Software Engineering Lab 02: Python Setup and Microgrid Resilience Analysis

Part 1: Setting up Python and Jupyter Lab

1.1 Install Python

- **Windows:**
    1. Visit python.org and download the latest version of Python for Windows.
    2. Run the installer and ensure that you check the box "Add Python to PATH".
    3. After installation, open Command Prompt and type `python --version` to check if Python was installed successfully.
- **Mac OSX:**
    1. Open the Terminal and check if Python is already installed using the command: `python3 --version`.
    2. If Python is not installed, visit python.org and download the latest version for Mac.
    3. Run the installer and follow the instructions. After installation, type `python3 --version` in the Terminal to verify.

1.2 Install Jupyter Lab and Pandas

- **About pip:**
    1. `pip` is a package manager for Python that allows you to install, update, and manage software packages easily.
- **Installing Jupyter Lab and Pandas:**
    1. Open your Command Prompt (Windows) or Terminal (Mac).
    2. Install Jupyter Lab and Pandas using the following command:

        ```
        pip install jupyterlab pandas
        ```

    3. Once the installation is complete, you can verify Jupyter Lab by running: `jupyter-lab`

        This will open Jupyter Lab in your default web browser.

1.3 Testing Your Setup

Create a new Jupyter notebook from Jupyter Lab and execute the following command in a new code cell to verify Pandas:

```
import pandas as pd

pd.__version__
```

Part 2: Creating and Formatting a Jupyter Notebook

1. **Create a Markdown Cell**:
    - Add a new markdown cell at the top of your notebook. Include the title of your lab, a brief description, and use the following markdown features:
        - **Bullets**
        - **Bold**
        - *Italic*

Example Markdown Syntax:

```
# Lab 01: Microgrid Resilience Analysis

This lab investigates the resilience of microgrids using power data.

**Objectives**:

- Learn about Python and Jupyter Lab

- Use Pandas to manipulate data

- Analyze microgrid power data


*Resilience* is **critical** for microgrids operating in islanded mode.
```

Part 3: Analyzing Microgrid Power Data

1. **Load Data**:
    - A folder `data/` contains power data files:
        - `constant_14days.csv`
        - `workweek_14days.csv`
        - `power_load_data.csv`

    Use Pandas to load one of these CSV files into a DataFrame:

    ```
    import pandas as pd

    df = pd.read_csv('data/constant_14days.csv')
    ```

2. **Plot Power Data**:
    - Use the `plot_power_load_df` method from the provided `power_loads.py` module to visualize the data:

        ```
        from power_loads import plot_power_load_df
        ```

```
plot_power_load_df(df)
```

3. **Generate Statistics**:
    - Use ChatGPT to help you generate Python code to display statistics for your data in a table.
4. **Refer to Example Notebook**:
    - Check the `help/` folder for the `power_loads.ipynb` notebook to see examples of how to use the provided module.

Part 4: Submitting Your Work

1. **Save Your Notebook**:
    - Save your notebook in the course GitHub repository under the `labs/lab_01/` folder.
    - Name your file: `<lastname_power>.ipynb`.
2. **Commit and Push to GitHub**:
    - Commit your work to the `labs` branch periodically using GitHub Desktop or from the command line:

```
git add labs/lab_01/<lastname>_power.ipynb
git commit -m "Lab 01 submission"
git push origin labs
```

References

- **Python**: A high-level programming language used for general-purpose programming.
- **Jupyter Lab**: An interactive development environment for notebooks, code, and data.
- **Pandas**: A Python library providing data structures and functions for data analysis.
- **pip**: A package installer for Python, used to install libraries and dependencies.

Markdown Example:

```
# My Lab Title

This is a description of my lab.

**Key Points**:

- First bullet

- Second bullet

*Italic text* and **Bold text** are supported in markdown.
```