

## **Final Report**

Title: Measuring the Distribution of Air Quality Hazards Across Minneapolis

Notice: Dr. Bryan Runck

Author: Rob Hendrickson

Date: 12/21/2022

Project Repository:

[https://github.com/RwHendrickson/GIS5571/blob/main/Final\\_Project](https://github.com/RwHendrickson/GIS5571/blob/main/Final_Project)

Time Spent: 70+ hours

The EPA defines environmental justice as:

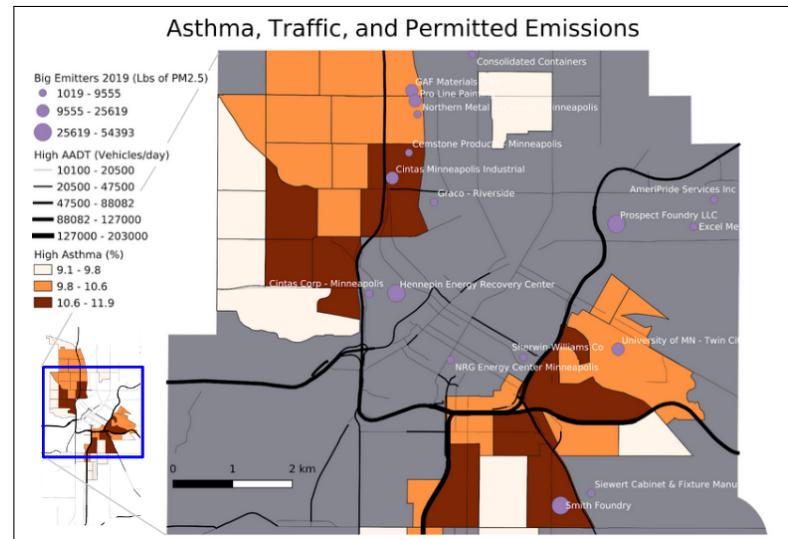
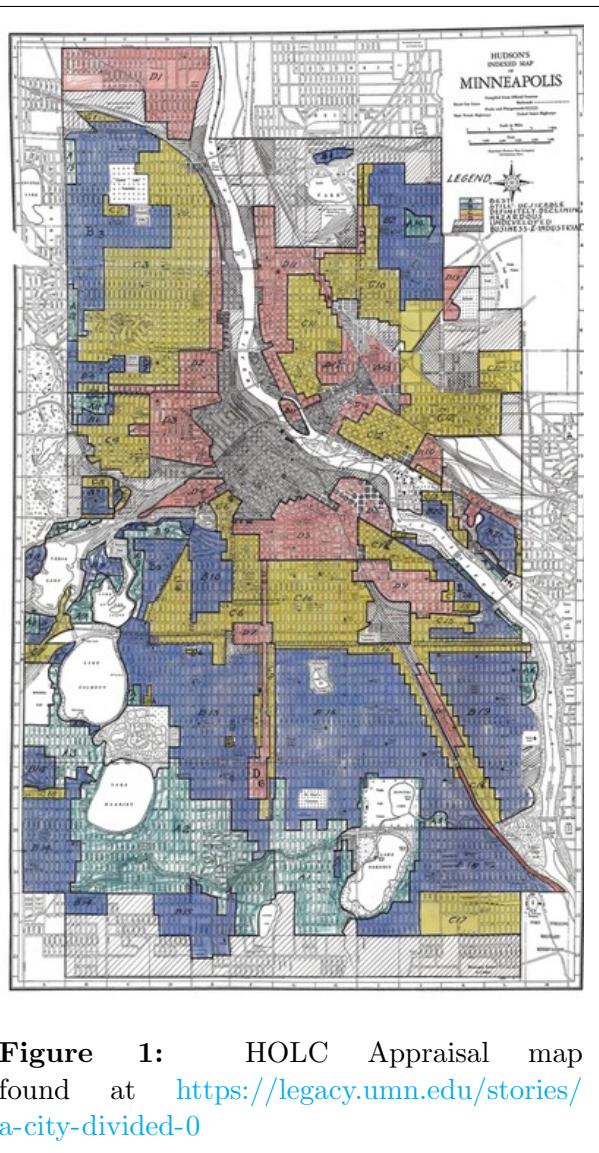
“...the fair treatment and meaningful involvement of all people regardless of race, color, national origin, or income with respect to the development, implementation and enforcement of environmental laws, regulations and policies. Fair treatment means no group of people should bear a disproportionate share of the negative environmental consequences resulting from industrial, governmental and commercial operations or policies.” [1]

## **Abstract**

It is understood that some parts of Minneapolis experience a greater burden of environmental hazard than others. Anecdotally and visually, this can be correlated to restrictive housing practices of the early to mid 20th century. This project aims to quantify the cumulative environmental harms across Minneapolis at a fine spatial resolution with the intention of spatially correlating this with modern demographics and restrictive housing practices.

# 1 Problem Statement

From the colonization of the sacred Bdote Region of the Dakota people to the first racially restrictive covenant in 1910 to the "redlining" of the Home Owners' Loan Corporation (HOLC) appraisal maps of the 1930's (see figure 1), Minneapolis has a long environmental justice history that is rooted in forcing communities of color and non-Christians into the more industrial parts of town. Though the Fair Housing Act of 1968 ended these restrictive housing practices, they continue to shape the distribution of peoples across our city today.



**Figure 2:** Contextual information on air pollution in Minneapolis. Health data from [2].

How do we spatially measure this environmental harm? There are a plethora of factors involved in measuring environmental risk including extreme temperature exposure, tree canopy, lead in soil, water, and paint, and access to natural spaces. This project will focus on air quality, specifically particulate matter 2.5 (PM2.5), though there are a number of other pollutants of concern such as volatile organic compounds (VOCs), sulfur oxides (SO<sub>2</sub>), and nitrogen oxides (NO<sub>x</sub>) [3].

PM2.5 are generic particles of 2.5 microns (micrometers) in diameter that are created during the combustion processes of cars, energy production, manufacturing, and trash incineration. The American Heart Association has established a causal link between these particles and heart and lung disease [3]. These impacts manifest themselves in missed days of school and work, hospitalizations, and premature mortality [4].

Two sources of PM2.5 exposure were considered as factors in determining risk. Current Annual Average Daily Traffic ( $T_{2022}$ ) and current industrial emissions ( $I_{2020}$ ). These and other project requirements can be found in table 1.

	Requirement	Defined As	Data (Spatial)	Data (Attribute)	Dataset	Preparation
1	Data Acquisition	Request municipal boundaries (CTUs), traffic, industrial emissions, observed air quality, and HOLC data	Polygons, polylines, points	City Name AADT PM2.5 Emissions (lbs) PM2.5 Observed ( $\frac{\mu g}{m^3}$ ) HOLC Classification	Metropolitan Council (CTUs) Minnesota Dept. of Transportation (AADT_MN) Minnesota Pollution Control Agency (Emissions_MN) PurpleAir Air Quality Monitors (PurpleAir) Home Owners' Loan Corporation (HOLC Grades)	Navigated API trees, requested PurpleAir API
2	Process Data	Select Minneapolis, define extent, and clip datasets Query emissions Omit faulty air monitors, average historic PM2.5 observations	Polygons, polylines, points	AADT, PM2.5 Emissions, PM2.5 Observed, HOLC Classification	Cleaned and clipped current traffic, current industrial emissions, annual average observed PM2.5, HOLC Classifications	Explored Data
3	Create Raster Template	Use extent to define a regular grid (the blank canvas)	Rasters		Template Raster	
4	Air Quality Data Interoperability	Make all air quality data interoperable by rasterizing or interpolating onto template raster	Rasters	AADT, PM2.5 Emissions, PM2.5 Observed	Rasterized traffic, Rasterized industrial emissions, Interpolated average observed PM2.5	
5	Emissions Dispersion	Disperse emissions sources	Rasters	AADT Exposure Industrial PM2.5 Exposure	AADT Exposure Surface Industrial Emissions Exposure Surface	Studied convolution & dispersion methodologies
6	Normalize Air Quality Data	Divide values by maximum exposure to get in a comparable scale of 0 to 1	Rasters	Normalized AADT exposure, Industrial PM2.5 Exposure, & Interpolated Average Observed PM2.5	Normalized AADT & Industrial PM2.5 Exposure Surfaces, Normalized Average Observed PM2.5 Surface	Explore distributions of values
7	Simulate Index Surface Creation	Iterate through linear combinations between the rasters with weights that sum up to 1	Raster	Air Quality Hazard Index	Index Surfaces	
8	Validate Index Surfaces	Look at RMSE and spatial distribution of indices' residuals from observed PM2.5		Residuals, RMSE		
9	Sensitivity Analysis	Vary parameters in emissions dispersion. Check resulting surface accuracy and zonal statistics.	Rasters	RMSE	Index Surfaces	
10	Zonal Statistics	Get zonal statistics of the most realistic model by HOLC zones	Polygons	Mean, Minimum, & Maximum Index Values	HOLC Zonal Statistics	

Table 1: Project requirements

## 2 Input Data

The 2010 County, City, and Township Boundaries of the Twin Cities (CTUs) from the Metropolitan Council (MetCouncil) were acquired to define the extent of the study. These polygons included the 'CTU\_NAME' which was used to identify each boundary.

Current Annual Average Daily Traffic segments of Minnesota (AADT\_MN) from the Minnesota Dept. of Transportation (MnDoT) were acquired to model annual traffic exposure. The notable attribute from these polylines was 'CURRENT\_VO' which is the most recent measurement for AADT at each road segment.

Permitted industrial emissions of Minnesota (Emissions\_MN) from Minnesota Pollution Control Agency (MPCA) were acquired to model annual industrial exposure. This CSV included notable attributes: 'YEAR', 'FACILITY\_NAME', 'POLLUTANT', 'EMISSIONS (LB)', 'COUNTY', 'LONGITUDE', and 'LATITUDE'. Each facility had multiple entries for each year and permitted pollutant and the emissions were annual values. It spans from 2006 to 2020.

Particulate matter 2.5 observations from the past year across the study area (contributed by PurpleAir) were acquired to validate the model. These CSVs included the notable attributes 'created\_at' and 'PM2.5\_ATM\_ug/m<sup>3</sup>' which were the date the sensor was created and PM2.5 observed values in micrograms per cubic meter ( $\frac{\text{micrograms}}{\text{meter}^3}$ ), respectively. Furthermore, each CSV's filename had information on their location (latitude/longitude), sensor name, whether the sensor was outside or indoors, and if it was a backup sensor.

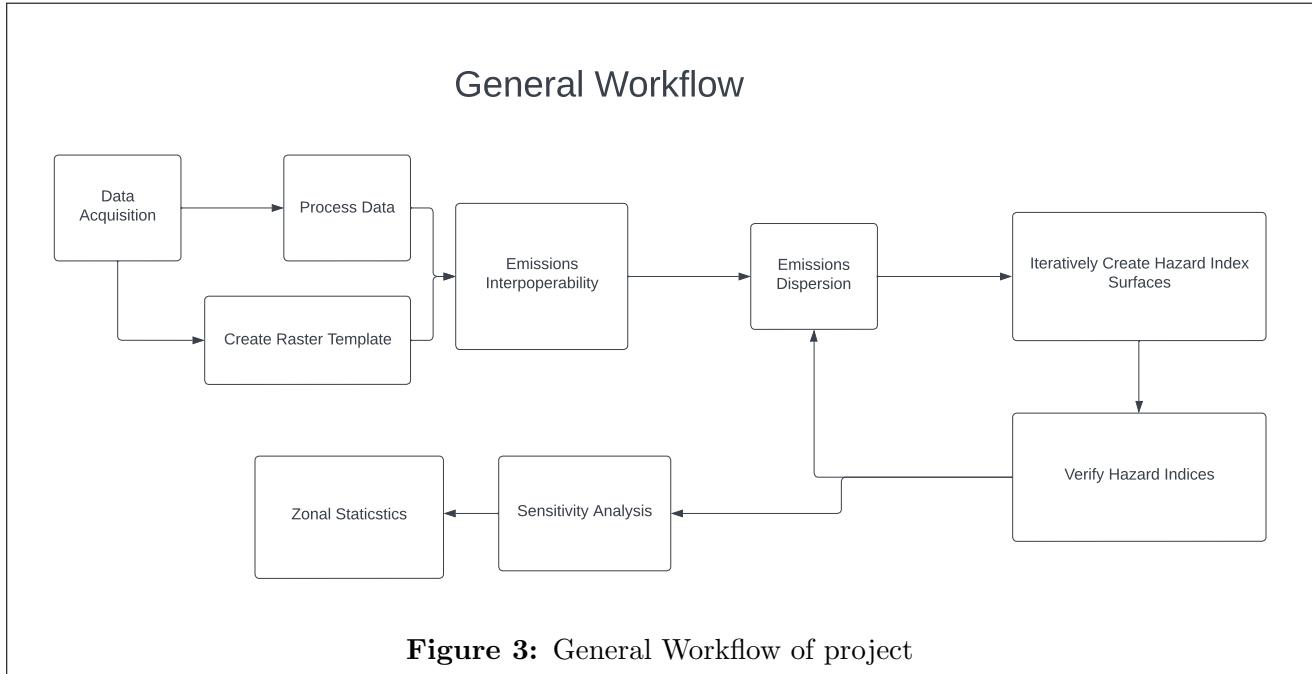
Home Owners' Loan Corporation Grades by parcel of the Twin Cities (HOLC) from MetCouncil were acquired for zonal statistics and to explore correlations between restrictive housing practices and present pollution levels. The notable attribute from these polygons was 'HSG\_SCALE' which correspond to the land classifications and HOLC grades of the 1930's (see [here](#) for more information).

