Computational Statistics DA-5

2. Consider the following data

y_1	10	5	7	19	11	8	9	11	10	12
y_2	15	9	3	25	7	13	12	13	7	10
x	12	10	7	5	12	20	7	8	12	13

Construct a one way MANOVA by considering y_1, y_2 as dependent variables and x as a independent variable. Then write the conclusion.

Code:

```
import csv
with open('one_way_manova.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["x", "y1", "y2"])
    writer.writerow([12, 10, 15])
    writer.writerow([10, 5, 9])
    writer.writerow([7,7,3 ])
    writer.writerow([7,19,25 ])
    writer.writerow([7,11,7 ])
    writer.writerow([7,8,13 ])
    writer.writerow([7,9, 12])
    writer.writerow([7,11,3 ])
    writer.writerow([7,10,7 ])
    writer.writerow([7,12, 10])
with open('one_way_manova.csv', 'r', newline='') as file:
    reader=csv.reader(file)
    for row in reader:
        print(row)
import numpy as np
import pandas as pd
from statsmodels.multivariate.manova import MANOVA
from statsmodels.stats.multicomp import pairwise_tukeyhsd
df = pd.read_csv("one_way_manova.csv")
df=df.dropna ()
maov = MANOVA.from_formula('y1 + y2 ~ x', data=df)
print(maov.mv_test())


print("PRANAV MURTHY 21BBS0059”)
```

Output:

```
pranav@Pranavs-MacBook-Air-2 pythoncode % python3 DA3_1.py
['x', 'y1', 'y2']
['12', '10', '15']
['10', '5', '9']
['7', '7', '3']
['7', '19', '25']
['7', '11', '7']
['7', '8', '13']
['7', '9', '12']
['7', '11', '3']
['7', '10', '7']
['7', '12', '10']
                Multivariate linear model
============================================================

------------------------------------------------------------
       Intercept        Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.4148 2.0000 7.0000  4.9378 0.0460
         Pillai's trace 0.5852 2.0000 7.0000  4.9378 0.0460
 Hotelling-Lawley trace 1.4108 2.0000 7.0000  4.9378 0.0460
    Roy's greatest root 1.4108 2.0000 7.0000  4.9378 0.0460
------------------------------------------------------------

------------------------------------------------------------
           x            Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.6850 2.0000 7.0000  1.6097 0.2660
         Pillai's trace 0.3150 2.0000 7.0000  1.6097 0.2660
 Hotelling-Lawley trace 0.4599 2.0000 7.0000  1.6097 0.2660
    Roy's greatest root 0.4599 2.0000 7.0000  1.6097 0.2660
============================================================

PRANAV MURTHY 21BBS0059
pranav@Pranavs-MacBook-Air-2 pythoncode %
```

3. Consider the following data

y_1	10	5	7	19	11	8	9	11	10	12
y_2	15	9	3	25	7	13	12	13	7	10
x_1	12	10	7	5	12	20	7	8	12	13
x_2	10	8	6	23	12	11	13	11	10	12

Construct a two way MANOVA by considering y_1, y_2 as dependent variables and x_1, x_2 as a independent variables. Then write the conclusion.

Input:

```
import csv
with open('two_way_manova.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["x1", "x2", "y1", "y2"])
    writer.writerow([12, 10, 10, 15])
    writer.writerow([10, 8, 5, 9])
    writer.writerow([7, 6, 7, 3])
    writer.writerow([7, 23, 19, 25])
```

```
    writer.writerow([7, 12, 11, 7])
    writer.writerow([7, 11, 8, 13])
    writer.writerow([7, 13, 9, 12])
    writer.writerow([7, 11, 11, 3])
    writer.writerow([7, 10, 10, 7])
    writer.writerow([7, 12, 12, 10])
with open('two_way_manova.csv', 'r', newline='') as file:
    reader=csv.reader(file)
    for row in reader:
        print(row)
import numpy as np
import pandas as pd
from statsmodels.multivariate.manova import MANOVA
from statsmodels.stats.multicomp import pairwise_tukeyhsd
df = pd.read_csv("two_way_manova.csv")
df=df.dropna ()
maov = MANOVA.from_formula('y1 + y2 ~ x1 + x2', data=df)
print(maov.mv_test())

print("PRANAV MURTHY 21BBS0059”)
```

