Supplementary Wildfire Analysis Training Manual

# Supplementary Wildfire Analysis Training Manual (Extended Challenges & Learnings)

\*\*Purpose:\*\* This document expands on the original training manual by capturing additional key challenges, workaround strategies, troubleshooting steps, and learning experiences that occurred throughout your wildfire analysis journey.

---

# 🔁 File Handling & Notebook Launch Challenges

## ❌ Problem:

Opening the `.ipynb` file via Anaconda Navigator or from a directory other than the data folder resulted in path errors (e.g., missing shapefiles).

## ✅ Fix:

You learned that Jupyter notebooks must be opened \*\*directly from the data folder\*\* (via right-click > Open With > Jupyter Notebook). This maintains relative path references.

## 🔄 Learning:

- Jupyter's working directory defaults to the launch path.

- Opening from an unrelated location results in `FileNotFoundError` when using `gpd.read\_file()` or raster functions.

## 🧠 Tip:

Always verify with `os.getcwd()` or check `!ls` in the notebook to confirm file visibility.

---

# 🧭 Running Cells: Run All vs. Sequential Execution

## ⚠️ Misstep:

After running `Restart & Run All`, downstream cells failed because earlier errors blocked valid variable definitions.

## 🧪 Adjustment:

Instead of re-running everything after an error, you manually executed each cell \*\*sequentially\*\* post-fix.

## 📌 Key Insight:

- Errors high in the cell order block outputs later on.

- If restarting the kernel, rerun everything only if all errors have been corrected.

---

# 📎 File Extension Issues & Default App Behavior

## Confusion:

Windows did not recognize `.ipynb` files with Jupyter Notebook.

## ✅ Resolution:

You selected `jupyter-notebook.exe` from Anaconda Scripts, and manually checked \*\*"Always use this app"\*\*.

## 📌 Best Practice:

Once associated, notebooks will open consistently with correct environment context.

---

# 🧼 Distance Errors & Geometry Fixes

## 🔍 Observation:

Distances showed a value of `1.797693e+305` — a floating-point max value, indicating a calculation problem.

## Root Cause:

- Lat/lon data was used without projecting to a local CRS.

- Some geometries were \*\*invalid\*\*, causing distance methods to return junk values.

## 🔧 Fix Strategy:

- Use `.is\_valid.value\_counts()` to identify and filter invalid geometries.

- Reproject all data to EPSG:32611 (UTM Zone 11N).

fire\_centroids = fire\_centroids[fire\_centroids.is\_valid]  
fire\_centroids\_proj = fire\_centroids.to\_crs('EPSG:32611')

# 🎨 Scatterplot Rendering Failures

## Error:

## 🔍 Root Cause:

## ✅ Fix:

# 🧪 Statistical Correlation (Pearson r) Returns NaN

## 🔍 Diagnosis:

## ✅ Fix:

corr, p\_value = pearsonr(  
 analysis\_df['nearest\_dist\_km'],  
 analysis\_df['area\_km2']  
)

# 📦 Manual & Output Management

## You Asked:

## ✅ Answer:

# 🔧 Strategy Recap: Countering Common Pitfalls

# ✅ Final Reflection