Table of Contents

[Question 1: Data Wrangling 2](#_Toc145961402)

[Solutions: 3](#_Toc145961403)

[Question 1a) 3](#_Toc145961404)

[Question 1b) 4](#_Toc145961405)

[Question 1C) 8](#_Toc145961406)

[Question 2: Generating data & the normal distribution 9](#_Toc145961407)

[Solutions: 10](#_Toc145961408)

[Question 2a) 10](#_Toc145961409)

[Question 2b) 11](#_Toc145961410)

[Question 2c) 12](#_Toc145961411)

[Question 2d) 12](#_Toc145961412)

[Question 2e) 13](#_Toc145961413)

[Appendix: 14](#_Toc145961414)

[Question 1: 14](#_Toc145961415)

[Question 2: 15](#_Toc145961416)

# Question 1: Data Wrangling

Consider two possible features for a new fruit classification system: weight and diameter. Sample data for each feature is provided in assigData2.tsv

100 weight and diameter measurements are given for three types of fruit: apple, orange, and grape. (File can be easily viewed in Excel or MATLAB. Columns are: W\_apl W\_orng W\_grp D\_apl D\_orng D\_grp)

1. To develop a Bayesian classifier, we need to estimate the parameters of the class-conditional distribution for each feature and for each class. Assuming the class-conditional distributions follow normal distributions with unknown mean and variance for each class, estimate the six means and the six estimates of variance.
2. Plot the histograms for each feature showing the distribution of each feature over each class. For each feature, you should have a single plot (single axis) with three potentially overlapping histograms representing the three fruit types.
3. Use transparency and a different color and/or line style for each class and make sure you can see all the data (i.e., that bars are not completely occluding each other in your figure).
4. Which feature would you prefer and why? (150 words)
5. Illustrate results using at least two bin widths when generating your histograms.
6. Provide a plot visualizing apple weight vs. diameter. Add a line of best fit and report the Pearson Correlation Coefficient.

## Solutions:

### Question 1a)

Table 1 contains the mean, variance, and standard deviation (STD) *(additional info)* of the 6 features in the dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Mean (µ) | Variance (σ2) | STD (σ) |
| W\_apl | 0011.003084 | 0001.401103 | 01.183682 |
| W\_orng | 0011.944999 | 0006.806620 | 02.608950 |
| W\_grp | 0008.733358 | 0024.787095 | 04.978664 |
| D\_apl | 1006.707200 | 1621.374432 | 40.266294 |
| D\_orng | 1114.833850 | 0383.405810 | 19.580751 |
| D\_grp | 0832.546227 | 8356.381727 | 91.413247 |

Table : Mean, Variance, and STD for each column in the dataset

Please note, that the question assumes there is a normal distribution in the data, but from the solution in Q1b) it is known that the data is not normally distributed for all the columns.

The formula for each is provided below:

### Question 1b)

Figure 1 shows the histogram of weight distribution for the three classes with 30 bins for each class. Figure 2 shows the histogram of diameter distribution for the three classes with 20 bins for each class.

Figures 3-6 have a constant bin width for each class. Figures 3 and 4 are histograms for the weight distributions with the bin width being 1 and 2 units of width respectively. (The units are not provided for the data)

Figures 5 and 6 are histograms for the diameter distribution with the bin width being 30 and 60 respectively.

In all of the figures “**line, transparency, and different colours**“ have been maintained.

The legend for each figure is provided in the top-right hand corner for all the figures in this section.

For all the figures the class for apples is shown in “Red”, the class for oranges is shown in “orange”, and the class for grapes is shown in “purple”.

Please note for Figures 1 and 2 the “bin count” is constant for each class in their respective figures. In the case of Figures 3-6 the “bin width” is constant for each class in their respective figures.

**The feature “Diameter” is the ideal candidate classification**. The reason is that there is minimal overlap between the classes in the case of Diameter. When considering the “Weight” we see that there is a very significant **overlap** between “apples” and “oranges”, hence models will have a hard time distinguishing between the two in a **generalized** setting.

Note: Classifiers may still manage to give good accuracy on **train data** when using only “Weight” but they will perform poorly on **test or validation data.** This is because of overfitting.