More information on these data sources can be found in table 2.

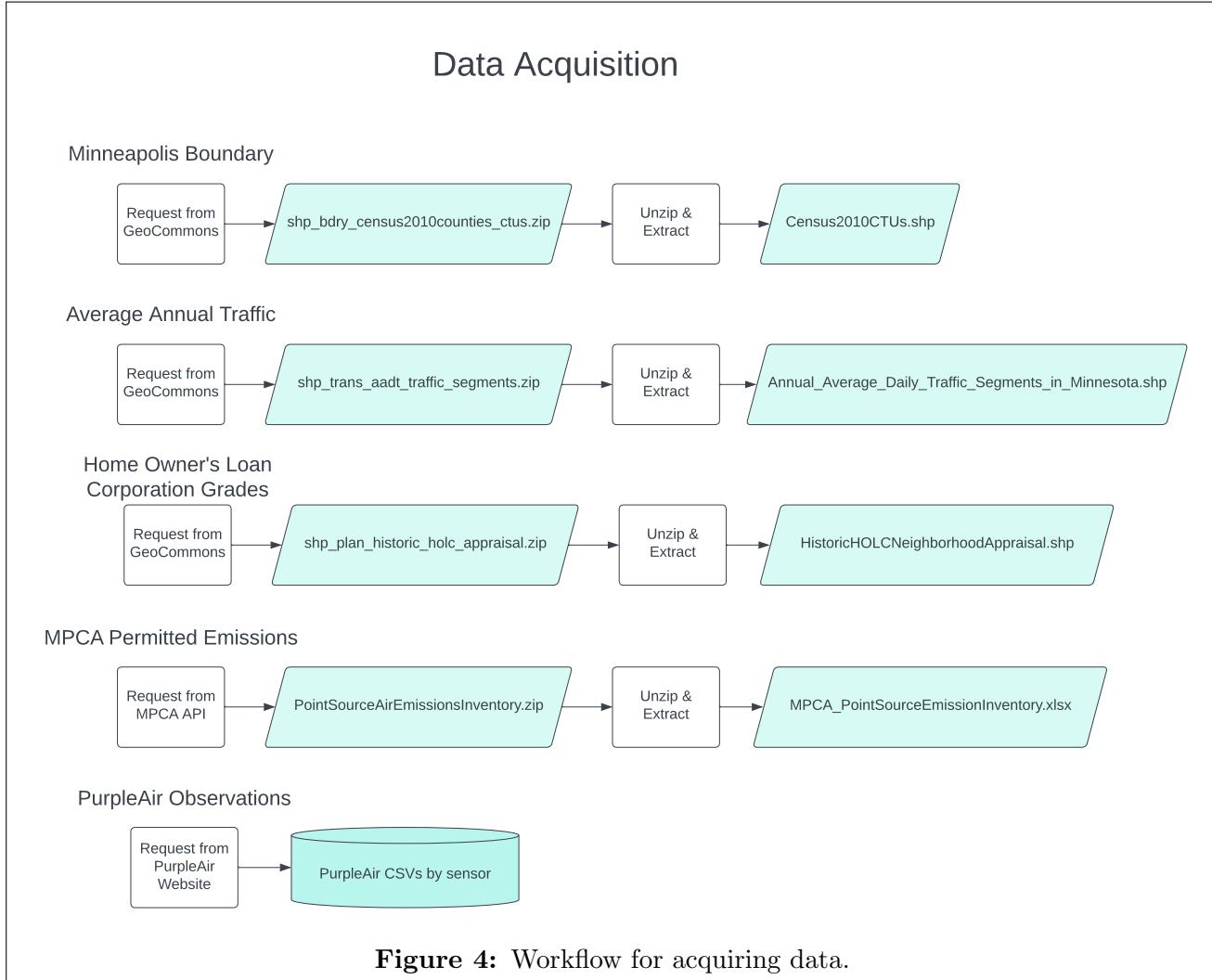
	<b>Title</b>	<b>Purpose in Analysis</b>	<b>Spatial Datatype</b>	<b>CRS</b>	<b>Link to Source</b>
1	MetCouncil's 2010 County, City, & Township Boundaries of the Twin Cities [5]	Defining extent	Polygons	EPSG:26915 (UTM 15N, NAD83)	<a href="https://gisdata.mn.gov/dataset/us-mn-state-metc-bdry-census2010counties-ctus">https://gisdata.mn.gov/dataset/ us-mn-state-metc-bdry-census2010counties-ctus</a>
2	MnDoT's Current AADT Segments [6]	Modeling Hazard Index	Polylines	EPSG:26915 (UTM 15N, NAD83)	<a href="https://gisdata.mn.gov/dataset/trans-aadt-traffic-segments">https://gisdata.mn.gov/dataset/ trans-aadt-traffic-segments</a>
3	MPCA's Permitted Facility Emission [7]	Modeling Hazard Index	Points	EPSG:4326 (WGS84)	<a href="https://www.pca.state.mn.us/air/permitted-facility-air-emissions-data">https://www.pca.state.mn.us/air/ permitted-facility-air-emissions-data</a>
4	PurpleAir Observed Air Quality [8]	Validating Model	Points	EPSG:4326 (WGS84)	<a href="https://api.purpleair.com/">https://api.purpleair.com/</a>
5	HOLC Grades [9]	Zonal Statistics	Polygons	EPSG:26915 (UTM 15N, NAD83)	<a href="https://gisdata.mn.gov/dataset/us-mn-state-metc-plan-historic-holc-appraisal">https://gisdata.mn.gov/dataset/ us-mn-state-metc-plan-historic-holc-appraisal</a>

**Table 2:** Data Sources

### 3 Methods



### 3.1 Data Acquisition



The code that performs these functions can be found in the folder, `1_Data_IO`, at the beginning of each dataset's respective notebook.

#### 3.1.1 Minnesota GeoCommons

The datasets for CTUs, AADT\_MN, and HOLC Grades were acquired from the Minnesota GeoCommons using a function seen in figure 5. These zipfiles each contained a shapefile which can be read into a Geopandas GeoDataFrame using the function, `geopandas.read_file()`.

```

def extract_zip_from_url(url=None):
    '''Extract a zipfile from the internet
    then unpack it in to its own folder
    within the working directory.
    Takes a single url (string).'''

    if type(url) == str: # Single url
        # Create folder name for file
        folder_name = url.split('/')[-1][-4]
        # Make folder for files
        path = os.path.join(os.getcwd(), folder_name)
        if folder_name not in os.listdir():
            os.mkdir(path)
        # Unload zip into the new folder
        response = urllib.request.urlopen(url) # Get a response
        zip_folder = zipfile.ZipFile(BytesIO(response.read())) # Read Response
        zip_folder.extractall(path=path) # Extract files
        zip_folder.close() # Close zip object
    else:
        print('Error Extracting: Invalid Input')

```

**Figure 5:** A function that extracts and unpacks a zipfile from a url.

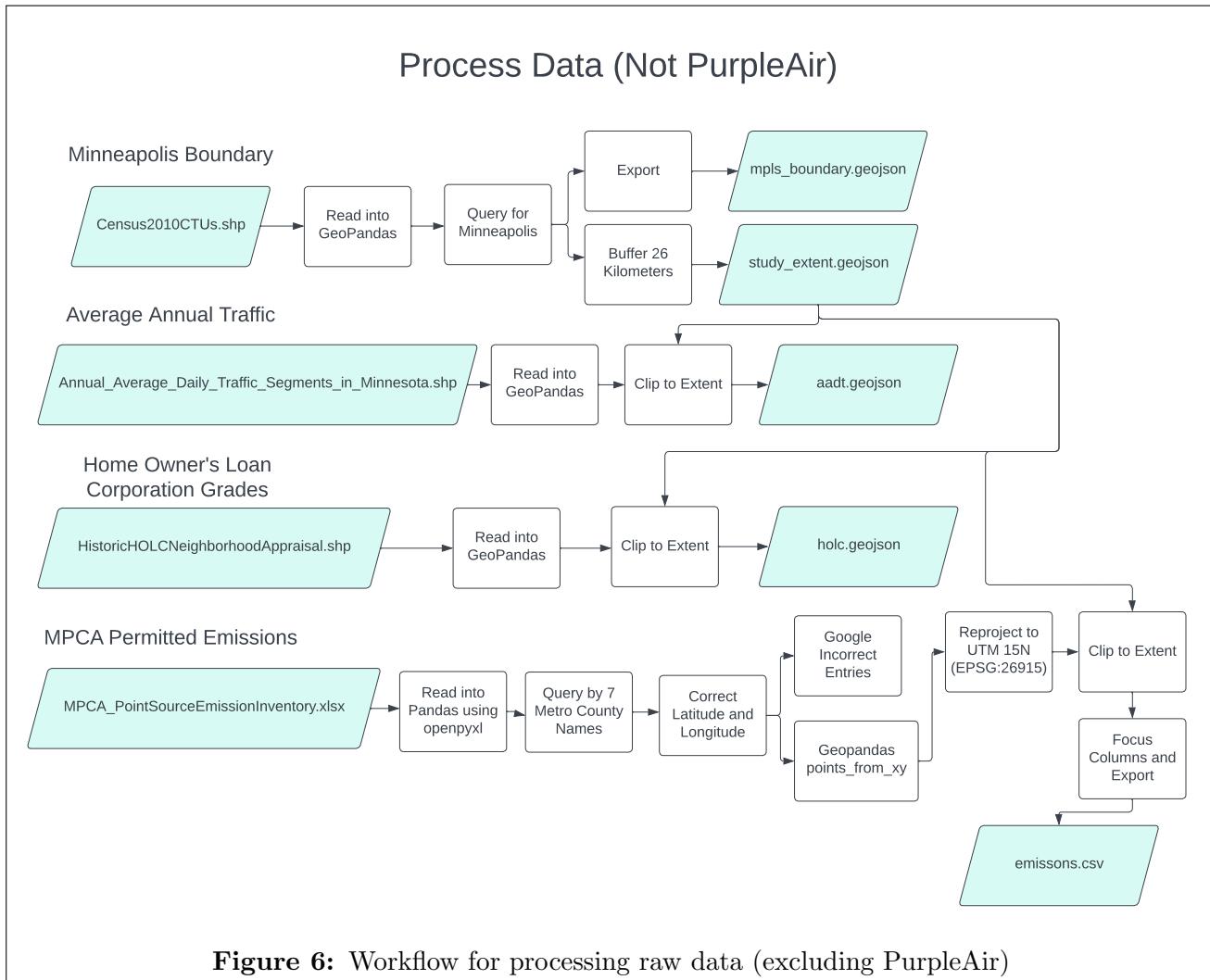
### 3.1.2 Minnesota Pollution Control Agency API

The dataset, Emissions\_MN, was acquired from the MPCA's API. It is a file stored away at this url, '[https://files.pca.state.mn.us/pub/file\\_requests/datasets/Air/PointSourceAirEmissionsInventory.zip](https://files.pca.state.mn.us/pub/file_requests/datasets/Air/PointSourceAirEmissionsInventory.zip)'. This zipfile can be retrieved and unpacked using the function in figure 5. Within this zipfile is an Excel Spreadsheet (.xlsx) which can be read into a Pandas DataFrame with the function, `pandas.read_excel()` and the open source package, `openpyxl`.