Output:

```
pranav@Pranavs-MacBook-Air-2 pythoncode % python3 DA3_1.py
['x1', 'x2', 'y1', 'y2']
['12', '10', '10', '15']
['10', '8', '5', '9']
['7', '6', '7', '3']
['7', '23', '19', '25']
['7', '12', '11', '7']
['7', '11', '8', '13']
['7', '13', '9', '12']
['7', '11', '11', '3']
['7', '10', '10', '7']
['7', '12', '12', '10']
                 Multivariate linear model
============================================================

------------------------------------------------------------
        Intercept        Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.4233 2.0000 6.0000  4.0874 0.0758
         Pillai's trace 0.5767 2.0000 6.0000  4.0874 0.0758
  Hotelling-Lawley trace 1.3625 2.0000 6.0000  4.0874 0.0758
     Roy's greatest root 1.3625 2.0000 6.0000  4.0874 0.0758
------------------------------------------------------------

------------------------------------------------------------
            x1           Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.4465 2.0000 6.0000  3.7191 0.0890
         Pillai's trace 0.5535 2.0000 6.0000  3.7191 0.0890
  Hotelling-Lawley trace 1.2397 2.0000 6.0000  3.7191 0.0890
     Roy's greatest root 1.2397 2.0000 6.0000  3.7191 0.0890
------------------------------------------------------------

------------------------------------------------------------
           x2            Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
          Wilks' lambda  0.0551 2.0000 6.0000 51.4181 0.0002
         Pillai's trace  0.9449 2.0000 6.0000 51.4181 0.0002
 Hotelling-Lawley trace 17.1394 2.0000 6.0000 51.4181 0.0002
    Roy's greatest root 17.1394 2.0000 6.0000 51.4181 0.0002
============================================================

PRANAV MURTHY 21BBS0059
pranav@Pranavs-MacBook-Air-2 pythoncode %
```

Pranav Murthy || 21BBS0059

Computational Statistics DA-4

1. Plot the bivariate normal distributions $\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}2 & 0\\0 & 3\end{bmatrix}\right)$ and $\mathcal{N}\left(\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}1 & 0.4\\0.4 & 2\end{bmatrix}\right)$. Then write the conclusion.

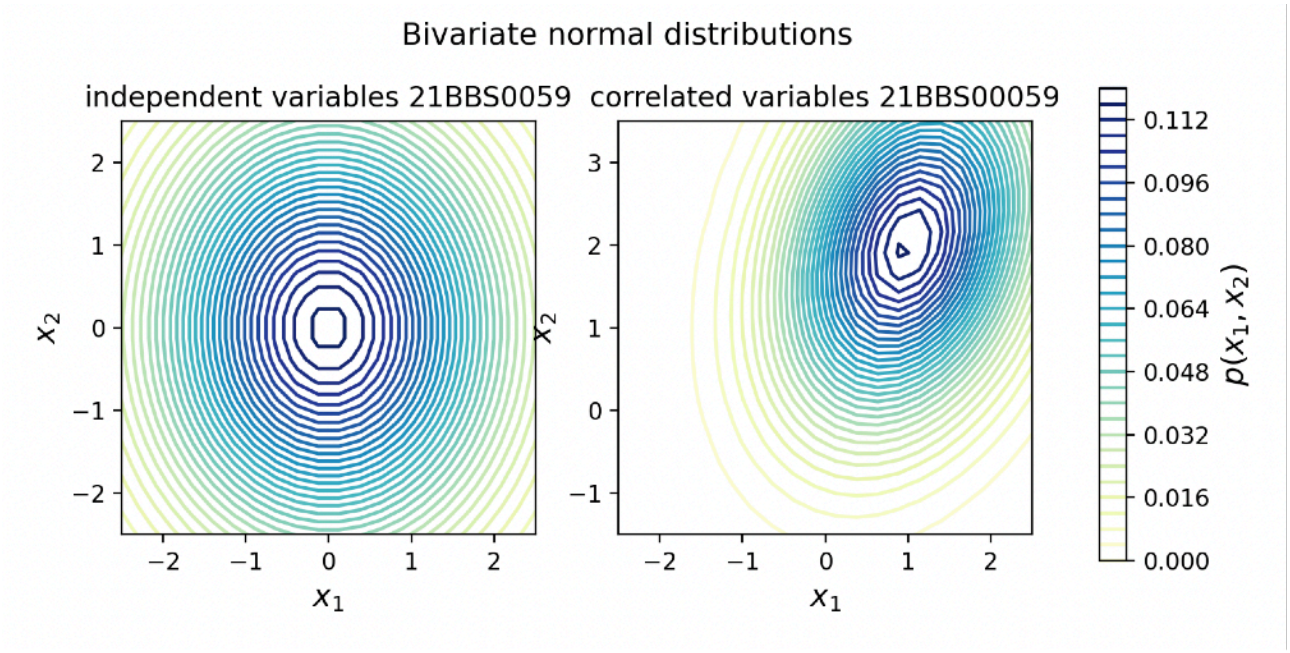
Code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal
from matplotlib import cm
def multivariate_normal(x ,d, mean, covariance):
    x_m=x-mean
    return(1./(np.sqrt((2*np.pi)**d*np.linalg.det(covariance)))*np.exp(-
(np.linalg.solve(covariance,x_m).T.dot(x_m))/2))
def generate_surface(mean, covariance, qd):
    nb_of_x = 64