A graph of a weight histogram

Description automatically generated

Figure : Histogram for "Weight" distribution for apples, oranges, and grapes. The bin count for each class is 30.

A graph of different colored bars

Description automatically generated

Figure : Histogram for "Diameter" distribution for apples, oranges, and grapes. The bin count for each class is 20.

A graph of weight loss

Description automatically generated with medium confidence

Figure : Histogram for "Weight" distribution for apples, oranges, and grapes. The bin width for each class is equal to 1 unit.

A graph with different colored bars

Description automatically generated

Figure : Histogram for "Weight" distribution for apples, oranges, and grapes. The bin width for each class is equal to 2 units.

A graph of different colored lines

Description automatically generated

Figure : Histogram for "Diameter" distribution for apples, oranges, and grapes. The bin with for each class is equal to 10.

A graph of a diagram

Description automatically generated with medium confidence

Figure : Histogram for "Diameter" distribution for apples, oranges, and grapes. The bin with for each class is equal to 20.

### Question 1C)

Figure 7 represents the scatter plot for weight of apple vs the diameter of apple. The points are marked with “a” in the figure. The line of best fit is shown in the figure in blue. The equation for the best-fit line is shown in the plot’s legend. The x-axis in the figure shows the weight of the apple while the y-axis shows the diameter of the apple.

The Pearson correlation for the weight and diameter of apples is “**0.085**”. The correlation between the remaining features is provided in Table 2. The cell intersection of column and rows provides the correlation among the features.

The line of best fit equation is

A graph with red dots and a blue line

Description automatically generated

Figure : Weight VS Diameter for Apple along with Line of Best Fit using Linear Regression



Table : Pearson Correlation among each column in the dataset.

# Question 2: Generating data & the normal distribution

1. Generate 1000 samples drawn from a trivariate normal distribution with , .

You do not need to provide the actual samples in your assignment submission. Instead, report estimates of the mean and variance of the first dimension based on your 1000 samples. Do your estimates agree with the actual values? (estimates + brief discussion)

1. Create two scatter plots of the data, ensuring that the scale of both axes are equal so that the true shape of the distribution is visible. The first scatter plot should visualize the first two dimensions of your data. The second scatter plot should visualize dimension 1 vs. dimension 3. Why do their shapes differ? (25 words)
2. What is its trace of Σ? Is Σ positive definite? Explain.
3. Calculate and report the eigenvectors and eigenvalues of Σ.
4. Lastly, plot the PDF and CDF for the third dimension of your distribution.

## Solutions:

### Question 2a)

The estimated mean and variance for the first dimension are as follows:

* Mean: 4.938153
* Variance: 3.67591658

Yes, the estimated mean agree with the generated data. They are off by a tiny fraction, but this is expected based on the random generation of data.

The remaining generated data’s dimension also follow the same mean, and covariance.

Following is a list of means for generated data:

* Dimension 1: 4.938
* Dimension 2: -0.409
* Dimension 3: 16.986

The Covariance matrix is as follows for the generated data

### Question 2b)

Figures 8 and 9 are the scatter plots required in the question. Figure 8 plots dimension 1 vs 2. Figure 9 plots dimension 1 vs 3. The scale of x and y axis are constant in both figures. The colour of markers in each figure has been made unique to distinguish between the two. The colour blue is used for markers in Figure 8 and green has been used in Figure 9.

A blue dot diagram with numbers

Description automatically generated with medium confidence

Figure : Scatter plot for dimension 1 vs 2

A green dot diagram with numbers

Description automatically generated with medium confidence

Figure : Scatter plot for dimension 1 vs 3d

Differ:

**The centre of the cluster in each figure is determined by the mean of the data. The inclination of the cluster is dependent on correlation.**

Note: Further explanation could be provided but the limit is 25 words.

### Question 2c)

Positive defined:

A matrix is positive definite when:

1. It is symmetric
2. Its eigen values are positive.

Σ is symmetric since Σ= ΣT and the eigen values are positive (from solution 2d). **Therefore, it is positive definite.**

### Question 2d)

Given a matrix “A” the eigenvalue “�,λ” and eigenvector “v” follow the formula: Av = λv.

