AGENT-BASED MODELING OF THE TRANSPORT OF TROPOSPHERIC OZONE GENERATED BY POINT-SOURCE EMISSIONS OF OZONE PRECURSORS

by

David Adam Mills, B.S.

A thesis submitted to the Graduate Council of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Geography
August 2018

Committee Members:

Yongmei Lu, Chair

John Tiefenbacher

Russell Weaver

FAIR USE AND AUTHOR’S PERMISSION STATEMENT

Fair Use

Duplication Permission

DEDICATION

To my intelligent and beautiful daughter, Faith. You were born while I was an undergraduate at Texas State University. The better part of my life has been because of you. I am a better person because of you. I love you.

"You're off to great places!
Today is your day!
Your mountain is waiting.
So…get on your way!"
- Dr. Seuss

ACKNOWLEDGEMENTS

I am extremely grateful to my advisor and the members of my committee: Dr. John Tiefenbacher, Dr. Russell Weaver, and Dr. Yongmei Lu. I am indebted to Dr. Tiefenbacher for his thorough and comprehensive review of my Thesis drafts. His expertise in air quality, incomparable editing, and punctilious reviews have tremendously improved my current work. Dr. Tiefenbacher's example is what I will strive for as I continue to improve my writing in the future. Dr. Weaver answered an unscheduled knock at his door countless times. Each time he would take time out of his busy day to listen and provide feedback in a manner that implied helping a student was the most important task on his list of things to do. I hope that I will provoke onto students a sense of importance about themselves and their research goals as Dr. Weaver has done for me. Dr. Lu has been a mentor to me since I took her Advanced GIS course as an undergraduate student. She has unwaveringly encouraged and pushed me to pursue my education and research. Dr. Lu's steady support and guidance have led me to pursue a future that I am passionate about and proud of.

I could not have hoped for a more professional, friendly, and knowledgeable group to be surrounded by than the staff, professors, students, and my peers at Texas State University's Department of Geography. I do not believe that a higher quality Geography department exists. Also, I am appreciative for the love and support of my friends and family during this time. There are too many of you, and too much that you have done to be mentioned here.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

1. INTRODUCTION

According to the World Health Organization, air quality is the single largest environmental risk to global health (WHO 2017). As such, numerous international, governmental, and non-governmental agencies exist to advocate for policies and practices designed to improve air quality. In the United States, for example, the Environmental Protection Agency (EPA) currently regulates standards for six common air pollutants, referred to as the “criteria air pollutants” (EPA, n.d.). Among the six criteria pollutants monitored by the EPA is tropospheric ozone (O3). Unlike most of the other criteria pollutants—namely, carbon monoxide, lead, nitrogen dioxide, and sulfur dioxide—tropospheric ozone is not emitted directly into the air. Rather, it is formed when volatile organic compounds (VOC) and nitrous oxides (NO_x) react to sunlight. For that reason, the geographic distributions and movements of these two ground-level ozone precursors contain valuable information on the spatial and temporal distribution of “bad ozone” across space and time (EPA, n.d.). With that in mind, this study employs agent-based modeling (ABM) of NO_x and VOCS from point-source emissions to visualize and model ozone transport in the troposphere.

More precisely, this study undertakes a two-step process to evaluate the geographical gaps that can be filled with an ABM. First, the locations of ozone sensors in the state of Texas is evaluated. Texas was selected as a study area, among other reasons, because there are currently 18 counties in the state that are classified as nonattainment areas with respect to the EPA’s National Ambient Air Quality Standards (NAAQS) 8-hour average concentration of ozone (they exceed 70 parts per billion).

Second, the spatial distribution of the potential for ozone formation and transport is assessed for a subarea in the state of Texas. The Texas Commission on Environmental Quality (TCEQ) monitors emissions of VOC and NO_x from approximately 2,000 point-sources. Guidance for the spacing of monitors has been developed by the National Research Council (NRC). According to the NRC, adequate coverage of a region requires at least two sensors that are placed in strategic positions relative to potentially exposed populations and predominant wind patterns: one sensor should be located upwind and the other located downwind of the areas of the largest or densest populations in the region. Areas with point-source emissions and without adequate monitoring are areas that could benefit from a regional ABM as a means of indicating ozone risk. An ozone-formation risk assessment map will be developed to reveal the patterns of potential for ozone formation from precursor gases (VOCs and NO_x) from point sources. Areas identified as high-risk and having sensors are important for the calibration of an ABM.

The objective of this research is thus to use ABM to model the transport of tropospheric ozone generated by point-source emissions of ozone precursors. The model produces simulated data indicating tropospheric ozone risk. The ABM considers the locations of point-source emissions, hourly wind speed, and hourly wind direction to determine the distribution of ozone formation and transport. Further analysis is then conducted using topographic variables to understand how plume height and elevation impact vertical ozone transport in the troposphere. VOCS and NO_x emissions are treated as agents originating from the point sources identified in the TCEQ 2010 emission point-source inventory. These two pollutants are precursor emissions whose interaction and chemical reactions generate ozone by exposure to sunlight, particularly ultraviolet (UV) radiation. Ozone monitor locations within TCEQ's Region 11 are emulated within the model. Historical hourly ozone readings from Region 11's ozone monitors are factored into the model. Each monitor's hourly ozone readings are visually compared with the ozone precursor gas concentration and predictions of ozone generated by the model. This study demonstrates the use of an ABM to evaluate and visualize tropospheric ozone formation and transport.

2. BACKGROUND

2.1 Criteria Air Pollutants

In 1970, the Environmental Protection Agency (EPA) identified six chemical compounds considered to be the most widespread and most detrimental atmospheric contaminants in the United States. These compounds make up the list of criteria air pollutants that would form the basis of air quality management according to the Clean Air Act. Tropospheric ozone, particulate matter, carbon monoxide, lead, sulfur dioxide, and nitrogen dioxide are used to evaluate air quality throughout the United States (Godish, et al. 2015). The 1970 Clean Air Act required states to reduce these emissions to meet the newly established National Ambient Air Quality Standards (NAAQS).

2.2 "Good" and "Bad" Ozone

Ozone is found in two of the Earth's atmospheric layers: the troposphere and the stratosphere. Ozone in the stratosphere is considered "good ozone" because it absorbs harmful ultraviolet (UV) rays. Ozone in the troposphere is considered "bad ozone" because it causes health effects in humans when inhaled, damages plants, and degrades materials in the built environment.

Scientists became aware that ozone was a vital component of the stratosphere in the 1960s after the increasing body of evidence that chlorofluorocarbons (CFCs) were increasing in the atmosphere since the time of their invention (the 1930s) and were found to be negatively impacting "good ozone." The evidence "propelled this simple molecule into the international spotlight and resulted in the investment of billions of dollars in the research and development of new industrial chemicals" (McElroy 2008:1). When ozone is not reduced by contaminants such as CFCs and other halogenated compounds, the lifetime of ozone molecules in the stratosphere is years.

In contrast to the helpful effects of ozone as it absorbs UV rays in the stratosphere, ozone in the troposphere, formed by chemical processes driven by that same UV radiation passing through the troposphere to the surface, poses health risks to people, animals, plants, and man-made materials. Ozone damages tissue in respiratory airways, decreases lung function. Additionally, it can decrease the body's ability to fight against infection, and it is recognized as a genotoxic substance that has a high potential to cause cancer (Godish 2015:185-186). To combat the harmful effects of ground-level ozone, the EPA requires each state in the U.S. to have a State Implementation Plan (SIP) that clearly explains the measures that will be taken to reduce ground-level ozone and to maintain stratospheric ozone. When ozone forms in the troposphere, its lifetime is between hours and days.

2.3 Ozone Formation and Transportation

Sources of the ingredients to produce ozone are biogenic and technological (e.g., automobiles and industries) and can take the form of areal or point-source emissions. Ozone is a secondary pollutant that forms in the atmosphere in a chemical reaction from the primary pollutants emitted by humans. VOCs and NO_x combine in a complex process driven by UV radiation (in sunlight) to form ozone. The reaction creates a photochemical smog that appears reddish-brown in color. Ozone concentrations tend to increase proportionally with ambient environmental temperatures, as heat promotes chemical reactions. Horizontal atmospheric transport—wind—can carry ozone hundreds of miles away from its point of origin.

2.4 Monitoring Ozone

Ozone monitors are owned and operated by private entities and governmental organizations of all sizes. The TCEQ monitors ozone in Texas in multi-month periods between March 1st and November 30th. The National Ambient Air Quality Standards (NAAQS) for ozone have changed twice since 1991: < 125 parts per billion (ppb) in 1991, < 85 ppb in 2004, and < 70 ppb in 2015. The Air Quality Index (AQI) is used to categorize air quality from "good" to "hazardous" using six intervals of descriptors. The descriptors are categories that reflect the danger of the ambient air for people from those most sensitive to those who are in good health. When a region does not meet the NAAQS, the EPA requires information to be reported for the area and the implementation of additional pollution control measures.

2.5 Continuous Air Monitoring Stations (CAMS)

The NRC has created criteria for the federal, state, and local agencies to use for determining where to place ozone-monitoring sites. Each area being monitored should have a minimum of two sensors. One sensor should be upwind and another sensor should be downwind of sites with the highest populations. Criteria further clarify that the sensors should be monitored at temporally relevant times when ozone concentrations will be at their annual peak. In 1991, the NRC found that "although each city complies with EPA criteria, the limited number of sites or the placement of monitors calls into question the validity of some city trends. For many rural areas there are no state or local ozone-monitoring requirements. It therefore is likely that there is insufficient monitoring to characterize rural areas and areas at the upwind boundary of many urban locations" (NRC, P. 78, 1991). In TCEQ Region 11, an additional fourteen sensors have been placed since 1999, nine of which are capable of measuring ozone. Despite, efforts to improve coverage using physical monitors, monetary costs make mass distribution of these monitors for collection of high-resolution data impractical. For example, Table 1 contains information from a TCEQ contract for maintaining two CAMS in Fort Worth. The budget has a cost of $35,000 annually for maintaining two CAMS that have already been constructed.

Table 1: Fort Worth CAMS Budget FY17

Budget Category	***Cost for Work to be Performed***
Salary / Wages	*$ 15,500.00*
Fringe Benefits	*$ 6,185.00*
Travel	*$ 500.00*
Supplies	*$ 4,207.06*
Equipment	*$ 0.00*
Contractual	*$ 0.00*
Construction	*$ 0.00*
Other	*$ 3,950.00*
Indirect Costs	*$ 4,657.94*
Total	*$ 35,000.00*

3. LITERATURE REVIEW

3.1 Evaluating Data from CAMS

Air quality monitoring stations are used to evaluate the influence of local and environmental factors on ozone concentrations (Hudak 2014; Munir 2013). Since Galton presented the first linear regression line during a lecture in 1877 (Pearson, 1930), there have been many advanced techniques developed to measure trends and to develop an improved understanding of how variables relate to one another. For instance, when data are sparse, and their resolution is low, agent-based modeling can be used to empirically fill gaps in data through the emulation of behavior-based interactions that affect the distributions and concentrations of pollutants.

Hudak (2014) used ozone monitoring sensors to examine the spatial pattern of ozone in the Dallas-Fort Worth area. The temperature, wind, areas of highest population, as well as the vehicle miles travelled were each considered as possible factors for predicting the distribution and concentration of ozone. While wind direction and temperature were associated with higher concentration of ozone indicated by sensors, the areas with highest populations and vehicle miles traveled were found to not be associated with concentrations (Hudak 2014). Hudak's (2014) research demonstrates that wind speed and wind direction are independent variables that influence the movement of pollution from point-source emissions. Coates (2016) underscores the role of temperature in the production of emissions, such as the VOCS isoprene from vegetation, and in the rate of chemical reactions in the atmosphere.

3.2 Agent-based Modeling and Ozone

Agent-based modeling (ABM) is a system in which agents possess unique characteristics and act according to a set of rules which influence their behaviors and interactions within the model. Through improving technologies and ever-increasing computational power, researchers can examine an increasingly complex set of relationships using agent-based modeling.

ABM is a bottom-up approach to understanding systems whereby individual agents and parameters are introduced to the system and drive the model. For users of Geographic Information Systems (GIS), ABM "provides the ability to model the emergence of phenomena through individual interactions of features within a GIS over time and space" (Crooks, 2015:67). Patterns that emerge from a model based on the interactions of agents can be compared to empirical data for calibration and validation. A calibrated model can predict how changes to the agents might alter the patterns. Data that describe agent behaviors the other relevant parameters can be input into a calibrated model to evaluate environmental impacts in areas that lack sensors or information regarding environmental impacts.

Agent-based modeling has been used to evaluate air quality in several studies. Oprea (2012) used an ABM to create an Air Quality Monitoring and Analysis System (AQMAS) for EPA regions. In a review of ABM for pollution studies, researchers noted that "environment pollution simulation and decision support tools can help decision-makers to set up environmental management policies in order to preserve the ecosystem and ensure public health" (Ghazi 2014:13). Researchers developed an ABM for forecasting air quality using data from monitoring stations and a multi-level architecture

(Papaleonidas et al. 2012). While ABM has been used to assess air pollution, examples of studies that leverage ABM to depict ozone formation and transport are not forthcoming in the literature.

3.3 Ozone – Formation and Transport

Emission events that contribute to the formation of ozone in the troposphere can vary greatly. Base-load power plants are an example of a constant emitter. The U.S. Energy Information Administration (EIA) defines a base load plant as, “a plant, usually housing high-efficiency steam-electric units, which is normally operated to take all or part of the minimum load of a system, and which consequently produces electricity at an essentially constant rate and runs continuously…” (EIA 2018:B).

After precursor gases are emitted form a source such as a base load power plant, early photochemical models suggested that “time for the photochemical production of surface O_3 required a few hours under favorable meteorological conditions” (Chung 1977:1132). Chung’s research made observations about ozone concentrations in response to changing weather systems and meteorological events. Accounting for these systems is important as they can provide a scientific explanation for the movement of ozone from the upper layers of atmosphere to the lower troposphere (Chung 1977).

Ryerson et. al (1988) also researched the transport of ozone caused by precursor emissions by measuring lower and upper limits to ozone production efficiency (OPE) using monitoring data acquired from an aircraft flight pattern ranging from 200 to 400 meters with plume transects approximately 500 meters above ground level (AGL). While decreasing concentrations were found with greater distances from emission point sources, it is important to note that the distances measured at their shortest range were 19 km from the plume site. Also, the data interpretation for this research maintained constant wind speed, direction, emission rates, as well as vertical uniformity. Averaging constants for the purposes of large-scale data crunching as Ryerson did is effective at greater distances from emission point sources.

However, more detailed hourly wind data are useful for the depiction of ozone formation and transport across shorter distances. Similar to Ryerson's measurements of ozone taken from a plane at altitude, tracking of ozone transport can occur at ground-level monitoring sites using elevations above sea level rather than through lightweight aircraft using altitudes above ground-level for the purposes of the research proposed herein.

Research has found that wind speed is correlated to the ascent or descent of gases in the atmosphere. According to Godish et al. (2015:90) "higher wind speeds decrease effective stack height. However, due to the increased volume of air associated with increasing wind speeds, ground-level concentrations are usually reduced. Higher wind speeds decrease the distance at which maximum ground-level (ozone) concentrations (MGLCs) occur." During exceptionally high winds exceeding 50 miles per hour, plume heights may be completely negligible as ozone drops immediately (Godish, 2015). In conjunction with wind direction, elevation, and plume height, wind-speed is also an important factor to help explain the variations between ozone monitor readings.

Munir (2013) studied tropospheric ozone as a secondary pollutant whose concentration could largely be associated with local meteorological variables (Munir 2013). The findings agree with Chung's (1977) research. Munir's dissertation work examined the impacts of ozone on human health, agricultural, and materials and buildings. Munir applied quantile regression modeling and generalized additive modeling. This type of modeling efficiently identifies correlations between variables. Variables used for this type of modeling are generally derived from large sets of aggregated data. When sources are spaced and distributed in close enough proximity, interpolation can be achieved at a high level of accuracy. This research creates representative point clouds based on the winds movement of precursor gas emissions in an ABM to interpolate ozone risk.

4. PROBLEM STATEMENT AND RESEARCH PURPOSE

To understand the impact of point-source pollutant emissions on our atmosphere, it is important that sensors are properly located from sources of pollution in relation to the dominant wind directions during times of peak ozone to capture the most accurate data. Allen (2004:3) found a strong correlation between ozone concentration and temperature. Higher temperatures produce "bad ozone" days. The TCEQ has an ozone-forecast season, a period of the year when ozone is more likely to form and when levels are likely to threaten human health. For instance, the Central Texas ozone-forecast season is April 1st to October 31st. With climates becoming progressively warmer in Texas due to climate change, it is likely that additional ozone monitors will continually need to be placed to survey the affected areas (human, wildlife, plants, and structures) of rising ozone content in the troposphere.

In 1991, the National Research Council (NRC) concluded that two monitors might be adequate to measure ozone concentrations if they are located upwind and downwind of the areas of highest population (NRC 1991). This recommendation from the NRC is intended to monitor emissions of ozone precursor gases. However, the point sources identified by organizations like the Texas Commission on Environmental Quality (TCEQ) are often not located in the areas of highest population density. Industrial sites are often distant from the highest population densities due to the nature of industrial and residential development patterns and land use regulations. In addition, prevailing wind patterns may vary directionally throughout the year or during particular seasons, even during the ozone-forecast periods. Winds can carry high concentrations of ozone away from monitors and measures of ozone at monitors may not accurately reflect the true problem of ozone in a metropolitan area. For these reasons and more, engagements with ABM may provide valuable insights for monitor placement, as well as for better understanding patterns of tropospheric ozone and its transport.

5. DATA AND METHODS

5.1 Conceptual Model and Analytical Approach

For present purposes, the first step toward developing an ABM involved creating a map depicting the patterns of the potential for ozone formation from point-source emission data from the TCEQ 2010 point-source emission inventory. These maps were used to identify the areas that would be used to calibrate the ABM. Meteorological data from the station at Austin-Bergstrom International Airport was used to reflect hourly wind speed measurements, wind directions, and hourly temperature data for the model. The model output was combined with elevation data to discern the relationship of emissions and transport of ozone precursors to ozone concentrations throughout the study area.

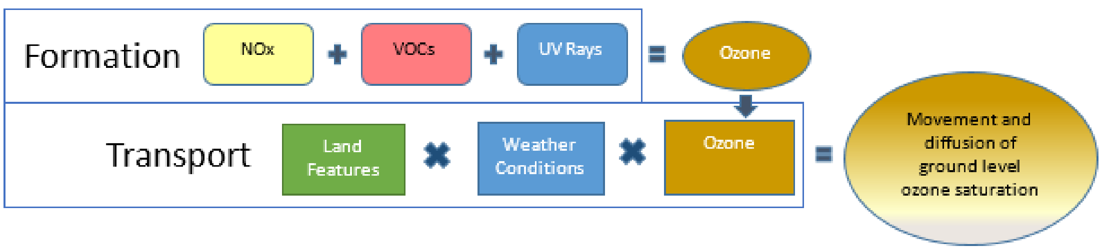


Figure 1: Simplified Conceptual Model of Ozone Behavior

The processes of ozone formation and transport are continuous and simultaneous (Figure 1). The ABM developed in this thesis is relatively simple and two dimensional, as it does not model detailed chemical reactions, nor does it factor in vertical movements of the atmosphere. Calculating temporal rates of ozone formation and decay are complex processes and many factors influence their behavior. Consequently, for this inquiry—which is designed to show above all the potential for ABM to model the transport of O_3—the following overarching assumption is adopted: ozone will form in the presence of NO_x and VOCs but the true location(s) and moment(s) at which formation occurs are indeterminate. On that backdrop, in this study's ABM, NO_x and VOCs are assumed to be emitted from point sources at a constant hourly rate based on the hourly average portion of the annual total from 2010. The pollutants are assumed to conform to simple horizontal movement based on normal average hourly wind-speed and wind-direction. Grid units in the modeled study area reflect the cumulative sum of NOx and VOCs that pass through them.

To create the ABM, this research relied on Netlogo (Wilensky, et al. 1999) and supplemental packages. Exporting data from the ABM into tables with coordinate location data allowed for an analysis of data using ArcMap 10.5.1 and for the production of evidentiary visualizations that depict the nature of ozone formation and transport throughout the troposphere. More precisely, upon building and running an ABM, outputs of the accumulation pattern described above were exported as CSV files in varying temporal units. To visualize the patterns, 8,688 sequential screenshots were captured and merged into an AVI video file to depict ozone formation in the study area for the entire year. This video displays hourly ozone concentrations, hourly windspeeds, hourly wind directions, hourly temperatures, and the temporal details to the hour. Each second of video corresponds to one hour of time. The video is approximately 20 minutes in length.

The ABM results were then imported into ArcMap to visualize the potential patterns of ozone. It is assumed that when the precursors are present in the same two-dimensional space, ozone forms. The temporal scope of the GIS analysis undertaken in this study is June through October 2010. The period between April and October is a period in which temperature tends to be higher and conducive to promoting ozone formation and UV rays are increasing in intensity in the troposphere in the northern hemisphere. April and May 2010, however, had only two ozone monitors in operation in the study area.

5.2 Data Acquisition, Data Processing, and Statewide Assessment

Data were acquired from the TCEQ, the Department of Transportation, and the U.S. Census Bureau. ArcMap 10.5.1 and 10.6 were used for initial data processing and area assessment. Shapefiles and datasets used for the initial planning and preparation of this model were available as aggregated sets encompassing the State of Texas.

