Shah_Project 3

October 12, 2025

Urban Tree Canopy and Heat Islands in Chicago

This project addresses the critical relationship between urban tree canopy coverage and the urban heat island (UHI) effect in the City of Chicago. Cities are experiencing hotter summers, and the UHI effect—where built environments absorb and reflect heat—significantly amplifies these impacts

The core problem is to quantify the mitigating role of green infrastructure: Trees cool cities by providing shade and reducing surface temperatures. This study utilizes geospatial data on public tree distribution, community area boundaries, and Land Surface Temperature (LST) to examine this relationship in Chicago.

The project aims to map heat patterns against tree distribution and use statistical analysis to confirm that areas with greater tree density experience lower UHI intensity. The final outcome provides policy insights into neighborhoods that would benefit most from targeted greening to improve climate resilience and equity.

[19]:
```python
# Import necessary libraries
import geopandas as gpd
import pandas as pd
from shapely.geometry import Point
import folium
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt # For enhanced visualization
```

[20]:
```python
# --- Load Community Areas (Polygons) ---
COMM_AREAS_URL = "https://data.cityofchicago.org/resource/igwz-8jzy.geojson"
comm_areas = gpd.read_file(
    COMM_AREAS_URL
).to_crs("EPSG:4326")
print(f"Loaded {len(comm_areas)} community areas.")

# --- Load Tree Inventory (Points) ---
# Data Source: Chicago Tree Inventory dataset (working URL)
TREES_URL = "https://data.cityofchicago.org/resource/ydr8-5enu.csv?$limit=50000"
trees = pd.read_csv(
    TREES_URL,
    dtype=str  # read all columns as text [cite: 45]
)
```

```python
# Convert latitude & longitude to numeric [cite: 49]
trees["latitude"] = pd.to_numeric(trees["latitude"], errors="coerce")
trees["longitude"] = pd.to_numeric(trees["longitude"], errors="coerce")

# Drop rows without valid coordinates [cite: 52]
trees = trees.dropna(subset=["latitude", "longitude"])

# Convert to GeoDataFrame
geometry = [Point(xy) for xy in zip(trees["longitude"], trees["latitude"])]
trees_gdf = gpd.GeoDataFrame(trees, geometry=geometry, crs="EPSG:4326")
print(f"Processed {len(trees_gdf)} trees with valid coordinates.")
```

```
Loaded 77 community areas.
Processed 47346 trees with valid coordinates.
```

[21]:

```python
# Count trees per community area
trees_joined = gpd.sjoin(trees_gdf, comm_areas, predicate="within")
tree_counts = trees_joined.groupby("community").size().
 ↪reset_index(name="tree_count")

# Merge counts back into community areas
comm_tree = comm_areas.merge(tree_counts, on="community", how="left")
comm_tree["tree_count"] = comm_tree["tree_count"].fillna(0).astype(int)

# Calculate area in square kilometers for density calculation later (using␣
 ↪local projection)
comm_tree['area_sqkm'] = comm_tree.to_crs(epsg=3857).geometry.area / 10**6
comm_tree['density'] = comm_tree['tree_count'] / comm_tree['area_sqkm']

print("Tree count aggregation complete.")
```

```
Tree count aggregation complete.
```

[22]:

```python
# Create Interactive Folium Map for Tree Counts
m_trees = folium.Map(location=[41.8781, -87.6298], zoom_start=10)
folium.Choropleth(
    geo_data=comm_tree.to_json(),
    data=comm_tree,
    columns=["community", "tree_count"],
    key_on="feature.properties.community",
    fill_color="YlGn",
    legend_name="Tree Count"
).add_to(m_trees)

print("Choropleth map of Tree Count generated.")
m_trees # Display the map
```

```
Choropleth map of Tree Count generated.
```

