Name ID

Ahmed Shaban 202200217

Abdullah Ahmed 202200206

Alaa Sabry 202200331

**Report on MATLAB App for Probability and Statistical Analysis**

Section 1

**Introduction**

The developed MATLAB application provides an interactive environment for conducting probability and statistical analysis. The app allows users to load a sample dataset, compute essential statistical properties, and visualize probability density functions (PDF), cumulative distribution functions (CDF), and moment-generating functions (MGFs). This report details the app's functionality, workflow, and implementation.

**Features and Workflow**

The app includes the following main components:

**1. Loading Data**

* **Button:** Load File Button
* **Purpose:** Allows users to load a .mat file containing the sample data.
* **Functionality:**
  + The user selects a .mat file via a file dialog.
  + The app reads the first variable in the file and stores it in the space property.
  + The filename is displayed in the File Path Edit Field for reference.
* **Validation:** Alerts the user if no file is selected.

A screenshot of a computer code

Description automatically generated

**2. Computing and Plotting the PDF**

* **Button:** PDF Button
* **Purpose:** Computes and visualizes the Probability Density Function.
* **Steps:**
  1. The user specifies the number of bins via Enter number of bins Edit Field.
  2. The app calculates the bin edges, centers, and histogram counts.
  3. The PDF is computed as: PDF=counts total counts ×b in width\text {PDF} = \frac{\text{counts}}{\text{total counts} \times \text{bin width}}
  4. The PDF is plotted as a smooth curve using interpolation.
* **Validation:** Alerts users if the number of bins is invalid (e.g., non-positive).

**A white background with black text

Description automatically generated**

**3. Computing and plotting the CDF**

* **Button:** CDF Button
* **Purpose:** Computes and visualizes the Cumulative Distribution Function.
* **Steps:**
  + Uses the computed PDF to calculate the CDF as: CDF(x)=∑ PDF (xi)×bin width\text {CDF}(x) = \sum \ text {PDF}(x\_i) \times \ text {bin width}
  + Plots the CDF with respect to bin centers.

A screenshot of a computer program

Description automatically generated

**4. Statistics Computation**

* **Button:** Statistics Button
* **Purpose:** Calculates statistical properties including the mean, variance, and third moment.
* **Steps:**
  1. Calculates the expected value (mean):
  2. Calculates the variance:
  3. Calculates the third moment
  4. Updates the respective labels in the UI to display these values.

**5. Moment-Generating Function (MGF)**

* **Button:** M\_t Button
* **Purpose:** Plots the Moment-Generating Function (MGF) and its derivatives.
* **Steps:**
  1. The user specifies the maximum value of tt via T\_maxEdit Field.
  2. The app computes:
     + M(t)=E [e^tX] M(t)
     + M′(t)=E[Xe^tX]M'(t)
     + M′′(t)=E[X^2e^tX]M''(t)
  3. Plots M(t)M(t), M′(t)M'(t), and M′′(t)M''(t) over the range of tt.
  4. Displays the values of M (0)M(0), M′(0)M'(0), and M′′(0)M''(0).

**Implementation Details**

**Properties**

* **Public Properties:**
  + UI Figure: Main container for the app.
  + Multiple UI Axes for plotting (PDF, CDF, MGFs).
* **Private Properties:**
  + filename, space, PDF, bin\_width, bin\_cen, prob\_of\_X for data handling and computations.

**Callback Functions**

* Each button and input field has an associated callback function to handle user actions and update the UI.
* Error handling and validation are implemented for robust functionality.

**Visualization**

* Each plot (PDF, CDF, MGFs) is displayed in a dedicated axis with customized labels, titles, and legends.

**User Interface Design**

* **UI Components:**
  + Buttons for actions (e.g., Load File, PDF, CDF, Statistics, MGF).
  + Input fields for parameters (e.g., file path, number of bins, T max T\_{max}).
  + Labels to display computed values (mean, variance, moments).
* **Axes:**
  + Dedicated panels for each visualization.
  + Organized layout for clarity.

**A screenshot of a computer

Description automatically generatedOutput**

* **Uniform distribution**

The hand analysis for this gave us the following results:

* Mean:
* Variance:
* 3rd Moment:

So the output matches these results with small deviations.

* **A screenshot of a computer

  Description automatically generatedNormal distribution**

The hand analysis for this gave us the following results:

* Mean:
* Variance:
* 3rd Moment:

So the output matches these results with small deviations.