### 3.1.3 PurpleAir API

This API was very similar to the Google Places API. An API key must be requested and various fields and variables can be utilized to query for real-time data. However, historical data requires special permissions to request by API. The past year of observations within the study area were instead installed manually using the PurpleAir map and download tool. These 524 individual CSV files with very descriptive names of nearby sensors could each be read into a Pandas DataFrame using, `pandas.read_csv()`.

## 3.2 Data Processing



**Figure 6:** Workflow for processing raw data (excluding PurpleAir)

The code that performs these functions can be found in the folder, `1_Data_IO`, at the end of each dataset's respective notebook.

### 3.2.1 Minneapolis Boundary and Study Extent

Once the CTUs were read into a GeoDataFrame, they could be queried by the field, '`CTU_NAME`'. Minneapolis was selected and buffered 26 kilometers. These shapes were saved as `mpls_boundary.geojson` and `study_extent.geojson`, respectively.

### 3.2.2 Surrounding AADT Segments

Once AADT\_MN was read into a GeoDataFrame, it was clipped using the function, `geopandas.clip(gdf, mask)`, where gdf is AADT\_MN and mask is the study extent read as a GeoDataFrame. These polylines were saved as `aadt.geojson` with all attributes.

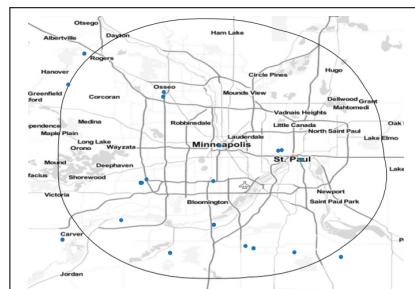
### 3.2.3 HOLC Grades by Parcel

HOLC Grades by Parcel were in the correct CRS and within the extent.

### 3.2.4 Surrounding MPCA Permitted Emissions

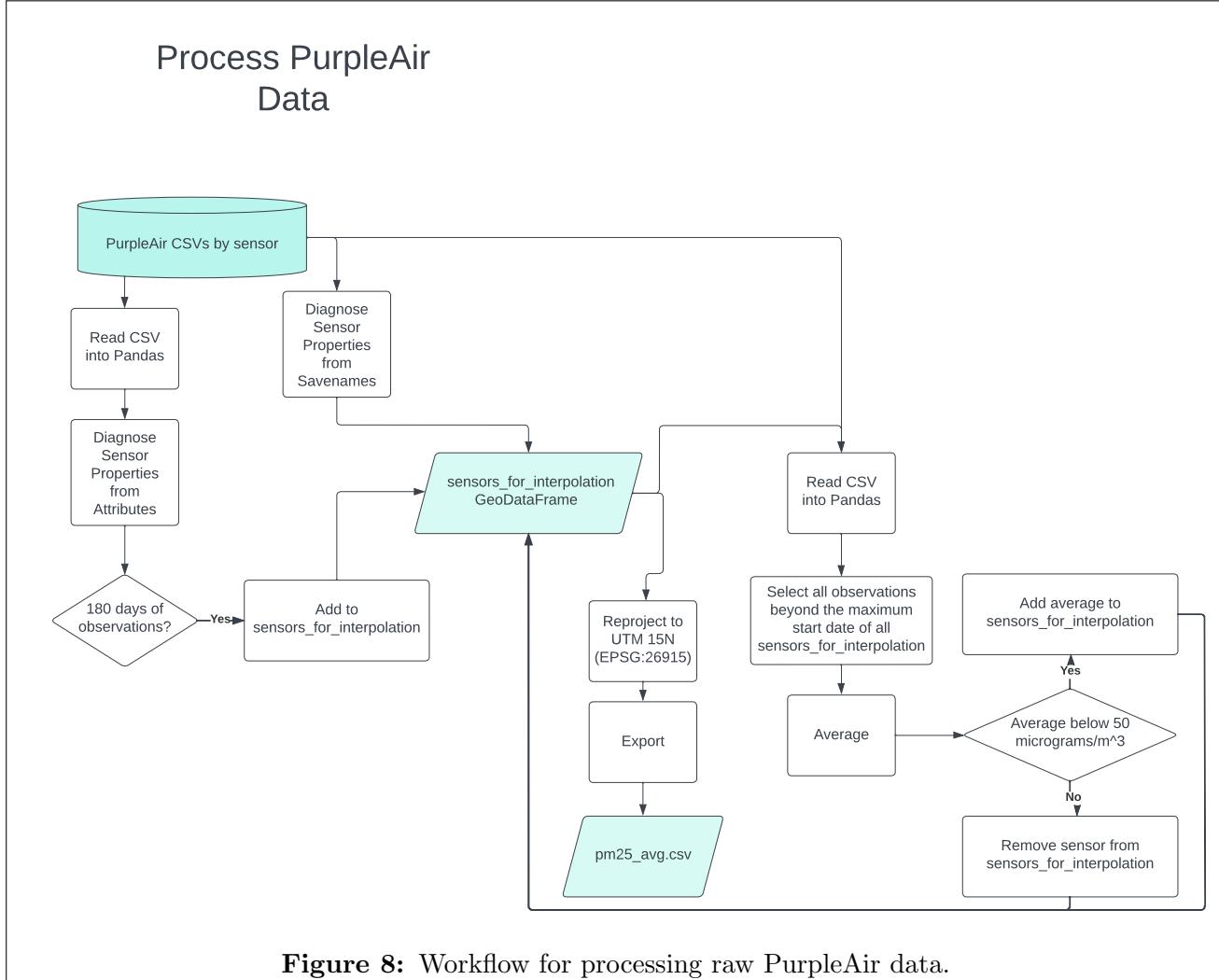
After the MPCA Permitted Emissions were loaded into a DataFrame, the data was explored extensively, noting the unique years (2006-2020), unique pollutants, and the various ways Portable Source was entered in the COUNTY column. The data was then queried by county. In exploring the latitude and longitude fields, they were found to be flipped. This was corrected and the MPCA notified. Some facilities had NaN coordinate values which sent me on a minor bout of investigative Google API requests. Ultimately, missed locational entries were omitted from the data but there are some leads, (see figure 7).

At this point the queried and corrected permitted emissions DataFrame was spatialized into a Geopandas GeoDataFrame using the function, `geopandas.points_from_xy()`, which creates a Geopandas geometry series. The coordinates were reprojected from WGS84 to UTM 15N, NAD83 and then clipped to the extent of the study. These annual emissions were then saved as a CSV, `emissions.csv`, with new columns 'EASTING' and 'NORTHING' that had values of their reprojected coordinates.



**Figure 7:** Suspected locations of missed locational entries found through the Google Places API.

### 3.2.5 PurpleAir



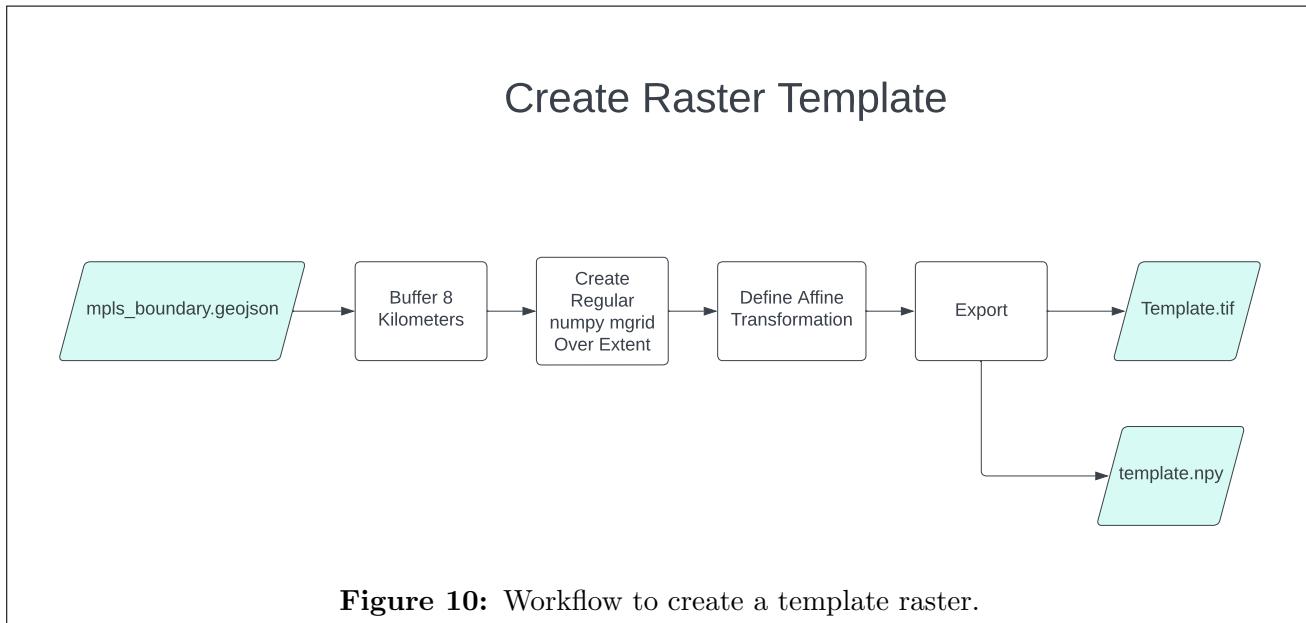
The 524 individual CSVs from PurpleAir had very descriptive names from which 7 columns were ascertained. This included: latitude, longitude, whether the sensor was outside or not, and if it was a 'B' sensor. The CSVs of the outdoor 'A' sensors were iteratively read into Pandas DataFrames and checked to see if they had a half a year of data. If they did, their indices were recorded for later selection. At this time, start dates, end dates, and number of observations were all recorded for each CSV with the information gathered from the filenames into a DataFrame that looked like figure 8.

	filename	name	is_active	is_outside	is_city	is_B	latitude	longitude	Start_Date	End_Date	Observations
6	Lee Circle (outside) (44.9073 -93.60277) Primar...	Lee Circle	Yes	outside	No	No	44.9073	-93.60277	2021-11-13 00:00:00	2022-10-25 12:00:00	1112.0
14	MPCA_Blaine (outside) (45.1407 -93.222) Primar...	MPCA_Blaine	Yes	outside	No	No	45.1407	-93.222	2021-11-13 00:00:00	2022-10-25 12:00:00	1021.0
44	City of Minneapolis Community Air Monitoring P...	City of Minneapolis Community Air Monitoring P...	Yes	outside	Yes	No	44.908063	-93.231281	2022-03-21 12:00:00	2022-10-25 12:00:00	592.0
48	Edina MN (outside) (44.878832 -93.354406) Prim...	Edina MN	Yes	outside	No	No	44.878832	-93.354406	2021-11-13 00:00:00	2022-10-25 12:00:00	1124.0
49	Cityscape (outside) (44.940389 -93.351256) Pri...	Cityscape	Yes	outside	No	No	44.940389	-93.351256	2021-11-13 00:00:00	2022-10-25 12:00:00	1146.0

**Figure 9:** Information on the Csv files downloaded from PurpleAir.

This DataFrame could be selected by the indices of the CSV files that had enough observations. These were considered to be candidates for interpolation. There were 71 using the data downloaded on 10/25/22. These were iterated through to find the average observed PM2.5 levels at each station between the maximum of all Start\_Date (7/24/22) and 10/25/22. There were some outlier sensor readings and these were each removed until 61 sensors were deemed worthy for interpolation. These were spatialized, reprojected, and saved as a CSV, pm25\_avg.csv.

