exp 9 Euclidian and manhattan

```
import csv
import math
A=[]
with open('iris.csv',newline='') as csvfile:
  for row in csv.reader(csvfile):
    A.append(row)
  t=[]
  for i in A:
    t.append(i[:-1])
  t.pop(0)
  print(len(t))
  res=[]
  res1=[]
  for i in range(0,len(t)):
    temp=[]
    temp1=[]
    for j in range(0,i+1):
      if i==j:
        temp.append(0)
        temp1.append(0)
      else:
```

```
        sum1=(float(t[i][0])-float(t[j][0]))**2+(float(t[i][1])-float(t[j][1]))**2+(float(t[i][2])-
float(t[j][2]))**2+(float(t[i][3])-float(t[j][3]))**2

        sum2=abs(float(t[i][0])-float(t[j][0]))+abs(float(t[i][1])-float(t[j][1]))+abs(float(t[i][2])-
float(t[j][2]))+abs(float(t[i][3])-float(t[j][3]))

        p=math.sqrt(sum1)

        #p=float("{:.2f}".format(p))

        p=round(p,2)

        temp.append(p)

        p1=round(sum1,2)

        temp1.append(p1)

    res.append(temp)

    res1.append(temp1)

  print("Euclidian matrix")

  for i in res:

    print(i)

  print("Manhattan matrix")

  for j in res1:

    print(j)
```

1 boxplot for each attribute

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read_csv("iris.csv")

#print (data.head(10))
```

```
new_data = data[["sepal.length", "sepal.width", "petal.length", "petal.width"]]

#print(new_data.head())

plt.figure(figsize = (10, 7))

new_data.boxplot()

# below one is not necessary

sns.set(style="ticks", palette="pastel")

f, axes = plt.subplots(2, 2, sharey=False, figsize=(12, 8))

#f, axes = plt.subplots(2, 2, sharey=False, figsize=(12, 8))

sns.boxplot(x="variety", y="petal.length",data=data, ax = axes[0,0])

sns.boxplot(x="variety", y="sepal.length", data=data, ax=axes[0,1])

sns.boxplot(x="variety", y="petal.width",hue = "variety",data=data, ax=axes[1,0])

sns.boxplot(x="variety", y="sepal.width", data=data, ax=axes[1,1])

# adding a title to the plot

f.suptitle("Boxplot of the Petal and Sepal measurements by Iris plant Species")

plt.show()
```

2 data distribution curve for each attribute of iris data set and check whether

#the attributes are unform distributed or positively / negatively skewed

```
import math,os,random

import pandas as pd

import numpy as np

import seaborn as sns

import scipy.stats as stat
```

```
import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D

col=['sepal.length','sepal.width','petal.length','petal.width','variety']

iris=pd.read_csv("iris.csv",names=col)

f, axes = plt.subplots(2, 2, figsize=(7, 7), sharex=True)


sns.distplot( iris["sepal.length"][1:] , color="red", ax=axes[0, 0])

sns.distplot( iris["sepal.width"][1:] , color="pink", ax=axes[0, 1])

sns.distplot( iris["petal.length"][1:] , color="grey", ax=axes[1, 0])

sns.distplot( iris["petal.width"][1:] , color="blue", ax=axes[1, 1])
```

3 mean median and standard deviation of iris data

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

data = pd.read_csv("iris.csv")

#sepallength

sum_data = data["sepal.length"].sum()

mean_data = data["sepal.length"].mean()

median_data = data["sepal.length"].median()

std=data["sepal.length"].std()

print("Sepal Length:\n Sum:",sum_data, "\nMean:", mean_data,
"\nMedian:",median_data,"\nStandardDeviation:",std)

print()

#sepal. width

sum_data = data["sepal.width"].sum()
```

```
mean_data = data["sepal.width"].mean()

median_data = data["sepal.width"].median()

std=data["sepal.width"].std()

print("Sepal Width:\n Sum:",sum_data, "\nMean:", mean_data,
"\nMedian:",median_data,"\nStandardDeviation:",std)

print()

#petal length

sum_data = data["petal.length"].sum()

mean_data = data["petal.length"].mean()

median_data = data["petal.length"].median()

std=data["sepal.width"].std()

print("Petal Length:\n Sum:",sum_data, "\nMean:", mean_data,
"\nMedian:",median_data,"\nStandardDeviation:",std)

print()

#petal width

sum_data = data["petal.width"].sum()

mean_data = data["petal.width"].mean()

median_data = data["petal.width"].median()

std=data["sepal.width"].std()


print("Petal Width:\n Sum:",sum_data, "\nMean:", mean_data,
"\nMedian:",median_data,"\nStandardDeviation:",std)
```

4 and 5 the scatter plot for each pair of attributes of iris data set

correlation coefficient for every pair attribute of iris data set

```
import pandas as pd
```

```
import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import statsmodels.api as sm

data = pd.read_csv("iris.csv")

sns.FacetGrid(data,hue="variety")\

.map(plt.scatter,"sepal.length","sepal.width")\

.add_legend()

plt.show()

sns.FacetGrid(data,hue="variety")\

.map(plt.scatter,"petal.length","petal.width")\

.add_legend()

plt.show()

#correlation coefficients between measurement variables:


data.groupby("variety").corr()


#qqnorm(data["petal.length"])  #q-q plot for petal length

#qqline(data["petal.length"])
```

