Assignment 10

**Scipy:**

Q1: We have the min and max temperatures in a city In India for each months of the year.

We would like to find a function to describe this and show it graphically, the dataset

given below.

Task:

1. fitting it to the periodic function

2. plot the fit

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

temp\_max = np.array([39, 41, 43, 47, 49, 51, 45, 38, 37, 29, 27, 25])

temp\_min = np.array([21, 23, 27, 28, 32, 35, 31, 28, 21, 19, 17, 18])

from scipy import optimize

def yearly\_temps(times, avg, ampl, time\_offset):

return (avg

+ ampl \* np.cos((times + time\_offset) \* 2 \* np.pi / times.max()))

res\_max, cov\_max = optimize.curve\_fit(yearly\_temps, months,

temp\_max, [20, 10, 0])

res\_min, cov\_min = optimize.curve\_fit(yearly\_temps, months,

temp\_min, [-40, 20, 0])

days = np.linspace(0, 12, num=365)

plt.figure()

plt.plot(months, temp\_max, 'ro')

plt.plot(days, yearly\_temps(days, \*res\_max), 'r-')

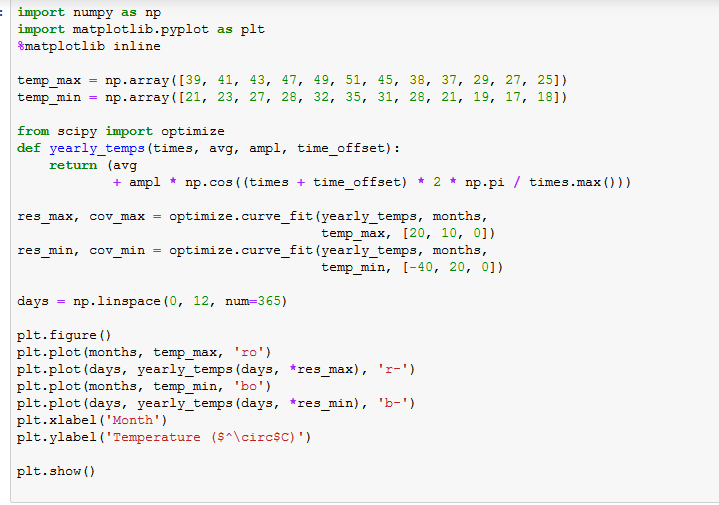
plt.plot(months, temp\_min, 'bo')

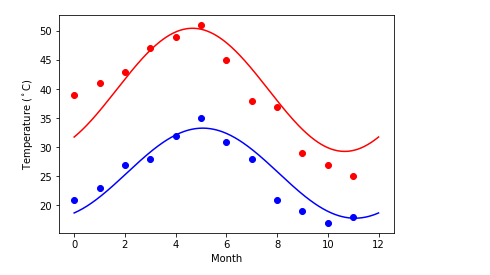
plt.plot(days, yearly\_temps(days, \*res\_min), 'b-')

plt.xlabel('Month')

plt.ylabel('Temperature ($^\circ$C)')

plt.show()





**Matplotlib:**

Q1. Create a pie chart presenting the male/female proportion

Answer:

import matplotlib as mpl

import matplotlib.pyplot as plt

url='https://raw.githubusercontent.com/Geoyi/Cleaning-Titanic-Data/master/titanic\_original.csv'

titanic = pd.read\_csv(url)

sdis=scatg.describe().counts

scatg=pd.Categorical(titanic.sex).dropna()

label=scatg.unique()

plt.pie(sdis,labels=label,startangle=90, autopct='%.1f%%')

plt.title('Male/Female proportion')

plt.show()



Q2. Create a scatterplot with the Fare paid and the Age, differ the plot color by gender

Answer:

x=titanic['fare']

y=titanic['age']

#z=titanic['sex']

z=np.array(titanic.sex.dropna(axis=0).unique())

colors = np.where(titanic['sex']=='female','r','g')

fig = plt.figure()

ax = fig.add\_subplot(1, 1, 1)

ax.scatter(x,y,s=30,alpha=0.5,edgecolors='none',c=colors,label=z)

plt.title('Scatter Plot')

plt.xlabel('Fare')

plt.ylabel('Age')

plt.show()