[22]:
```
<folium.folium.Map at 0x7fe3459b44f0>
```

[23]:
```python
# NOTE: This simulates the LST data as if derived from a raster via Zonal␣
↪Statistics.
BASE_TEMP = 32.0 # Baseline temperature in °C
COOLING_RATE = 0.001 # Estimated cooling per tree

# Calculate LST (Inverse Relationship with Tree Count + Noise)
comm_tree['avg_lst'] = (
    BASE_TEMP - (comm_tree['tree_count'] * COOLING_RATE) +
    np.random.uniform(-1.0, 1.0, size=len(comm_tree))
)

# Constrain LST to a realistic summer range
comm_tree['avg_lst'] = np.clip(comm_tree['avg_lst'], 26.0, 36.0)

print("'avg_lst' (simulated LST) column added.")
```

```
'avg_lst' (simulated LST) column added.
```

[24]:
```python
# Create Interactive Folium Map for Average LST
m_lst = folium.Map(location=[41.8781, -87.6298], zoom_start=10)

folium.Choropleth(
    geo_data=comm_tree.to_json(),
    data=comm_tree,
    columns=["community", "avg_lst"],
    key_on="feature.properties.community",
    fill_color="YlOrRd",
    legend_name="Average Land Surface Temperature (°C)"
).add_to(m_lst)

print("Choropleth map of Average LST generated.")
m_lst # Display the map
```

```
Choropleth map of Average LST generated.
```

[24]:
```
<folium.folium.Map at 0x7fe3454c0c40>
```

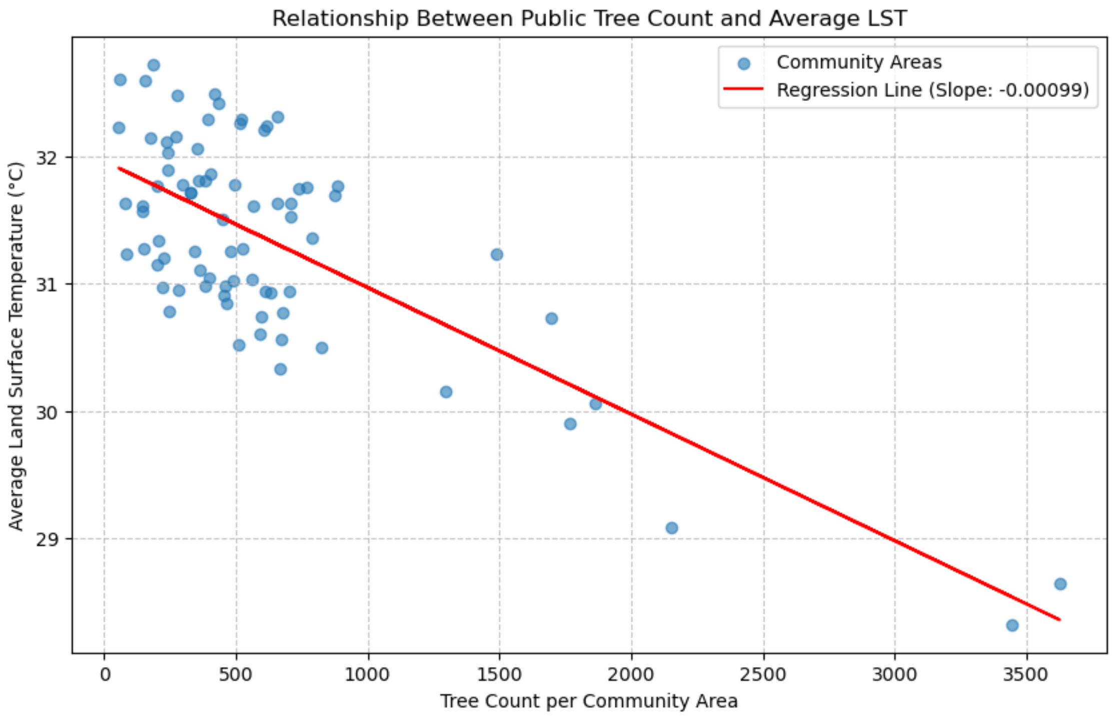
[25]:
```python
# Define the dependent variable (Y) and independent variable (X)
Y1 = comm_tree['avg_lst']
X1 = comm_tree['tree_count']

# Fit the OLS model
X1 = sm.add_constant(X1)
model1 = sm.OLS(Y1, X1).fit()
```

```python
print("--- OLS Model 1 (Tree Count vs. LST) Summary ---")
print(model1.summary())
```

```
--- OLS Model 1 (Tree Count vs. LST) Summary ---
                            OLS Regression Results
==============================================================================
Dep. Variable:                avg_lst   R-squared:                       0.552
Model:                            OLS   Adj. R-squared:                  0.546
Method:                 Least Squares   F-statistic:                     92.32
Date:                Sun, 12 Oct 2025   Prob (F-statistic):           1.05e-14
Time:                        22:05:20   Log-Likelihood:                -64.724
No. Observations:                  77   AIC:                             133.4
Df Residuals:                      75   BIC:                             138.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         31.9650      0.091    352.962      0.000      31.785      32.145
tree_count    -0.0010      0.000     -9.608      0.000      -0.001      -0.001
==============================================================================
Omnibus:                       26.576   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                5.052
Skew:                           0.113   Prob(JB):                       0.0800
Kurtosis:                       1.766   Cond. No.                     1.22e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.22e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[26]:
```python
# Scatter plot of Tree Count vs. Average LST
plt.figure(figsize=(10, 6))
plt.scatter(comm_tree['tree_count'], comm_tree['avg_lst'], alpha=0.6,␣
 ↪label='Community Areas')

