Execute the following script to submit the Spark job:

- **build_and_run_ingest_roof.sh**

**ingest_roof.py** job will run and achieve the following:

- Check each JSON file is valid, fixes minor issues, and stage copy under **roof_data_fixed/**
- Read files based on expected schema
- Perform appropriate transformations to extract into the following Dataframes:
  - **roof**
  - **building_mounting_plane**
  - **building_mounting_plane_penetration**
  - **building_mounting_plane_penetration_ring_edge**
  - **building_mounting_plane_polygon_exterior_ring**
  - **building_mounting_plane_polygon_interior_ring_edge**
  - **site_model_obstruction**
  - **site_model_obstruction_ring_edge**
- Write to CSV under **output_data/**

Relevant log messages will be available.

- **ingest_roof_yyyy-MM-dd_Hhmmss.log**

The following are my primary learnings:

- Although Spark is good at inferring JSON schemas, this only works when files are consistent. I experimented a lot to ultimately get the data read in correctly.
- Although I wanted to avoid “pure Python” functionality, I did so simply to fix bad/unexpected files. This would be a great opportunity to notify the upstream team to get these issues fixed where possible.

Thought process for choosing this solution:

- I chose Apache Spark as my data processing engine because it performs quickly and is especially helpful for scaling this out for larger sets of data (“linearly scalable”). Even if running on a single node, it will perform better than Pandas, for example.
- I chose to explode out nested JSON into tabular format because it is much simpler to consume by the different teams via SQL and other mechanisms.
- I partially normalized the data but not completely because less joins performs better for analytics and requires simpler queries.