The TCEQ divides Texas into sixteen management regions. Texas county boundary shapefiles were acquired from the Texas Department of Transportation. TCEQ maps were consulted to divide Texas by TCEQ region in the GIS after adjoining point-source data to the layer by county. Census tract shapefiles with 2015 population data were acquired from the U.S. Census Bureau via the American Fact Finder website.

The State of Texas Air Reporting System (STARS) tracks point-source emissions from sites such as power plants and industrial facilities. "Only those sites whose emission rates exceed the reporting applicability levels found in 30 Texas Administrative Code 101.10 are tracked. Currently, there are approximately 2,000 industrial sites updating emissions annually" (TCEQ 2015). The TCEQ maintains an annual inventory of point-source emissions. The point-source emission data for 2010 was downloaded as a spreadsheet from the TCEQ, and this dataset was joined to the county units by matching to county names.

The TCEQ lists 260 public and private air quality monitors in Texas. Of these, only 83 are equipped with ozone sensors. Most of these sensors are in central and southeastern Texas. The monitoring data were downloaded in a table containing pertinent data such as CAMS identification number, location description, operators name, longitude, latitude, activation date, and, if applicable, a deactivation date.

Areas with high likelihood for ozone formation were determined by using the 2010 TCEQ point-source emission inventory data. In the GIS, these data were joined to the county in which the sites are located. Emission data were summed by county to facilitate choropleth mapping of annual emissions of NO_x and VOCs from point-sources. Finally, NO_x and VOCs were separately standardized on zero-to-one scales and a county-scale ozone-production map was created. This map was used to identify the areas in each region that were suitable for calibrating the ABM.

According to the U.S. Census Bureau, referring to its administrative geographies, an urban area as "any block or block group having a population density of at least 1000 people per square mile." (NAL 2016). Census tracts in the state of Texas are divided into two distinct areas based on this definition: urban areas (in maroon) and non-urban areas, the latter of which are coded as "rural" in Figure 2.

The majority of the urban census tracts in Texas have ozone sensors located within the urban area and +/- 10 miles of the urban/non-urban boundary as suggested by the NRC's criteria (Figure 2). The four largest metropolitan areas—Dallas, Fort Worth, San Antonio, and Houston—are relatively well-covered by ozone sensors compared to other areas. Urban areas in West Texas that are not located on the US-Mexico border lack ozone monitoring stations. Indeed, Lubbock, Midland, and Amarillo have no ozone monitors. Jeannie Allen of NASA cautions that "ozone levels are generally higher downwind of ozone precursor sources, at distances of hundred or even thousands of

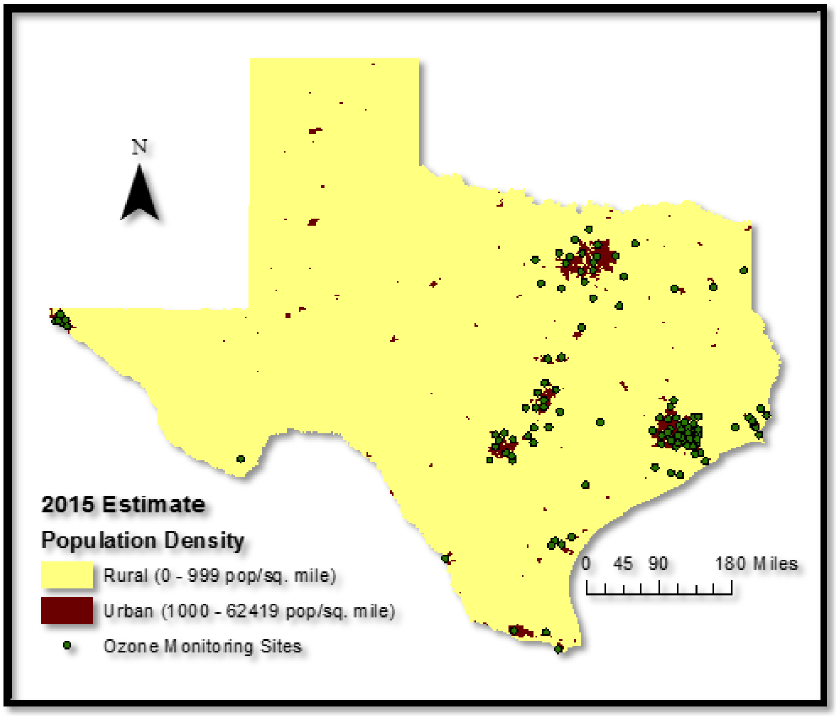


Figure 2: Population Density and Ozone Sensor Placement

kilometers, so ozone concentrations in rural areas can be higher than in urban areas" (Allen 2002). There are currently no ozone sensors in West Texas to record the potential paths of ozone as described by Allen.

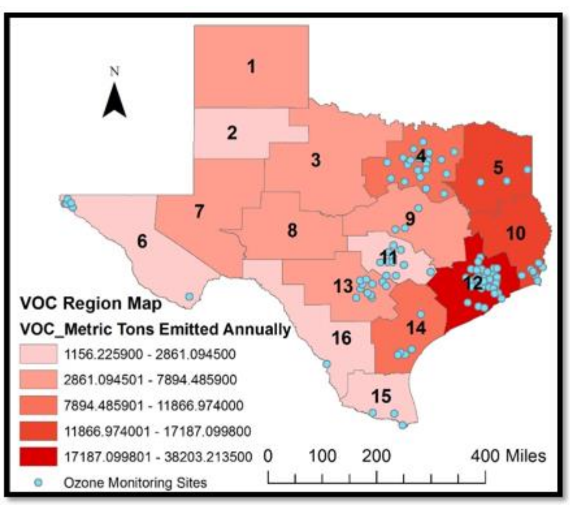


Figure 3: VOCs Region Map

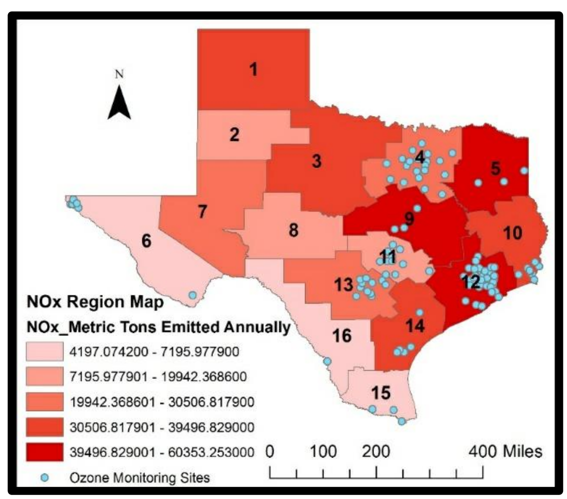


Figure 4: NO_x Region Map

VOCs emissions from industrial sites are required to report emission data to the TCEQ (Figure 3). Regions 4, 5, 10, 12, and 14 have the highest rates of emissions of VOC. The greatest emissions of NO_x from industrial sites are occur in regions 1, 3, 4, 5, 7, 9, 10, 12, 13, and 14 (Figure 4). Regions 4, 5, 10, 12, and 14 have high emissions of both VOCS and NO_x. The combination of these two precursor emissions make these regions susceptible to high concentrations of ozone.

County-level emission data allow one to look more closely at the spatial relationships between the geographies of precursor emissions and of ozone sensors (Figures 5 and 6). The clustered placement of stations in the southeastern part of Region 10, for instance, appears to coincide with the high number of emission sources of VOCs in Jefferson and Orange counties. Jefferson and Orange counties are at the southern edge of Region 10, where annual emissions are high and there are many monitors.

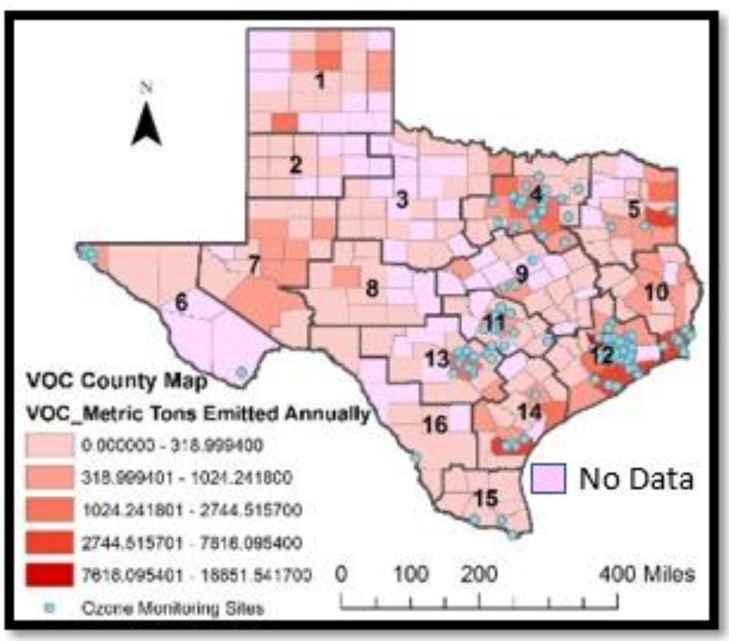


Figure 3: VOCs County Map

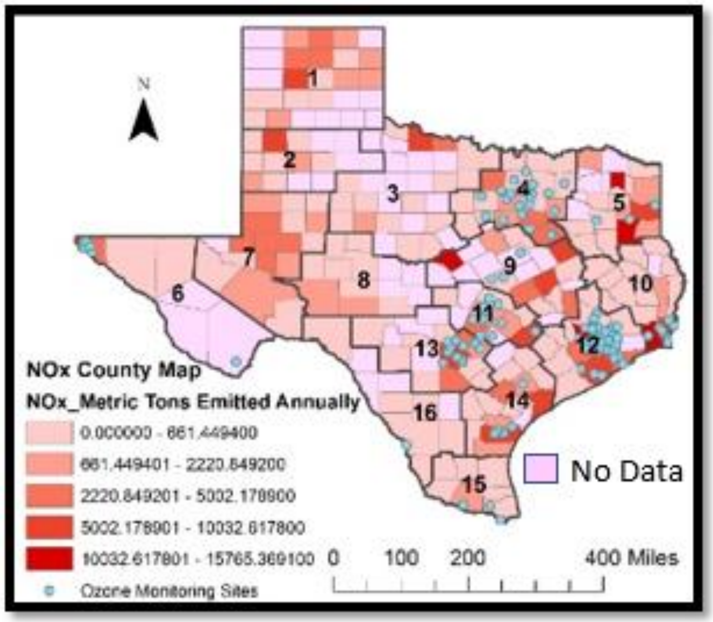


Figure 4: NOx County Map

County-level vector data for VOCs and NO_x emissions were converted into raster format. The data were standardized for each gas on a concentration-scale from zero to one. Then, the standardized values were summed to produce a map of patterns of ozone formation (Figure 7).

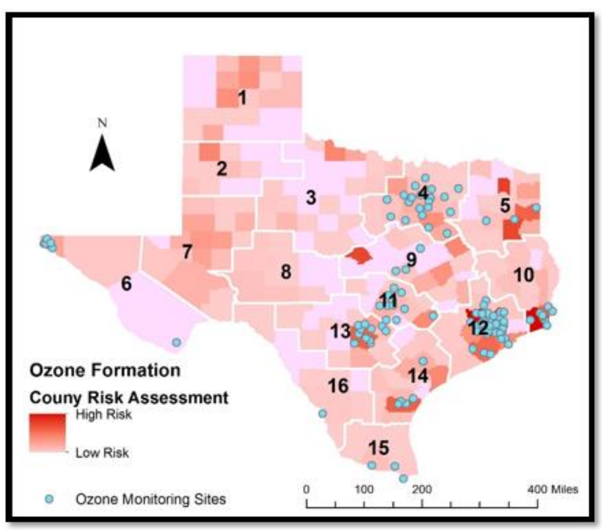


Figure 5: Ozone Formation County Risk Assessment

Five counties can be identified by their mid-red color indicating moderate levels of emissions: Hutchinson and Castro counties (Region 1), Wise County (Region 4), Cass County (Region 5), and Calhoun County (Region 14) (Figure 7). Crucially, these areas, which are characterized by moderate emissions levels, have no ozone monitors. Such counties therefore stand to gain from alternative means of visualizing and studying emissions via ABM.

Several additional areas with relatively high levels of emissions also lacked ozone monitors. Based on the data employed in this study, Mills County in Region 9 is characterized by some of the highest potential for ozone formation. Titus County, Harrison County, and Rusk County (all in Region 5) were similarly likely to have high ozone levels. In Region 9, Freestone County, Robertson County, and Mills County were areas of high ozone potential. At the same time, none of the counties mentioned in the preceding sentences contain ozone monitors.

5.3 Agent-Based Modelling to Depict Ozone Transport.

Region 11 is used as a case study herein to demonstrate how an ABM can be useful for modeling and visualizing ozone formation patterns. There are seven ozone monitors in Region 11. Within Region 11, Fayette County is the county with the greatest emission of the ozone precursors (Figure 8). The Fayette County Coal Plant produces more criteria air pollutant tons per year than all other point-source emitters in Region 11 combined. This particular generating station was used below to calibrate the study's ABM. An advantage to using the Fayette County Coal Plant to calibrate the ABM is that it is a base-load power plant, which means that emissions are produced at a constant rate throughout the year with little variation. Additionally, there is a monitor 5.18 km (3.22

miles) from the plant generally located downwind in the average ozone-forecast-season wind direction.

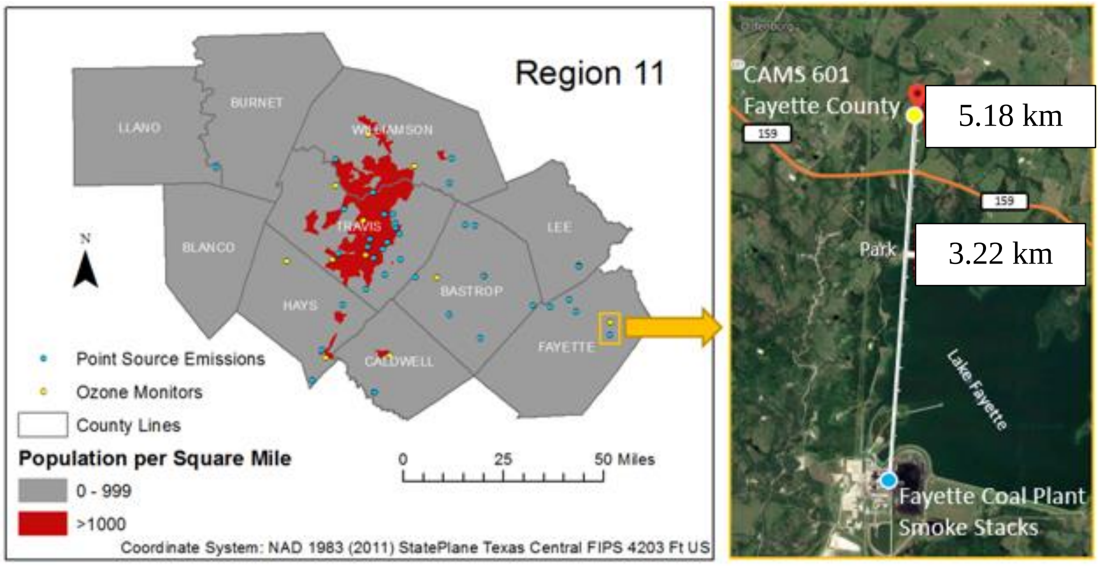


Figure 6: Fayette County Coal Plant

The ABM has a .32 km (.2-mile) resolution and the pollutants are emitted according to TCEQ annual point-source data at a constant rate based on annual TPY every 12 minutes. Hourly wind speed and direction recorded at the Austin-Bergstrom International Airport were global variables that are assumed to indicate the exact movement of all pollutants present in the model. The wind data are processed at a rate equivalent to every 12 minutes. For example, if the wind speed is 5 miles per hour and the wind direction is blowing north, the pollutants all move 1 mile in the north direction every tick (12-minutes). Data were exported as CSV files for analysis.

The ABM assigns a value equivalent to 12 minutes of the annual production of NO_x and VOCs from each pollutant's emission point-source to the pollutant agent. It is assumed that these assigned values do not decay or otherwise change until the emission agent departs the study area, at which point the agent ceases to exist in the model. These emission values are copied to patches on which the ozone arrives during transport. The patches accumulate the sum of the values from NO_x and VOCs pollutant emissions. For visualization purposes, the model maintains a visual tracer effect that gradually changes color from a white to a dark red until over 5 hours, at which time the color of the patch returns to its original color. Any emissions arriving at the fading patch will restart the fading effect. Additional information about the model can be found in the NetLogo code provided in Appendix A.

5.4 Analyzing ABM Results in GIS

Data from the ABM output were imported to ArcMap as CSV files. From there, the data were plotted using the latitude and longitude coordinates of each patch with or without the pollutants recorded by patches within the ABM. Approximately 300,000 points were analyzed using a natural-neighbor analysis, once for NO_X and once for VOCs. Th natural-neighborhood tool in ArcMap finds the closest group of input samples from query points and applies a proportional weight to them based on the areas of Voronoi polygon used to interpolate values (Sibson 1981). This tool allows incorporation of a magnitude field. The accumulated NO_x and accumulated VOCs values were used in the magnitude field to create their respective maps.

After this mapping process, the NO_X and VOCs raster files were standardized based on the maximum magnitude value. All values for each raster were divided by their raster’s maximum value using the ArcMap Raster Calculator tool to standardize values from zero to one. Finally, these standardized values were summed using the ArcMap Raster Calculator, and proof-of-concept maps displaying transport routes of ozone across the troposphere were generated.

6. RESULTS

6.1 Summary

Annual point-source emissions data for NO_x and VOCs from 2010 were agents in the agent-based model of ozone transport in TCEQ Region 11. The model included seven ozone monitors that recorded the hourly ozone concentration at their sites. The model output can be used to visualize patterns for each of the seven monitors (Figure 9.)

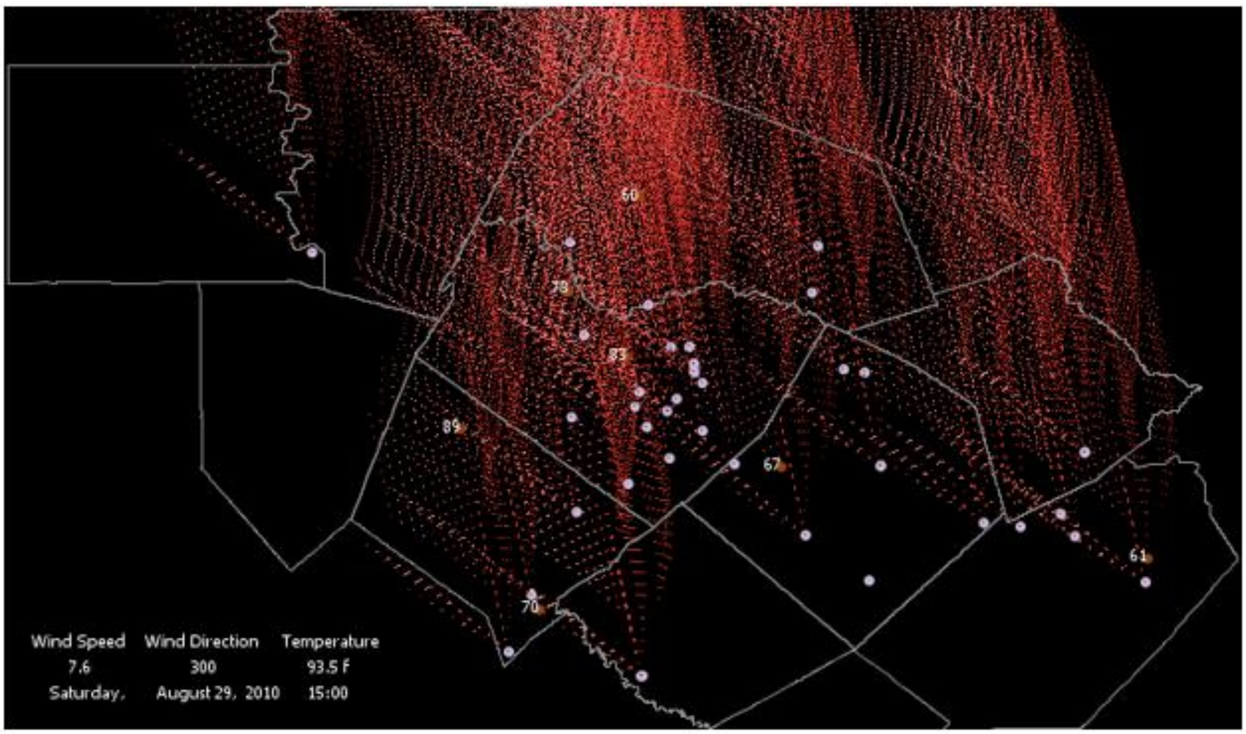


Figure 7: Agent Based Model Output Window - An example of the agent-based model output window is used to visualize data from August 29, 2010 at 15:00, when ozone concentrations were recorded at their highest concentration in 2010 for the Dripping Springs sensor located at northwest Hays County.

The emissions appear as small white-to-red colored dots on the screen. Each emission is assigned a value that is proportional to the amount of NO_x and VOCs produced by the point source from which it was emitted. The resolution of the patches are 0.32 km (0.2 miles). Each 0.32 km (0.2 miles) patch maintains a sum of the NO_x and VOCs agents that have arrived at its location.

A natural-neighbor analysis of the monthly data exported from the ABM can be used to produce continuous raster layers with standardized values to show the modeled tropospheric ozone transport patterns and relative concentration levels (Figure 10).