    x1s = np.linspace(-8, 8, num=nb_of_x)
    x2s = np.linspace(-8, 8, num=nb_of_x)
    x1, x2 = np.meshgrid(x1s, x2s)
    pdf = np.zeros((nb_of_x, nb_of_x))
    for i in range(nb_of_x):
        for j in range(nb_of_x):
            pdf[i,j] = multivariate_normal(np.matrix([[x1[i,j]],[x2[i,j]]]),d, mean,
covariance)
    return x1, x2, pdf
fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(8,4))
d=2
bivariate_mean=np.matrix([[0.],[0.]])
bivariate_covariance=np.matrix([[2.,0.],[0.,3.]])
x1,x2,p=generate_surface(bivariate_mean,bivariate_covariance,d)
con = ax1.contour(x1,x2,p,33,cmap=cm.YlGnBu)
ax1.set_xlabel('$x_1$', fontsize=13)
ax1.set_ylabel('$x_2$', fontsize=13)
ax1.axis([-2.5,2.5,-2.5,2.5])
ax1.set_aspect('equal')
ax1.set_title('independent variables 21BBS0252',fontsize=12)
bivariate_mean=np.matrix([[1.],[2.]])
bivariate_covariance=np.matrix([[1.,0.4],[0.4,2.]])
x1,x2,p=generate_surface(bivariate_mean,bivariate_covariance,d)
con = ax2.contour(x1,x2,p,33,cmap=cm.YlGnBu)
ax2.set_xlabel('$x_1$', fontsize=13)
```

```
ax2.set_ylabel('$x_2$', fontsize=13)
ax2.axis([-2.5,2.5,-1.5,3.5])
ax2.set_aspect('equal')
ax2.set_title('correlated variables 21BBS0252', fontsize=12)
fig.subplots_adjust(right=0.8)
cbar_ax=fig.add_axes([0.85,0.15,0.02,0.7])
cbar=fig.colorbar(con,cax=cbar_ax)
cbar.ax.set_ylabel('$p(x_1, x_2)$', fontsize=13)
plt.suptitle('Bivariate normal distributions', fontsize=13,y=0.95)
plt.show()
```

Output:



2. Consider the following data

y_1	10	5	7	19	11	8	9	11	10	12
y_2	15	9	3	25	7	13	12	13	7	10
x	12	10	7	5	12	20	7	8	12	13

Construct a one way MANOVA by considering y_1, y_2 as dependent variables and x as a independent variable. Then write the conclusion.

Code:

```
import numpy as np
import pandas as pd
```

```
from statsmodels.multivariate.manova import MANOVA
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import csv
with open('one_way_manova.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["x", "y1", "y2"])
    writer.writerow([12, 10, 15])
    writer.writerow([10, 5, 9])
    writer.writerow([7,7,3 ])
    writer.writerow([7,19,25 ])
    writer.writerow([7,11,7 ])
    writer.writerow([7,8,13 ])
    writer.writerow([7,9, 12])
    writer.writerow([7,11,3 ])
    writer.writerow([7,10,7 ])
    writer.writerow([7,12, 10])
df = pd.read_csv("one_way_manova.csv")
df=df.dropna ()
maov = MANOVA.from_formula('y1 + y2 ~ x', data=df)
print(maov.mv_test())
with open('one_way_manova.csv', 'r', newline='') as file:
    reader=csv.reader(file)
    for row in reader:
        print(row)
print("PRANAV MURTHY 21BBS0059")
```

Output:

```
pranav@Pranavs-MacBook-Air-2 pythoncode % python3 DA3_1.py
                 Multivariate linear model
============================================================

------------------------------------------------------------
       Intercept        Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.4148 2.0000 7.0000  4.9378 0.0460
         Pillai's trace 0.5852 2.0000 7.0000  4.9378 0.0460
 Hotelling-Lawley trace 1.4108 2.0000 7.0000  4.9378 0.0460
    Roy's greatest root 1.4108 2.0000 7.0000  4.9378 0.0460
------------------------------------------------------------

------------------------------------------------------------
           x            Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.6850 2.0000 7.0000  1.6097 0.2660
         Pillai's trace 0.3150 2.0000 7.0000  1.6097 0.2660
 Hotelling-Lawley trace 0.4599 2.0000 7.0000  1.6097 0.2660
    Roy's greatest root 0.4599 2.0000 7.0000  1.6097 0.2660
============================================================

['x', 'y1', 'y2']
['12', '10', '15']
['10', '5', '9']
['7', '7', '3']
['7', '19', '25']
['7', '11', '7']
['7', '8', '13']
['7', '9', '12']
['7', '11', '3']
['7', '10', '7']
['7', '12', '10']
PRANAV MURTHY 21BBS0059
pranav@Pranavs-MacBook-Air-2 pythoncode %
```

Conclusion: -

Let significance level = 95 %. Thus, alpha = 0.05 In output, p-value for the intercept is 0.0460. Therefore, alpha > p-value. That means the differences between the means are not statistically significant

3. Consider the following data

y_1	10	5	7	19	11	8	9	11	10	12
y_2	15	9	3	25	7	13	12	13	7	10
x_1	12	10	7	5	12	20	7	8	12	13
x_2	10	8	6	23	12	11	13	11	10	12

Construct a two way MANOVA by considering y_1, y_2 as dependent variables and x_1, x_2 as a independent variables. Then write the conclusion.

Code:

```
import numpy as np
import pandas as pd
from statsmodels.multivariate.manova import MANOVA
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import csv
with open('two_way_manova.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["x1", "y1", "y2", "x2"])
    writer.writerow([12, 10, 15, 10])
    writer.writerow([10, 5, 9, 8])
    writer.writerow([7, 7, 3 , 6])
    writer.writerow([7, 19, 25, 23 ])
    writer.writerow([7, 11, 7, 12 ])
    writer.writerow([7, 8, 13, 11 ])
    writer.writerow([7, 9, 12, 13])
    writer.writerow([7, 11, 3, 11 ])
    writer.writerow([7, 10, 7, 10])
    writer.writerow([7, 12, 10, 12 ])
df = pd.read_csv("two_way_manova.csv")
df=df.dropna ()
maov = MANOVA.from_formula('y1 + y2 ~ x1 + x2', data=df)
print(maov.mv_test())
with open('two_way_manova.csv', 'r', newline='') as file:
    reader=csv.reader(file)
```

```
    for row in reader:
        print(row)

print("PRANAV MURTHY 21BBS0059")
```

Output:

```
pranav@Pranavs-MacBook-Air-2 pythoncode % python3 DA3_1.py
                 Multivariate linear model
============================================================

------------------------------------------------------------
        Intercept        Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
           Wilks' lambda 0.4233 2.0000 6.0000  4.0874 0.0758
          Pillai's trace 0.5767 2.0000 6.0000  4.0874 0.0758
  Hotelling-Lawley trace 1.3625 2.0000 6.0000  4.0874 0.0758
     Roy's greatest root 1.3625 2.0000 6.0000  4.0874 0.0758
------------------------------------------------------------

------------------------------------------------------------
            x1           Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
           Wilks' lambda 0.4465 2.0000 6.0000  3.7191 0.0890
          Pillai's trace 0.5535 2.0000 6.0000  3.7191 0.0890
  Hotelling-Lawley trace 1.2397 2.0000 6.0000  3.7191 0.0890
     Roy's greatest root 1.2397 2.0000 6.0000  3.7191 0.0890
------------------------------------------------------------

