import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

cell\_df=pd.read\_csv("C:/Users/MADHURA/Downloads/cell\_samples.csv")

cell\_df.head()

cell\_df.tail()

import matplotlib.pyplot as plt

%matplotlib inline

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

plt.matshow(cell\_df.corr(), cmap="summer")

plt.colorbar()

plt.xticks(list(range(len(cell\_df.columns))),cell\_df.columns,rotation='vertical')

plt.yticks(list(range(len(cell\_df.columns))),cell\_df.columns,rotation='horizontal')

plt.show()

cell\_df.plot.box()

plt.xticks(list(range(len(cell\_df.columns))), cell\_df.columns, rotation='vertical')

plt.plot(cell\_df.MargAdh.values)

plt.hist(cell\_df.MargAdh.values)

axes = pd.plotting.scatter\_matrix(cell\_df, alpha=0.2)

plt.tight\_layout()

plt.show()

pd.plotting.scatter\_matrix(cell\_df, alpha=0.2, figsize=(12,12), diagonal='kde')

plt.show()

cell\_df.plot()

cell\_df.describe()

low = 0.01

high = 0.99

cell\_df.quantile([low,high])

qdf = cell\_df.quantile([low,high])

qdf.MargAdh

qdf.MargAdh[low]

cell\_df.MargAdh.dropna(inplace=True)

cell\_df.describe()

plt.boxplot(cell\_df.MargAdh)

cell\_df.tail()

cell\_df.shape

cell\_df.tail()

cell\_df.shape

cell\_df.size

cell\_df=pd.read\_csv("C:/Users/MADHURA/Downloads/cell\_samples.csv")

cell\_df.tail()

cell\_df.shape

cell\_df.size

cell\_df.count()

cell\_df["Class"].value\_counts()

benign\_df=cell\_df[cell\_df["Class"]==2][0:200]

malignant\_df=cell\_df[cell\_df["Class"]==4][0:200]

benign\_df=cell\_df[cell\_df["Class"]==2][0:200]

malignant\_df=cell\_df[cell\_df["Class"]==4][0:200]

help(benign\_df.plot)

benign\_df=cell\_df[cell\_df["Class"]==2][50:100]

malignant\_df=cell\_df[cell\_df["Class"]==4][50:100]

axes = benign\_df.plot(kind='scatter',x='Clump',y='UnifSize',color='blue',label='malignant')

malignant\_df.plot(kind='scatter',x='Clump',y='UnifSize',color='red',label='Benign',ax=axes)

cell\_df.dtypes

cell\_df=cell\_df[pd.to\_numeric(cell\_df['BareNuc'],errors='coerce').notnull()]

cell\_df['BareNuc']=cell\_df['BareNuc'].astype('int')

cell\_df.dtypes

cell\_df.columns

cell\_df.columns

feature\_df=cell\_df[['Clump','UnifSize','UnifShape','MargAdh','SingEpiSize',

'BareNuc','BlandChrom','NormNucl','Mit']]

x=np.asarray(feature\_df)

y=np.asarray(cell\_df['Class'])

x[0:5]

cell\_df.columns

feature\_df=cell\_df[['Clump','UnifSize','UnifShape','MargAdh','SingEpiSize',

'BareNuc','BlandChrom','NormNucl','Mit']]

X=np.asarray(feature\_df)

y=np.asarray(cell\_df['Class'])

y[0:5]

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=4)

X\_train.shape

y\_train.shape

X\_test.shape

y\_test.shape

from sklearn import svm

classifier=svm.SVC(kernel='linear', gamma='auto',C=2)

classifier.fit(X\_train,y\_train)

y\_predict=classifier.predict(X\_test)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,y\_predict))