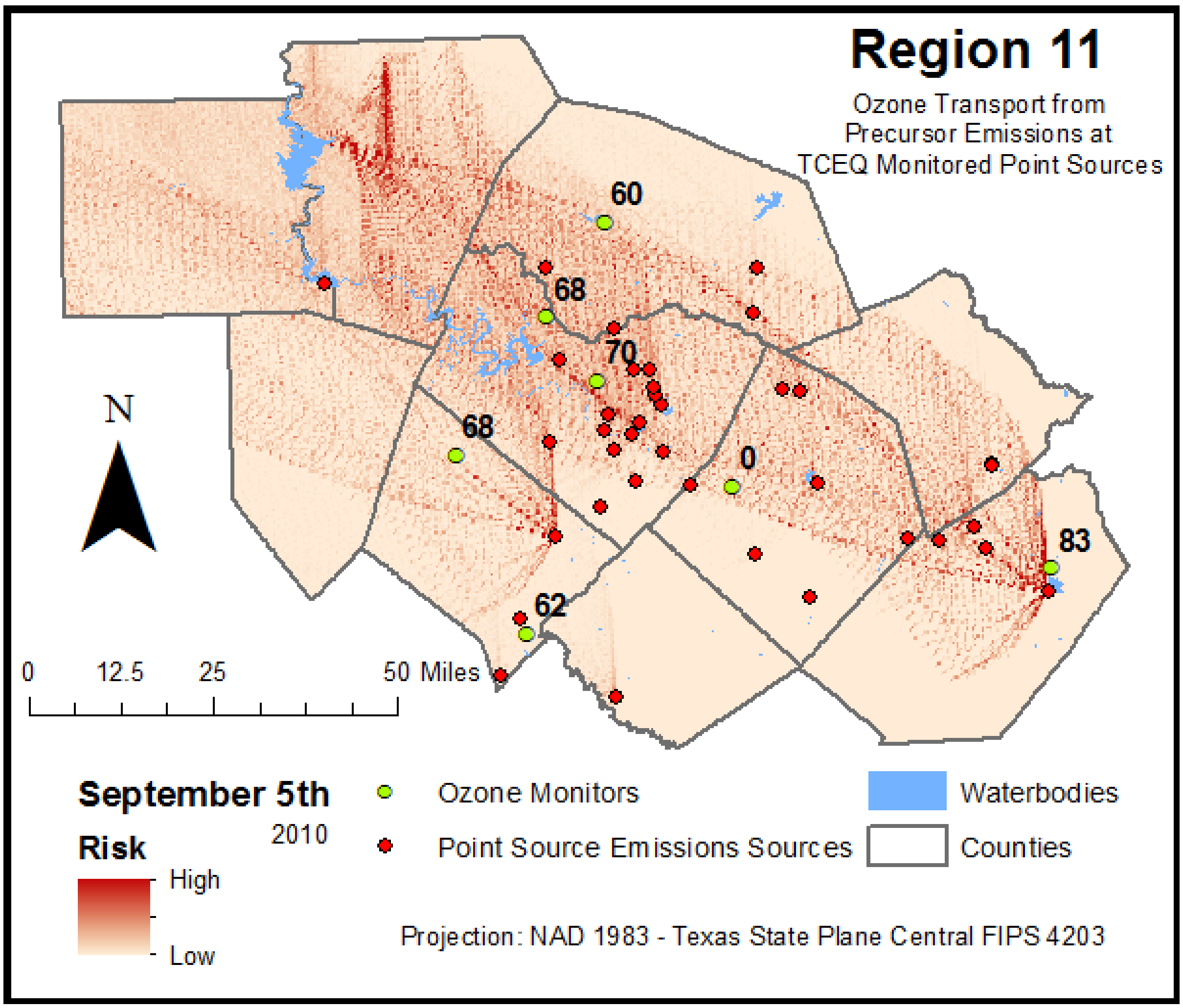


Figure 8: ABM Paths for Ozone Transport - Probable paths for ozone transport based on wind direction and wind speed for 24-hours on September 5th, 2010.

6.2 Topographic Influence on Transport Behavior

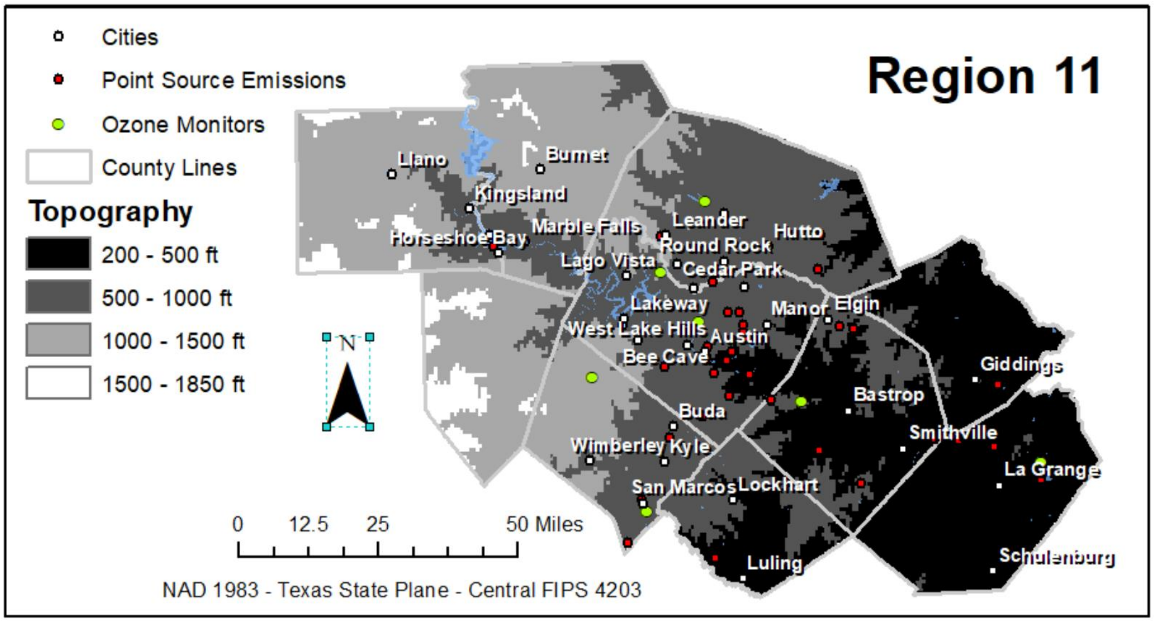


Figure 9: Region 11 Elevation Map

The elevation of Region 11 increases gradually from east to west (Figure 11). A monitoring station is located in Dripping Springs in the northwestern corner of Hays County, west of Bee Cave. Ozone monitoring in the ABM becomes active when winds push NO_x and VOCs emissions southward and westward. When this occurs, there tends to be a substantial increase in ozone concentration at the Dripping Springs monitor. The rise and fall of the ozone concentration appears to coincide with the transport of emissions. The monitor is located at least 500 feet higher in elevation than most of the point-source emissions in the Austin area and more than 1000 feet higher in elevation than the power plant in Fayetteville. The Dripping Springs monitor is higher than the plume height of every point-source in the region. When wind is flowing westward or southward, Dripping Springs becomes more susceptible to ozone precursors and ozone.

6.3 *Wind Influence on Transport Behavior*

Wind is predominately from the south during the ozone-forecast season. This pattern is interrupted by occasional eastward flows sending the pollution to the east in a wave-like pattern swinging from westward and northward (Figure 12).

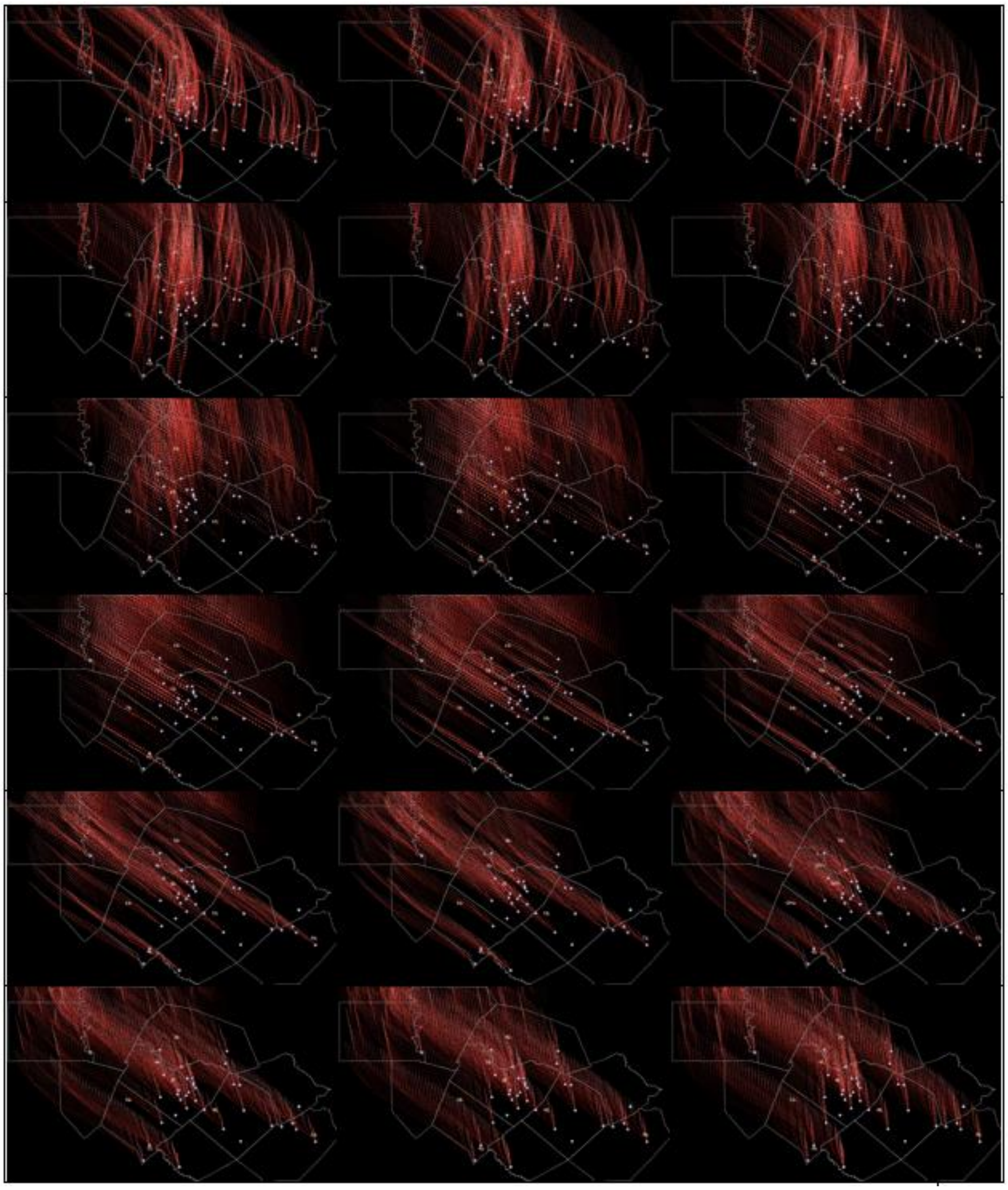


Figure 10:Example of Wind Fluctuation Pattern

Data from the patches were aggregated by month to analyze the emissions intersecting each patch in June, July, August, September, and October 2010. Two natural neighbor analyses were performed using NO_x and VOCs as the magnitude variable, respectively. Monthly averages of ozone monitoring data from each monitor were ranked from highest to lowest ozone concentrations (Figure 13). The ranks remain stable from month to month among the sensors with the highest four concentrations for the months of June, July, and August, when the wind maintained a consistent northerly pattern. There is a significant shift in the monitor's ozone concentration rankings when the predominant wind flow shifted to the west during September and October. During the early autumn, the prevailing wind direction tracks point-source emissions from downtown Austin and the Fayetteville power plant directly toward Dripping Springs.

The shift in historical ozone concentration rankings appears to coincide with the shift in the ozone generation predicted by the ABM (Figure 13). The modeling and visualization of data using the ABM and GIS reveals the relationship between ozone precursor emissions from and ground-level ozone concentrations at monitors.

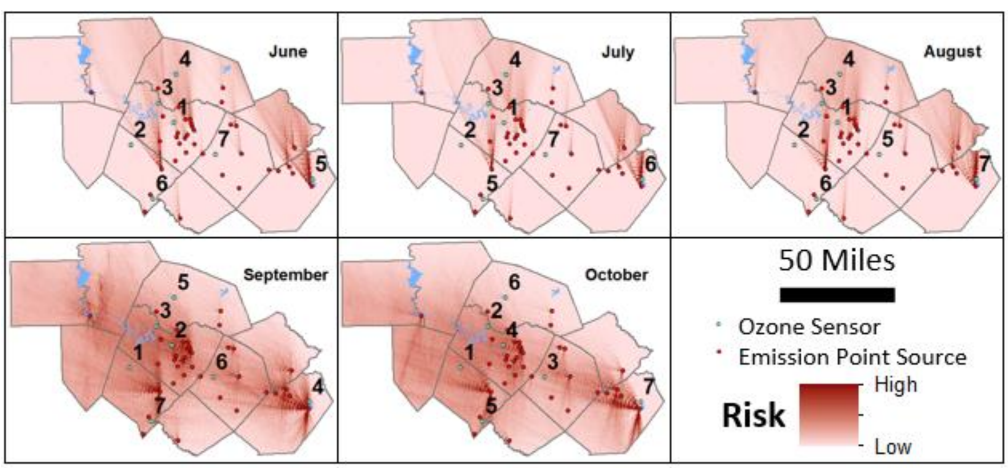


Figure 11: Relationship of Point Source Emissions to Ground-Level Ozone Concentration

7. EVALUATION OF ABM AND RESULTS

7.1 Evaluation Using Peak Daily One-Hour Ozone

This section compares historical peak daily one-hour ozone monitor readings that exceed 80 parts per billion (ppb) to the ABM's daily arrival of emissions from identified point-sources (Appendix B). An emission is considered an arrival if it is located within a 1-mile radius of the monitoring site (Appendix C) on the same day.

Table 2: 2010, Days with Peak Daily One-Hour Ozone Concentrations Higher than 80

Date	Peak Daily One-Hour Ozone	Monitor
May 4th	85	CAMS 3
May 5th	90 84	CAMS 3 CAMS 38
May 28th	86 81	CAMS 3 CAMS 614
May 29th	85	CAMS 3
June 4th	86 82	CAMS 3 CAMS 614
August 12th	87 81	CAMS 3 CAMS 38
August 17th	86 80	CAMS 3 CAMS 38
August 27th	82	CAMS 614
August 28th	89 83 81	CAMS 614 CAMS 3 CAMS 38
September 5th	83	CAMS 601
October 8th	81 80	CAMS 3 CAMS 614

CAMS 3 has the most peak daily one-hour ozone concentrations over 80 and was further examined for that reason. For the dates of May 4th through August 28th, the model found eleven point-source emissions to have contributed to emission arrivals at CAMS 3 to varying degrees on each of these days. October 8th was excluded from this analysis because it presented a far different makeup of point-source emission contributors.

TH0787H and BC0041K were excluded because their emissions of NO_x and VOCs are zero tons per year. TH0191A was excluded because it's emissions were negligible. Table 4 and Figure 14 show the counts of emission arrivals to CAMS 3 from each point source. Figure 14 also shows the ozone concentration on each day for CAMS 3.

Table 3: CAMS 3 Emission Arrival Counts

Account	May 4	May 5	May 28	May 29	Jun 4	Aug 12	Aug 17	Aug 28
TH0015V	38	32	34	29	33	34	36	5
TH0065G	10	12	2	2	0	0	0	47
TH0104V	117	123	95	97	96	35	47	42
TH0142N	57	66	51	55	64	20	20	43
TH0502F	38	46	16	22	25	15	11	9
TH0760E	15	14	4	3	5	0	7	26
HK0014M	0	0	0	0	0	10	11	0
BC0057S	13	9	0	0	0	0	0	4
BC0083R	17	13	0	0	0	0	0	11
CA0011B	0	0	0	0	4	5	4	16
CA0027J	0	0	0	0	5	7	5	18

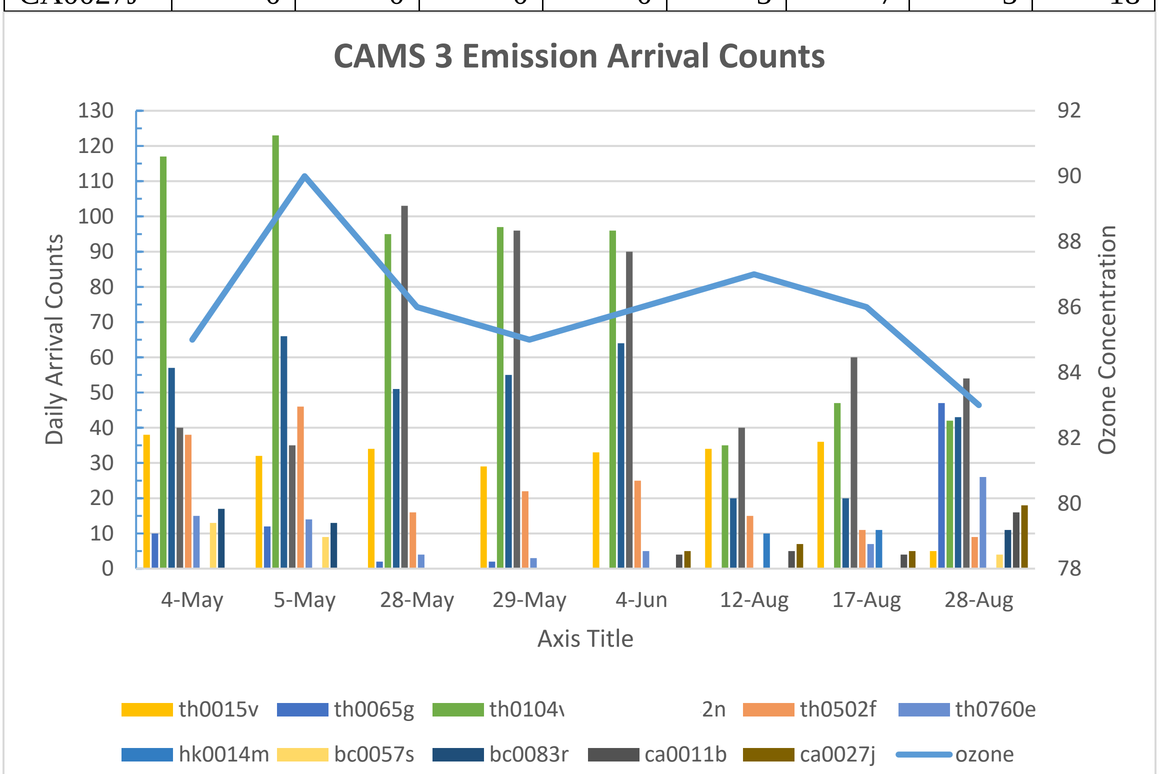


Figure 12: CAMS 3 Emission Arrival Counts

Next, the arrival counts for the twelve point-sources were given NO_x, VOCs, and NO_x/VOCs combined impact scores.

Table 4: NOx, VOCs, and Combined Impact Scores

NO_x Impact Scores								
Account	May 4	May 5	May 28	May 29	Jun 4	Aug 12	Aug 17	Aug 28
TH0015V	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.00
TH0065G	0.02	0.03	0.00	0.00	0.00	0.00	0.00	4.63
TH0104V	7.39	7.77	6.00	6.13	6.06	2.21	2.97	2.65
TH0142N	0.06	0.07	0.05	0.06	0.07	0.02	0.02	0.04
TH0502F	0.25	0.30	0.10	0.14	0.16	0.10	0.07	0.06
TH0760E	0.30	0.28	0.08	0.06	0.10	0.00	0.14	0.53
HK0014M	0.00	0.00	0.00	0.00	0.00	3.35	3.68	0.00
BC0057S	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
BC0083R	0.50	0.38	0.00	0.00	0.00	0.00	0.00	0.33
CA0011B	0.00	0.00	0.00	0.00	0.11	0.14	0.11	0.44
CA0027J	0.00	0.00	0.00	0.00	0.03	0.04	0.03	0.10
VOCs Impact Scores								
Account	May 4	May 5	May 28	May 29	Jun 4	Aug 12	Aug 17	Aug 28
TH0015V	6.83	5.75	6.11	5.21	5.93	6.11	6.47	0.90
TH0065G	0.96	1.15	0.19	0.19	0.00	0.00	0.00	4.51
TH0104V	5.31	5.58	4.31	4.40	4.36	1.59	2.13	1.91
TH0142N	3.98	4.61	3.56	3.84	4.47	1.40	1.40	3.01
TH0502F	1.49	1.80	0.63	0.86	0.98	0.59	0.43	0.35
TH0760E	0.26	0.24	0.07	0.05	0.09	0.00	0.12	0.44
HK0014M	0.00	0.00	0.00	0.00	0.00	7.99	8.79	0.00
BC0057S	0.38	0.26	0.00	0.00	0.00	0.00	0.00	0.12
BC0083R	1.39	1.06	0.00	0.00	0.00	0.00	0.00	0.90
CA0011B	0.00	0.00	0.00	0.00	0.48	0.59	0.48	1.90
CA0027J	0.00	0.00	0.00	0.00	0.60	0.84	0.60	2.15
NO_x/VOCs Combined Impact Scores								
Account	May 4	May 5	May 28	May 29	Jun 4	Aug 12	Aug 17	Aug 28
TH0015V	6.85	5.77	6.13	5.23	5.95	6.13	6.49	0.90
TH0065G	0.98	1.18	0.20	0.20	0.00	0.00	0.00	4.63
TH0104V	12.70	13.35	10.31	10.53	10.42	3.80	5.10	4.56
TH0142N	4.04	4.68	3.62	3.90	4.54	1.42	1.42	3.05
TH0502F	1.74	2.10	0.73	1.01	1.14	0.69	0.50	0.41
TH0760E	0.56	0.52	0.15	0.11	0.19	0.00	0.26	0.97
HK0014M	0.00	0.00	0.00	0.00	0.00	11.34	12.47	0.00
BC0057S	0.38	0.26	0.00	0.00	0.00	0.00	0.00	0.12
BC0083R	1.89	1.45	0.00	0.00	0.00	0.00	0.00	1.23
CA0011B	0.00	0.00	0.00	0.00	0.59	0.73	0.59	2.34
CA0027J	0.00	0.00	0.00	0.00	0.63	0.88	0.63	2.26

NO_x, VOCs, and combined impact scores were developed using the following equations:

$$S.NO_x = NOxTPY_i \, / \, MaxNO_xTPY$$

$$S.VOCs = VOCsTPY_i \, / \, MaxVOCsTPY$$

$$NO_xImpact = S.NO_x * Arrivals$$

$$VOCsImpact = S.VOCs * Arrivals$$

$$CombinedImpact = (S.VOCs + S.NO_x) * Arrivals$$

S (Standardized), TPY (Tons Per Year), Max (Highest Ozone Concentration)

The results (Table 5) were multiplied by ten to aid in interpretation.(Figures 15, 16, and 17).

