("https://raw.githubusercontent.com/Julia-almeida/Covid/main/cd
c_covid_deaths_9_17_22.csv")
#Shrinking df by excluding unnecessary data:
startDate = df[df["Start Date"] == "01/01/2020"] #all the rows from df
that have 01/01/2020 as starting date
endDate = startDate[startDate["End Date"] == "09/17/2022"]
totalDeaths = endDate[endDate["Place of Death"] == "Total - All Places of
Death"]
#print(totalDeaths)

vax =
pd.read_csv("https://raw.githubusercontent.com/Julia-almeida/Covid/main/CO
VID-19_Vaccinations_in_the_United_States_County_Sept_to_Dec_2022.csv")
number_ppl_1dose = [0]*51

newFile = open("rowsNeeded.csv", 'w') #creating and opening the csv file
for future data entry
```

```
writeToFile = csv.writer(newFile) #using writer() method for future data
entry to newly created file

states = []
deaths = []
#since totalDeaths is a shrunk data frame (which is more complicated data
structure than regular list,
#we have to use more complex for loop with it)
for i,row in totalDeaths.iterrows():
 writeToFile.writerow(row) #writing rows to rowsNeeded.csv, which will be
a 2D array consisting of all these rows as separate lists
 states.append(row["State"])
 deaths.append(int(row["COVID-19 Deaths"]))
 newFile.close()

print("COVID Deaths by State:")
states.pop(0) #removes the first element from the list
deaths.pop(0)
for i in range(len(states)):
 print(states[i]+':', deaths[i])
print()

abbrStates =
['AL','AK','AZ','AR','CA','CO','CT','DE','DC','FL','GA','ID','IL','IN','IA
','KS','KY','LA','ME','MD','MA','MI','MN','MS','MO','MT','NE','NV','NH','N
J','NM','NY','NC','ND','OH','OK','OR','PA','RI','SC','SD','TN','TX','UT','
VT','VA','WA','WV','WI','WY','PR']

print(len(abbrStates))
# filters data to only include date 9/7/2022
vax_9_7_22 = vax[vax["Date"] == "9/7/2022"]

# goal: loop over all the rows of vax_9_7_22 and add
Administered_Dose1_Recip to number_ppl_1dose for the corresponding state
for i,row in vax_9_7_22.iterrows():
 state_abbr = row["Recip_State"]
 if (state_abbr in abbrStates) and (not
np.isnan(row["Administered_Dose1_Recip"])):
   state_idx = abbrStates.index(state_abbr)
   number_ppl_1dose[state_idx] += row["Administered_Dose1_Recip"]
```

```
   #pop_vax_cty[state_idx] += row["??"]


 print(number_ppl_1dose)

x = []
for i in range(1, len(states)+1):
 x.append(i)
#print(x)

colors = []
for i in range(len(states)):
 r = random.random()
 g = random.random()
 b = random.random()
 colors.append((r,g,b))
#print(colors)

#Plot COVID Deaths for each state:
plt.figure(figsize=(11, 5), dpi=100)
plt.bar(x, deaths, width=0.8, color=colors)
plt.xticks(x, abbrStates, rotation='vertical')
plt.xlabel('States')
plt.ylabel('Deaths')
plt.title('COVID-Related Deaths by State')
plt.show()

#The population.csv uses data from
https://data.census.gov/cedsci/table?tid=PEPPOP2021.NST_EST2021_POP&hidePr
eview=false
population_df =
pd.read_csv("https://raw.githubusercontent.com/Julia-almeida/Covid/main/po
pulations.csv")
population = []
for i,row in population_df.iterrows():
 population.append(int(row[1]))

deathPercent = []
#Proportions:
#population = 100
```

```
#deaths     =  ?
#? = deaths * 100 / population
for i in range(len(deaths)):
 deathPercent.append(deaths[i]*100/population[i])
#print(deathPercent)

dose1_percent = []
for i in range(len(number_ppl_1dose)):
   dose1_percent.append(number_ppl_1dose[i]*100/population[i])

print()

#Plot COVID Death Rates for each state:
plt.figure(figsize=(11, 5), dpi=100)
plt.bar(x, deathPercent, width=0.8, color=colors)
plt.xticks(x, abbrStates, rotation='vertical')
plt.xlabel('States')
plt.ylabel('Death Rates (in %)')
plt.title('COVID Death Rates by State')
plt.show()


# Scatter plot
plt.scatter(deathPercent, dose1_percent)
plt.xlabel('Death Rates')
plt.ylabel('Average Number of Vaccines per Person')

# cut out HI
x = x[0:11] + x[12:51]
states = states[0:11]+states[12:51]
abbrStates = abbrStates[0:11] + abbrStates[12:51]
deaths = deaths[0:11] + deaths[12:51]
number_ppl_1dose = number_ppl_1dose[0:11] + number_ppl_1dose[12:51]
population = population[0:11] + population[12:51]

#covariance, standard deviation, and mean
print("correlation")
corr = np.corrcoef(deathPercent, dose1_percent)
print(corr)
covariance = cov(deathPercent, dose1_percent)
```

```
print()
print("covariance")
print(covariance)
print()
print('standard deviation')
print(np.std(deathPercent))
mean = np.average(deathPercent)
print()
print('mean/average')
print(mean)
```