* **A screenshot of a computer

  Description automatically generatedPoisson distribution**

The hand analysis for this gave us the following results:

* Mean:
* Variance:
* 3rd Moment:

So the output matches these results with small deviations; however, the value of the third moment was not the same as the hand analysis; this could be due to that calculating the moment in the Poisson distribution relies on empirical data and sampling.

* **Binomial distribution**
* **Given sample**

**A screenshot of a graph

Description automatically generated**

Section 2

**Features and Workflow**

The app contains multiple interactive components, including buttons, input fields, and plots. Below is a detailed explanation of each feature and its workflow:

**1. Loading Data**

* **Button:** Load File Button
* **Purpose:** Enables the user to load a .mat file containing the sample data.
* **Workflow:**
  + The app opens a file selection dialog when the button is clicked.
  + It extracts the first variable from the selected .mat file and assigns it to the space property.
  + The file name is displayed in the File Path Edit Field for user reference.
  + If no file is selected, the app remains idle without proceeding further.

A screenshot of a computer code

Description automatically generated

**2. Plotting the Joint Probability Distribution**

* **Button:** Plot Button
* **Purpose:** Visualizes the joint probability distribution of two variables (X and Y).
* **Workflow:**
  1. Extracts X and Y values from the loaded data (space).
  2. Prompts the user to enter the number of bins in the Enter number of bins Edit Field.
  3. Computes a 2D histogram to estimate the joint probability distribution.
  4. Plots the joint probability distribution on UI Axes as a 3D surface plot.
  5. Adds a colormap for better visualization and enables smooth shading.

A screenshot of a computer screen

Description automatically generated

**3. Plotting Marginal Distributions**

* **Axes:** UIAxes2 (X Marginal) and UIAxes3 (Y Marginal)
* **Workflow:**
  + Computes histograms for X and Y values using the specified number of bins.
  + Normalizes the histograms to estimate marginal probabilities.
  + Uses interpolation to create smooth curves for marginal distributions.
  + Displays the marginal distributions on separate axes for clarity.

**4. Computing Statistics**

* **Button:** Statistics Button
* **Purpose:** Calculates and displays covariance and correlation coefficients.
* **Workflow:**
  1. Extracts X and Y values from the loaded data.
  2. Computes the expected values (means) of X and Y using the marginal distributions.
  3. Calculates the covariance as:
  4. Computes the standard deviations of X and Y.
  5. Calculates the correlation coefficient as:
  6. Updates the Co Value and Corr Value labels to display the computed values.

**Implementation Details**

**Properties**

* **Public Properties:**
  + UI Figure: Main container for the app.
  + Components like UI Axes, Buttons, and Labels to interact with the user.
* **Private Properties:**
  + space: Stores the loaded sample data.
  + Centers of X, centers of Y: Stores bin centers for marginal distributions.
  + X marginal, Y marginal: Stores computed marginal probabilities.

**Callback Functions**

* Each UI component has a callback function that executes its specific functionality upon user interaction.
* Key callback functions include:
  + Load File Button Pushed: Loads the data and updates the file path.
  + Plot Button Pushed: Generates joint and marginal probability distributions.
  + Statistics Button Pushed: Calculates statistical metrics and updates the UI.

**Visualization**

* **Joint Distribution:**
  + Displayed as a 3D surface plot on UI Axes with labeled axes and a color-coded surface.
* **Marginal Distributions:**
  + Displayed as 2D line plots with smooth interpolation.
  + Each plot is displayed on separate axes (UIAxes2 and UIAxes3).

