Assignment 2 - MTL766: Multivariate Statistical Methods

Due Date: 11:59pm Friday, 14th October, 2023

Instructions

Please adhere to the following guidelines while completing this assignment:

1. Your submission should consist of a single Jupyter notebook containing all code implementations and corresponding explanations in Markdown format.
2. For each question, provide a proper descriptions that outline the mathematical and implementation details of the code.
3. Ensure that your code is thoroughly documented with inline comments to explain key steps and logic.
4. Evaluation will consider both the quality of your submitted work and your ability to articulate your code's functionality.
5. All code must be independently authored from scratch, without any reference to external solutions.
6. You are allowed to use only the following libraries: `SciPy`, `NumPy`, and `Pandas`.

Your adherence to these instructions will contribute to a comprehensive evaluation of your assignment.

Feel free to address any further questions or concerns you may have on the assignment by posting them in the Microsoft Teams assignment channel.

Description

In this assignment, you will work with multivariate normal distributions, properties of the distribution, and hypothesis testing using Python. You are allowed to use only the following libraries: SciPy, NumPy, and Pandas.

Questions [20 marks]

Question 1: Multivariate Normal Distribution Generation

1. Write a Python function that generates n samples from a multivariate normal distribution with a given mean vector μ and covariance matrix Σ. Your function should return a NumPy array where each row represents a sample. [**1**]

Question 2: Distribution Properties

Write Python functions to calculate the following properties for the generated samples. The sample mean and sample variance-covariance matrix should be calculated from scratch without using direct functions available in `numpy` for computation. Specifically functions like `np.mean` and `np.cov` are not allowed.

1. Mean Vector: Write a function to compute the sample mean vector. [**1**]

2. Covariance Matrix: Write a function to calculate the sample covariance matrix. [**1.5**]

3. Moment Generating Function: Write a function that computes the moment generating function of the distribution for a given t. [**1.5**]

Question 3: Conditional Distribution

1. Write a Python function that calculates the conditional distribution of the i-th variable given the other variables. The function should return the mean vector and covariance matrix of the conditional distribution. In the test section try this function for each variable in the multivariate normal distribution, evaluated at the value of remaining variables according to any of the generated samples. [**5**]

Question 4: Mahalanobis Distance

1. Mahalanobis Distance: Write a function that calculates the Mahalanobis distance between each sample and the mean vector μ. [**1.5**]

Question 5: Sampling from Wishart Distribution

1. Write code to sample from a Wishart Distribution from scratch, including density and sampling functions. Specifically you cannot directly use `scipy.stats.wishart`. Other functionalities from `scipy` and `numpy` can be used. Write a function to generate samples from the Wishart distribution with specified degrees of freedom and scale matrix. [**5**]

2. Implement a function to calculate the Hotelling's T^2 statistic for a given set of multivariate data. [**2**]

Question 6: Sampling Distribution

1. Write a Python function to compute the distribution of the sample mean for the multivariate normal distribution. Generate samples from this distribution. You can use previously defined functions. [**1.5**]

Submission

Submit your Code (.ipynb file) containing the functions and any necessary explanations or comments.

Also, provide a **test case section** that demonstrates the usage of these functions with sample data. The number of variables in the multivariate case should be at least 6. Make use of the following data.

$$\Sigma = \begin{bmatrix} 4.3 & -0.1 & 0.7 & 1.2 & -0.5 & 0.4 \\ -0.1 & 3.8 & 0.3 & 0.9 & 0.2 & 0.1 \\ 0.7 & 0.3 & 5.5 & -0.3 & 0.8 & -0.2 \\ 1.2 & 0.9 & -0.3 & 6.1 & 0.1 & -0.6 \\ -0.5 & 0.2 & 0.8 & 0.1 & 4.7 & 0.5 \\ 0.4 & 0.1 & -0.2 & -0.6 & 0.5 & 3.9 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 1.5 \\ 2.8 \\ -0.6 \\ 4.1 \\ -1.0 \\ 3.2 \end{bmatrix}$$

You can use your own data as well for additional testing but ensure that the covariance matrix is symmetric positive definite.