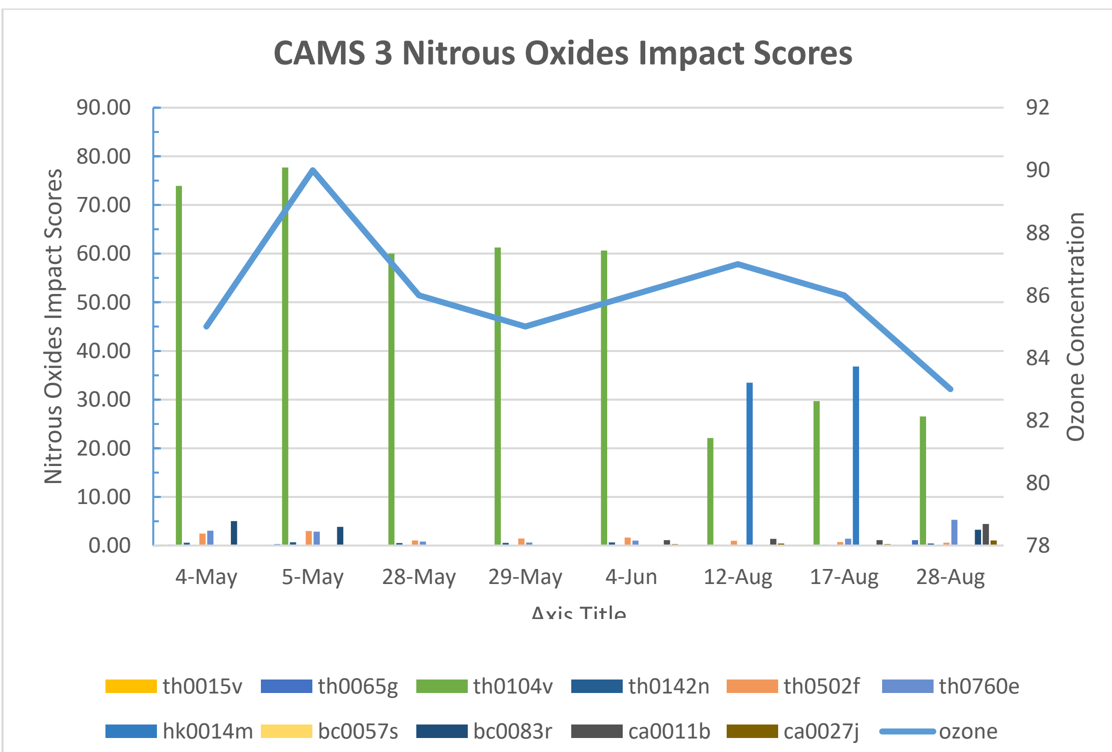


Figure 13: CAMS 3 Nitrous Oxides Impact Scores

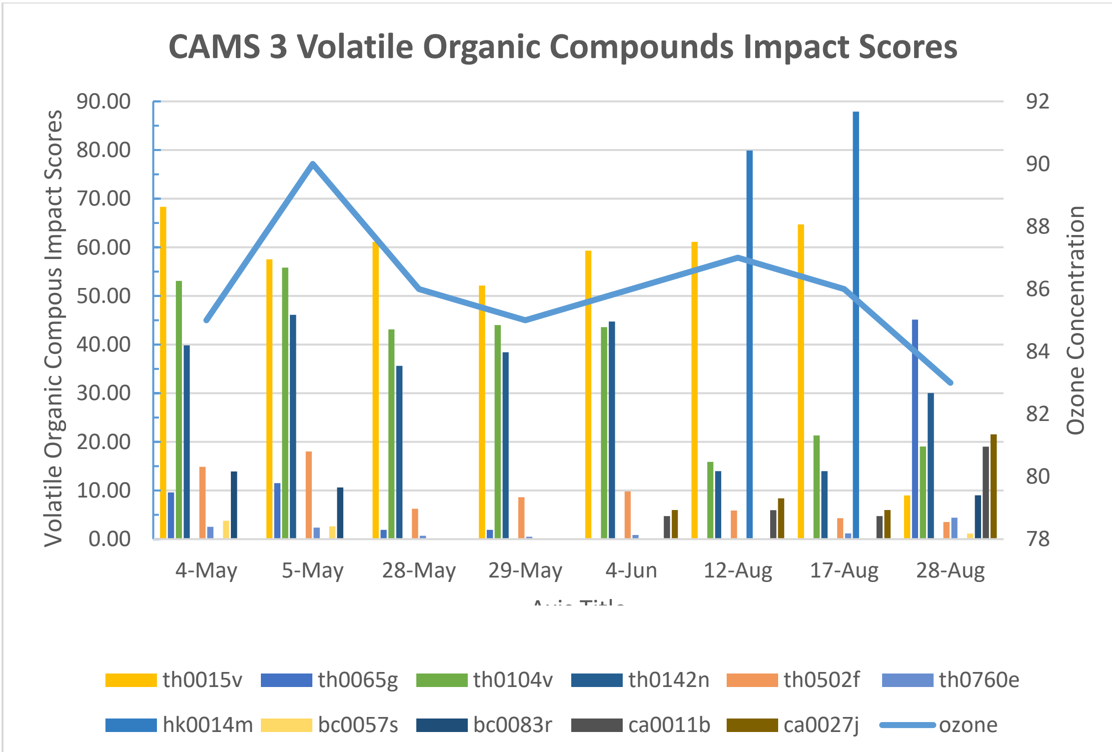


Figure 14: CAMS 3 Volatile Organic Compounds Impact Scores

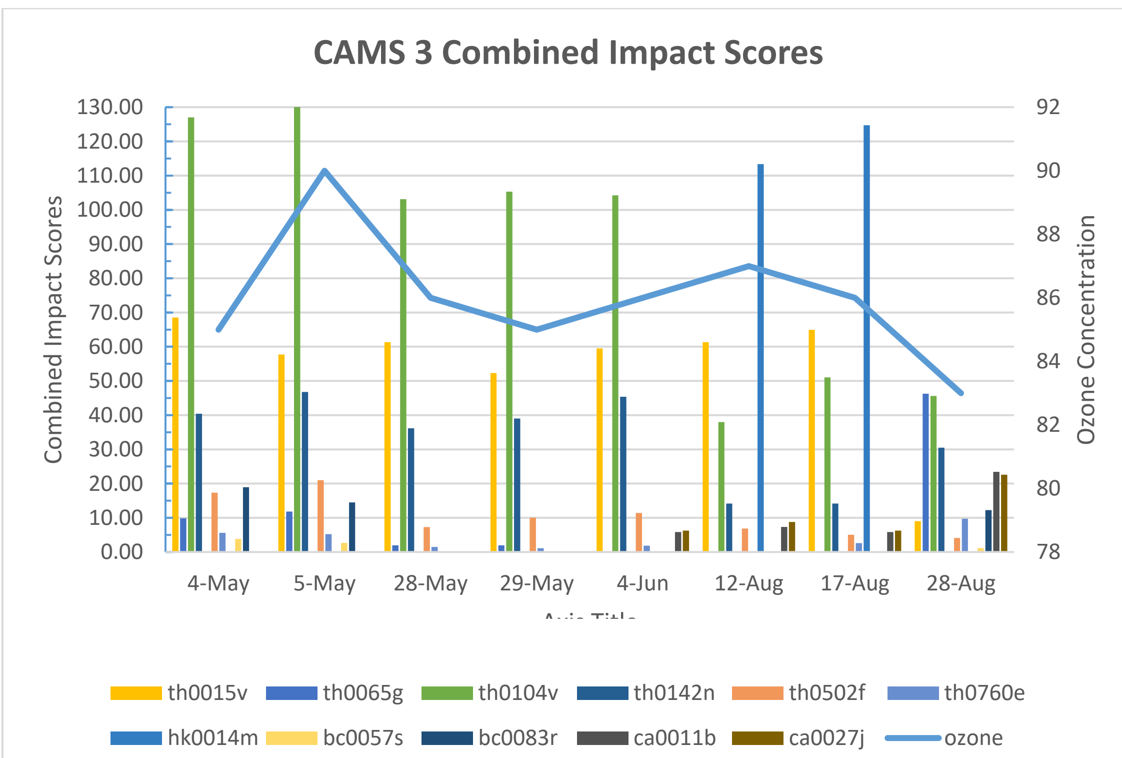


Figure 15: CAMS 3 Combined Impact Scores

A scatterplot was used to develop trendlines with correlation coefficients to evaluate whether combined impact scores were correlated with higher or lower ozone concentrations at CAMS 3(Figure 18).

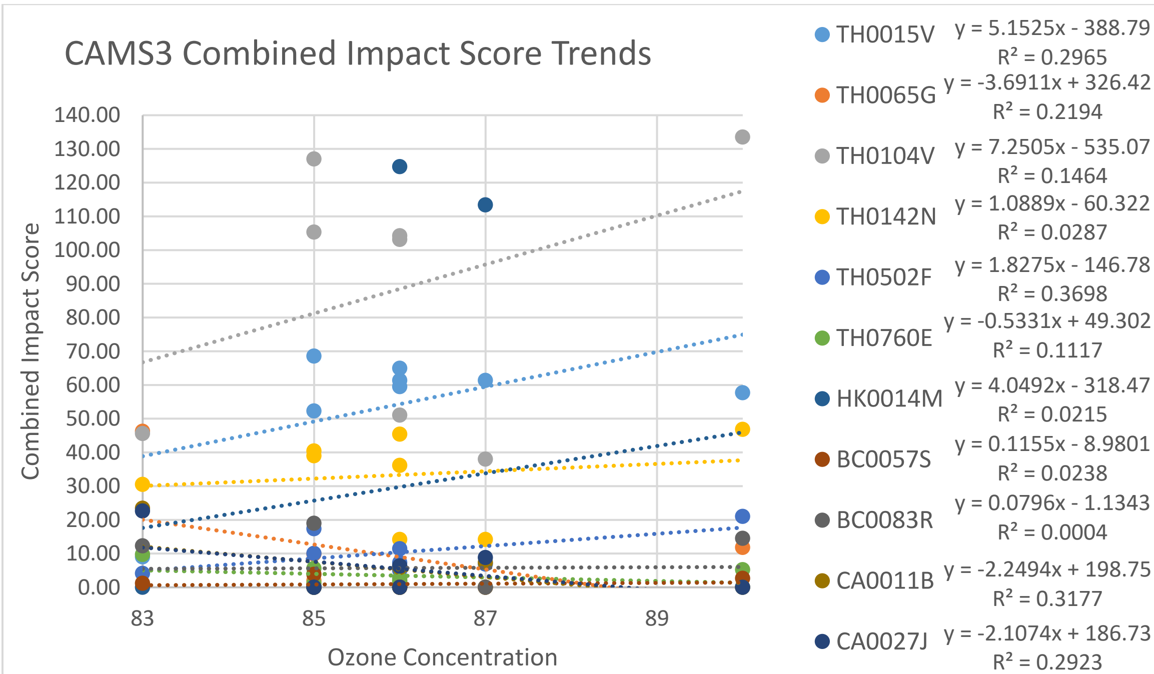


Figure 16: CAMS3 Combined Impact Score Trends

Next, the elevation of the point-source emissions in relationship to the elevation of CAMS3 was considered. The ABM represents ozone crossing the x, y location of a monitor across a 2-dimensional plane. Because the model did not factor in elevation, the ozone that is predicted to have formed from point-source emissions may not actually have reached the monitors, despite crossing their two-dimensional space at the x, y location (Table 6).

Table 5: Elevation of CAMS and Point-Source Emissions

Place	Elevation (Meters)	Distance (KM)	Combined Impact R^2
CAMS 3	232		
TH0015V	136	13.9	.2965
TH0065G	148	13	.2194

Table 5. Continued

Place	Elevation (Meters)	Distance (KM)	Combined Impact R^2
TH0104V	174	7.9	.1464
TH0142N	188	15.5	.0287
TH0502F	166	23.2	.3698
TH0760E	125	21.4	.1117
HK0014M	227	34.7	.0215
BC0057S	132	51	.0238
BC0083R	166	30.7	.0004
CA0011B	138	68.8	.3177
CA0027J	138	68.8	.2923

Wind speed, direction, and temperature were considered next (Table 7).

Table 6: Wind Speed, Wind Direction, and Temperature Model Screenshots

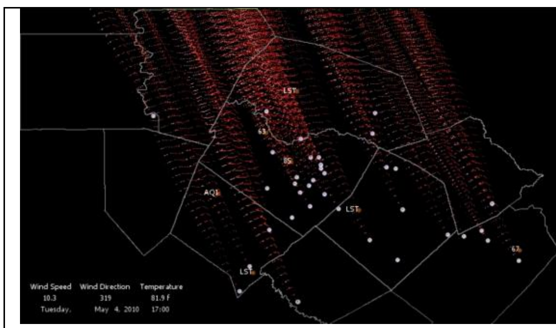
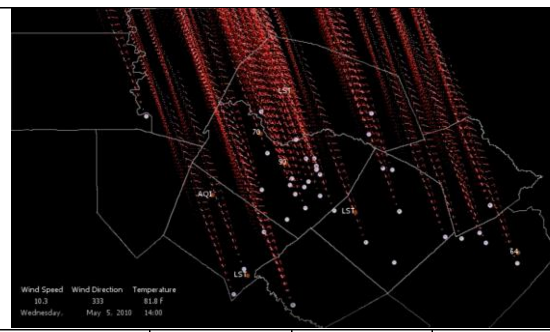
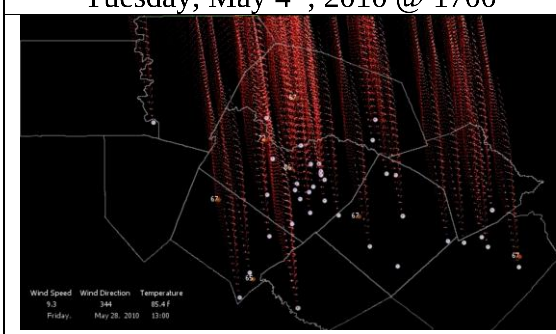
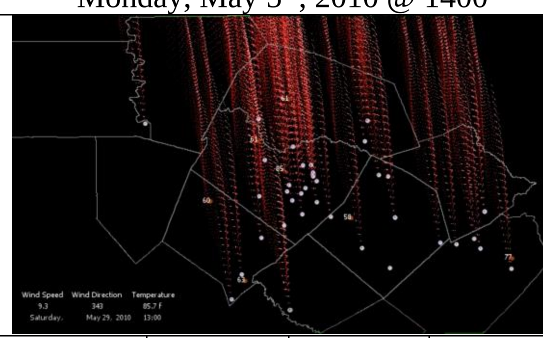
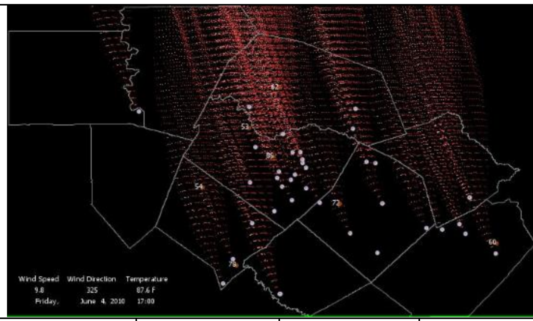
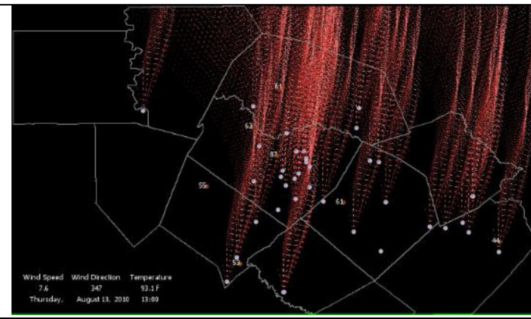
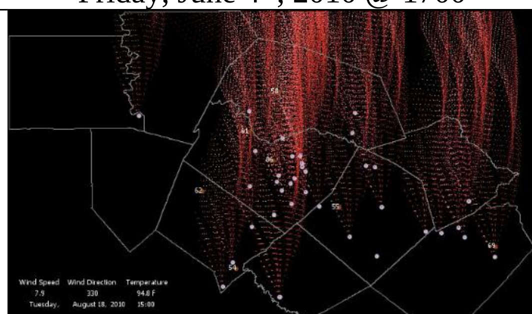
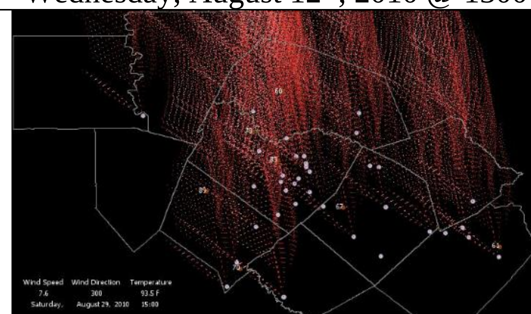
<table>
<tr><td colspan="4"></td><td colspan="4"></td></tr>
<tr><td>Ozone</td><td>Wind Spd</td><td>Wind Dir</td><td>Temp</td><td>Ozone</td><td>Wind Spd</td><td>Wind Dir</td><td>Temp</td></tr>
<tr><td>85</td><td>10.3 mph</td><td>319</td><td>81.9 f</td><td>90</td><td>10.3 mph</td><td>333</td><td>81.1 f</td></tr>
<tr><td colspan="4">Tuesday, May 4th, 2010 @ 1700</td><td colspan="4">Monday, May 5th, 2010 @ 1400</td></tr>
<tr><td colspan="4"></td><td colspan="4"></td></tr>
<tr><td>Ozone</td><td>Wind Spd</td><td>Wind Dir</td><td>Temp</td><td>Ozone</td><td>Wind Spd</td><td>Wind Dir</td><td>Temp</td></tr>
<tr><td>86</td><td>9.3 mph</td><td>344</td><td>85.4 f</td><td>85</td><td>9.3 mph</td><td>343</td><td>85.7 f</td></tr>
<tr><td colspan="4">Friday, May 28th, 2010 @ 1300</td><td colspan="4">Saturday, May 29th, 2010 @ 1300</td></tr>
</table>

Table 6. Continued

							
Ozone	Wind Spd	Wind Dir	Temp	Ozone	Wind Spd	Wind Dir	Temp
86	9.8 mph	325	87.6 f	87	7.6 mph	347	93.1 f
Friday, June 4th, 2010 @ 1700				Wednesday, August 12th, 2010 @ 1300			
							
Ozone	Wind Spd	Wind Dir	Temp	Ozone	Wind Spd	Wind Dir	Temp
86	7.9 mph	330	94.8 f	83	7.6 mph	300	93.5 f
Monday, August 17th, 2010 @ 1500				Friday, August 28th, 2010 @ 1500			

A scatterplot compares wind speed, wind direction, and temperature to the ozone concentration (Figure 19).

The sample size was not large enough meaningfully evaluate the significance of the relationships shown. However, there are trends can be described. Of the ABM's 41 emission point-sources, eleven potentially influenced the monitored ozone concentrations. The ABM suggested that four point-sources (TH0015V, TH0104V, TH0142N, and TH0502F) generated emissions that arrived at CAMS 3 on each of the 8 peak daily one-hour ozone days exceeding 80 ppb (Table 8). Specifically, these point-sources are: a pavement mixture company named Austin Hot Mix (TH0015V), the Hal C.