------------------------------------------------------------
            x2           Value  Num DF Den DF F Value Pr > F
------------------------------------------------------------
           Wilks' lambda  0.0551 2.0000 6.0000 51.4181 0.0002
          Pillai's trace  0.9449 2.0000 6.0000 51.4181 0.0002
  Hotelling-Lawley trace 17.1394 2.0000 6.0000 51.4181 0.0002
     Roy's greatest root 17.1394 2.0000 6.0000 51.4181 0.0002
============================================================

['x1', 'y1', 'y2', 'x2']
['12', '10', '15', '10']
['10', '5', '9', '8']
['7', '7', '3', '6']
['7', '19', '25', '23']
['7', '11', '7', '12']
['7', '8', '13', '11']
['7', '9', '12', '13']
['7', '11', '3', '11']
['7', '10', '7', '10']
['7', '12', '10', '12']
PRANAV MURTHY 21BBS0059
pranav@Pranavs-MacBook-Air-2 pythoncode %
```

Pranav Murthy || 21BBS0059

Computational Statistics DA-3

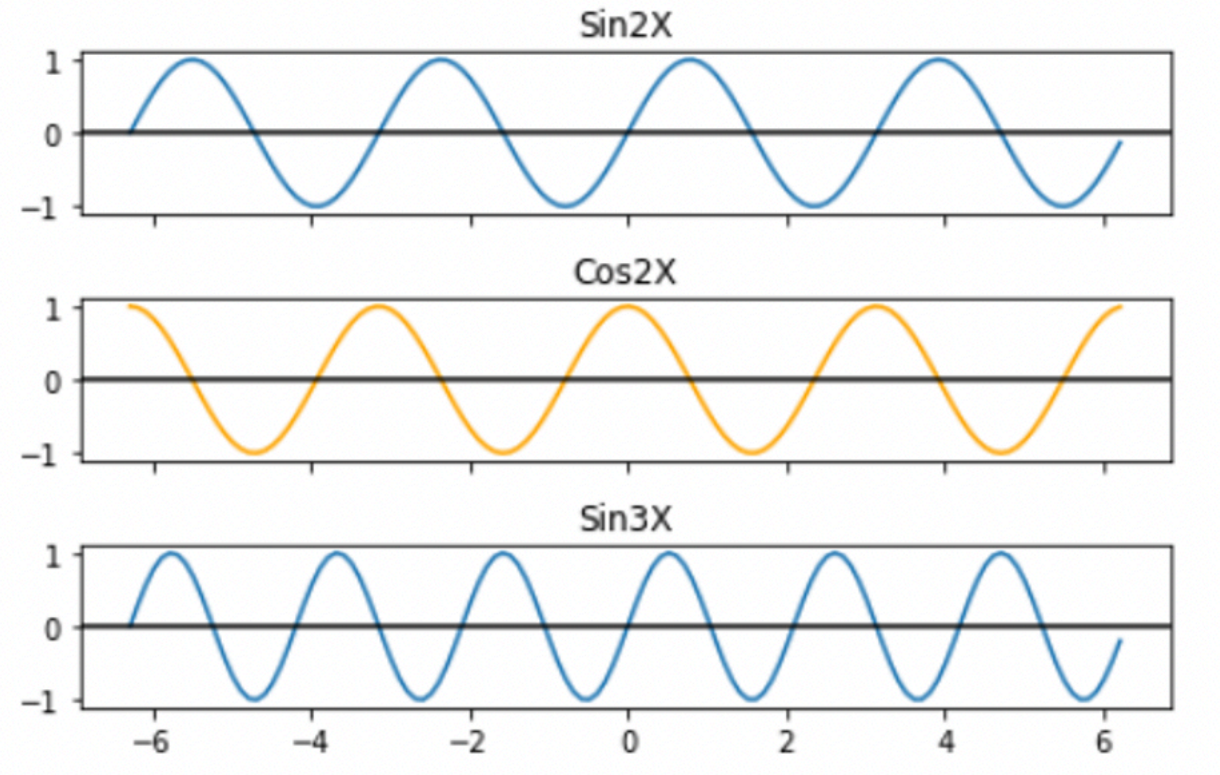
1. Write a python program to plot the functions $\sin(2x)$, $\cos(2x)$ and $\sin(3x)$. Use subplot command to plot these functions.

Code:

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

x = np.arrange(-2*np.pi,2*np.pi,0.1)
y1 = np.sin(2*x)
y2 = np.cos(2*x)
y3 = np.sin(3*x)
fig, (ax1,ax2,ax3) = plt.subplots(nrows=3,ncols=1,sharex=True)
ax1.set_title("Sin2X")
ax1.plot(x,y1,label="sin2x")
ax1.axhline(y=0,color="#000000")
ax2.set_title("Cos2X")
ax2.plot(x,y1,label="Cos2X")
ax2.axhline(y=0,color="#000000")
ax3.set_title("Sin3X")
ax3.plot(x,y1,label="sin3x")
ax3.axhline(y=0,color="#000000")
fig.tight_layout()
plt.show()
```