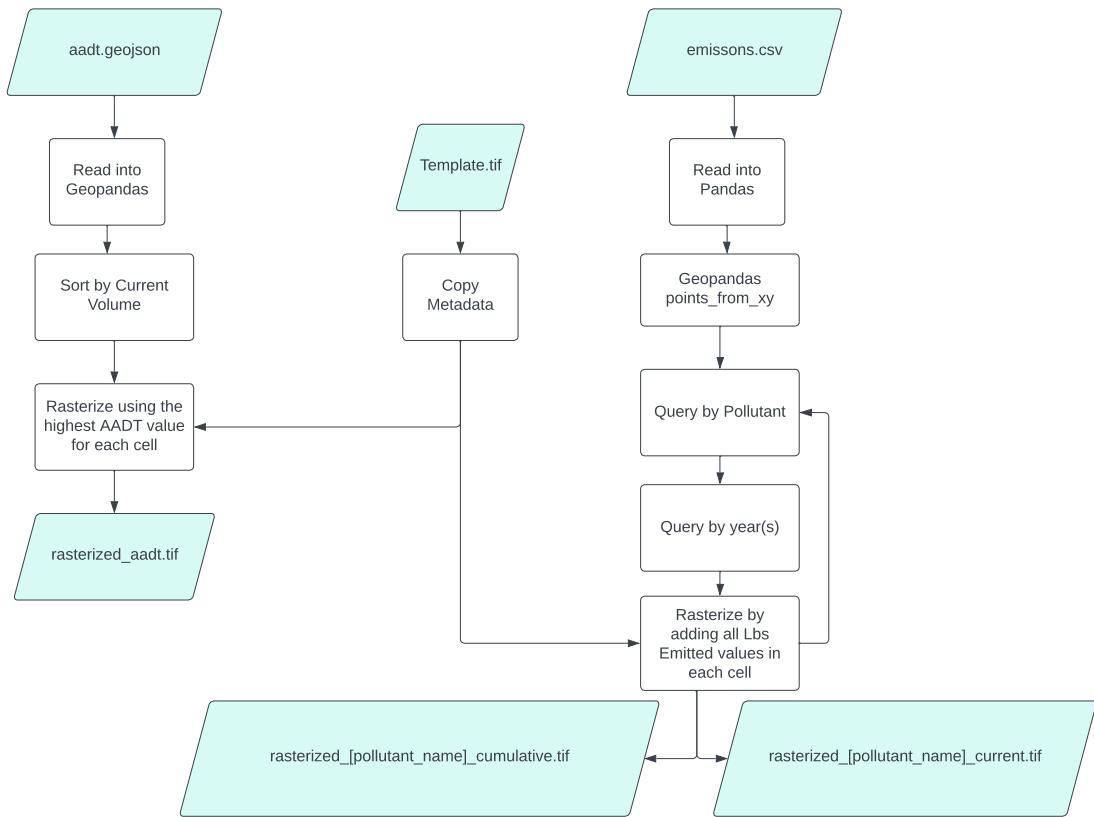
### 3.3 Create Template Raster



In the folder 2\_Model\_Pollutant\_Exposure there is a notebook named, 1\_Raster\_Template that describes how to perform these steps in more detail. The raster resolution chosen was 50m with 533 x\_cells and 678 y\_cells.

### 3.4 Rasterize Emissions Sources

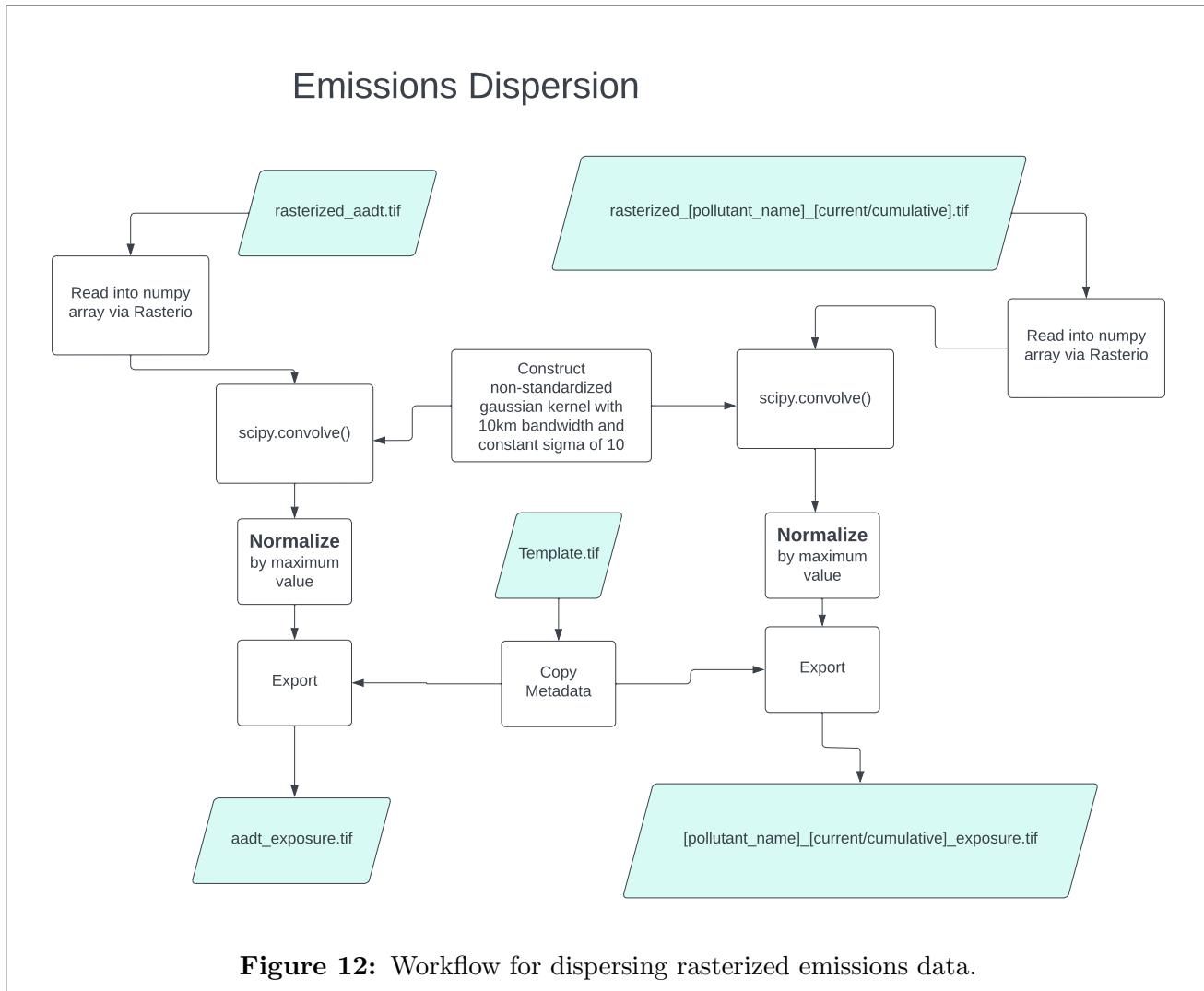
Emissions Interoperability



**Figure 11:** Workflow for rasterizing emissions source data.

The notebook that rasterizes the pollution sources can be found at `2_Model_Pollutant_Exposure/2_Rasterize_Sources.ipynb`. Notably during this process, the Rasterio function `features.rasterize()` was found to only have two types of merge algebra options, `rasterio.enums.MergeAlg.add` and the default, `rasterio.enums.MergeAlg.replace`. The merge algebra was set to `.replace` for AADT and `.add` for the current and cumulative pollutant emissions from this list of pollutants: 'Sulfur Dioxide', 'PM Primary', 'PM10 Primary', 'PM2.5 Primary', 'Volatile Organic Compounds'. These rasters were exported as GeoTiffs into the `Rasterized_Sources` folder.

### 3.5 Emissions Dispersion



Example code that performs these calculations can be found in the notebooks within the folder, `2_Model_Pollutant_Exposure/3_Model_Exposure`. Generally, this involved constructing a non-standardized, Gaussian kernel using the function shown in figure 13 with a fixed diameter,  $l$ , of 200 cells (10km at 50m resolution) and a sigma (10) to start. This kernel was passed along with a rasterized pollution source raster read as a numpy array into the Scipy function, `scipy.ndimage.convolve()`. This effectively emulated a weighted focal sum across the pollution source raster where the sigma represents a sort of density or fullness of the kernel (weights) used for dispersion.

```

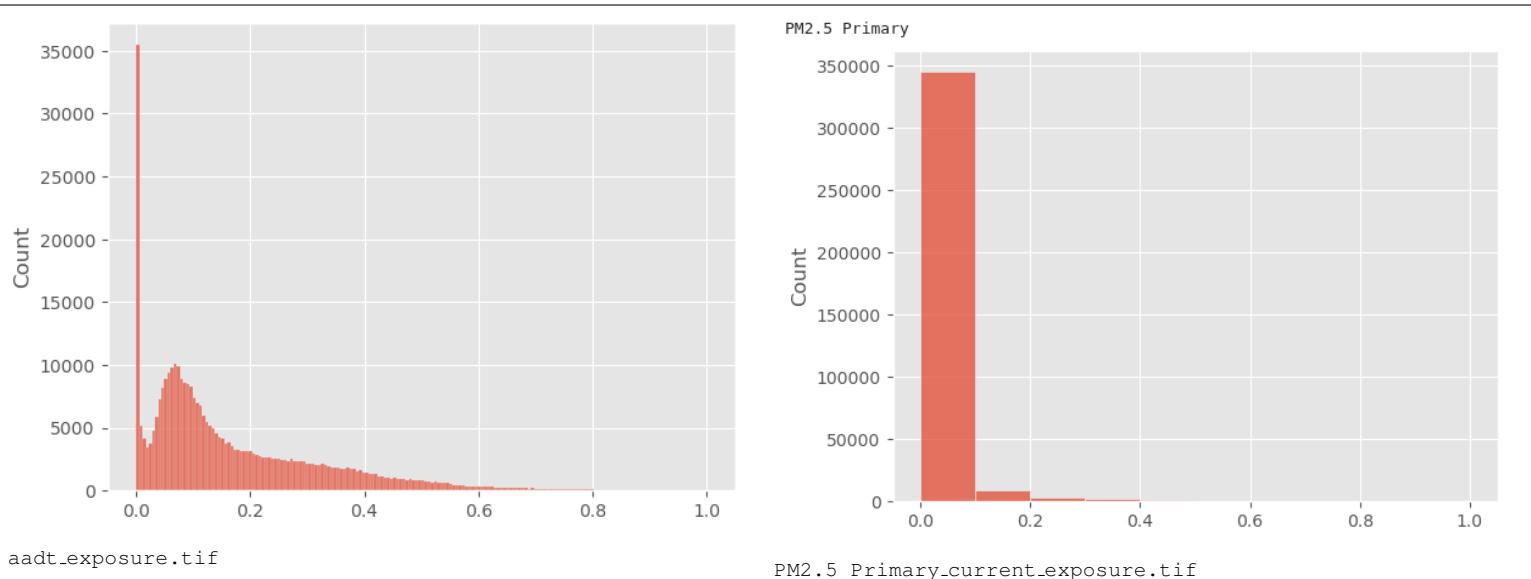
# Define weights kernel

def gkern(l=5, sig=1.):
    """
    Creates gaussian kernel with side length `l` and a sigma of `sig`
    """
    ax = np.linspace(-(l - 1) / 2., (l - 1) / 2., l)
    gauss = np.exp(-0.5 * np.square(ax) / np.square(sig))
    kernel = np.outer(gauss, gauss)
    return kernel #/ np.sum(kernel) <- this is for standardizing.. like a focal sum