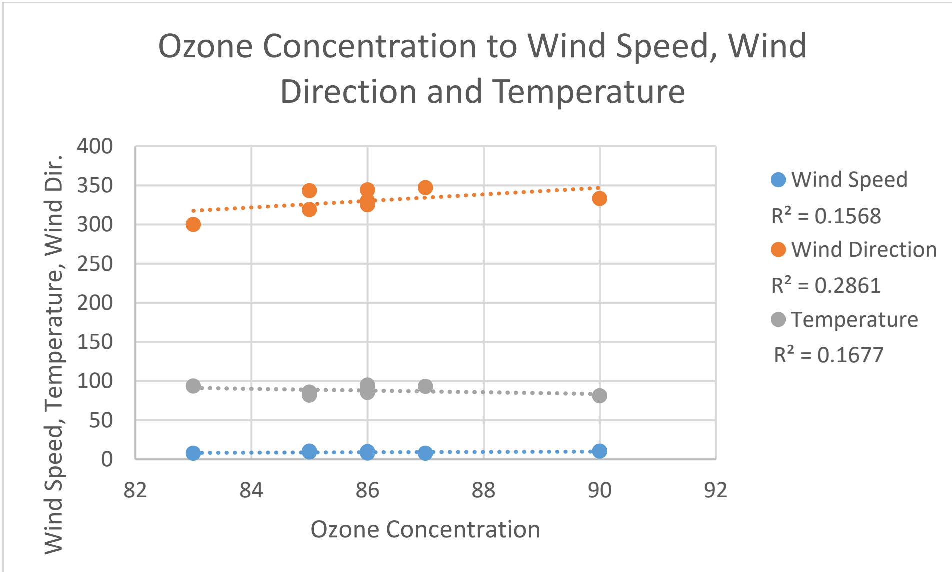


Figure 17: Ozone Concentration to Wind Speed, Wind Direction and Temperature

Weaver Power Plant at the University of Texas (TH104V), a semiconductor manufacturer named Spansion (TH0142N), and the Austin Community Landfill (TH0502F). However, the correlations to the ozone data are not perfectly linear within this set. However, the outlier is clearly Spansion and a few details which are notable. It has the lowest combined NO_x and VOCs TPY. Additionally, it is the farthest west point-source and, among the point-sources under consideration, it is at the highest elevation.

Table 7: Emission Point-Sources with Consistent Emission Arrivals

Account #	R^2	NO_x (TPY)	VOC (TPY)	Distance (km)	Elevation (m)	Direction to CAMS 3
TH0015V	.2965	3.64	40.72	13.9	136	327°
TH0104V	.1464	407.6003	10.2819	7.9	174	343°
TH0142N	.0287	6.5657	15.8305	15.5	188	346°
TH0502F	.3698	42.055	8.8725	23.2	166	340°

7.2 Evaluation Using Daily Mean Ozone

Another evaluation used daily mean ozone with 362 days, days higher than 70 ppb, 75 ppb, and 80 ppb ozone concentration. For this analysis, only ABM arrival counts of emissions were considered against ozone concentration.

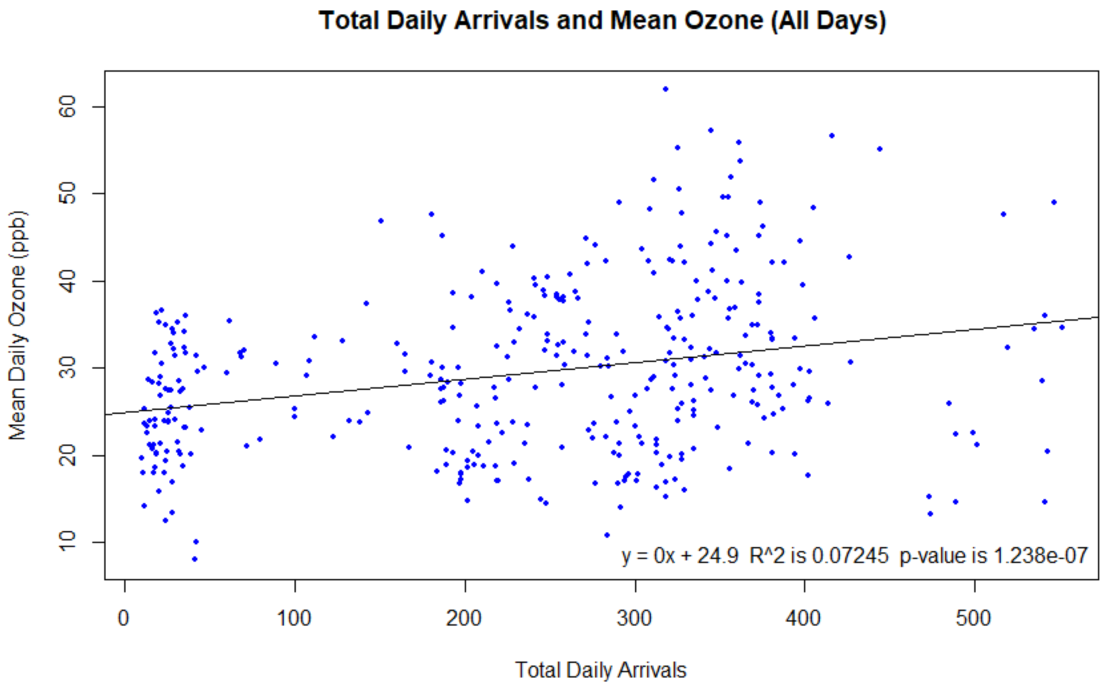


Figure 18: Total Daily Arrivals and Mean Ozone (362 Observations)

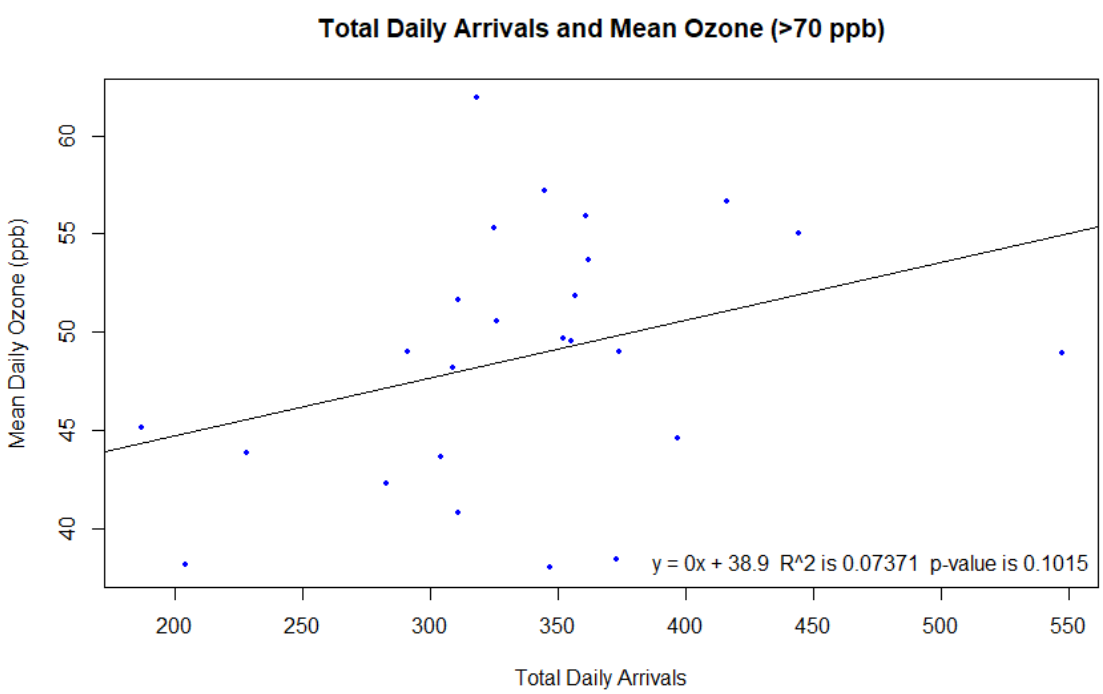


Figure 19: Total Daily Arrivals and Mean Ozone (25 Observations)

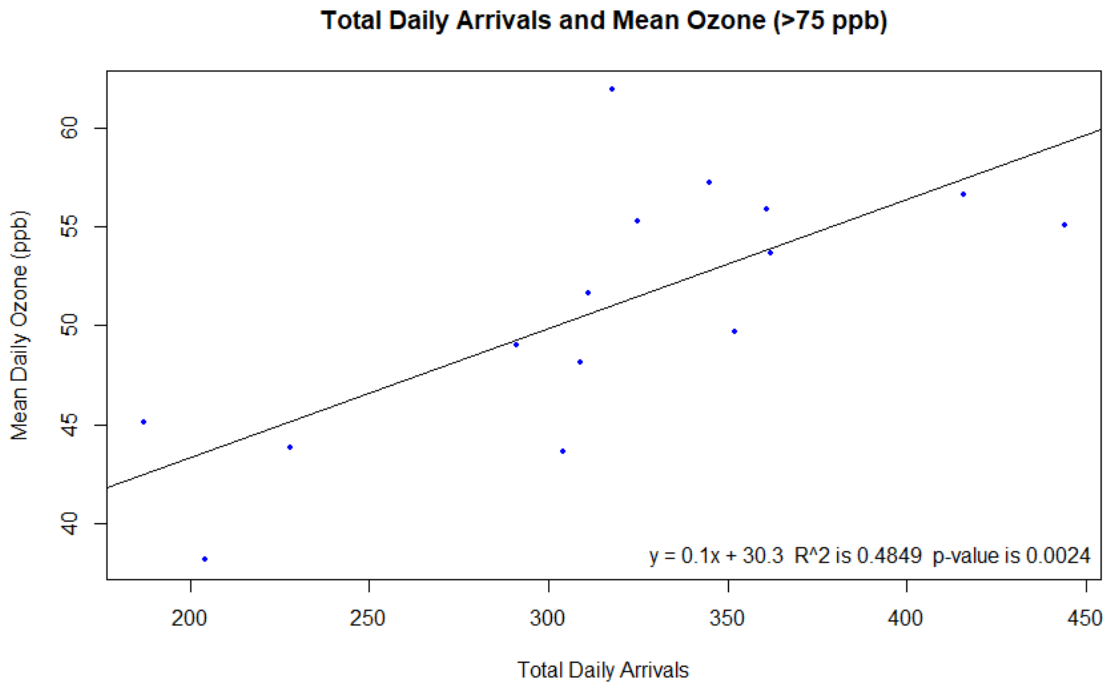


Figure 20: Total Daily Arrivals and Mean Ozone (15 Observations)

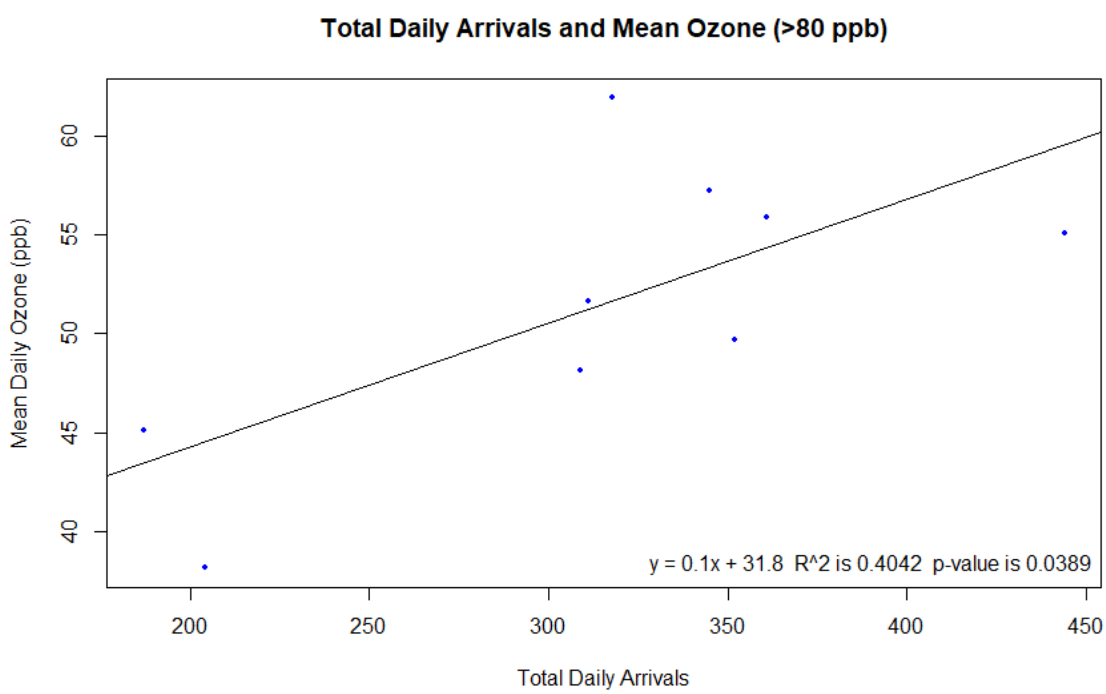


Figure 21: Total Daily Arrivals and Mean Ozone (9 Observations)

Impact of point-source emissions on mean daily ozone concentration appears to be more highly correlated with ozone concentration variance on days with a peak daily ozone concentration of greater than 75 ppb. However, the sample size becomes smaller with higher peak daily ozone concentration and high correlations are less significant. This model fits well with a shorter range of ozone concentrations because of the net impact of point-source emissions is more apparent within a shorter range. Also, the higher concentrations within the range are indicative of ideal days for ozone formation.

The higher R-square values indicated that mean daily ozone variation is a better fit for correlating point-source emission impact on ground-level ozone concentrations than peak-daily ozone variation. R-square values increasing when days with higher peak-daily ozone concentrations are considered may be an indicator of the magnitude of the impact of point-source emissions. Correlation appears higher with a decreased range between the minimum and maximum daily ozone concentration. Increasing the amount of data available on days with 75ppb or higher ozone concentration would improve the significance of the ABM results.

8. DISCUSSION AND CONCLUSION

8.1 Discussion

This thesis utilizes the agent-based modeling software and proprietary programing language of Netlogo (Wilensky, et al. 1999) and supplemental packages that allow for the incorporation of historically significant CSV datasets. Exporting data from an agent-based model into spreadsheets with coordinate location data allows for a comprehensive analysis of data using ArcMap 10.5.1 and for the production of evidentiary visualizations that depict the nature of ozone formation and transport throughout the troposphere. The locations of ozone precursor emitting point-sources and the insufficient presence of ozone monitoring data sites depicted leaves large areas without data, creating challenges for interpolating ozone pollution data.

8.2 Limitations

There were several generalizations and assumptions made in the development and use of this model. The agent-based model is not intended to imply precise transport paths and concentrations of ozone. Plume heights and other topographic elements have not been modeled in the ABM. Variations in molar mass, chemical processes, cloud cover, and the vertical dimension of the atmosphere are also not reflected in this model.

There are other important sources of ozone precursor gases that were not included in this model, but that could skew the results during certain times and in specific places. For instance, ozone levels may be significantly higher in hourly historical readings than the model predicts during high-traffic hours such as during peak-commuting periods. If higher concentrations from historical data during high traffic periods were an easily identifiable and distinct phenomenon within a future agent-based model, then this ABM could successfully validate ozone levels based on the point-sources emissions.

The model does not including molecular mass differences in transport. For example, nitrous dioxide (N_2O) has a molar mass of 44.013 mol/g, natural gas has a molar mass of 19 mol/g, and ozone has a molar mass of 48 mol/g. Gas with lower molar mass would move differently than gases with higher molar mass: increased temperature proportionally increases the speed at which gas molecules move (NSF 2017). Due to increased velocity in higher temperatures, clusters of VOCs and NO_x within a specified area are more likely interact with one another in a smaller space over a shorter period. The formation of large quantities of ozone is more likely, which may increase the amount of ozone cumulatively reaching ozone sensors. Incorporating these features in the future could improve the ABM accuracy. Increased concentrations of ozone produced through chemical reactions between precursor gases within the agent-based model could then be weighted proportional to the number of VOCs and NO_x clustered together within a specified distance.

The detailed chemical processes that produce ozone are not included in this model. Hourly temperature data are only used for visual reference and do not directly affect the model. Including other variables into the model parameters such as cloud cover and temperature might increase the accuracy of pollutant transport analysis. Only one meteorological station was used to represent wind direction and speed across the region. Data from additional meteorological stations might also improve the predictive accuracy of the model.

8.3 Conclusion

This research developed and tested an agent-based model (ABM) for tropospheric transport of ozone-precursor pollutants from point-source locations in TCEQ Region 11. The results demonstrate a novel approach to using an ABM to model and visualize tropospheric ozone formation and transport. Analysis of ABM results within ArcMap demonstrated that there is a relationship between precursor gas emissions from point-sources and ground-level ozone monitoring data within study area. This provides evidence that ozone-precursors emitted from point-sources are significantly related to fluctuations of ground-level ozone measurements within the region. ABM could be a useful tool for policy-makers, decision-makers, and other audiences for visualizing and predicting the phenomena related to air pollution in areas of interest. Further development of methodologies such as this one can further our understanding of the natural and anthropogenic dangers of our planet.