**User Interface Design**

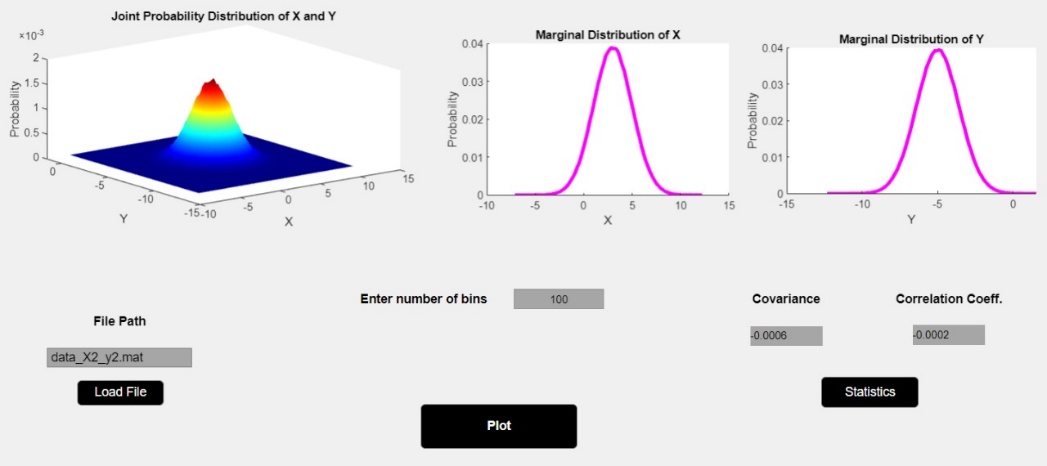
**UI Components**

* **Buttons:**
  + Load File Button for loading data.
  + Plot Button for generating plots.
  + Statistics Button for computing metrics.
* **Input Fields:**
  + File Path Edit Field to display the loaded file name.
  + Enter number of bins Edit Field to specify the number of bins.
* **Labels:**
  + Display computed covariance and correlation coefficient values.

**Layout**

* Organized into three sections:
  + **Top Section:** Displays the joint probability distribution.
  + **Middle Section:** Shows marginal distributions for X and Y.
  + **Bottom Section:** Includes controls for file loading and statistical computation.

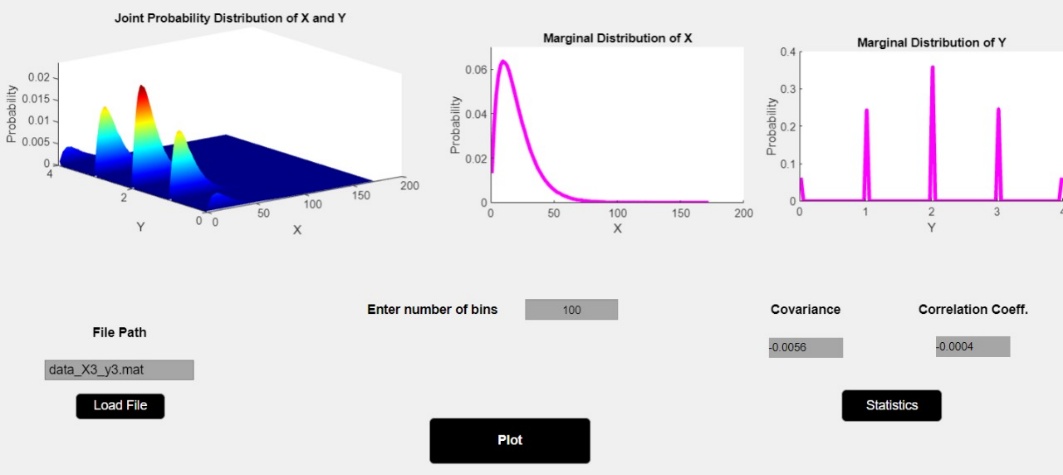
**Output**

* **X2**

The hand analysis for this gave us the following results:

* Covariance:
* Corr. Coeff.:

So the output matches these results with small deviations.

* **X3**

The hand analysis for this gave us the following results:

* Covariance:
* Corr. Coeff.:

So the output matches these results with small deviations.

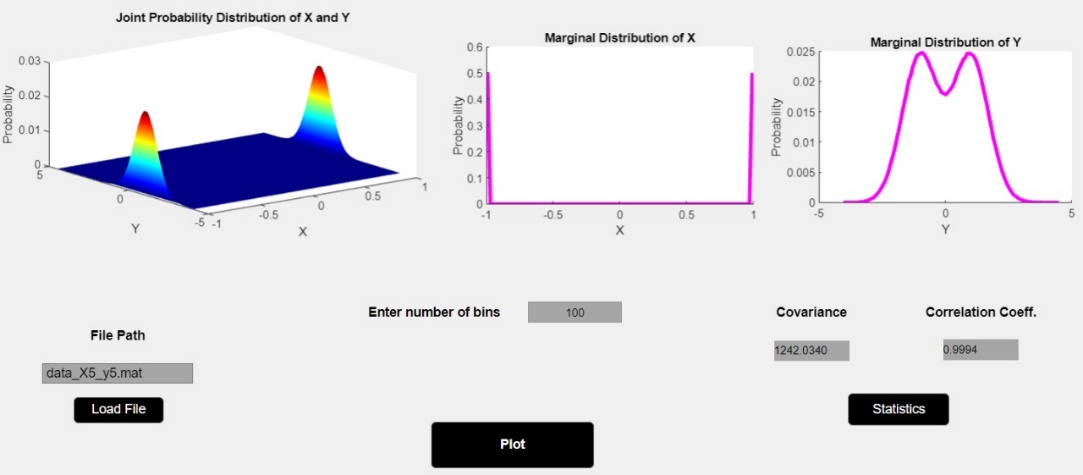
* **A screenshot of a computer

  Description automatically generatedX4**

The hand analysis for this gave us the following results:

* Covariance:
* Corr. Coeff.:

So the output matches these results with small deviations.

