# #plots

**#load data using pandas DF**

import pandas as pd

**#loading glass data**

url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/glass/glass.data'

col\_names = ['id','ri','na','mg','al','si','k','ca','ba','fe','glass\_type']

glass = pd.read\_csv(url, names=col\_names, index\_col='id')

glass.sort('al', inplace=True)

glass.head()

**#loading Drinks data**

drink\_cols = ['country', 'beer', 'spirit', 'wine', 'liters', 'continent']

url = 'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/drinks.csv'

drinks = pd.read\_csv(url, header=0, names=drink\_cols, na\_filter=False)

**#loading ufo data**

url = 'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/ufo.csv'

ufo = pd.read\_csv(url)

ufo['Time'] = pd.to\_datetime(ufo.Time) 🡨 changing to python datetime

ufo['Year'] = ufo.Time.dt.year <- Extracting Year from Time column

**#loading beer data set**

import pandas as pd

url = 'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/beer.txt'

beer = pd.read\_csv(url, sep=' ')

beer

**#plotting data in 2D using seabros with MATPLOTLIB**

import seaborn as sns

import matplotlib.pyplot as plt

sns.set(font\_scale=1.5)

sns.lmplot(x='al', y='ri', data=glass, ci=None)

**#to visualize via heat map**

# Download the file and save it in python working directory <https://raw.githubusercontent.com/TVMKISHORE/Analytics/master/data/yelp.json>

#we can load directly with the Url, but it dint work for some reason.

with open(‘yelp.json.txt’, 'rU') as k:

data = [json.loads(row) for row in k]

yelp=pd.DataFrame(data)

# add DataFrame columns for cool, useful, and funny

yelp['cool'] = [row['votes']['cool'] for row in data]

yelp['useful'] = [row['votes']['useful'] for row in data]

yelp['funny'] = [row['votes']['funny'] for row in data]

sns.heatmap(yelp.corr())

**# multiple scatter plots**

sns.pairplot(yelp, x\_vars=['cool', 'useful', 'funny'], y\_vars='stars', size=6, aspect=0.7, kind='reg')

**#plotting data in 2D 🡸using pandas with MATPLOTLIB**

glass.plot(kind='scatter', x='al', y='ri')

glass.plot(kind='scatter', x='al', y='ri',alpha=0.3) -- Adding transparency

pd.scatter\_matrix(drinks[['beer', 'spirit', 'wine']]) – scatter matrix

pd.scatter\_matrix(drinks[['beer', 'spirit', 'wine']], figsize=(10, 8))—SM with Figure size

drinks.continent.value\_counts().plot(kind='bar') -- Plotting categorical variables

ufo.Year.value\_counts().sort\_index().plot()-- Plotting categorical variables

drinks.groupby('continent').mean().plot(kind='bar') – plot for numeric drinks.groupby('continent').mean().drop('liters', axis=1).plot(kind='bar')—-Drop 1 col

drinks.groupby('continent').mean().drop('liters', axis=1).plot(kind='bar', stacked=True)—Stacked plot

drinks.spirit.plot(kind='box') 🡸 boxplot

drinks.beer.plot(kind='density', xlim=(0, 500)) <- Density plot

**# HISTOGRAM plot**

import pandas as pd

import matplotlib.pyplot as plt

# increase default figure and font sizes for easier viewing

plt.rcParams['figure.figsize'] = (8, 6)

plt.rcParams['font.size'] = 14

# add title and labels

drinks.beer.plot(kind='hist', bins=20, title='Histogram of Beer Servings')

plt.xlabel('Beer Servings')

plt.ylabel('Frequency')

# histogram of beer servings grouped by continent

drinks.hist(column='beer', by='continent')

drinks.hist(column='beer', by='continent', sharex=True, sharey=True)<- Share axis options

**# visualization using pure Matplotlib**

import pandas as pd

import matplotlib.pyplot as plt

**# SCATTER plot using Matplotlib**

plt.scatter(glass.al, glass.ri)

plt.xlabel('al')

plt.ylabel('ri')

**#SCATTER PLOT WITH COLORS 🡨 used to plot clusters**

import pandas as pd

import matplotlib.pyplot as plt

d = {'hipsize' :pd.Series([32,31,42,45,60,59]),

'Sholder' :pd.Series([41,40,46,45,46,47]),

'cluster' :pd.Series([1,1,2,2,3,3])}

df = pd.DataFrame(d)

colors = np.array(['red', 'green', 'blue', 'yellow'])

plt.scatter(df.hipsize,df.Sholder,c=colors[df.cluster], s=50)

**#SIMPLE plot for DENSITY and PREDECTION line.**

**# fit a linear regression model**

from sklearn.linear\_model import LinearRegression

linreg = LinearRegression()

feature\_cols = ['al']

X = glass[feature\_cols]

y = glass.ri

linreg.fit(X, y)

**# make predictions for all values of X**

glass['ri\_pred'] = linreg.predict(X)

glass.head()

plt.plot(glass.al, glass.ri\_pred, color='red')

plt.xlabel('al')

plt.ylabel('Predicted ri')

**#Surface3d using Mapplotlib 🡨 Can be used to plot error value VS thetas**

from mpl\_toolkits.mplot3d import Axes3D

from matplotlib import cm

from matplotlib.ticker import LinearLocator, FormatStrFormatter

import matplotlib.pyplot as plt

import numpy as np

fig = plt.figure()

ax = fig.gca(projection='3d')

X = np.arange(-5, 5, 0.25)

Y = np.arange(-5, 5, 0.25)

X, Y = np.meshgrid(X, Y)

R = np.sqrt(X\*\*2 + Y\*\*2)

Z = np.sin(R) **<- Replace with error or function, X, Y being theta values**

surf = ax.plot\_surface(X, Y, Z, rstride=1, cstride=1, cmap=cm.coolwarm,

linewidth=0, antialiased=False)

ax.set\_zlim(-1.01, 1.01)

ax.zaxis.set\_major\_locator(LinearLocator(10))

ax.zaxis.set\_major\_formatter(FormatStrFormatter('%.02f'))

fig.colorbar(surf, shrink=0.5, aspect=5)

plt.show()

**#Other plots can be referred below**

<http://matplotlib.org/users/screenshots.html>