```

**Figure 13:** A function that creates a non-standardized, Gaussian Kernel with specified diameter and sigma.

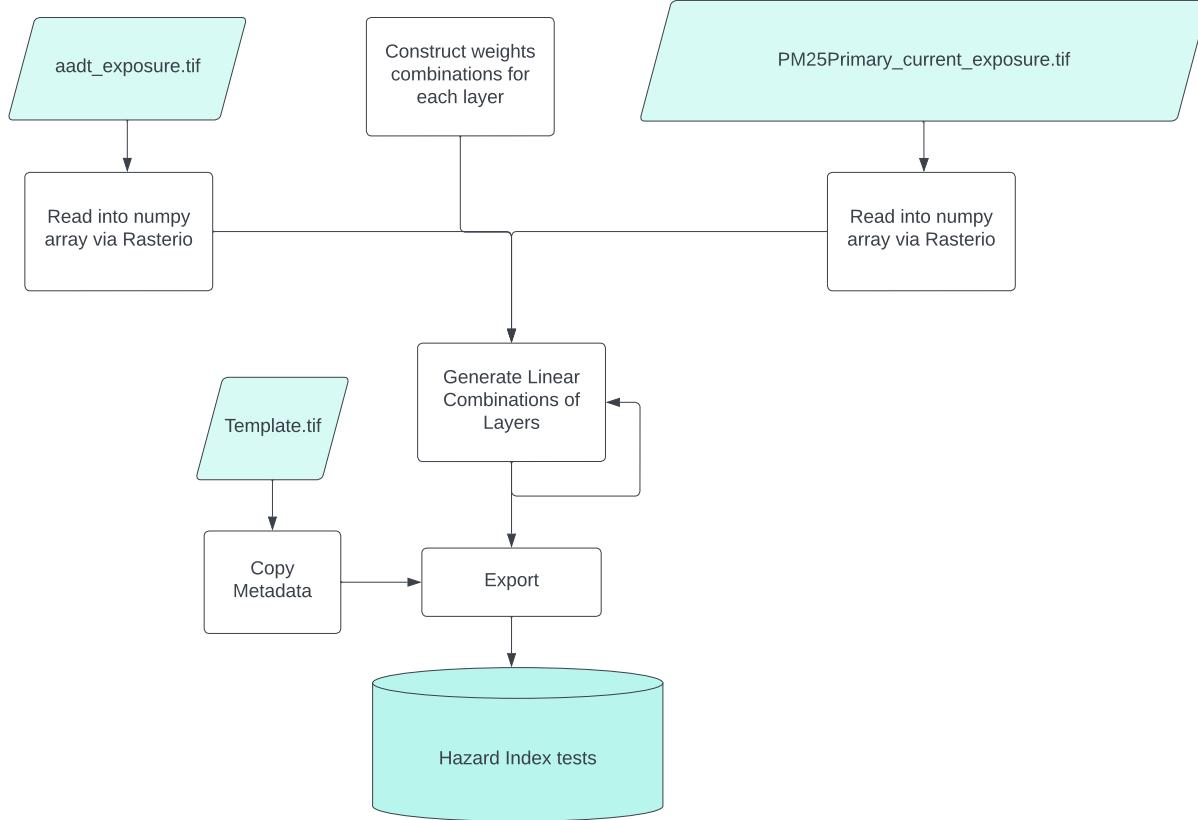
After the dispersion took place, the resulting values were divided by their maximum value to normalize the data. This was considered to be the exposure raster for each respective source and all were saved in the folder, 2\_Model\_Pollutant\_Exposure/\_Modeled\_Exposure. The distribution of the raster values for aadt\_exposure.tif and PM2.5\_Primary\_current\_exposure.tif can be seen in figure 14.



**Figure 14:** Distributions of pollution source exposure raster values.

### 3.6 Create Hazard Index

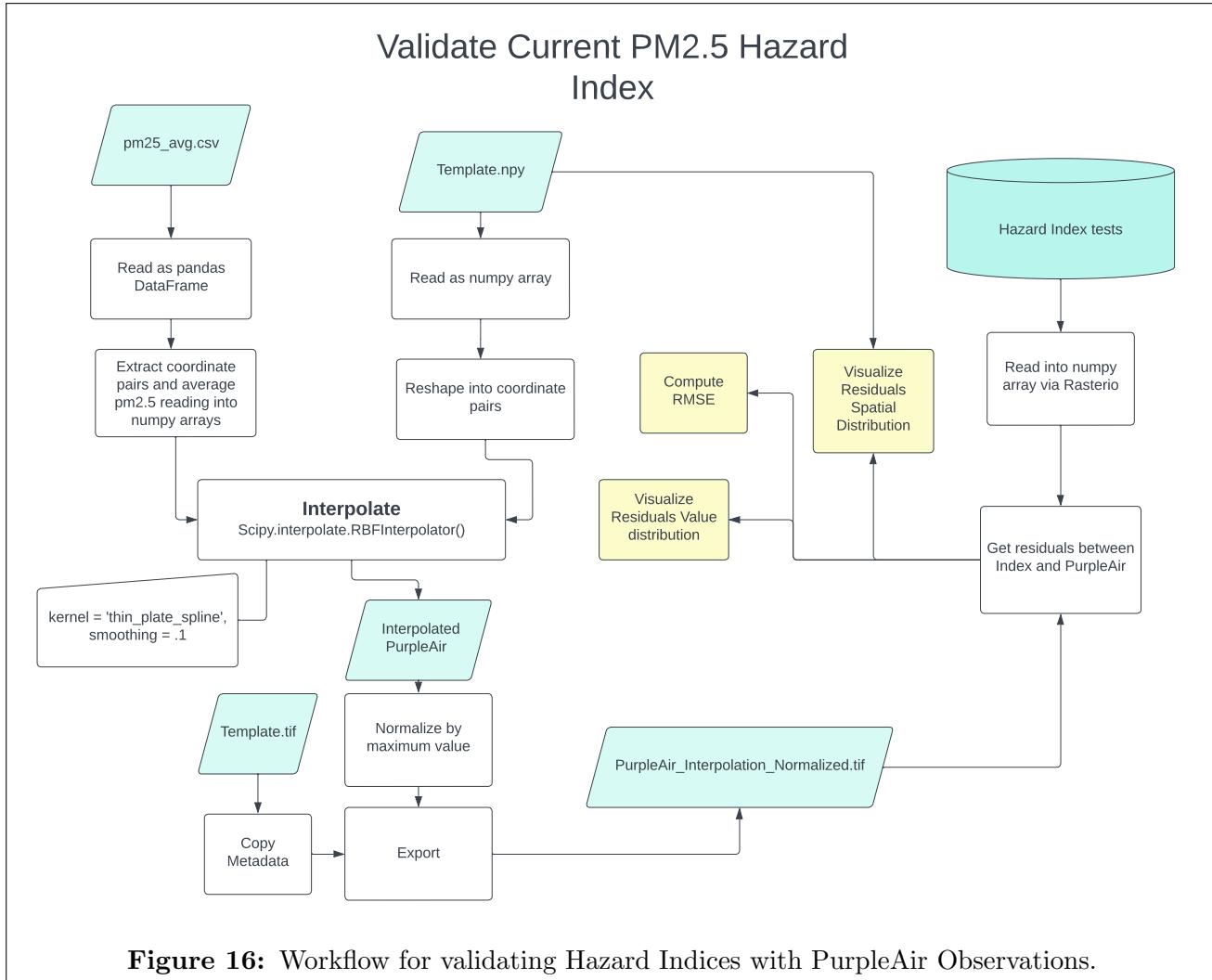
#### Current PM2.5 Hazard Index Creation



**Figure 15:** Workflow for constructing Current PM2.5 Hazard Indices.

The code for this portion of the project can be found in the notebooks within, `3_Hazard_Index`. At this stage, it was decided to only use the layers `aadt_exposure.tif`,  $T$ , and `PM2.5 Primary_current_exposure.tif`,  $I$ , to construct current PM2.5 hazard indices. Linear combinations of these layers were computed using numpy and layer weights between 0% and 100% such that  $T_{weight} + I_{weight} = \%100$ . These test hazard indices were saved in the folder, `3_Hazard_Index/_Tests`, with the naming format of,  $[I_{weight}] \text{PM-}[T_{weight}] \text{T\_test.tif}$ .

### 3.7 Validate Hazard Index



Code for this section can be found within the notebooks in the folder, `4_Validate_Index_Surface`.

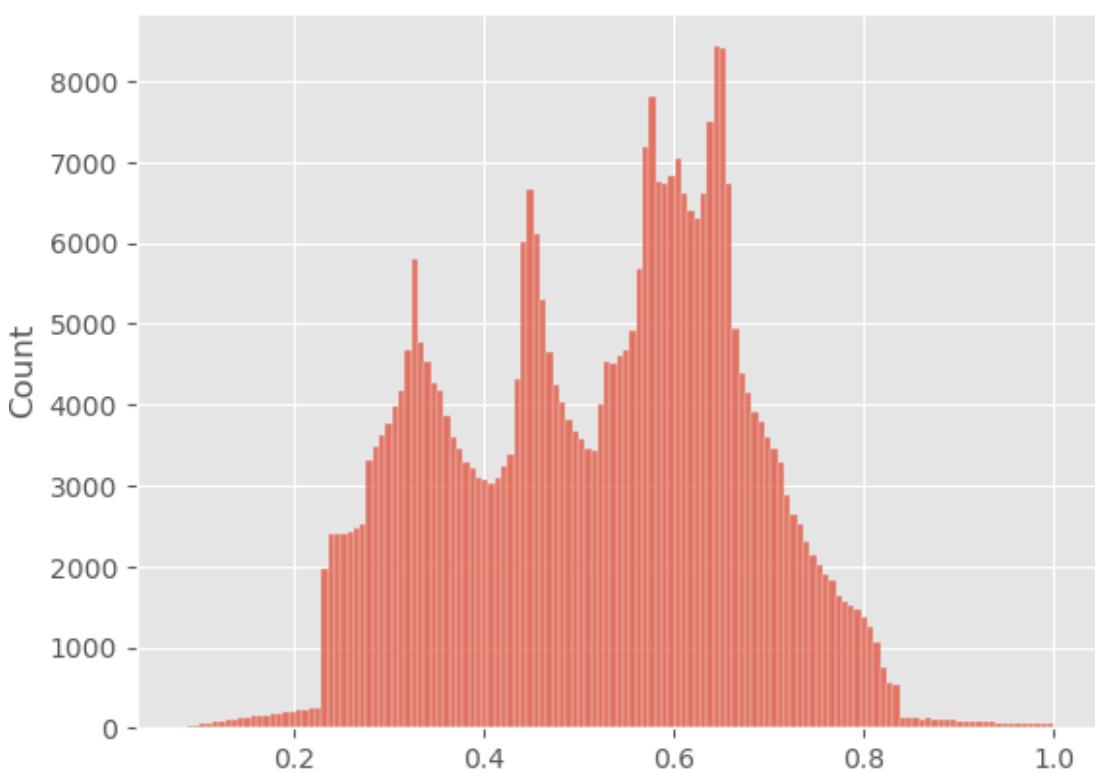
#### 3.7.1 Interpolation

The validation process began with constructing an interpolator for the PurpleAir observed average PM2.5 readings using the Scipy function, `interpolate.RBFInterpolator()` and the parameters:  $y$ ,  $d$ ,  $kernel = 'thin_plate_spline'$ , and  $smoothing = 0.1$  where  $y$  is the coordinate pairs from the PurpleAir Stations as a numpy array of shape, (Number of Stations, 2), and  $d$  was an array of the 'Avg\_pm2.5' values. This interpolator could be used with the coordinates from `template.npy` to construct a surface (see figure 17). The

resulting values were normalized by their maximum value (see figure 18) and saved as the file, PurpleAir\_Interpolation\_Normalized.tif.

```
# Get Coordinates  
  
coords = PurpleAir[['EASTING', 'NORTHING']].values  
  
# Interpolate  
  
interpolator = interp.RBFInterpolator(y = coords,  
                                         d = PurpleAir['Avg_pm2.5'].values,  
                                         kernel = 'thin_plate_spline',  
                                         smoothing = .1)  
                                         # smoothing = 0.1,  
                                         # kernel = 'inverse_quadratic',  
                                         # epsilon = .004)  
raster_coords = raster.reshape(2, -1).T  
  
Z = interpolator(raster_coords).reshape(raster[0].shape)
```