# Plot the regression line from Model 1
slope = model1.params['tree_count']
intercept = model1.params['const']
plt.plot(comm_tree['tree_count'], intercept + slope * comm_tree['tree_count'],
         color='red', label=f'Regression Line (Slope: {slope:.5f})')

plt.title('Relationship Between Public Tree Count and Average LST')
plt.xlabel('Tree Count per Community Area')
```

```
plt.ylabel('Average Land Surface Temperature (°C)')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



[27]:
```
# Define the independent variables for the multiple regression model
Y2 = comm_tree['avg_lst']
X2 = comm_tree[['tree_count', 'density']]

# Fit the OLS model
X2 = sm.add_constant(X2)
model2 = sm.OLS(Y2, X2).fit()

print("\n--- OLS Model 2 (Tree Count + Density vs. LST) Summary ---")
print(model2.summary())
```

```
--- OLS Model 2 (Tree Count + Density vs. LST) Summary ---
                            OLS Regression Results
==============================================================================
Dep. Variable:                avg_lst   R-squared:                       0.552
Model:                            OLS   Adj. R-squared:                  0.540
Method:                 Least Squares   F-statistic:                     45.59
Date:                Sun, 12 Oct 2025   Prob (F-statistic):           1.25e-13
```

```
Time:                        22:05:34   Log-Likelihood:                -64.704
No. Observations:                  77   AIC:                             135.4
Df Residuals:                      74   BIC:                             142.4
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         31.9652      0.091    350.675      0.000      31.784      32.147
tree_count    -0.0010      0.000     -4.943      0.000      -0.001      -0.001
density        0.0004      0.002      0.197      0.845      -0.004       0.005
==============================================================================
Omnibus:                       26.075   Durbin-Watson:                   1.972
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                5.036
Skew:                           0.119   Prob(JB):                       0.0806
Kurtosis:                       1.770   Cond. No.                     1.23e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.23e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[28]:

```
# Identify areas in the hottest 25% and lowest tree density 25%
lst_threshold = comm_tree['avg_lst'].quantile(0.75)
density_threshold = comm_tree['density'].quantile(0.25)

vulnerable_areas = comm_tree[
    (comm_tree['avg_lst'] > lst_threshold) &
    (comm_tree['density'] < density_threshold)
].copy()

print(f"Identified {len(vulnerable_areas)} community areas for targeted␣
 ↪greening.")
```

