Quick Guide: How to Use These Scripts

This guide explains **how to run, modify, and automate** the scripts efficiently.

📌 1. Setting Up Your Environment

Before running any script, ensure you have **Python** and required libraries installed.

🔧 Install Required Libraries

Open your terminal or command prompt and run:

```
pip install pandas numpy requests beautifulsoup4 schedule matplotlib seaborn flask dash plotly sklearn pdfplumber openpyxl
```

This installs all the dependencies required for the scripts.

📌 2. Running a Script

Each script is a **standalone Python file** that you can run individually.

🔹 Example: Running a Script

If you want to remove duplicates from a dataset:

1. Save the script as `remove_duplicates.py`
2. Open **Terminal** or **Command Prompt**

Run the script using:

```
python remove_duplicates.py
```

3.

📌 Running Scripts with Input Files

Some scripts need **CSV, JSON, or Excel** files as input.

Example: Running `export_data.py` to save a DataFrame as JSON

```
python export_data.py --file sample_data.csv --format json
```

📌 3. Modifying Scripts for Your Needs

Most scripts are designed to be easily **customized**.

🔹 **Example: Change Method for Handling Missing Values**

In `handle_missing_values.py`, modify:

```
df = handle_missing_values(df, method='median')  # Change to 'mode' if needed
```

🔹 **Example: Change API URL for Data Extraction**

In `fetch_api_data.py`, modify:

```
url = "https://api.example.com/data"  # Replace with your API
```

📌 4. Automating Scripts with Task Scheduler

You can **schedule** scripts to run **automatically**.

🔹 **Windows: Using Task Scheduler**

1. Open **Task Scheduler**
2. Create a **New Basic Task**

Set the action to:

```
python C:\path\to\your_script.py
```

3.
4. Choose **Daily, Hourly, or Custom Interval**
5. Click **Finish**

🔹 **Mac/Linux: Using Cron Jobs**

Open Terminal and type:

```
crontab -e
```

1.

Add a job to run a script every hour:

```
0 * * * * python /path/to/your_script.py
```

2.

📌 5. Running a Live Dashboard

Scripts like **Live Data Monitoring Dashboard** require a web server.

🔹 **Start the Dashboard**

```
python live_data_dashboard.py
```

Then, open **http://127.0.0.1:8050/** in your browser.

📌 6. Best Practices for Using These Scripts

✅ **Keep Data Backups** before running any transformation scripts.
✅ **Modify Parameters** to fit your dataset and business needs.
✅ **Use Virtual Environments** (`venv` or `conda`) to avoid conflicts.
✅ **Automate** using `schedule` or `cron` for continuous data processing.
✅ **Test with Sample Data** before applying scripts to large datasets.

📌 7. Common Issues & Troubleshooting

Issue	Possible Fix
`ModuleNotFoundError: No module named X`	Run `pip install X`
`FileNotFoundError`	Check file path and permissions
`API request failed`	Ensure correct API key and endpoint
`Dash app not starting`	Change port `app.run_server(port=8080)`

Final Thoughts

- These scripts provide a **ready-to-use automation framework**.
- You can **modify, integrate, and schedule** them for **real-world tasks**.
- Keep **learning & experimenting** to improve your workflow!