**Figure 17:** Code snippet showing the interpolation method used.



**Figure 18:** Value distribution of PurpleAir\_Interpolation\_Normalized.tif.

### 3.7.2 Accuracy Checks

The interpolation (observed) and hazard indices (predicted) were masked by the total bounds of Minneapolis using the function, `numpy.ma.array()`. Example code can be seen in figure 19. The residuals between these two masked arrays were computed to find the RMSE of each index within Minneapolis and the residuals spatial and value distributions were visualized. These initial comparisons can be explored further in the notebook, `4_Validate_Index_Surface/2_Compare_Model_and_PurpleAir.ipynb`.

```
# Mask PurpleAir to just Minneapolis (for accuracy checks)

# Boundary

mpls_path = os.path.join(os.getcwd(), '...', '1_Data_IO', 'Data', 'mpls_boundary.geojson')
mpls = gpd.read_file(mpls_path)

# Bounds

extent = mpls.geometry.total_bounds # minx, miny, maxx, maxy

minx = extent[0]
miny = extent[1]
maxx = extent[2]
maxy = extent[3]

# Find indices of raster within bounds

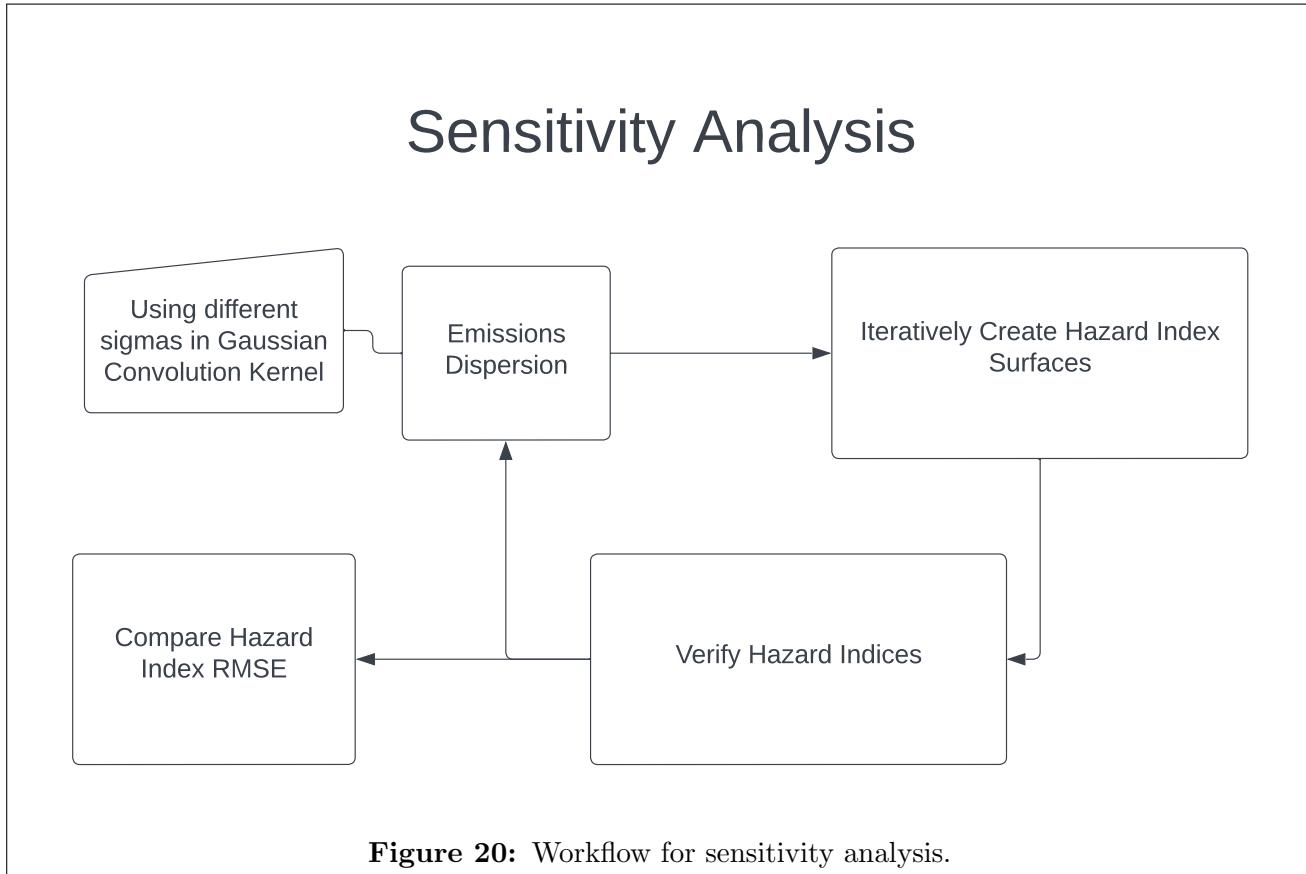
in_bounds_x = np.logical_and(raster[0]>(minx), raster[0]<(maxx))
in_bounds_y = np.logical_and(raster[1]>(miny), raster[1]<(maxy))
in_bounds = np.logical_and(in_bounds_x, in_bounds_y)

# Select these from PurpleAir

PurpleAir_clipped = ma.array(PurpleAir.T, mask = np.invert(in_bounds))
```

**Figure 19:** Masking the PurpleAir interpolation by the total bounds of Minneapolis.

### 3.8 Sensitivity Analysis



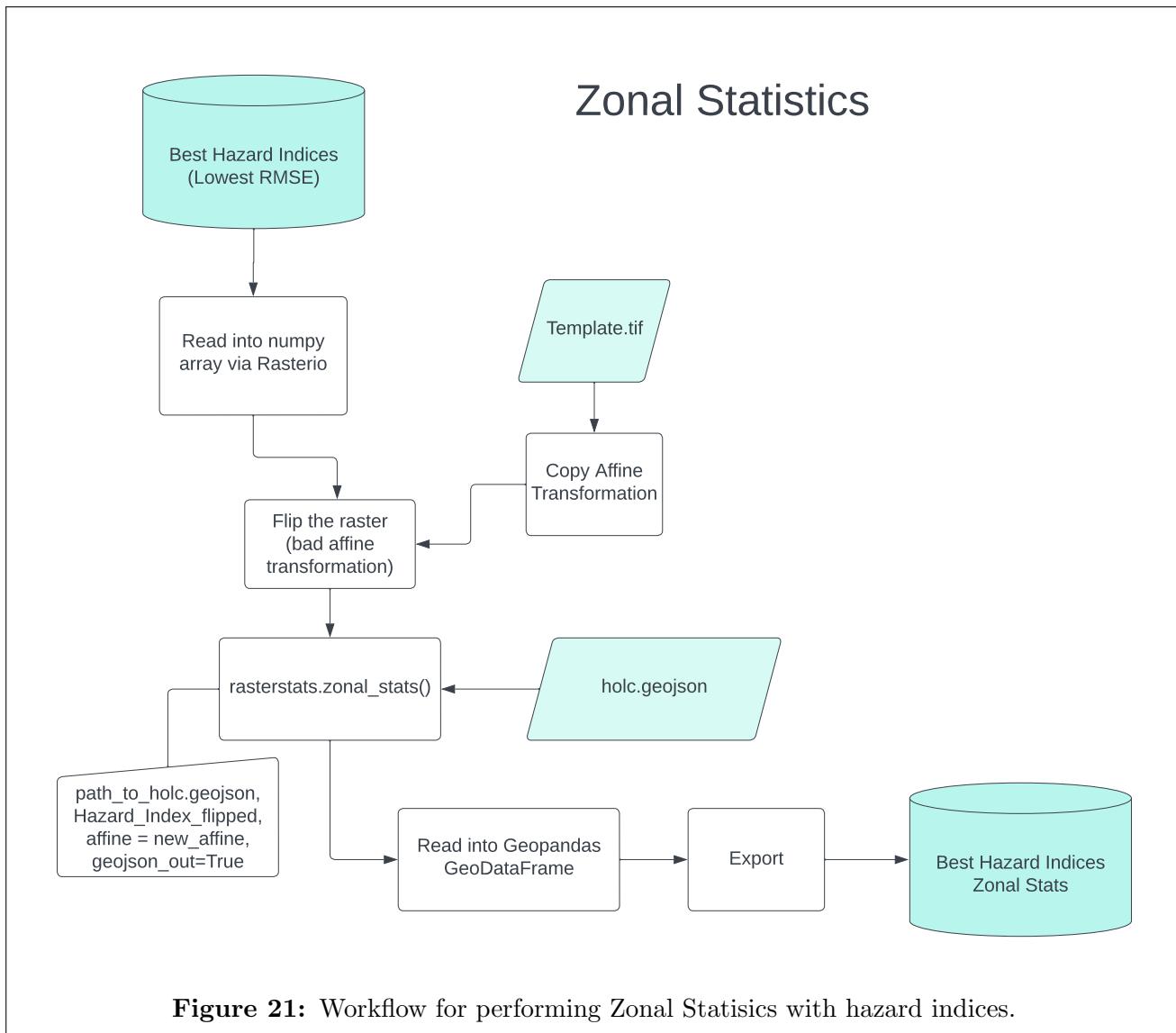
**Figure 20:** Workflow for sensitivity analysis.

The script that performs this analysis is in the main Final\_Project Directory called, `7_Sensitivity_Analysis_Prep.py`. On top of varying the weights for the exposure layers, the Sigma ( $\sigma$ ) in the traffic dispersion Gaussian Kernel was allowed to vary between 1 and 15 by 0.5 intervals with a constant 10km diameter. The naming convention for these hazard indices was changed to:

`[ $\sigma$ ]sig-[ $I_{weight}$ ]I-[ $T_{weight}$ ]T_HazardIndex.tif.`

Code that explores these findings can be found in the folder `7_Sensitivity_Analysis`.