Output:



2. Write a python program to find the minimum of a function $f(x) = x^2 + 15\cos(x)$.

Code:

```
import numpy as np
import scipy.optimize as spo

def f(x):
        y=x**2 + 15*np.cos(x)
        return y
x_start = -3
result = spo.minimize(f,x_start,options={"disp:":True})
if result.success:
        print("minimum of the function is at x={} and y={}".format(result.x,result.fun))
else:
        printf("no minimum found")
```

Output:

```
Optimization terminated successfully.
         Current function value: -6.303627
         Iterations: 3
         Function evaluations: 10
         Gradient evaluations: 5
mimimum of the function is at x=[-2.76414175] and y = -6.303627327052955
```

3. Draw the samples from a normal distribution with mean 2 and variance 3. Display the histograms for the samples.

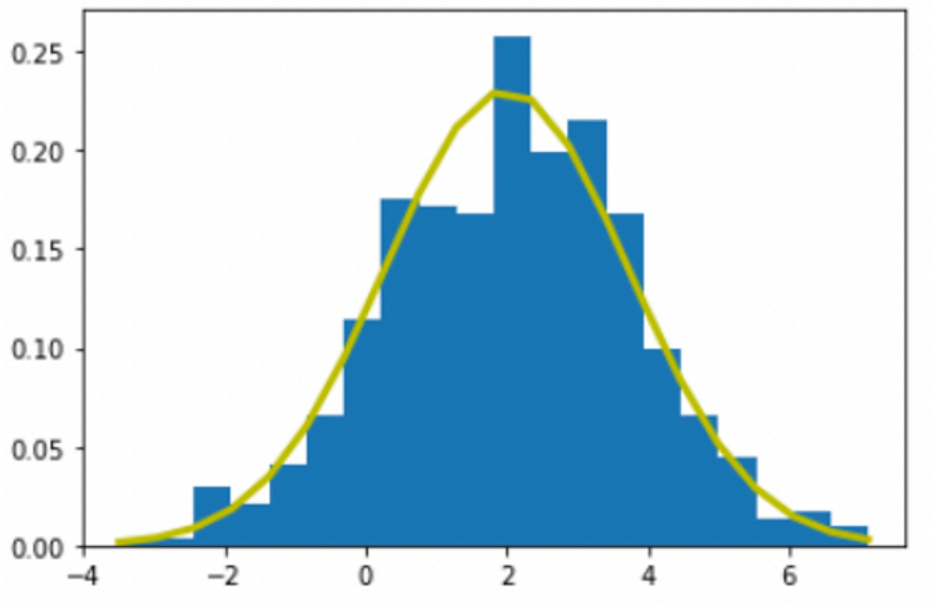
Code:

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

mu,sigma =2, (3**(1/2))

