Data Visualization & Dashboards in Agriculture

ABE 6933 – Special Topics in Agricultural and Biological Engineering

Hands-On Group Exercise: EDA, Visualization, and Project Management with Git/GitHub

Objective:
Students will collaboratively explore a dataset using Git/GitHub for version control and complete specific EDA tasks. By the end of the exercise, each student will contribute to the analysis, learn GitHub collaboration techniques, and generate meaningful insights through data visualization.

Part 1: Git/GitHub Setup

Step 1: Initializing the Git Repository (to be done by one group member)

1. **Choose a team leader** who will initialize the project.
2. **Download the provided dataset** and place it in a folder for the project.
    a) **Data Elements:**
        i. **Year**: from 2010 to 2020.
        ii. **Average_Temperature**: Average annual temperature (°C).
        iii. **Total_Rainfall**: Total annual rainfall (mm).
        iv. **Wheat_Yield**: Wheat yield (tons per hectare).
        v. **Corn_Yield**: Corn yield (tons per hectare).
3. Initialize a new Git repository in that folder:

```
git init
```

4. Add the dataset file to the repository:

```
git add <dataset_filename.csv>
```

5. Create an initial commit:

```
git commit -m "Initial commit with dataset"
```

6. Create a new repository on GitHub and push the local repository to GitHub:

```
git remote add origin <repository_URL>
git push -u origin main
```

Step 2: Collaborating via GitHub (all group members)

1. Each group member must **fork the repository** from the team leader’s GitHub.
2. Clone the forked repository locally:

```
git clone <your_forked_repository_URL>
```

3. Each member creates their own branch to work on specific parts of the EDA:

```
git checkout -b <branch_name>
```

4. After completing the assigned task (Part 2, below), commit and push your changes to GitHub:

```
git add .
git commit -m "Completed task <describe task>"
git push origin <branch_name>
```

5. Open a **Pull Request (PR)** on GitHub and assign the team leader for review.
6. The **team leader** reviews and merges the PRs into the main branch.

Part 2: Exploratory Data Analysis (EDA)

After setting up the Git repository, each student will focus on a specific EDA task. Here's the breakdown of tasks for each member.

Task Assignments (to be divided among group members):

Student 1: Descriptive Statistics & Summary

- **Objective**: Perform summary statistics for the key variables in the dataset.

1. Use R to calculate:
   - Mean, median, mode, standard deviation, min, and max for:
     - Temperature
     - Rainfall
     - Wheat yield
     - Corn yield
2. Create a script (e.g., summary_stats.r) to output these statistics to the console.
3. Add and commit the script and results to your branch:

```
git add summary_stats.r
git commit -m "Added summary statistics script"
git push origin <branch_name>
```

4. Open a **Pull Request** and wait for the team leader to merge it.

Student 2: Data Distribution & Outliers

- **Objective**: Analyze data distribution and detect outliers.

1. Create histograms for:
    - Average temperature
    - Total rainfall
    - Wheat yield
    - Corn yield
2. Generate boxplots to identify outliers in the same variables.
3. Save the script (e.g., data_distribution.r) and export the graphs' plots in individual files.
4. Add and commit the script and graphs to your branch:

```
git add data_distribution.r <graphs>
git commit -m "Added histograms and boxplots"
git push origin <branch_name>
```

5. Open a **Pull Request** for review and merge.

Student 3: Correlation Analysis

- **Objective**: Investigate the correlation between climate variables and crop yield.

1. Create a correlation matrix between:
    - Temperature, Rainfall, Wheat Yield, Corn Yield.
2. Visualize the correlation matrix using a heatmap (or similar) to illustrate relationships.
3. Save the script (e.g., correlation_analysis.r) and generate a heatmap PNG file.
4. Add and commit the script and graph to your branch:

```
git add correlation_analysis.r <correlation_matrix_image>
git commit -m "Added correlation matrix and heatmap"
git push origin <branch_name>
```

5. Open a **Pull Request** for review and merge.

Student 4: Time Series Analysis (Optional, if groups of 4)

- **Objective**: Explore the time series trends in the data.

1. Create line plots showing the trends over time (2010-2020) for:
    - Temperature
    - Rainfall
    - Wheat yield
    - Corn yield
2. Save the script (e.g., time_series_analysis.r) and export the plots to PNG files.
3. Add and commit the script and line plots to your branch:

```
git add time_series_analysis.r <line_plots>
git commit -m "Added time series analysis"
git push origin <branch_name>
```

4. Open a **Pull Request** for review and merge.

Part 3: Pull Request Merging and Final Review

1. The **team leader** reviews the Pull Requests from each team member, ensures there are no conflicts, and merges them into the main branch.
2. Once all branches are merged, the entire group should review the final project and verify that all tasks have been completed successfully.
3. **Update the README.md file** to document:
    - The purpose of the project
    - Brief descriptions of each EDA task
    - How to run the code
    - Instructions for collaborators on how to contribute
4. All students must Pull from upstream (team leader repository) to update their own local copy and then push the new code to their origin.

Part 4: Submission

Each group member will submit the link to the **GitHub repository** containing the following:

- The final EDA scripts (merged into the main branch).
- **All graphs and outputs (stored in a folder like outputs/).**
- A detailed README.md file.

Reflection Questions:

1. How did Git/GitHub help manage collaboration within the group?
2. What challenges did you face while working collaboratively using Git/GitHub, and how did you solve them?
3. How did dividing the EDA tasks improve the efficiency of the project?