The calculated eigenvalues are: [5.21773098 3.79611128 1.98615774]

The calculated eigenvectors are:

* [ 0.37920187 -0.9251552 -0.01714037]
* [ 0.92353174 0.37725742 0.06903598]
* [-0.05740267 -0.04200825 0.99746691]

### Question 2e)

Figures 10 and 11 provide the PDF and CDF respectively for the 3rd dimension of randomly generated data.

A diagram of a function

Description automatically generated

Figure : PDF for dimension 3

A graph of a distribution function

Description automatically generated

Figure CDF for dimension 3

# Appendix:

Please note, all code is written in Python. Content written in “<>” will change based on the individual.

## Question 1:

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

from sklearn.linear\_model import LinearRegression

data\_loc = “<Your dataset location>"

dataset = pd.read\_csv(data\_loc, sep="   ")

# Get the mean and variance

dataset.describe()

# Code for plotting histogram

# Repeat this code to plot multiple figures

dataset[“<column\_name>"].hist(

    bins=<list or range for constant width | integer for constant number of bins>,

    color="<provide colour name>",

    histtype=u'step',

    figsize=(12,5)

    )

# Build a line of best fit using “Linear regression

x = dataset[["W\_apl"]]

y = dataset[["D\_apl"]]

model = LinearRegression()

model.fit(x,y)

r2\_score = model.score(x, y)

print(f"""

      Linear Regression:

      \t R2: {r2\_score}

      \t m(slope): {model.coef\_}

      \t c(intercept: {model.intercept\_})

      """)

# visualize apple’s weight vd diameter

ax = dataset[["W\_apl","D\_apl"]].plot(

    x="W\_apl",

    y="D\_apl",

    kind="scatter",

    marker="$a$",

    color="red",

    label="Apple",

    s=50,

)

ax.plot(

    x,

    m\*x+c,

    label=f"Y={m:0.2}X+{c:0.2f}",

    )

plt.legend(loc='upper left')

ax.set\_title("Weight of Apple Vs Diameter of Apple")

# calculate Pearson correlation

dataset[["W\_apl","D\_apl"]].corr(method='pearson')

## Question 2:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

mean = [5, -0.5, 17]

cov = [[4,0.5,0],[0.5,5,-0.2],[0,-0.2,2]]

sample\_size = 1000

# generate dataset

 pd.DataFrame(

    np.random.multivariate\_normal(

        mean=mean,

        cov=cov,

        size=sample\_size

    ),

    columns=["dimension\_1","dimension\_2","dimension\_3"]

)

# get the mean and variance for generate data

dataset.describe()

dataset.corr()

# plot scatter plot

dataset.plot.scatter(

    x="dimension\_1",

    y="dimension\_2",

    label="Dim 1 vs 2",

    color="blue",

    xlim = [-5,15],

    ylim=[-10,25]

)

dataset.plot.scatter(

    x="dimension\_1",

    y="dimension\_3",

    label="Dim 1 vs 3",

    color="green",

    xlim = [-5,15],

    ylim=[-10,25]

)

# Calculate Trace

trace = np.trace(cov)

print(f"trace of Sigma: {trace}")

# Calculate eigenvalues and eigenvectors

eigenvalues, eigenvectors = np.linalg.eig(cov)

print(f"eigen values = {eigenvalues}")

print(f"eigen vectors:\n{eigenvectors}")

# plot PDF

ax = dataset['dimension\_3'].plot.kde(

    title="Probablity Density Function (PDF) for Dimension 3",

    label="PDF-Dimension 3",

    legend=True

)

ax.set\_xlabel("values")

# plot CDF

for i in range(3,4):

    count, bins\_count = np.histogram(dataset[f'dimension\_{i}'], bins=1000)

    pdf = count / sum(count)

    cdf = np.cumsum(pdf)

    plt.plot(

        bins\_count[1:],

        cdf\*100,

        label=f"CDF - Dimension {i}"

        )

plt.legend(loc="upper left")

plt.ylabel("Percentage")

plt.xlabel("Values")

plt.title("Cumulative Distribution Function (CDF) Dimension 3")