fire_index_mapping_with_fire_boundaries_refined.py

Explanation:

- **Layer Control Box Functionality**
  - For Part 0:
    - Create a base map without setting tiles (to manually add later)
      - m = folium.Map(location=[34.0522, -118.2437], zoom_start=8, tiles=None) #Zoom out from 9 to 8
    - Add CartoDB Positron as a FeatureGroup so it behaves like LA County Boundary
      - base_map_layer = folium.FeatureGroup(name="CartoDB Positron Base Map", control=True, overlay=True)
      - folium.TileLayer("cartodbpositron").add_to(base_map_layer)
      - base_map_layer.add_to(m)
    - Create LA County Boundary FeatureGroup
      - la_county_layer = folium.FeatureGroup(name="Los Angeles County Boundary", control=True, overlay=True)
    - Return the Los Angeles County Bounday and Base Map layers
      - return m, la_county_layer, base_map_layer # Return both layers
  - Part 3
    - Ensure both CartoDB Positron and LA County Boundary are added before other layers
      - la_county_layer.add_to(combined_map)
      - base_map_layer.add_to(combined_map)
    - Create a FeatureGroup and assign it to "Fire Indices"
      - feature_group = folium.FeatureGroup(name=index_combination, control=True, overlay=False).add_to(combined_map)
    - Add the actual layer to the FeatureGroup
      - layer.add_to(feature_group)
    - Add layer to base_layers with the combination as the key
      - base_layers[index_combination] = feature_group # Store FeatureGroup instead of raw layer
    - Add LayerControl to the combined map with only base layers
      - control = folium.LayerControl(collapsed=False, exclusive_groups=["Fire Indices"])
      - combined_map.add_child(control)
- **Popup Functionality**
  - Part 3

- The original code assumed that the values list would always have the same length as the indices list. However, in cases where a particular index might not have a corresponding value for a given data point (because it's None), this would lead to an IndexError.
- The changed code iterates through indices and uses data.get(index_type) to safely access the value for that index. If the value is None, it skips adding that index to the popup content. This makes the popup generation more robust.
- **Using the interactive property of CircleMarker** to address the situation when a click event is captured by the topmost circle, preventing it from propagating down to the underlying circles where the popups are attached.
    - interactive=False for i > 0: We set the interactive property to False for all circles except the outermost one (where i is 0). This makes these inner circles effectively "transparent" to click events, allowing the events to pass through to the outermost circle.
    - interactive=True (default) for i == 0: The outermost circle retains its default interactive=True behavior, so it will capture click events, and its popup will be displayed.
- **Z-index:** the stacking order (z-index) of the circles might be such that a smaller circle is visually on top but its popup is hidden behind a larger circle. Explicitly set the zIndexOffset property of the CircleMarker - zIndexOffset=1000 - i: assign a higher zIndexOffset to the outermost circle (where i is 0) and decreasing values for inner circles. This should ensure that the outermost circle is always on top in terms of z-order. The value 1000 is arbitrary and can be adjusted as needed.