### 3.9 Zonal Statistics



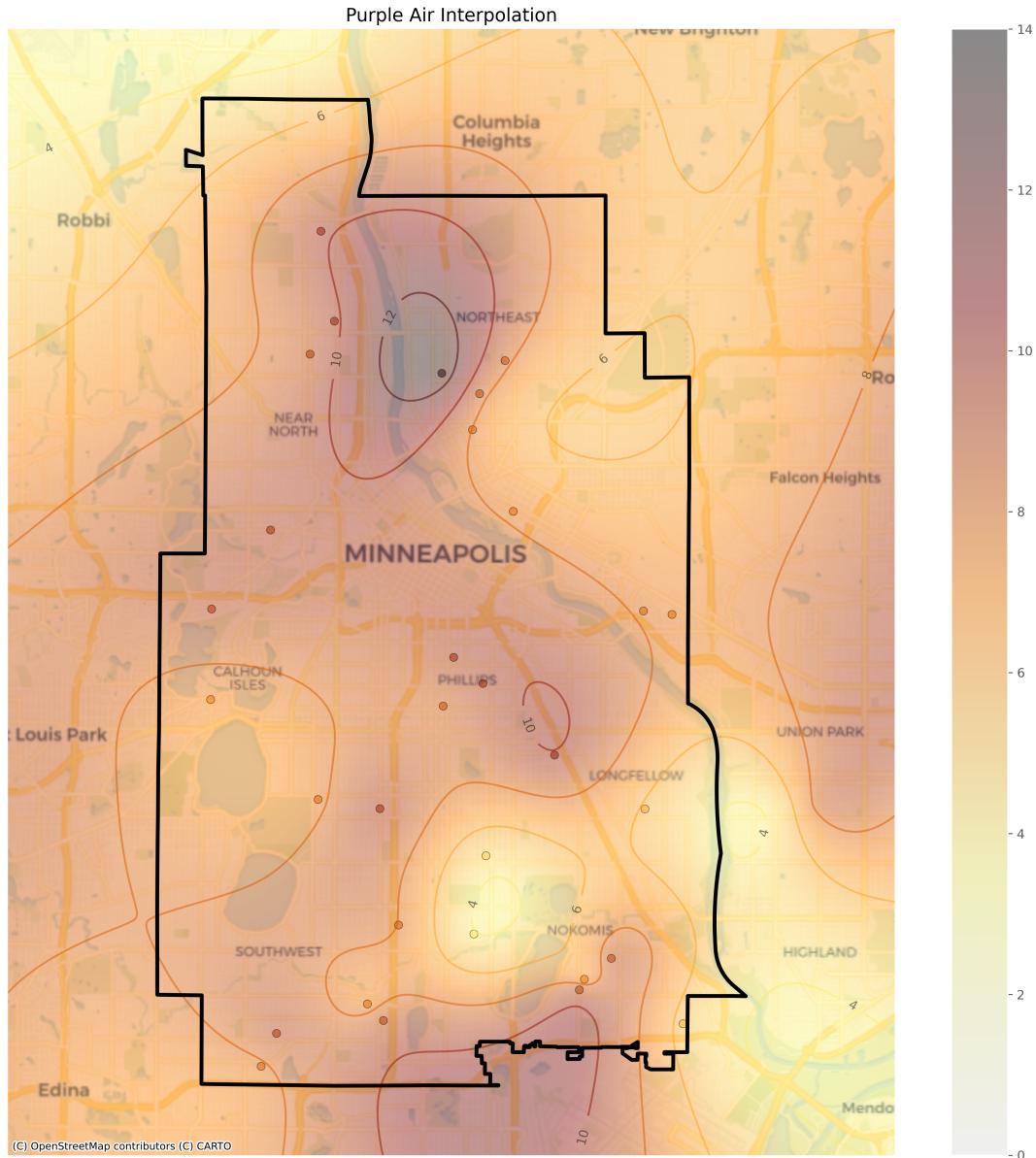
The Zonal Means of some select hazard indices were calculated using the HOLC Parcels as zones and the resulting GeoJSONs can be found in the folder, 7\_Sensitivity\_Analysis/Zonal\_Stats. Example code that accomplishes this step can be found in 5\_Zonal\_Statistics/1\_HOLC\_Zones.ipynb and was included in the sensitivity analysis script. Notably, it was discovered that the Rasterstats function, `zonal_stats()` does not accept affine transformations with an inverted y-axis. This can be corrected with a function shown in figure 22.

```
# Flip the raster
# From: https://github.com/perrygeo/python-rasterstats/issues/98

def flipud(raster, affine):
    raster = np.flipud(raster)
    affine = Affine(
        affine.a,
        affine.b,
        affine.c,
        affine.d,
        -1 * affine.e,
        affine.f + (affine.e * (raster.shape[0] - 1)),
    )
    return raster, affine
```

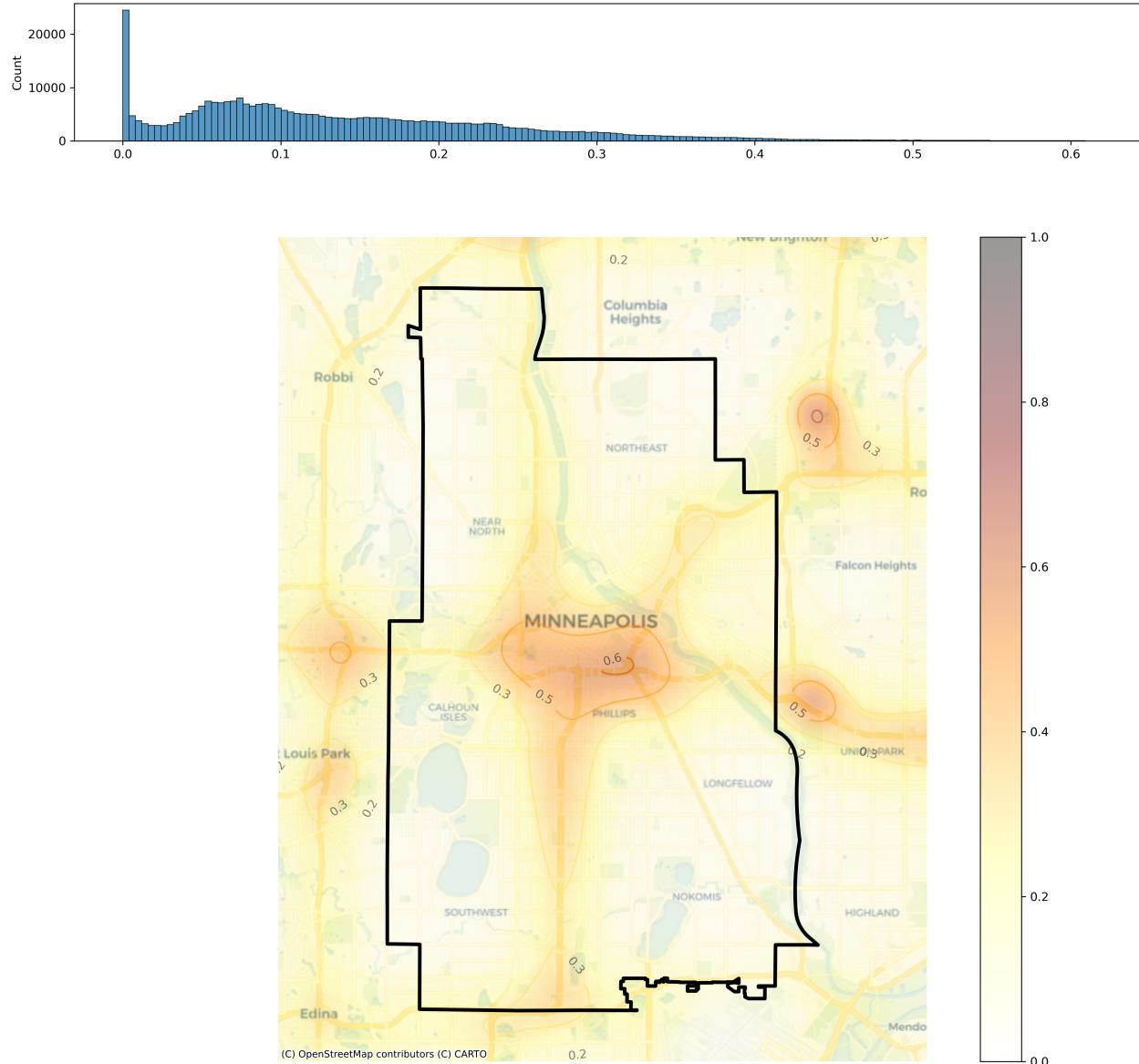
**Figure 22:** A function which flips a raster and affine transformation such that North is "up."

## 4 Results

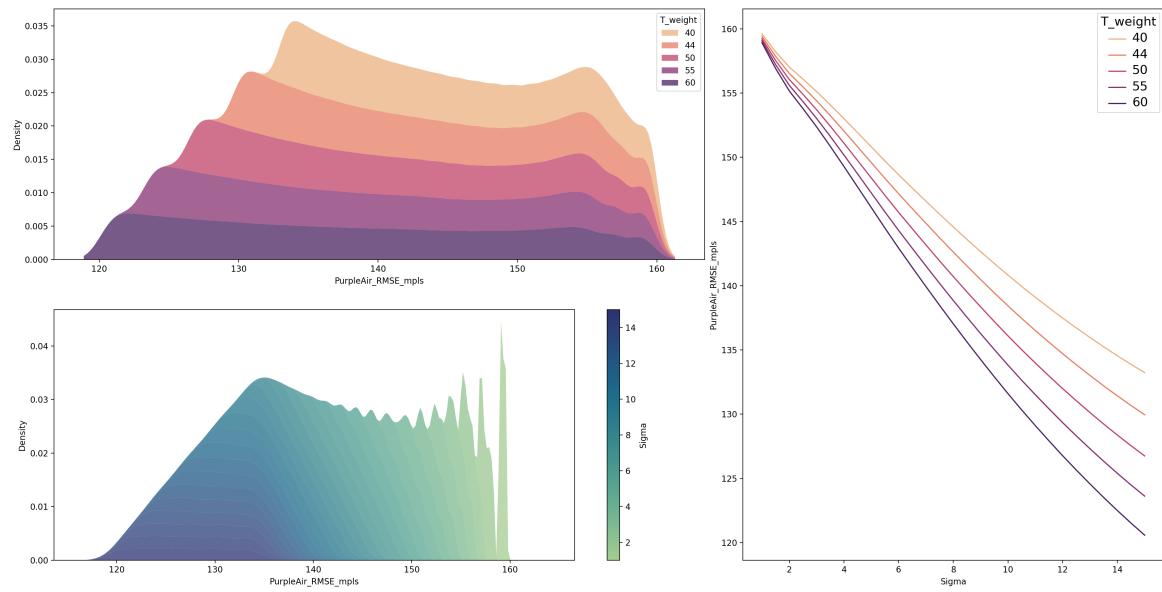


**Figure 23:** Interpolation of Average PM<sub>2.5</sub> Observations with sample points displayed ( $\frac{\mu\text{g}}{\text{m}^3}$ ).

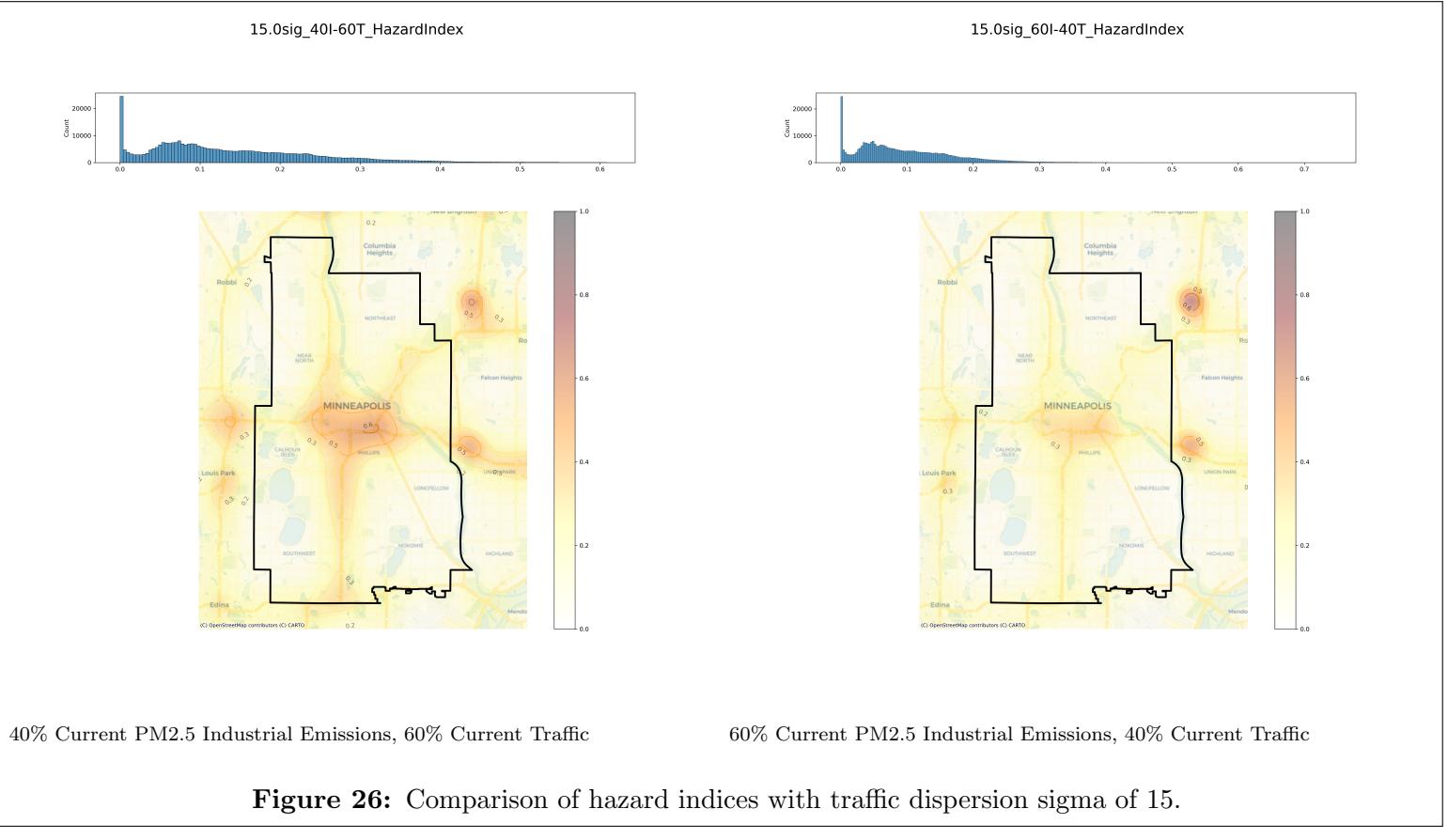
## 15.0sig\_40I-60T\_HazardIndex



**Figure 24:** Hazard index with traffic dispersion sigma of 15 (40% Current PM2.5 Industrial Emissions, 60% Current Traffic).