s = np.random.normal(mu,sigma,1000)
count,bins, ignored = plt.hist(s,20,density=True)
plt.plot(bins,1/(sigma*np.sqrt(2*np.pi))*np.exp(-(bins-mu)**2/(2*sigma**2)),linewidth=3,color='y')
plt.show()
```

Output:

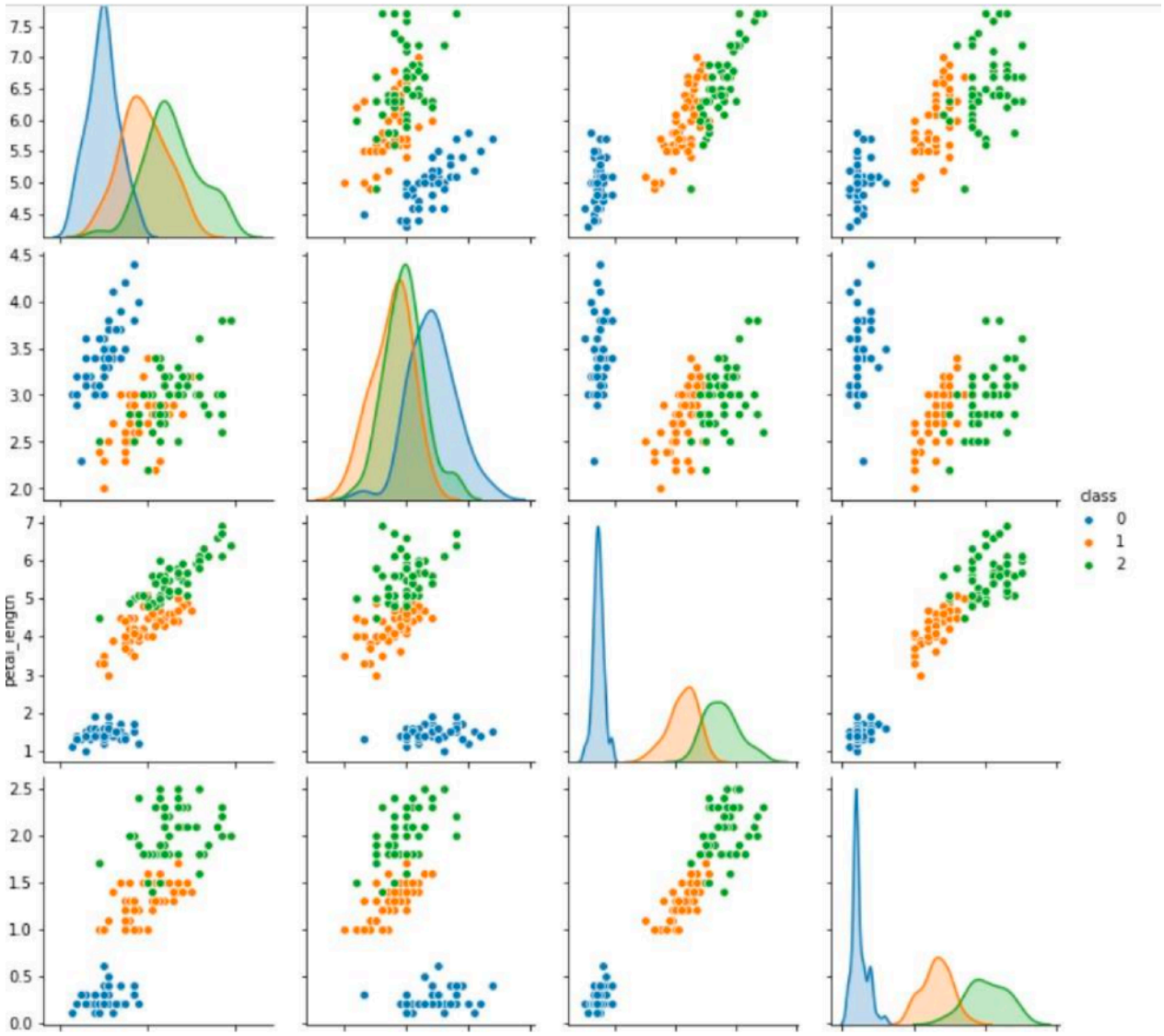


4. Read the data given in "http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data" which contains 178 rows and 14 columns. Arrange these data with the columns name as $S1, S2, \ldots, S14$. Extract the variables $S4$, $S5$, $S6$, $S7$, $S8$ and make matrix scatter plot for these five variables.

Code:

```
import pandas as pd
import matplotlib.pyplot as mp
datadata = pd.read_csv("http://archive.ics.uci.edu/ml/machine-learning-databases
/wine/wine.data")
dataframe data = data.head()
df = pd.DataFrame(data, columns=["Name", "Price", "User Rating"])
df.plot(x="Name", y=["Price", "User Rating"], kind="bar", figsize=(9,8))
mp.show()
```

Output:


petal_length
class
0
1
2