APPENDIX SECTION

APPENDIX A

```
extensions [
      gis
      csv
     ]
globals [
     county-dataset
     monitor-dataset
     emissions-dataset
     noxtotal
     voctotal
     counter              ;; Counter variable used to rotate through each emission source point.
     emissionsource-ID    ;; Used to determine emission-source the emission was produced at.
     data                 ;; Used to import 1 row of wind-data from CSV
     weather              ;; Holds weather data
     wind-speed           ;; Tracks Hourly Historical Wind-Speed from CSV file.
     wind-direction       ;; Tracks Hourly Historical Wind-Direction from CSV file.
     max-nox              ;; Used to store the max number of nox particles from a point-source for
proportional coloring according to amount of emissions.
     max-voc              ;; Used to store the max number of voc particles from a point-source for
proportional coloring according to amount of emissions.
     normalized-color     ;; Temporarily stores normalized ranges for assigning colors (0-1)
     jan                  ;; Used during setup to set the day of the week.
     feb                  ;; Used during setup to set the day of the week.
     mar                  ;; Used during setup to set the day of the week.
     apr                  ;; Used during setup to set the day of the week.
     may                  ;; Used during setup to set the day of the week.
     jun                  ;; Used during setup to set the day of the week.
     jul                  ;; Used during setup to set the day of the week.
     aug                  ;; Used during setup to set the day of the week.
     sept                 ;; Used during setup to set the day of the week.
     oct                  ;; Used during setup to set the day of the week.
     nov                  ;; Used during setup to set the day of the week.
     dec                  ;; Used during setup to set the day of the week.
     track-iterations     ;; Tracks the iterations that the program has run.
     average-temperature  ;; For selecting the appropriate monthly temperature each month.
     hourly-temperature   ;; Fluctuates the temperature +/- 10 degrees daily from the average
temperature.
     month                ;; Tracks Months.
     month-num            ;; Converts and stores Month to a numerical value for certain
applications.
     year                 ;; Tracks Years.
     days                 ;; Tracks Days.
     hours                ;; Tracks Hours.
     minutes              ;; Tracks Minutes.
     output-day           ;; Outputs the day-of-week in text form, ("Sunday, Monday... etc.")
     counts-days          ;; Counts the days to progressively move days of week until reaching
```

```
starting day of the week.
        calendar-counter      ;; Counts days of the month to progress through months.
        feb-leap-year         ;; Stores 28 days or 29 days for February depending on leap year.
        calendar-days         ;; For storing correct Gregorian calendar days in a year to accomodate
leap year when necessary.
        day-of-week           ;; Tracks which day of the week it is (numerical form). ("Sunday 7,
Monday 1... etc.")
        holiday-counter       ;; Global variable counter for determining holidays that fall on a
certain day of week every year (ie. Thanksgiving).
        holiday?              ;; True or False for the day being a holiday.
        highest-nox           ;; Stores the highest nox values recorded at the monitors for printing
(ONLY USED FOR CALIBRATION).
        hours-counter         ;; Stores + 1 for each hour to move iteratively through the wind and
temperature data.
  ;;; The following variables may be created by an automated system for each region in future
developments. ;;;
        cam3
        cam38
        cam601
        cam614
        cam675
        cam684
        cam690
       ]
breed [ emissions emission ]
breed [ emission-sources emissions-source ]
breed [ monitors monitor ]
;;; The following breeds are used for the outputs at the bottom-left hand corner of the screen.
breed [ monthfields monthfield ]
breed [ dayfields dayfield ]
breed [ yearfields yearfield ]
breed [ hourfields hourfield ]
breed [ minutefields minutefield ]
breed [ dayofweekfields dayofweekfield ]
breed [ daycountfields daycountfield ]       ;; This is a spaceholder if the user wants to count the #
of days.
breed [ temperaturefields temperaturefield ]
breed [ templabelfields templabelfield ]
breed [ winddirfields winddirfield ]
breed [ winddirlabelfields winddirlabelfield ]
breed [ windspdfields windspdfield ]
breed [ windspdlabelfields windspdlabelfield ]
patches-own
[
  longitude            ;; Patches store the longitude
  latitude             ;; Patches store the latitude
  model-nox            ;; Sums the NOx from point-source emissions at patch-locations.
  model-voc            ;; Sums the VOC from point-source emissions at patch-locations.
  patch-NOXTPY         ;; The corresponding patch variable the the Shapefile's vector NOX TPY
variable is copied to.
  patch-VOCTPY         ;; The corresponding patch variable the the Shapefile's vector VOC TPY
```

```
variable is copied to.
  original-color         ;; Stores the original color variable for the patches.
]

turtles-own
[
  turtle-COUNTERID       ;; Stores the turtle-COUNTERID variable that is used to rotate through
turtles emitting particles.
  turtle-ACCOUNT         ;; The corresponding turtle variable that the Shapefile's vector
ACCOUNT variable is copied to.
  turtle-RN              ;; The corresponding turtle variable that the Shapefile's vector RN variable is
copied to.
  turtle-COMPANY         ;; The corresponding turtle variable that the Shapefile's vector
COMPANY variable is copied to.
  turtle-SITE            ;; The corresponding turtle variable that the Shapefile's vector SITE variable
is copied to.
  turtle-YEAR            ;; The corresponding turtle variable that the Shapefile's vector REPORTING
YEAR variable is copied to.
  turtle-COUNTY          ;; The corresponding turtle variable that the Shapefile's vector COUNTY
variable is copied to.
  turtle-REGION          ;; The corresponding turtle variable that the Shapefile's vector REGION
variable is copied to.
  turtle-SIC             ;; The corresponding turtle variable that the Shapefile's vector SIC variable is
copied to.
  turtle-DESCRIPTION     ;; The corresponding turtle variable that the Shapefile's vector SIC
DESCRIPTION variable is copied to.
  turtle-COTPY           ;; The corresponding turtle variable that the Shapefile's vector CO TPY
variable is copied to.
  turtle-NOXTPY          ;; The corresponding turtle variable that the Shapefile's vector NOX TPY
variable is copied to.
  turtle-PBTPY           ;; The corresponding turtle variable that the Shapefile's vector PB TPY
variable is copied to.
  turtle-PM10TPY         ;; The corresponding turtle variable that the Shapefile's vector PM10 TPY
variable is copied to.
  turtle-PM25TPY         ;; The corresponding turtle variable that the Shapefile's vector PM2.5
TPY variable is copied to.
  turtle-SO2TPY          ;; The corresponding turtle variable that the Shapefile's vector SO2 TPY
variable is copied to.
  turtle-VOCTPY          ;; The corresponding turtle variable that the Shapefile's vector VOC TPY
variable is copied to.
  turtle-LONGITUDE       ;; The corresponding turtle variable that the Shapefile's vector
LONGITUDE variable is copied to.
  turtle-LATITUDE        ;; The corresponding turtle variable that the Shapefile's vector
LATITUDE variable is copied to.
  turtle-CAMS            ;; The corresponding turtle variable that the Shapefile's vector CAMS
variable is copied to.
  NOx-hourly             ;; Stores number representing the total number of NOx particles on the map
per hour.
  VOC-hourly             ;; Stores number representing the total number of VOC particles on the map
per hour.
  NOx-daily              ;; Stores number representing the total number of NOx particles on the map
```

```
per day.
  VOC-daily              ;; Stores number representing the total number of VOC particles on the map
per day.
  annual-NOx-particles   ;; Stores number representing the total number of NOx particles on the
map per year.
  annual-VOC-particles   ;; Stores number representing the total number of VOC particles on the
map per year.
  turtle-COUNTER         ;; Used for counting the number of emissions produced from an
emission-source.
  particle-count         ;; Stores the number of particles produced every 12 minutes from an
emission source.
  particle-age           ;; Tracks the age of the agent.
  monitor-nox            ;; Stores hourly pollution levels at monitor site.
  hourly-ozone           ;; Holds list of hourly ozone monitor records each day for each monitor.
]
to setup
  clear-all              ;; Clears the contents of the model window
  reset-ticks            ;; Resets the Ticks
  file-close-all         ;; Close any files open from last run
  ;;; Linear Interpolation to convert patch coordinates to latitude and longitude.
  ask patches
    [
      set longitude (((pxcor - -360) / (360 - -360)) * (-96.5664814816 - -98.9646296296) + -
98.9646296296)
      set latitude (((pycor - -210) / (210 - -210)) * (31.035462963 - 29.6280555556) +
29.6280555556)
    ]
  set month start-month                    ;; Sets the calendar to the pre-selected starting month.
  set year start-year                      ;; Sets the calendar to the pre-selected starting year.
  set calendar-counter start-day           ;; Sets the calendar to the pre-selected starting day.
  set minutes 0                            ;; Resets the Minutes to 0.
  set hours 0                              ;; Resets the Hours to 0.
  set days 0                               ;; Resets the Days to 0.
  set counter 0                            ;; Counter is set to 0 for use calculating the day of the week.
  set counts-days 1                        ;; Sets counts-days to 1.
  set track-iterations track-iterations + 1  ;; Adds 1 to the # of iterations (To begin recording at the
1st iteration as opposed to the 0th).
  set holiday? false                       ;; Initially sets the holiday value to false.
  set holiday-counter 0                    ;; Resets the holiday-counter to 1.
  setup-calendar         ;; Runs commands to set up the calendar
  ; Load all of the datasets
  set county-dataset gis:load-dataset "data/Counties.shp"
  set monitor-dataset gis:load-dataset "data/2010_Ozone_Monitors.shp"
  set emissions-dataset gis:load-dataset "data/Region_11_Point_Source_Emissions.shp"
  ; Set the world envelope to the union of all of our dataset's envelopes
  gis:set-world-envelope (gis:envelope-union-of (gis:envelope-of county-dataset)
                                                (gis:envelope-of monitor-dataset)
                                                (gis:envelope-of emissions-dataset))
  set-default-shape turtles "circle"
  set hours-counter 0
  set weather [ ]
```

```
  set weather csv:from-file "/data/wind-data-2010.csv" ;; Creates dataset from the CSV file with
the wind data.
  set data (item hours-counter weather)              ;; Reads the first line of the wind data.
  set hourly-temperature item 0 data         ;; Sets the starting hourly-temperature data.
  set wind-speed item 1 data                ;; Sets the starting wind-speed data.
  set wind-direction item 2 data             ;; Sets the starting wind-direction data.
  ifelse wind-direction < 180               ;; Checks to see if the wind-direction is less than 180
degrees
  [ set wind-direction wind-direction + 180 ] ;; The wind-direction data indicates the direction
from which the wind is blowing, adding 180 degrees show the direction the wind is heading.
  [ set wind-direction wind-direction - 180 ] ;; The wind-direction dta indicates the direction from
which the iwnd is blowing, subtracting 180 degrees shows the direction the wind is heading
end
;;; THIS CAN BE RE-WRITTEN FOR AUTOMATION OF OTHER REGION FILES ;;
to load-csv-data
  set cam3 []
  set cam3 csv:from-file "results/real_world/CAMS/cams_3.csv"
  file-close
  set cam38 []
  set cam38 csv:from-file "results/real_world/CAMS/cams_38.csv"
  file-close
  set cam601 []
  set cam601 csv:from-file "results/real_world/CAMS/cams_601.csv"
  file-close
  set cam614 []
  set cam614 csv:from-file "results/real_world/CAMS/cams_614.csv"
  file-close
  set cam684 []
  set cam684 csv:from-file "results/real_world/CAMS/cams_684.csv"
  file-close
  set cam690 []
  set cam690 csv:from-file "results/real_world/CAMS/cams_690.csv"
  file-close
  set cam675 []
  set cam675 csv:from-file "results/real_world/CAMS/cams_675.csv"
  file-close
  ask monitors with [turtle-CAMS = 3]
  [
    set hourly-ozone (item 1 cam3)
    set label (item 1 hourly-ozone)
  ]
  ask monitors with [turtle-CAMS = 38]
  [
    set hourly-ozone (item 1 cam38)
    set label (item 1 hourly-ozone)
  ]
  ask monitors with [turtle-CAMS = 601]
  [
    set hourly-ozone (item 1 cam601)
    set label (item 1 hourly-ozone)
  ]
```

```
  ask monitors with [turtle-CAMS = 614]
  [
    set hourly-ozone (item 1 cam614)
    set label (item 1 hourly-ozone)
  ]
  ask monitors with [turtle-CAMS = 675]
  [
    set hourly-ozone (item 1 cam675)
    set label (item 1 hourly-ozone)
  ]
  ask monitors with [turtle-CAMS = 684]
  [
    set hourly-ozone (item 1 cam684)
    set label (item 1 hourly-ozone)
  ]
  ask monitors with [turtle-CAMS = 690]
  [
    set hourly-ozone (item 1 cam690)
    set label (item 1 hourly-ozone)
  ]
end
; Displays the Time and Date in the Model.
; Labels align right.
; Leave 5 spaces per character + a buffer.
to display-date-time-in-model
    create-dayofweekfields 1
    [
      setxy -290 -190
      set color black
    ]
    create-monthfields 1
    [
      setxy -240 -190
      set color black
    ]
    create-dayfields 1
    [
      setxy -225 -190
      set color black
    ]
    create-yearfields 1
    [

    setxy -200 -190
      set color black
    ]
    create-hourfields 1
    [
      setxy -170 -190
      set color black
    ]
```

```
    create-minutefields 1
    [
      setxy -160 -190
      set color black
    ]
    create-windspdfields 1
    [
      setxy -310 -175
      set color black
    ]
    create-winddirfields 1
    [
      setxy -237 -175
      set color black
    ]
    create-temperaturefields 1
    [
      setxy -160 -175
      set color black
    ]
    create-windspdlabelfields 1
    [
      setxy -290 -160
      set color black
      set label "Wind Speed"
    ]
    create-winddirlabelfields 1
    [
      setxy -212 -160
      set color black
      set label "Wind Direction"
    ]
    create-templabelfields 1
    [
      setxy -142 -160
      set color black
      set label "Temperature"
    ]
end
; Drawing point data from a shapefile.
to display-current-monitors
  set counter 0
  foreach gis:feature-list-of monitor-dataset [ vector-feature ->
    gis:set-drawing-color 23
    gis:fill vector-feature 3.0
    let location gis:location-of (first (first (gis:vertex-lists-of vector-feature)))
    let CAMS gis:property-value vector-feature "ozone_acti"   ;; Creates variable called "CAMS"
to store the CAMS # for the Monitor being drawn by the iteration.
      if not empty? location
      [
      create-monitors 1
```

```
        [
        set xcor item 0 location
        set ycor item 1 location
        set turtle-COUNTERID counter
        set turtle-CAMS CAMS
        set counter (counter + 1)
        set size 1
        ]
      ]
    ]
end

; Drawing polygon data from a shapefile.
to display-county-lines
  gis:set-drawing-color gray
  gis:draw county-dataset 1
end

to display-emissions-point-sources
  set counter 0
  foreach gis:feature-list-of emissions-dataset [ vector-feature ->
    gis:set-drawing-color violet + 3
    gis:fill vector-feature 3.0
    ; a feature in a point dataset may have multiple points, so we
    ; have a list of lists of points, which is why we need to use
    ; first twice here
    let location gis:location-of (first (first (gis:vertex-lists-of vector-feature)))
    let ACCOUNT gis:property-value vector-feature "ACCOUNT"                              ;; Creates
variable called "ACCOUNT" to store the account # for the Emission-Point-Source being drawn
by the iteration.
    let NOXTPY gis:property-value vector-feature "NOX_TPY"                               ;; Creates
variable called "NOXTPY" to store the NOx Tons Per Year for the Emission-Point-Source being
drawn by the iteration.
    let VOCTPY gis:property-value vector-feature "VOC_TPY"                               ;; Creates
variable called "VOCTPY" to store the VOC Tons Per Year for the Emission-Point-Source being
drawn by the iteration.
      if not empty? location
      [
      create-emission-sources 1
        [
        set xcor item 0 location
        set ycor item 1 location
        set turtle-ACCOUNT ACCOUNT                                              ;; Assigns the account #
to the turtle representing the Emission-Point-Source.
        set turtle-NOXTPY NOXTPY                                                ;; Assigns the NOx TPY to
the turtle representing the Emission-Point-Source.
        set turtle-VOCTPY VOCTPY                                                ;; Assigns the VOC TPY
to the turtle representing the Emission-Point-Source.
        set patch-NOXTPY NOXTPY
        set patch-VOCTPY VOCTPY
        set turtle-COUNTERID counter
```

```
        set counter (counter + 1)
        set size 1
        set turtle-COUNTER 0
        set NOXTotal (NOXTotal + turtle-NOXTPY)          ;; Adds NOXTPY of the new
emissions-source to the NOXTotal
        set VOCTotal (VOCTotal + turtle-VOCTPY)          ;; Adds VOCTPY of the new emissions-
source to the VOCTotal
        set annual-NOx-particles (turtle-NOXTPY * 20617)  ;; With each NOx molecule being 44
moles, there are 20,617 NOx molecules per ton.
        set annual-VOC-particles (turtle-VOCTPY * 20617)  ;; With each VOC molecule being 44
moles, there are 20,617 VOC molecules per ton.
        if max-nox < turtle-NOXTPY                     ;; Checks to see if the emissions-source is the
new max-nox (highest value).
          [ set max-nox turtle-NOXTPY ]                ;; If the new emissions source is the higher
value, it sets it as the max-nox.
        if max-voc < turtle-VOCTPY                     ;; Checks to see if the emissions-source is the
new max-nox (highest value).
          [ set max-voc turtle-VOCTPY ]                ;; If the new emissions source is the higher
value, it sets it as the max-voc.
        ]
      ]
  ]
  ask emission-sources
  [
    set NOx-daily (annual-NOx-particles / 365)
    set VOC-daily (annual-VOC-particles / 365)
    set NOx-hourly (NOx-daily / 24)
    set VOC-hourly (VOC-daily / 24)
  ]
end

to go
  if year = 2011
    [ stop ]
;  pollution-monitor       ;; Runs commands to monitor the pollution.
  calendar-commands
  clock-commands
  tick
  if ticks mod 12 = 0      ;; These commands run every 12 ticks (minutes)
    [
;;;  Enable the next 4 lines to take a screenshot of the model window every hour.
;;;   if ticks mod 60 = 0
;;;   [
;;;   export-view  (word "/results/model_screenshots/Output-" month-num "-" calendar-counter "-"
hours ".png")  ;; Outputs the worldview screen as a PNG.
;;;   ]
      daily-high-commands  ;; Runs commands for acquiring daily level result maps.
      create-NOX          ;; Runs commands for creation of the emission NOx
      create-VOC          ;; Runs commands for creation of the emission VOC
      particles           ;; Runs commands for particle movement.
```

```
      pollution-mapping    ;; Runs commands to change color of pathes based on pollution
presence.
    ]
end

to daily-high-commands
    if month = "May" and calendar-counter = 4 and hours = 0 and minutes = 0 ;;
        [
          ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
        ]
    if month = "May" and calendar-counter = 5 and hours = 0 and minutes = 0 ;;
        [
        export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
        ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
        ]
    if month = "May" and calendar-counter = 6 and hours = 0 and minutes = 0 ;;
        [
        export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
        ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
        ]
    if month = "May" and calendar-counter = 28 and hours = 0 and minutes = 0 ;;
        [
          ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
        ]
    if month = "May" and calendar-counter = 29 and hours = 0 and minutes = 0 ;;
        [
        export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
          ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
```

```
      ]
    if month = "May" and calendar-counter = 30 and hours = 0 and minutes = 0 ;;
      [
      export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
        ask patches
        [
          set model-nox 0
          set model-voc 0
        ]
      ]
    if month = "June" and calendar-counter = 4 and hours = 0 and minutes = 0 ;;
      [
        ask patches
        [
          set model-nox 0
          set model-voc 0
        ]
      ]
    if month = "June" and calendar-counter = 5 and hours = 0 and minutes = 0 ;;
      [
      export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
        ask patches
        [
          set model-nox 0
          set model-voc 0
        ]
      ]
    if month = "June" and calendar-counter = 28 and hours = 0 and minutes = 0 ;;
      [
        ask patches
        [
          set model-nox 0
          set model-voc 0
        ]
      ]
    if month = "June" and calendar-counter = 29 and hours = 0 and minutes = 0 ;;
      [
      export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
        ask patches
        [
          set model-nox 0
          set model-voc 0
        ]
      ]
    if month = "August" and calendar-counter = 12 and hours = 0 and minutes = 0 ;;
      [
        ask patches
        [
```

```
        set model-nox 0
        set model-voc 0
      ]
    ]
  if month = "August" and calendar-counter = 13 and hours = 0 and minutes = 0 ;;
    [
    export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
    ask patches
      [
        set model-nox 0
        set model-voc 0
      ]
    ]
  if month = "August" and calendar-counter = 17 and hours = 0 and minutes = 0 ;;
    [
      ask patches
      [
        set model-nox 0
        set model-voc 0
      ]
    ]
  if month = "August" and calendar-counter = 18 and hours = 0 and minutes = 0 ;;
    [
    export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
    ask patches
      [
        set model-nox 0
        set model-voc 0
      ]
    ]
  if month = "August" and calendar-counter = 27 and hours = 0 and minutes = 0 ;;
    [
      ask patches
      [
        set model-nox 0
        set model-voc 0
      ]
    ]
  if month = "August" and calendar-counter = 28 and hours = 0 and minutes = 0 ;;
    [
    export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
    ask patches
      [
        set model-nox 0
        set model-voc 0
      ]
    ]
  if month = "August" and calendar-counter = 29 and hours = 0 and minutes = 0 ;;
```

```
        [
        export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
        ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
        ]
    if month = "September" and calendar-counter = 5 and hours = 0 and minutes = 0 ;;
        [
          ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
        ]
    if month = "September" and calendar-counter = 6 and hours = 0 and minutes = 0 ;;
        [
        export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
        ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
        ]
    if month = "October" and calendar-counter = 8 and hours = 0 and minutes = 0 ;;
        [
          ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
        ]
    if month = "October" and calendar-counter = 9 and hours = 0 and minutes = 0 ;;
        [
        export-world (word "Output-" month-num calendar-counter ".csv");; Outputs the worlds data
for this month.
        ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
        ]
end

to pollution-mapping
  if ticks = 12
    [
      ask patches
```

```
      [
        set original-color pcolor
      ]
  ]
if ticks mod 12 = 0      ;; These commands run every 12 ticks (minutes)
  [
    ask emissions        ;; This asks every emission to run these commands.
      [
        set pcolor 17    ;; This command will only affect the patch that the emission is on.
        set model-nox model-nox + (NOX-hourly / 5)
        set model-voc model-voc + (VOC-hourly / 5)
      ]
    if ticks mod 60 = 0
      [
        ask patches with [pcolor <= 17 AND pcolor >= 13]
        [ set pcolor (pcolor - 1) ]
        ask patches with [pcolor = 12]
        [ set pcolor original-color ]
        ask monitors with [turtle-CAMS = 3]
        [ set label (word "   " (item (hours + 1) hourly-ozone)) ]
        ask monitors with [turtle-CAMS = 38]
        [ set label (item (hours + 1) hourly-ozone) ]
        ask monitors with [turtle-CAMS = 601]
        [ set label (item (hours + 1) hourly-ozone) ]
        ask monitors with [turtle-CAMS = 614]
        [ set label (item (hours + 1) hourly-ozone) ]
        ask monitors with [turtle-CAMS = 675]
        [ set label (item (hours + 1) hourly-ozone) ]
        ask monitors with [turtle-CAMS = 684]
        [ set label (item (hours + 1) hourly-ozone)]
        ask monitors with [turtle-CAMS = 690]
        [ set label (item (hours + 1) hourly-ozone)]
      ]
    if ticks mod 1440 = 0
      [
        ask monitors with [turtle-CAMS = 3]
        [ set hourly-ozone (item (days + 1) cam3) ]
        ask monitors with [turtle-CAMS = 38]
        [ set hourly-ozone (item (days + 1) cam38) ]
        ask monitors with [turtle-CAMS = 601]
        [ set hourly-ozone (item (days + 1) cam601) ]
        ask monitors with [turtle-CAMS = 614]
        [ set hourly-ozone (item (days + 1) cam614) ]
        ask monitors with [turtle-CAMS = 675]
        [ set hourly-ozone (item (days + 1) cam675) ]
        ask monitors with [turtle-CAMS = 684]
        [ set hourly-ozone (item (days + 1) cam684)]
        ask monitors with [turtle-CAMS = 690]
        [ set hourly-ozone (item (days + 1) cam690)]
    ]
]
```

```
end

;;; COMMANDS FOR CREATION OF NOx ;;;
to create-NOX
  ask emission-sources
  [
    if turtle-NOXTPY != 0                              ;; Only runs if there is an emission.
    [
    set normalized-color (turtle-NOXTPY / max-nox)          ;; Assigns value according to a
normalized range (0-1) for the emission-source.
    set turtle-counter turtle-counter + NOx-hourly          ;; Counts emissions at a per hour constant
interval according to annual TPY produced.
    set color 12 + (3 * normalized-color)                  ;; Highest count of NOx will have darkest
color. Lowest count of NOx will have lightest color.
    set size 1                                    ;; Size is a standard 1 pixel
    set heading wind-direction                          ;; Sets the heading based on the wind-data.csv
file
    set particle-count (NOX-hourly / 5)                   ;; Sets particle count to the number of
emissions produced hourly divided by 5 (per 12 minutes or 1 tick).
    set particle-age 0
    hatch-emissions 1                               ;; Hatches an emission circle that represents the
number of emission produced.
    ]
  ]
end

;;; COMMANDS FOR CREATION OF NOx ;;;
to create-VOC
  ask emission-sources
  [
    if turtle-VOCTPY != 0                              ;; Only runs if there is an emission.
    [
    set normalized-color (turtle-VOCTPY / max-voc)          ;; Assigns value according to a
normalized range (0-1) for the emission-source.
    set turtle-counter turtle-counter + VOC-hourly          ;; Counts emissions at a per hour constant
interval according to annual TPY produced.
    set color 12 + (3 * normalized-color)                  ;; Highest count of NOx will have darkest
color. Lowest count of NOx will have lightest color.
    set size 1                                    ;; Size is a standard 1 pixel
    set heading wind-direction                          ;; Sets the heading based on the wind-data.csv
file
    set particle-count (VOC-hourly / 5)                   ;; Sets particle count to the number of
emissions produced hourly divided by 5 (per 12 minutes or 1 tick).
    set particle-age 0
    hatch-emissions 1                               ;; Hatches an emission circle that represents the
number of emission produced.
    ]
  ]
end

;;; COMMANDS FOR PARTICLE MOVEMENT ;;;
```

```
to particles
  if ticks mod 60 = 0                    ;; These commands run every 60 ticks (hour), runs inside of
move-particles because 60 is divisible by 12. (Improves Computational efficiency.)
    [
    set hours-counter hours-counter + 1
    ask emissions
      [
      set heading wind-direction          ;; Calculates the new heading and speed using the data from
the CSV file
      if particle-age = 24
        [ die ]
      set particle-age particle-age + 1
      ]
    set data (item hours-counter weather)  ;; Reads the next line of the weather dataset.
      if (item 0 data) != ""
      [
      set hourly-temperature item 0 data    ;; Sets the hourly temperature from the next line of the
CSV file each hour.
      ]
      if (item 1 data) != ""
      [
      set wind-speed item 1 data            ;; Sets the hourly wind-speed from the next line of the CSV
file each hour.
      ]
      if (item 2 data) != ""
      [
      set wind-direction item 2 data                ;; Sets the hourly wind-direction from the next line of
the CSV file each hour.
        ifelse wind-direction < 180                ;; Checks to see if the wind-direction is less than 180
degrees
        [ set wind-direction wind-direction + 180 ] ;; The wind-direction data indicates the direction
from which the wind is blowing, adding 180 degrees show the direction the wind is heading.
        [ set wind-direction wind-direction - 180 ] ;; The wind-direction dta indicates the direction
from which the iwnd is blowing, subtracting 180 degrees shows the direction the wind is heading
      ]
    ]
  ask emissions
    [
    fd wind-speed                         ;; Moves particles forward at x MPH (Occurs every 12 Minutes,
5 patches = 1 mile, no equation necessary.)
    if patch-ahead 1 = nobody [ die ]     ;; If particles reach the world edge, they die.
    ]
end

;;; CLOCK COMMANDS ;;;
to clock-commands
  set minutes minutes + 1                        ;; 1 Minute is accumulated to the clock per every tick.
  if minutes = 60                              ;; Commands for when minutes reach 60.
    [
    set hours hours + 1                          ;; 1 Hour is accumulated per 60 minutes.
    set minutes 0                              ;; Minutes are reset to 0 minutes.
```

```
   ]
  if hours = 24                              ;; Commands for 24 hours.
   [
   set days days + 1                         ;; 1 Day is accumulated per 24 hours.
   set hours 0                               ;; Hours are reset to 0 hours.
   set calendar-counter calendar-counter + 1       ;; 1 Calendar-Counter day progresses for the
calendar per day.
   if calendar-counter >= 28                 ;; On or after the 28th of the month, the following
commands run.
     [
     calendar-commands                       ;; Runs Calendar Commands (For changing the month
on the 28th, 29th, 30th or 31st.).
     ]
   set day-of-week day-of-week + 1
   if day-of-week > 7
     [ set day-of-week 1 ]
   holidays                                  ;; Commands for determining holidays.
  ifelse day-of-week = 1                     ;; Checks to see if output day is the numerical day for
Monday.
   [ set output-day "Monday" ]               ;; Sets the output day to Monday.
[ ifelse day-of-week = 2                     ;; Checks to see if output day is the numerical day for
Tuesday.
   [ set output-day "Tuesday" ]              ;; Sets the output day to Tuesday.
[ ifelse day-of-week = 3                     ;; Checks to see if output day is the numerical day for
Wednesday.
   [ set output-day "Wednesday" ]            ;; Sets the output day to Wednesday.
[ ifelse day-of-week = 4                     ;; Checks to see if output day is the numerical day for
Thursday.
   [ set output-day "Thursday" ]             ;; Sets the output day to Thursday.
[ ifelse day-of-week = 5                     ;; Checks to see if output day is the numerical day for
Friday.
   [ set output-day "Friday" ]               ;; Sets the output day to Friday.
[ ifelse day-of-week = 6                     ;; Checks to see if output day is the numerical day for
Saturday.
   [ set output-day "Saturday" ]             ;; Sets the output day to Saturday.
[ if day-of-week = 7                         ;; Checks to see if output day is the numerical day for
Sunday.
   [ set output-day "Sunday" ]               ;; Sets the output day to Sunday.
] ] ] ] ] ]                                  ;; Closes out 6 ifelse queries above.
   ]

  ask dayofweekfields
   [
     set label (word output-day ",")
   ]
  ask monthfields
   [
     set label month
   ]
  ask dayfields
   [
```

```
      set label (word calendar-counter ",")
    ]
  ask yearfields
    [
      set label year
    ]
  ask hourfields
    [
      ifelse hours >= 10
        [ set label (word hours ":") ]
        [ set label (word "0" hours ":") ]
    ]
  ask minutefields
    [
      ifelse minutes >= 10
        [ set label minutes ]
        [ set label (word "0" minutes) ]
    ]
  ask windspdfields
    [
      set label wind-speed
    ]
  ask winddirfields
    [
      set label wind-direction
    ]
  ask temperaturefields
    [
      set label (word hourly-temperature " f")
    ]
end

;;; CALENDAR COMMANDS ;;;
to calendar-commands
  ifelse month = "January"                          ;; Commands for the month of January.
    [