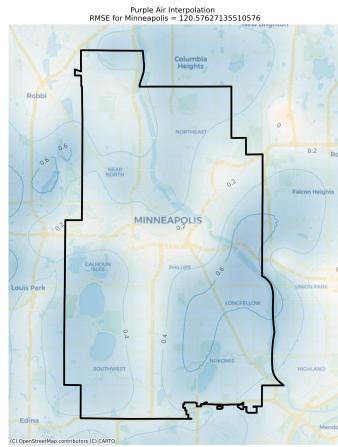
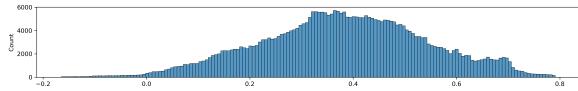


**Figure 25:** Graphs showing relationships between RMSE of hazard indices over normalized interpolated values, sigma of traffic dispersion, and weight of the traffic exposure layer.

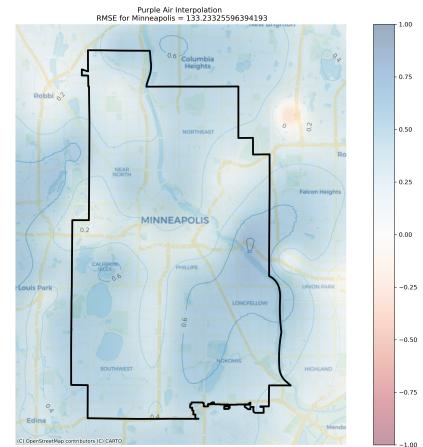
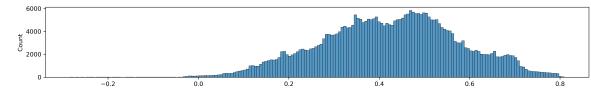


**Figure 26:** Comparison of hazard indices with traffic dispersion sigma of 15.

15.0sig\_40I-60T\_HazardIndex Residuals



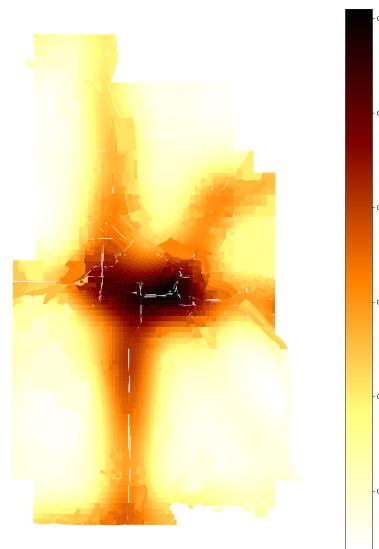
15.0sig\_60I-40T\_HazardIndex Residuals



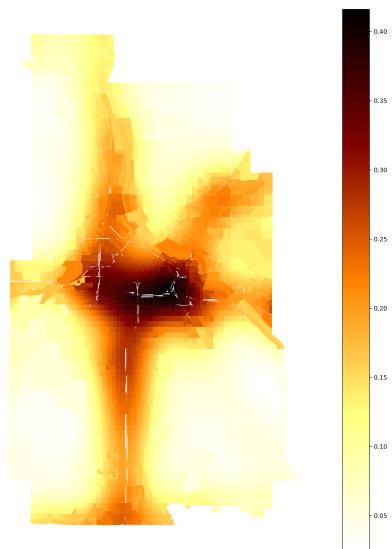
40% Current PM2.5 Industrial Emissions, 60% Current Traffic

60% Current PM2.5 Industrial Emissions, 40% Current Traffic

**Figure 27:** Comparison of hazard indices' residuals (traffic dispersion sigma of 15) from normalized average PurpleAir observations.



40% Current PM2.5 Industrial Emissions, 60% Current Traffic



60% Current PM2.5 Industrial Emissions, 40% Current Traffic

**Figure 28:** Comparison of hazard indices zonal means (traffic dispersion sigma of 15) with HOLC parcels.



**Figure 29:** Hazard index zonal mean average value by HOLC Grade (traffic dispersion sigma of 15, 40% Current PM2.5 Industrial Emissions, 60% Current Traffic).



**Figure 30:** Hazard index zonal mean average value by HOLC Grade (traffic dispersion sigma of 15, 60% Current PM2.5 Industrial Emissions, 40% Current Traffic).

## 5 Discussion and Future Directions

This project utilized a wide range of geographic information science techniques. Due to this, there are a number of steps in the workflow that could be improved.

In reviewing the PurpleAir interpolation visually (figure 23), we can see that the spline technique made semi-reasonable predictions. However, upon reviewing literature on modeling air pollution at street level, interpolating air quality data at this fine of scale can be quite difficult. Forehead and Huynh [10] states, "Concentrations can be highly variable over short distances and intervals of time, due to fleet composition, congestion, weather (mainly wind) and the shape of street canyons." Even at this temporal resolution, there can be many confounding factors not included in the interpolator used. This part of the analysis could certainly be improved upon, and future directions of this research will definitely involve further scrutiny of this step.

The hazard index itself was a bit of an experiment. The ultimate goal was to try and include traffic levels in air quality dispersion modeling which are typically only factored into the ambient air quality in MPCA [Dispersion Modeling](#). As can be seen in figure 26, the higher index values were primarily along roadways with much of the study area having an index less than 0.3. Looking at the residual plots (figure 27), we can see that the index was consistently lower than the normalized interpolated values (positive residuals) leading me to wonder about including more sources (such as rail yards) and perhaps including an ambient hazard index level based on the distribution of the residuals (see figure 27 histograms).

Other clues to improve the index come from looking at the results of the sensitivity analysis (see figure 25). Here, we can see two definite patterns. One, the greater the sigma value for traffic, the more accurate the model appeared. Two, the indices that weighted traffic appeared to perform better as they generally had higher index values. When I ran the dispersion models on my computer, my memory (16Gb) maxed out when using a sigma of 15 at the constant diameter of 10km. Perhaps utilizing a supercomputer to perform sensitivity analysis with larger sigmas, larger diameters, different sigmas for industrial pollution, and more refined weights could lead to a more accurate hazard index.

The zonal statistics were a good way to tie this analysis back to the problem statement.

We did see in figures 29 and 30 that the different HOLC grades tended to have average indices that aligned with what we could hypothesize from history. However, I think that improving the model should take precedence before giving these values much consideration. Future directions could include ANOVA analysis of the HOLC grades and using different zones (such as Census Block Groups) to conduct correlation analyses.

Further directions for this work should include a spatio-temporal investigation of the index to better understand temporally-cumulative impacts of PM2.5 and whether or not things are improving. Community engagement is also key, as the pollutants to focus on in the future should largely reflect what people care about (co-creation). Educational resources should also be developed further to assist in the big campaign to tackle these disparities in pollution exposure that the City of Minneapolis' offices of Sustainability and Health are starting.

Environmental justice (EJ) continues to expand and incorporate different conceptions of space, environment, and justice. Contemporary EJ writers often cite a need for more community outreach, education, and inclusive decision-making as well as innovative collaboration between planners, policy-makers, academics, and citizens to achieve profound environmental justice [11, 12, 13]. Pearsall and Pierce [13] emphasize a need for well defined indicators that are spatially focused, not aggregate measures, to gauge a policy's success at implementing environmental justice. This new workflow and index are a couple more tools we can use to collectively reckon with our history, assess our current situation, and work toward reparations.

## References

- [1] U.S. Environmental Protection Agency. URL: <https://www.epa.gov/environmentaljustice/learn-about-environmental-justice> (visited on 02/07/2022).
- [2] *500 Cities: Census Tract-level Data*. Centers for Disease Control and Prevention. 2018. URL: <https://chronicdata.cdc.gov/500-Cities-Places/500-Cities-Census-Tract-level-Data-GIS-Friendly-Fo/k25u-mg>.
- [3] *Air Quality in Minneapolis: A Neighborhood Approach*. Report. Health Professionals for a Healthy Climate, 2021. URL: [https://hpforhc.org/wp-content/uploads/2021/11/HPHC\\_factsheet\\_AirPollution.pdf](https://hpforhc.org/wp-content/uploads/2021/11/HPHC_factsheet_AirPollution.pdf).
- [4] Bryan Hubbell, Neal Fann, and Jonathan I. Levy. “Methodological considerations in developing local-scale health impact assessments: balancing national, regional, and local data”. In: *Air Quality, Atmosphere & Health* 2.2 (June 2009), pp. 99–110. DOI: [10.1007/s11869-009-0037-z](https://doi.org/10.1007/s11869-009-0037-z).
- [5] Metropolitan-Council. *Counties and Cities & Townships 2010, Twin Cities Metropolitan Area*. 2010. URL: <https://gisdata.mn.gov/dataset/us-mn-state-metc-bdry-census2010counties-ctus>.
- [6] Minnesota Department of Transportation. *Annual Average Daily Traffic Segments in Minnesota*. Periodically Updated. URL: <https://gisdata.mn.gov/dataset/trans-aadt-traffic-segments>.
- [7] *Permitted Facility Air Emissions Data*. Minnesota Pollution Control Agency. 2006–2019. URL: <https://www.pca.state.mn.us/air/permitted-facility-air-emissions-data>.
- [8] *PurpleAir API*. PurpleAir. 2022. URL: <https://api.purpleair.com/>.
- [9] Metropolitan-Council. *Historic Home Owners’ Loan Corporation Neighborhood Appraisal Map*. 1934. URL: <https://gisdata.mn.gov/dataset/us-mn-state-metc-plan-historic-holc-appraisal>.
- [10] H. Forehead and N. Huynh. “Review of modelling air pollution from traffic at street-level - The state of the science”. In: *Environmental Pollution* 241 (2018), pp. 775–786. ISSN: 0269-7491. DOI: <https://doi.org/10.1016/j.envpol.2018.06.019>. URL: <https://www.sciencedirect.com/science/article/pii/S0269749118300216>.
- [11] Gordon Walker. “Beyond Distribution and Proximity: Exploring the Multiple Spatialities of Environmental Justice”. In: *Antipode* 41.4 (Aug. 2009), pp. 614–636. DOI: [10.1111/j.1467-8330.2009.00691.x](https://doi.org/10.1111/j.1467-8330.2009.00691.x).
- [12] Jason Corburn. “Bringing Local Knowledge into Environmental Decision Making: Improving Urban Planning for Communities at Risk”. In: *Journal of Planning Education and Research* 22.4 (2003), pp. 420–433. DOI: [10.1177/0739456X03022004008](https://doi.org/10.1177/0739456X03022004008).

- [13] Hamil Pearsall and Joseph Pierce. "Urban Sustainability and Environmental Justice: Evaluating the Linkages in Public Planning/Policy Discourse". In: *Local Environment* 15.6 (July 2010), pp. 569–580. DOI: [10.1080/13549839.2010.487528](https://doi.org/10.1080/13549839.2010.487528).

## Self Score

Category	Description	Points Possible	Score
<b>Structural Elements</b>	All elements of a lab report are included (2 points each): Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score	28	26
<b>Clarity of Content</b>	Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level (12 points). There is a clear connection from data to results to discussion and conclusion (12 points).	24	20
<b>Reproducibility</b>	Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified.	28	20
<b>Verification</b>	Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated (10 points), the method of comparison is clearly stated (5 points), and the result of verification is clearly stated (5 points).	20	17
		100	83