* **X5**

The hand analysis for this gave us the following results:

* Covariance:
* Corr. Coeff.:

So the output matches these results with small deviations. However, in the covariance calculations it deviated too much; this could be due to an error in sampling this particular set of samples.

* **Given sampleA screenshot of a computer

  Description automatically generated**

Section 3

**1. Introduction**

This report describes the design and functionality of a MATLAB GUI-based application created for visualizing joint and marginal probability distributions. The app allows users to load sample data, compute the joint probability distribution of two variables (Z and W), and display both the joint distribution and its marginal distributions.

The application is built using MATLAB App Designer, which enables the creation of interactive UI components. The user can define mathematical expressions for Z and W, and specify the number of bins for histograms. The app provides plots for the joint and marginal distributions and uses the .mat file format for input data.

**2. Application Overview**

The application allows users to:

* Load a .mat file containing data.
* Enter mathematical expressions to define Z and W using the input variables X and Y.
* Specify the number of bins for histograms.
* View the joint probability distribution of Z and W as a 3D surface plot.
* View the marginal distributions of Z and W as interpolated histograms.

The app consists of several components:

* **File Selection:** The user can load a .mat file using a file picker dialog.
* **Input Fields:** Users can enter mathematical expressions for Z and W, and specify the number of bins for histograms.
* **Plotting:** The app generates the joint and marginal probability distributions and displays them on separate axes.

**3. Components and Functions**

**3.1 UI Elements**

The following UI components are created in the app:

* **UI Figure:** The main window of the application.
* **UI Axes (3 instances):** Display the joint probability distribution and marginal distributions.
* **Edit Fields:**
  + File Path Edit Field: Allows the user to display the loaded file path.
  + Zex2X1EditField: Input for the mathematical expression of Z.
  + Wex23YEditField: Input for the mathematical expression of W.
  + Enter number of bins Edit Field: Input for the number of bins for histograms.
* **Buttons:**
  + Load File Button: Allows the user to load a .mat file containing the sample data.
  + Plot Button: Triggers the plotting of joint and marginal distributions.

**3.2 Core Functions**

The core functions of the app are implemented as callback methods in the Section3 class:

**A screenshot of a computer program

Description automatically generated**

**3.2.1 File Loading Function (Load File Button Pushed)**

This function allows the user to load a .mat file. The user can select the file using a file dialog, which sets the filename and loads the sample space (data) from the file into the space property of the app.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

**3.2.2 Plotting Function (Plot Button Pushed)**

This function generates and displays the joint probability distribution and marginal distributions based on user input:

* **Joint Distribution:** A 3D surface plot is generated for the joint distribution of Z and W.
* **Marginal Distributions:** Interpolated histograms of Z and W are plotted separately.

First, the app extracts the data for X and Y and evaluates the mathematical expressions for Z and W entered by the user.

**4. Error Handling and Validation**

The app includes several validation steps to ensure that the user input is valid:

* **Number of Bins Validation:** The number of bins must be a positive number. If the user enters an invalid number, the app will display a message prompting them to enter a valid value.
* **Mathematical Expression Validation:** The app uses the eval function to evaluate the expressions for Z and W. If the expressions are invalid or cause an error, the app will display an error message.

**5. User Interface Layout**

The layout of the app is designed to be user-friendly, with clear labels and inputs for easy navigation:

* The input fields for Z, W, and the number of bins are placed centrally for easy access.
* The axes for plotting the distributions are arranged around the input fields.
* Buttons for loading the file and generating the plots are clearly labeled and placed for convenient access.

**Output**

* **A screenshot of a computer

  Description automatically generatedX2**
* **X3A screenshot of a computer

  Description automatically generated**
* **X4**

A screenshot of a computer

Description automatically generated

* **X5**

A screenshot of a computer

Description automatically generated

**Given sample**A screenshot of a computer

Description automatically generated

The outputs of section 3 are the same as section 2, but scaled by the values multiplied by X and Y.