      if calendar-counter > 31                     ;; Commands if the calendar has progressed through 31
days of January.
        [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
          ask patches
          [
            set model-nox 0
            set model-voc 0
          ]
          set calendar-counter 1                   ;; Resets the calendar counter to 1 just prior to changing
months.
          set month "February"                     ;; The month becomes February.
          set month-num 2 ]                        ;; The numberical month is 2.
    ]
    [
  ifelse month = "February"                         ;; Commands for the month of February.
```

```
   [
     if calendar-counter > feb-leap-year          ;; Commands if the calendar has progressed through
28 days of February except in leapyear it has 29.
       [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
         ask patches
         [
           set model-nox 0
           set model-voc 0
         ]
         set calendar-counter 1                  ;; Resets the calendar counter to 1 just prior to changing
months.
         set month "March"                       ;; The month becomes March.
         set month-num 3 ]                       ;; The numberical month is 3.
   ]
   [
  ifelse month = "March"                         ;; Commands for the month of March.
   [
     if calendar-counter > 31                    ;; Commands if the calendar has progressed through 31
days of March.
       [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
         ask patches
         [
           set model-nox 0
           set model-voc 0
         ]
         set calendar-counter 1                  ;; Resets the calendar counter to 1 just prior to changing
months.
         set month "April"                       ;; The month becomes April.
         set month-num 4 ]                       ;; The numberical month is 4.
   ]
   [
  ifelse month = "April"                         ;; Commands for the month of April.
   [
     if calendar-counter > 30                    ;; Commands if the calendar has progressed through 30
days of April.
       [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
         ask patches
         [
           set model-nox 0
           set model-voc 0
         ]
         set calendar-counter 1                  ;; Resets the calendar counter to 1 just prior to changing
months.
         set month "May"                         ;; The month becomes May.
         set month-num 5 ]                       ;; The numberical month is 5.
   ]
   [
  ifelse month = "May"
   [
     if calendar-counter > 31                    ;; Commands if the calendar has progressed through 31
days of May.
```

```
      [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
        ask patches
        [
          set model-nox 0
          set model-voc 0
        ]
        set calendar-counter 1                    ;; Resets the calendar counter to 1 just prior to changing
months.
        set month "June"                         ;; The month becomes June.
        set month-num 6 ]                        ;; The numberical month is 6.
    ]
    [
  ifelse month = "June"
    [
      if calendar-counter > 30                    ;; Commands if the calendar has progressed through 30
days of June.
      [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
        ask patches
        [
          set model-nox 0
          set model-voc 0
        ]
        set calendar-counter 1                    ;; Resets the calendar counter to 1 just prior to changing
months.
        set month "July"                         ;; The month becomes July.
        set month-num 7 ]                        ;; The numberical month is 7.
    ]
    [
  ifelse month = "July"
    [
      if calendar-counter = 31                    ;; Commands if the calendar has progressed through 31
days of July.
      [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
        ask patches
        [
          set model-nox 0
          set model-voc 0
        ]
        set calendar-counter 1                    ;; Resets the calendar counter to 1 just prior to changing
months.
        set month "August"                       ;; The month becomes August.
        set month-num 8 ]                        ;; The numberical month is 8.
    ]
    [
  ifelse month = "August"
    [
      if calendar-counter > 31                    ;; Commands if the calendar has progressed through 31
days of August.
      [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
        ask patches
        [
```

```
          set model-nox 0
          set model-voc 0
         ]
         set calendar-counter 1                  ;; Resets the calendar counter to 1 just prior to changing
months.
         set month "September"                   ;; The month becomes September.
         set month-num 9 ]                       ;; The numberical month is 9.
   ]
   [
  ifelse month = "September"
   [
     if calendar-counter > 30                    ;; Commands if the calendar has progressed through 30
days of September.
       [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
         ask patches
         [
          set model-nox 0
          set model-voc 0
         ]
         set calendar-counter 1                  ;; Resets the calendar counter to 1 just prior to changing
months.
         set month "October"                     ;; The month becomes October.
         set month-num 10 ]                      ;; The numberical month is 10.
   ]
   [
  ifelse month = "October"
   [
     if calendar-counter > 31                    ;; Commands if the calendar has progressed through 31
days of October.
       [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
         ask patches
         [
          set model-nox 0
          set model-voc 0
         ]
         set calendar-counter 1                  ;; Resets the calendar counter to 1 just prior to changing
months.
         set month "November"                    ;; The month becomes November.
         set month-num 11 ]                      ;; The numberical month is 11.
   ]
   [
  ifelse month = "November"
   [
     if calendar-counter > 30                    ;; Commands if the calendar has progressed through 30
days of November.
       [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
         ask patches
         [
          set model-nox 0
          set model-voc 0
         ]
```

```
        set calendar-counter 1                  ;; Resets the calendar counter to 1 just prior to changing
months.
        set month "December"                      ;; The month becomes December.
        set month-num 12 ]                      ;; The numberical month is 12.
   ]
   [
  if month = "December"
   [
     if calendar-counter > 31                   ;; Commands if the calendar has progressed through 31
days of October.
       [ export-world (word "Output-" month-num ".csv");; Outputs the worlds data for this month.
        ask patches
        [
          set model-nox 0
          set model-voc 0
        ]
        set calendar-counter 1                  ;; Resets the calendar counter to 1 just prior to changing
months.
        set year year + 1
          ifelse year = 2028 OR year = 2024 OR year = 2020 OR year = 2016 OR year = 2012 OR
year = 2008 OR year = 2004 OR year = 2000   ;; Checks to see if it is a leap year.
           [
             set calendar-days 366              ;; If it is a leap year calendar-days are set to 366 for
aging.
             set feb-leap-year 29              ;; If it is a leap year there are 29 days.
           ]
           [
             set calendar-days 365              ;; If it is not a leap year calendar-days are set to 365 for
aging.
             set feb-leap-year 28              ;; If it is not a leap year there are 28 days.
           ]
        set month "January"                     ;; The month becomes January.
        set month-num 1 ]                      ;; The numberical month is 1.
   ] ] ] ] ] ] ] ] ] ] ] ]                     ;; Closes 11 ifelse queries and 1 if query.
end

;;; CALENDAR SETUP ;;;
to setup-calendar
  ifelse year = 2028 OR year = 2024 OR year = 2020 OR year = 2016 OR year = 2012 OR year =
2008 OR year = 2004 OR year = 2000   ;; Checks to see if it is a leap year.
   [
     set calendar-days 366                     ;; If it is a leap year calendar-days are set to 366 for aging
and initial day of the week equation.
     set feb-leap-year 29                     ;; If it is a leap year there are 29 days.
   ]
   [
     set calendar-days 365                     ;; If it is not a leap year calendar-days are set to 365 for
aging and initial day of the week equation.
     set feb-leap-year 28                     ;; If it is not a leap year there are 28 days.
   ]
     ifelse year = 2001 OR year = 2007 OR year = 2018     ;; Checks the year.
```

```
      [ set day-of-week 1 ]                    ;; January 1st is Monday (2001, 2007, 2018).
   [  ifelse year = 2002 OR year = 2008 OR year = 2013     ;; Checks the year.
      [ set day-of-week 2 ]                    ;; January 1st is Tuesday (2002, 2008, 2013).
   [ ifelse year = 2003 OR year = 2014           ;; Checks the year.
      [ set day-of-week 3 ]                    ;; January 1st is Wednesday (2003, 2014).
   [ ifelse year = 2004 OR year = 2009 OR year = 2015      ;; Checks the year.
      [ set day-of-week 4 ]                    ;; January 1st is Thursday (2004, 2009, 2015).
   [ ifelse year = 2010 OR year = 2016           ;; Checks the year.
      [ set day-of-week 5 ]                    ;; January 1st is Friday (2010, 2016).
   [ ifelse year = 2000 OR year = 2005 OR year = 2011      ;; Checks the year.
      [ set day-of-week 6 ]                    ;; January 1st is Saturday (2000, 2005, 2011).
   [ if year = 2006 OR year = 2012 OR year = 2017 ;; Checks the year.
      [ set day-of-week 7 ]                    ;; January 1st is Sunday (2006, 2012, 2017).
   ] ] ] ] ] ]                           ;; Closes out five ifelse queries and one if query above.
     ifelse start-month = "January"              ;; Checks to see if the start-month is "January".
      [
      set month-num 1                          ;; Sets the Month # to 1
     while [ counts-days < start-day ]           ;; Runs this loop until counts-days is greater than the
start-day.
         [
           set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
           set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
             if day-of-week > 7                ;; This happens when the day-of-week is greater than
seven.
              [ set day-of-week 1 ]           ;; The day of week restarts at Monday.
         ]
      ]
      [
     ifelse start-month = "February"             ;; Checks to see if the start-month is "February".
      [
      set month-num 2
     ifelse year = 2001 OR year = 2007 OR year = 2018      ;; Checks the year.
      [ set day-of-week 4 ]                    ;; February 1st is Thursday.
   [ ifelse year = 2002 OR year = 2008 OR year = 2013      ;; Checks the year.
      [ set day-of-week 5 ]                    ;; February 1st is Friday (2002, 2008, 2013).
   [ ifelse year = 2003 OR year = 2014           ;; Checks the year.
      [ set day-of-week 6 ]                    ;; February 1st is Saturday (2003, 2014).
   [ ifelse year = 2004 OR year = 2009 OR year = 2015      ;; Checks the year.
      [ set day-of-week 7 ]                    ;; February 1st is Sunday (2004, 2009, 2015).
   [ ifelse year = 2010 OR year = 2016           ;; Checks the year.
      [ set day-of-week 1 ]                    ;; February 1st is Monday (2010, 2016).
   [ ifelse year = 2000 OR year = 2005 OR year = 2011      ;; Checks the year.
      [ set day-of-week 2 ]                    ;; February 1st is Tuesday (2000, 2005, 2011).
   [ if year = 2006 OR year = 2012 OR year = 2017 ;; Checks the year.
      [ set day-of-week 3 ]                    ;; February 1st is Wednesday (2006, 2012, 2017).
   ] ] ] ] ] ]                           ;; Closes out five ifelse queries and one if query above.
        while [ counts-days < start-day ]        ;; Runs this loop until counts-days is greater than the
start-day.
         [
           set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
```

```
        set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
         if day-of-week > 7                ;; This happens when the day-of-week is greater than
seven.
          [ set day-of-week 1 ]            ;; The day of week restarts at Monday.
       ]
     ]
     [
    ifelse start-month = "March"                ;; Checks to see if the start-month is "March".
     [
     set month-num 3
    ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
     [ set day-of-week 4 ]                    ;; March 1st is Thursday (2001, 2007, 2012 [leap year],
2018).
  [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
     [ set day-of-week 5 ]                    ;; March 1st is Friday (2002, 2013).
  [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
     [ set day-of-week 6 ]                    ;; March 1st is Saturday (2003, 2008 [leap year], 2014).
  [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
     [ set day-of-week 7 ]                    ;; March 1st is Sunday (2004 [leap year], 2009, 2015).
  [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
     [ set day-of-week 1 ]                    ;; March 1st is Monday (2010, 2016 [leap year]).
  [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
     [ set day-of-week 2 ]                    ;; March 1st is Tuesday (2000 [leap year], 2005, 2011).
  [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
     [ set day-of-week 3 ]                    ;; March 1st is Wednesday (2006, 2012 [leap year], 2017).
  ] ] ] ] ] ]                              ;; Closes out five ifelse queries and one if query above.
       while [ counts-days < start-day ]        ;; Runs this loop until counts-days is greater than the
start-day.
       [
        set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
        set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
         if day-of-week > 7                ;; This happens when the day-of-week is greater than
seven.
          [ set day-of-week 1 ]            ;; The day of week restarts at Monday.
       ]
     ]
     [
    ifelse start-month = "April"                ;; Checks to see if the start-month is "April".
     [
     set month-num 4
    ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
     [ set day-of-week 7 ]                    ;; April 1st is Sunday (2001, 2007, 2012 [leap year],
2018).
  [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
     [ set day-of-week 1 ]                    ;; April 1st is Monday (2002, 2013).
  [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
     [ set day-of-week 2 ]                    ;; April 1st is Tuesday (2003, 2008 [leap year], 2014).
  [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
     [ set day-of-week 3 ]                    ;; April 1st is Wednesday (2004 [leap year], 2009, 2015).
```

```
    [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
        [ set day-of-week 4 ]                     ;; April 1st is Thursday (2010, 2016 [leap year]).
    [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
        [ set day-of-week 5 ]                     ;; April 1st is Friday (2000 [leap year], 2005, 2011).
    [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
        [ set day-of-week 6 ]                     ;; April 1st is Saturday (2006, 2012 [leap year], 2017).
    ] ] ] ] ] ]                            ;; Closes out five ifelse queries and one if query above.
          while [ counts-days < start-day ]       ;; Runs this loop until counts-days is greater than the
start-day.
            [
              set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
              set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
                if day-of-week > 7                ;; This happens when the day-of-week is greater than
seven.
                  [ set day-of-week 1 ]           ;; The day of week restarts at Monday.
            ]
        ]
        [
      ifelse start-month = "May"                  ;; Checks to see if the start-month is "May".
        [
        set month-num 5
      ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
        [ set day-of-week 2 ]                     ;; May 1st is Tuesday (2001, 2007, 2012 [leap year],
2018).
    [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
        [ set day-of-week 3 ]                     ;; May 1st is Wednesday (2002, 2013).
    [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
        [ set day-of-week 4 ]                     ;; May 1st is Thursday (2003, 2008 [leap year], 2014).
    [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
        [ set day-of-week 5 ]                     ;; May 1st is Friday (2004 [leap year], 2009, 2015).
    [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
        [ set day-of-week 6 ]                     ;; May 1st is Saturday (2010, 2016 [leap year]).
    [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
        [ set day-of-week 7 ]                     ;; May 1st is Sunday (2000 [leap year], 2005, 2011).
    [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
        [ set day-of-week 1 ]                     ;; May 1st is Monday (2006, 2012 [leap year], 2017).
    ] ] ] ] ] ]                            ;; Closes out five ifelse queries and one if query above.
          while [ counts-days < start-day ]       ;; Runs this loop until counts-days is greater than the
start-day.
            [
              set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
              set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
                if day-of-week > 7                ;; This happens when the day-of-week is greater than
seven.
                  [ set day-of-week 1 ]           ;; The day of week restarts at Monday.
            ]
        ]
        [
      ifelse start-month = "June"                 ;; Checks to see if the start-month is "June".
```

```
      [
      set month-num 6
     ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
      [ set day-of-week 5 ]                    ;; June 1st is Friday (2001, 2007, 2012 [leap year], 2018).
    [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
      [ set day-of-week 6 ]                    ;; June 1st is Saturday (2002, 2013).
    [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
      [ set day-of-week 7 ]                    ;; June 1st is Sunday (2003, 2008 [leap year], 2014).
    [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
      [ set day-of-week 1 ]                    ;; June 1st is Monday (2004 [leap year], 2009, 2015).
    [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
      [ set day-of-week 2 ]                    ;; June 1st is Tuesday (2010, 2016 [leap year]).
    [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
      [ set day-of-week 3 ]                    ;; June 1st is Wednesday (2000 [leap year], 2005, 2011).
    [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
      [ set day-of-week 4 ]                    ;; June 1st is Thursday (2006, 2012 [leap year], 2017).
    ] ] ] ] ] ]                              ;; Closes out five ifelse queries and one if query above.
        while [ counts-days < start-day ]       ;; Runs this loop until counts-days is greater than the
start-day.
          [
           set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
           set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
             if day-of-week > 7                ;; This happens when the day-of-week is greater than
seven.
               [ set day-of-week 1 ]           ;; The day of week restarts at Monday.
          ]
      ]
      [
     ifelse start-month = "July"                 ;; Checks to see if the start-month is "July".
      [
      set month-num 7
     ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
      [ set day-of-week 7 ]                    ;; July 1st is Sunday (2001, 2007, 2012 [leap year], 2018).
    [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
      [ set day-of-week 1 ]                    ;; July 1st is Monday (2002, 2013).
    [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
      [ set day-of-week 2 ]                    ;; July 1st is Tuesday (2003, 2008 [leap year], 2014).
    [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
      [ set day-of-week 3 ]                    ;; July 1st is Wednesday (2004 [leap year], 2009, 2015).
    [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
      [ set day-of-week 4 ]                    ;; July 1st is Thursday (2010, 2016 [leap year]).
    [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
      [ set day-of-week 5 ]                    ;; July 1st is Friday (2000 [leap year], 2005, 2011).
    [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
      [ set day-of-week 6 ]                    ;; July 1st is Saturday (2006, 2012 [leap year], 2017).
    ] ] ] ] ] ]                              ;; Closes out five ifelse queries and one if query above.
        while [ counts-days < start-day ]       ;; Runs this loop until counts-days is greater than the
start-day.
          [
           set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
```

```
          set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
            if day-of-week > 7                ;; This happens when the day-of-week is greater than
seven.
              [ set day-of-week 1 ]           ;; The day of week restarts at Monday.
          holidays                           ;; Runs the commands to determine holidays
         ]
      ]
      [
    ifelse start-month = "August"              ;; Checks to see if the start-month is "August".
      [
      set month-num 8
    ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
      [ set day-of-week 3 ]                   ;; August 1st is Wednesday (2001, 2007, 2012 [leap year],
2018).
   [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
      [ set day-of-week 4 ]                   ;; August 1st is Thursday (2002, 2013).
   [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
      [ set day-of-week 5 ]                   ;; August 1st is Friday (2003, 2008 [leap year], 2014).
   [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
      [ set day-of-week 6 ]                   ;; August 1st is Saturday (2004 [leap year], 2009, 2015).
   [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
      [ set day-of-week 7 ]                   ;; August 1st is Sunday (2010, 2016 [leap year]).
   [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
      [ set day-of-week 1 ]                   ;; August 1st is Monday (2000 [leap year], 2005, 2011).
   [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
      [ set day-of-week 2 ]                   ;; August 1st is Tuesday (2006, 2012 [leap year], 2017).
   ] ] ] ] ] ]                               ;; Closes out five ifelse queries and one if query above.
        while [ counts-days < start-day ]       ;; Runs this loop until counts-days is greater than the
start-day.
         [
          set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
          set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
            if day-of-week > 7                ;; This happens when the day-of-week is greater than
seven.
              [ set day-of-week 1 ]           ;; The day of week restarts at Monday.
         ]
      ]
      [
    ifelse start-month = "September"            ;; Checks to see if the start-month is "September".
      [
      set month-num 9
    ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
      [ set day-of-week 6 ]                   ;; September 1st is Saturday (2001, 2007, 2012 [leap
year], 2018).
   [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
      [ set day-of-week 7 ]                   ;; September 1st is Sunday (2002, 2013).
   [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
      [ set day-of-week 1 ]                   ;; September 1st is Monday (2003, 2008 [leap year],
2014).
```

```
   [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
      [ set day-of-week 2 ]                    ;; September 1st is Tuesday (2004 [leap year], 2009,
2015).
   [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
      [ set day-of-week 3 ]                    ;; September 1st is Wednesday (2010, 2016 [leap year]).
   [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
      [ set day-of-week 4 ]                    ;; September 1st is Thursday (2000 [leap year], 2005,
2011).
   [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
      [ set day-of-week 5 ]                    ;; September 1st is Friday (2006, 2012 [leap year], 2017).
   ] ] ] ] ] ]                           ;; Closes out five ifelse queries and one if query above.
        while [ counts-days < start-day ]       ;; Runs this loop until counts-days is greater than the
start-day.
         [
           set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
           set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
             if day-of-week > 7                ;; This happens when the day-of-week is greater than
seven.
               [ set day-of-week 1 ]           ;; The day of week restarts at Monday.
           holidays                           ;; Runs the commands to determine holidays
         ]
      ]
      [
     ifelse start-month = "October"              ;; Checks to see if the start-month is "October".
      [
      set month-num 10
     ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
      [ set day-of-week 1 ]                    ;; October 1st is Monday (2001, 2007, 2012 [leap year],
2018).
   [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
      [ set day-of-week 2 ]                    ;; October 1st is Tuesday (2002, 2013).
   [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
      [ set day-of-week 3 ]                    ;; October 1st is Wednesday (2003, 2008 [leap year],
2014).
   [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
      [ set day-of-week 4 ]                    ;; October 1st is Thursday (2004 [leap year], 2009, 2015).
   [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
      [ set day-of-week 5 ]                    ;; October 1st is Friday (2010, 2016 [leap year]).
   [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
      [ set day-of-week 6 ]                    ;; October 1st is Saturday (2000 [leap year], 2005, 2011).
   [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
      [ set day-of-week 7 ]                    ;; October 1st is Sunday (2006, 2012 [leap year], 2017).
   ] ] ] ] ] ]                           ;; Closes out five ifelse queries and one if query above.
        while [ counts-days < start-day ]       ;; Runs this loop until counts-days is greater than the
start-day.
         [
           set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
           set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
```

```
          if day-of-week > 7               ;; This happens when the day-of-week is greater than
seven.
            [ set day-of-week 1 ]           ;; The day of week restarts at Monday.
        ]
      ]
      [
    ifelse start-month = "November"             ;; Checks to see if the start-month is "November".
      [
      set month-num 11
    ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
      [ set day-of-week 4 ]                   ;; November 1st is Thursday (2001, 2007, 2012 [leap
year], 2018).
   [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
      [ set day-of-week 5 ]                   ;; November 1st is Friday (2002, 2013).
   [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
      [ set day-of-week 6 ]                   ;; November 1st is Saturday (2003, 2008 [leap year],
2014).
   [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
      [ set day-of-week 7 ]                   ;; November 1st is Sunday (2004 [leap year], 2009, 2015).
   [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
      [ set day-of-week 1 ]                   ;; November 1st is Monday (2010, 2016 [leap year]).
   [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
      [ set day-of-week 2 ]                   ;; November 1st is Tuesday (2000 [leap year], 2005,
2011).
   [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
      [ set day-of-week 3 ]                   ;; November 1st is Wednesday (2006, 2012 [leap year],
2017).
   ] ] ] ] ] ]                                ;; Closes out five ifelse queries and one if query above.
       while [ counts-days < start-day ]       ;; Runs this loop until counts-days is greater than the
start-day.
        [
          set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
          set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
          if day-of-week > 7               ;; This happens when the day-of-week is greater than
seven.
            [ set day-of-week 1 ]           ;; The day of week restarts at Monday.
          holidays                         ;; Runs the commands to determine holidays.
        ]
      ]
      [
    if start-month = "December"               ;; Checks to see if the start-month is "December".
      [
      set month-num 12
    ifelse year = 2001 OR year = 2007 OR year = 2012 OR year = 2018      ;; Checks the year.
      [ set day-of-week 6 ]                   ;; December 1st is Saturday (2001, 2007, 2012 [leap year],
2018).
   [ ifelse year = 2002 OR year = 2013           ;; Checks the year.
      [ set day-of-week 7 ]                   ;; December 1st is Sunday (2002, 2013).
   [ ifelse year = 2003 OR year = 2008 OR year = 2014      ;; Checks the year.
```

```
      [ set day-of-week 1 ]                  ;; December 1st is Monday (2003, 2008 [leap year],
2014).
   [ ifelse year = 2009 OR year = 2015           ;; Checks the year.
      [ set day-of-week 2 ]                  ;; December 1st is Tuesday (2004 [leap year], 2009,
2015).
   [ ifelse year = 2004 OR year = 2010           ;; Checks the year.
      [ set day-of-week 3 ]                  ;; December 1st is Wednesday (2010, 2016 [leap year]).
   [ ifelse year = 2005 OR year = 2011 OR year = 2016      ;; Checks the year.
      [ set day-of-week 4 ]                  ;; December 1st is Thursday (2000 [leap year], 2005,
2011).
   [ if year = 2000 OR year = 2006 OR year = 2017 ;; Checks the year.
      [ set day-of-week 5 ]                  ;; December 1st is Friday (2006, 2012 [leap year], 2017).
   ] ] ] ] ] ]                           ;; Closes out five ifelse queries and one if query above.
        while [ counts-days < start-day ]       ;; Runs this loop until counts-days is greater than the
start-day.
         [
           set counts-days counts-days + 1      ;; Adds one to counts-days to progress the loop.
           set day-of-week day-of-week + 1      ;; Adds one to the day-of-week to change the day-
of-week.
             if day-of-week > 7               ;; This happens when the day-of-week is greater than
seven.
               [ set day-of-week 1 ]          ;; The day of week restarts at Monday.
           holidays                          ;; Runs the commands to determine holidays.
         ]
      ] ] ] ] ] ] ] ] ] ] ] ]                ;; Closes out 11 ifelse queries and 1 if query.

 ifelse day-of-week = 1                        ;; Checks to see if output day is the numerical day for
Monday.
   [ set output-day "Monday" ]                  ;; Sets the output day to Monday.
[ ifelse day-of-week = 2                       ;; Checks to see if output day is the numerical day for
Tuesday.
   [ set output-day "Tuesday" ]                 ;; Sets the output day to Tuesday.
[ ifelse day-of-week = 3                       ;; Checks to see if output day is the numerical day for
Wednesday.
   [ set output-day "Wednesday" ]               ;; Sets the output day to Wednesday.
[ ifelse day-of-week = 4                       ;; Checks to see if output day is the numerical day for
Thursday.
   [ set output-day "Thursday" ]                ;; Sets the output day to Thursday.
[ ifelse day-of-week = 5                       ;; Checks to see if output day is the numerical day for
Friday.
   [ set output-day "Friday" ]                  ;; Sets the output day to Friday.
[ ifelse day-of-week = 6                       ;; Checks to see if output day is the numerical day for
Saturday.
   [ set output-day "Saturday" ]                ;; Sets the output day to Saturday.
[ if day-of-week = 7                          ;; Checks to see if output day is the numerical day for
Sunday.
   [ set output-day "Sunday" ]                  ;; Sets the output day to Sunday.
] ] ] ] ] ]                                ;; Closes out 6 ifelse queries above.
end

to holidays
```

```
;;; THANKSGIVING ;;;
if month = "January"
 [
   ifelse calendar-counter >= 1
     [
       set holiday? true
     ]
     [
       set holiday? false
     ]
 ]
if month = "July"
 [
   ifelse calendar-counter = 4
     [
     set holiday? true
     ]
     [
     set holiday? false
     ]
 ]
if month = "September"
 [
   ifelse calendar-counter = 5
     [
       set holiday? true
     ]
     [
       set holiday? false
     ]
 ]

if month = "November"
 [
   if day-of-week = 4 and holiday-counter < 3
     [
     set holiday-counter holiday-counter + 1
     ]
   if holiday-counter = 4 and day-of-week = 3
     [
       set holiday? true
     ]
   if holiday-counter = 4 and day-of-week = 4
     [
       set holiday? true
     ]
   if holiday-counter = 4 and day-of-week = 5
     [
       set holiday? true
       set holiday-counter calendar-counter
     ]
```

```
    if calendar-counter = holiday-counter + 1
      [
        set holiday? false
      ]
    if day-of-week = 5 and holiday-counter = 3
      [
      set holiday-counter holiday-counter + 1
      ]
  ]
 if month = "December"
  [
    if calendar-counter >= 23
      [
        set holiday-counter 0
        set holiday? true
      ]
  ]
End
```