#6 qq-plot

```
'''import pandas as pd

import numpy as np

import seaborn as sns


import statsmodels.api as sm'''
```

```
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn import datasets
import numpy as np
iris = datasets.load_iris()
i=iris['data']
#data = pd.read_csv("iris.csv")
#data = data.apply(float)
sm.qqplot(i,line="45",fit=True)
plt.show()
```

7 covariance of every pair of attributes of iris data set

```
#covariance matrix
from sklearn import datasets
import numpy as np
iris = datasets.load_iris()
cov_data = np.corrcoef(iris.data.T)
print(cov_data)
```

8 Draw the histogram for every attribute of iris data set consider the width of histogram as 50

#histogram

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

```
data = pd.read_csv("iris.csv")


plt.figure(figsize = (50, 12))
x = data["sepal.length"]


plt.hist(x, bins = 20, color = "blue")
plt.title("Sepal Length ")
plt.xlabel("Sepal_Length")
plt.ylabel("Count")


plt.figure(figsize = (50, 12))
y = data["sepal.width"]
plt.hist(y, bins = 20, color = "green")
plt.title("Sepal Width ")
plt.xlabel("Sepal_Width")
plt.ylabel("Count")


plt.figure(figsize = (50, 12))
z= data["petal.length"]
plt.hist(z, bins = 20, color = "red")
plt.title("Petal Length ")
plt.xlabel("Petal_Length")
plt.ylabel("Count")
```

```
plt.figure(figsize = (50, 12))
q = data["petal.width"]
plt.hist(q, bins = 20, color = "yellow")
plt.title("Petal Width ")
plt.xlabel("Petal_Width")
plt.ylabel("Count")
```

#10. Construct dissimilarity matrix for weather nominal data set

#Dissimilarity matrix for weather

```
import csv
import math
A=[]
with open('weather.nominal.csv',newline='') as csvfile:
  for row in csv.reader(csvfile):
    A.append(row)
  t=[]
  for i in A:
    t.append(i[:-1])
  t.pop(0)
  col=len(t[0])


  res=[]
  for i in range(0,len(t)):
    temp=[]
    for j in range(0,i+1):
```

```
    m=0
    if i==j:
      temp.append(0)
      continue
    for k in range(col):
      if t[i][k]==t[j][k]:
        m=m+1
    h=(col-m)/col
    temp.append(round(h,2))
  res.append(temp)
for i in res:
  print(i)
```

#11. Consider each attribute of iris data set and divide values as equi-depth bins of size 20 each and smooth bins

```
#using bin means and bin boundaries
#binning
import numpy as np
import math
from sklearn.datasets import load_iris
from sklearn import datasets, linear_model, metrics


# load iris data set
dataset = load_iris()
a = dataset.data
```

```
b = np.zeros(150)


# take 1st column among 4 column of data set
for i in range (150):
        b[i]=a[i,1]


b=np.sort(b) #sort the array


# create bins
bin1=np.zeros((30,5))
bin2=np.zeros((30,5))
bin3=np.zeros((30,5))


# Bin mean
for i in range (0,150,5):
        k=int(i/5)
        mean=(b[i] + b[i+1] + b[i+2] + b[i+3] + b[i+4])/5
        for j in range(5):
                bin1[k,j]=mean
print("Bin Mean: \n",bin1)


# Bin boundaries
for i in range (0,150,5):
        k=int(i/5)
        for j in range (5):
```

```
                if (b[i+j]-b[i]) < (b[i+4]-b[i+j]):
                        bin2[k,j]=b[i]
                else:
                        bin2[k,j]=b[i+4]
print("Bin Boundaries: \n",bin2)

# Bin median
for i in range (0,150,5):
        k=int(i/5)
        for j in range (5):
                bin3[k,j]=b[i+2]
print("Bin Median: \n",bin3)
```

#12.Form a binary data set with 10 attributes and 20 records and find the dissimilarity matrix by considering attributes are symmetric and find

#dissimilarity matrix by considering attributes are asymmetric(1 is more important than 0)

#similarity dissimilarity

```
a = [0, 1, 1, 0]
b = [1, 1, 1, 0]
#p=[i for i, j in zip(a, b) if i==0 and j==0]#
#print(p)
def jaccard_similarity(A, B):
   #Find intersection of two sets
```

```
    #nominator = A.intersection(B)
    q=len([i for i, j in zip(a, b) if i==1 and j==1])
    r=len([i for i, j in zip(a, b) if i==1 and j==0])
    s=len([i for i, j in zip(a, b) if i==0 and j==1])
    t=len([i for i, j in zip(a, b) if i==0 and j==0])
    #print(q,r,s,t)
    disim_symmetric=(r+s)/(q+r+s+t)
    sim_asym=(q)/(q+r+s)
    print("disim_symmetric",disim_symmetric)
    print("disim_asym",1-sim_asym)
    #Find union of two sets
    #denominator = A.union(B)
    #Take the ratio of sizes
    #similarity = len(nominator)/len(denominator)
    #return similarity
jaccard_similarity(a, b)
```