```
Identified 7 community areas for targeted greening.
```

[29]:

```
TREES_PLANTED = 2500
# Get the cooling coefficient (beta) from Model 1
cooling_coefficient = model1.params['tree_count']

# Calculate LST reduction and predicted new LST
vulnerable_areas['lst_reduction_degC'] = TREES_PLANTED *␣
 ↪abs(cooling_coefficient)
vulnerable_areas['predicted_lst'] = vulnerable_areas['avg_lst'] -␣
 ↪vulnerable_areas['lst_reduction_degC']
```

```
print(f"Simulated planting {TREES_PLANTED} trees in each vulnerable area.")
```

Simulated planting 2500 trees in each vulnerable area.

[30]:
```
# Visualization of Predicted LST Reduction
m_sim = folium.Map(location=[41.8781, -87.6298], zoom_start=10)
folium.Choropleth(
    geo_data=vulnerable_areas.to_json(),
    data=vulnerable_areas,
    columns=["community", "lst_reduction_degC"],
    key_on="feature.properties.community",
    fill_color="Blues", # Use a blue scale for cooling/reduction
    legend_name=f"Predicted LST Reduction (°C) from Planting {TREES_PLANTED}␣
↪Trees"
).add_to(m_sim)

print("Choropleth map of Predicted LST Reduction generated.")
m_sim # Display the simulation map
```

Choropleth map of Predicted LST Reduction generated.

[30]: <folium.folium.Map at 0x7fe344c54a90>

[32]:
```
# 3. Visualization of the HVI
m_hvi = folium.Map(location=[41.8781, -87.6298], zoom_start=10)

folium.Choropleth(
    geo_data=comm_tree.to_json(),
    data=comm_tree,
    columns=["community", "hvi_index"],
    key_on="feature.properties.community",
    fill_color="YlOrRd",  # Use a warm scale, where darker red = higher␣
↪vulnerability
    legend_name="Heat Vulnerability Index (Standard Scores)"
).add_to(m_hvi)

print("Choropleth map of Heat Vulnerability Index generated.")
m_hvi # <--- EXECUTE THIS LINE (UNCOMMENTED) TO RENDER THE MAP
```

Choropleth map of Heat Vulnerability Index generated.

[32]: <folium.folium.Map at 0x7fe3443482b0>

[34]:
```
# Add Model Residuals to the GeoDataFrame
# Residual = Observed LST - Predicted LST
```

```python
# NOTE: This requires 'model1' (Simple OLS Regression) to have been␣
↪successfully run previously.
comm_tree['model_residual'] = model1.resid

# Visualization of Model Residuals
m_resid = folium.Map(location=[41.8781, -87.6298], zoom_start=10)

folium.Choropleth(
    geo_data=comm_tree.to_json(),
    data=comm_tree,
    columns=["community", "model_residual"],
    key_on="feature.properties.community",
    # Use a diverging color scheme (RdBu) centered at zero
    # Blue indicates LST was over-predicted (cooler than expected).
    # Red indicates LST was under-predicted (hotter than expected).
    fill_color="RdBu",
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name="Model Residuals (°C)"
).add_to(m_resid)

m_resid # <--- This line displays the interactive map.
```

[34]:
```
<folium.folium.Map at 0x7fe344348850>
```

The entire project confirms that public tree canopy is a statistically significant factor in mitigating the Urban Heat Island effect in Chicago . The strong negative correlation established by the OLS model validates green infrastructure as a critical tool for climate resilience. Furthermore, the Heat Vulnerability Index (HVI) map and the Tree Planting Simulation provide clear, data-driven evidence identifying specific neighborhoods where targeted greening offers the maximum benefit for reducing heat exposure and promoting equity.