APPENDIX B: TCEQ Region 11 Emission Point-Sources

Site	Account #	NO_x TPY	TPY	Longitude	Latitude
DECKER CREEK POWER PLANT	TH0004D	631.94	40.3533	-97.6126	30.30414
MCNEIL PLANT & QUARRY	TH0010I	417.76	5.6704	-97.7176	30.45622
AUSTIN HOT MIX	TH0015V	3.64	40.72	-97.6813	30.24939
ED BLUESTEIN SITE	TH0065G	15.3795	21.7553	-97.6627	30.27322
HAL C WEAVER POWER PLANT	TH0104V	407.6003	10.2819	-97.7353	30.28649
SPANSION AUSTIN FACILITY	TH0142N	6.5657	15.8305	-97.7207	30.21867
INTEGRATED CIRCUIT MFG OAK HILL FAB	TH0172E	8.1049	16.4759	-97.8673	30.23725
AUSTIN AMERICAN STATESMAN	TH0191A	0.0031	0.0008	-97.7438	30.2583
BFI SUNSET FARMS LANDFILL	TH0232L	10.32	15.937	-97.629	30.34077
3M AUSTIN CENTER	TH0243G	37.448	13.5842	-97.8426	30.39742
AUSTIN COUNTER TOPS	TH0247V	0	14.693	-97.6738	30.37391
AUSTIN TERMINAL	TH0310Q	5.13	65.3912	-97.628	30.32524
AUSTIN COMMUNITY LANDFILL	TH0502F	42.055	8.8725	-97.6767	30.15765
SUNSET FARMS ELECTRIC	TH0522W	29.59	0.45	-97.629	30.34077
AUSTIN FABRICATION FACILITY	TH0602A	68.0175	23.0461	-97.6375	30.37495
SAND HILL ENERGY CENTER	TH0760E	131.1013	3.851	-97.6121	30.21096

APPENDIX B CONTINUED

Site	Account #	NO_x TPY	TPY	Longitude	Latitude
CREEDMOOR LANDFILL	TH0787H	0	0	-97.7566	30.1076
TEXAS LEHIGH CEMENT CO	HK0014M	2159	181.02	-97.8576	30.05311
CENTRAL HEATING & UTILITIES	HK0036C	1.8028	0.9096	-97.9446	29.89123
HAYS ENERGY FACILITY	HK0108C	145.444	14.1089	-97.9897	29.78084
AQUATIC INDUSTRIES INC	WK0116E	0	3.0824	-97.87	30.57848
COUPLAND PUMP STATION	WK0148O	45.866	3.4891	-97.3989	30.48026
DURCON LABORATORY TOPS INCORPORATED	WK0171T	1.0189	21.8463	-97.3873	30.5708
SIM GIDEON POWER PLANT	BC0015L	266.17	21.4026	-97.2654	30.14289
HANSON BRICK ELGIN FACILITY	BC0018F	11.875	3.7209	-97.3374	30.33091
ROSANKY STATION	BC0041K	0	0	-97.2881	29.91974
HILBIG GAS STORAGE FACILITY	BC0057S	1.2309	6.618	-97.4115	30.00809
ELGIN PLANT	BC0059O	54.9823	44.3283	-97.2965	30.32459
LOST PINES 1 POWER PLANT	BC0082T	184.9	18.8077	-97.2654	30.14289
BASTROP ENERGY CENTER	BC0083R	191.0089	18.5373	-97.55	30.14629
CENTER UNION GAS COMPRESSOR STATION	BCA004D	15.5	6.41	-97.0648	30.03189
LULING GAS PLANT	CA0011B	178.3392	26.916	-97.7321	29.73383
PRAIRIE LEA COMPRESSOR STATION	CA0027J	37.618	27.117	-97.7292	29.73431
FAYETTE POWER PROJECT	FC0018G	6452.357	226.507	-96.7498	29.91628
GIDDINGS PLANT	FC0033K	437.37	100.8103	-96.9148	30.04928
LAGRANGE PLANT	FC0051I	138.149	17.856	-96.887	30.00599
WINCHESTER POWER PARK	FCA001A	12.44	3.2813	-96.9932	30.02472
MAIN PLANT	LF0053U	0	40.7832	-96.8679	30.16984
QFRP PLANT	LF0056O	0	30.1554	-96.8685	30.16865
TC FERGUSON POWER PLANT	LL0006O	449.956	47.4659	-98.3723	30.55806
HUNTER PLANT	CS0018B	896.135	52.5835	-98.0397	29.80516
BALCONES PLANT	CS0022K	2309.506	20.5118	-98.1834	29.67461

APPENDIX C: Continuous Air Monitoring Stations (CAMS) Present in Model

Site	CAMS #	Operator	Longitude	Latitude
Austin Northwest	3	TCEQ	-97.7603	30.35444
Austin Audubon Society	38	TCEQ	-97.8723	30.48317
Fayette County	601	Capital Area Council of Governments	-96.7459	29.96247
Dripping Springs School	614	Capital Area Council of Governments	-98.0833	30.21462
McKinney Roughs	684	Capital Area Council of Governments	-97.4589	30.14088
CAPCOG Lake Georgetown	690	Capital Area Council of Governments	-97.7346	30.66644
CAPCOG San Marcos Staples Road	675	Capital Area Council of Governments	-97.9289	29.86228

REFERENCES

Allen, Jeanni. 2002. "The Ozone We Breath." Research Features. NASA Earth Observatory. https://earthobservatory.nasa.gov/Features/OzoneWeBreathe/. Last Accessed: December 2, 2017.

Allen, Jeannie. 2004. "Tango in the Atmosphere: Ozone and Climate Change." Research Features. NASA Earth Observatory. https://www.giss.nasa.gov/research/features/200402_tango/. Last Accessed: December 1, 2017.

Axelrod, Robert. 2001. "The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration." Princeton University Press, Princeton.

Chung, Y. "Ground-level ozone and regional transport of air pollutants". 1977. SciTech Connect. EBSCOhost. Last Accessed: March 2, 2018.

Coates, J, Mar, K, Ojha, N, & Butler, T. 2016. "The influence of temperature on ozone production under varying NOx conditions - a modelling study". Atmospheric Chemistry & Physics. 16. 18. pp. 11601-11615. Sustainability Reference Center. EBSCOhost. Last Accessed: March 31, 2018.

Crooks, A. 2015. "Agent-Based Modeling and Geographical Information Systems". In *Geocomputation*, ed. Chris Brunsdon and Alex Singleton. 63-77. London: Sage.

Buhr, M, Ryerson, T, Frost, G, Goldan, P, Holloway, J, Hübler, G, Jobson, B, Kuster, W, McKeen, S, Parrish, D, Roberts, J, Sueper, D, Trainer, M, Williams, J, & Fehsenfeld, F. 1998. "Emissions lifetimes and ozone formation in power plant plumes". Journal Of Geophysical Research Atmospheres. 103, D17. p. 22569-22583. Scopus®, EBSCOhost. Last Accessed: February 19, 2018.

Ghazi S., Khadir T., Dugdale J. 2014. "Multi-Agent-based Simulation of Environmental Pollution Issues: A Review." In: Corchado J.M. et al. (eds) Highlights of Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection. PAAMS 2014. Communications in Computer and Information Science, vol 430. Springer, Cham

Godish, T, Davis, W, & Fu, J. 2015. "Air Quality." Boca Raton: CRC Press. Taylor & Francis Group. Texas State – Alkek Library's Catalog. Last Accessed: October 3, 2017.

Hudak, PF. 2014. "Spatial Pattern of Ground-Level Ozone Concentration in Dallas-Fort Worth Metropolitan Area". International Journal of Environmental Research. 8, 4, pp. 897-902. Environment Complete. EBSCOhost. Last Accessed: October 30, 2017.

Kuhn, H. W. & Neumann, J. & Morgenstern, O. & Rubinstein, A. 2007. "Theory of Games and Economic Behavior." Princeton: Princeton University Press. Project MUSE database. Last Accessed: December 27, 2017.

McElroy, C T & Fogal, P F. 2008. "Ozone: From discovery to protection". Atmosphere-Ocean. 46:1, 1-13, DOI: 10.3.3137/ao.460101.

Munir, S. 2013. “Spatial-temporal analysis of traffic-related ground level ozone”. British Library. EThOS. EBSCOhost. Last Accessed: October 30, 2017.

National Agricultural Library (NAL). 2016. “What is Rural?” https://www.nal.usda.gov/ric/what-is-rural. Last Accessed: December 3, 2017.

National Research Council (NRC). 1991. “3 Criteria for Designing and Evaluating Ozone Reduction Strategies”. Rethinking the Ozone Problem in Urban and Regional Air Pollution. Washington, DC: The National Academies Press. DOI: 10.17226/1889.

NRC. 1997. “Remote Sensing”. Rediscovering Geography: New Relevance for Science and Society. Chapter 4 Geography Techniques. Page 54. Paragraph 3. National Academy Press. Washington D.C.. https://www.nap.edu/read/4913/chapter/6#48. Last Accessed: December 7, 2017.

National Science Foundation (NSF). 2017. “Kinetic Theory of Gases”. Physical and Theoretical Chemistry. LibreTexts. https://chem.libretexts.org/Core/Physical_and_Theoretical_Chemistry/Physical_Properties_of_Matter/States_of_Matter/Properties_of_Gases/Kinetic_Theory_of_Gases/Kinetic_Theory_of_Gases. Last accessed: December 27, 2017.

Oprea M. 2012. “Agent-Based Modeling of an Air Quality Monitoring and Analysis System for Urban Regions.” In: Iliadis L., Maglogiannis I., Papadopoulos H., Karatzas K., Sioutas S. (eds) Artificial Intelligence Applications and Innovations. AIAI 2012. IFIP Advances in Information and Communication Technology, vol 382. Springer, Berlin, Heidelberg

Pearson, K. 1930. The Life, Letters and Labors of Francis Galton. Cambridge University Press. Cambridge, MA.

Ryerson, T. B., M. P. Buhr, G. J. Frost, P. D. Goldan, J. S. Holloway, G. Hübler, B. T. Jobson, W. C. Kuster, S. A. McKeen, D. D. Parrish, J. M. Roberts, D. T. Sueper, M. Trainer, J. Williams, and F. C. Fehsenfeld. 1998. "Emissions lifetimes and ozone formation in power plant plumes." Journal of Geophysical Research: Atmospheres 103:22569-22583.

Sibson, R. "A Brief Description of Natural Neighbor Interpolation," chapter 2 in Interpolating Multivariate Data. New York: John Wiley & Sons, 1981. 21–36.

Texas Commission on Environmental Quality (TCEQ). 2015. "Detailed Data from the Point Source Emissions Inventory". https://www.tceq.texas.gov/airquality/point-source-ei/psei.html.

U.S. Energy Information Administration. Glossary. "B." Base Load Plant. https://www.eia.gov/tools/glossary/index.php?id=B. Last Accessed February 19, 2018.

Wilensky, U. & Stroup, W., 1999. HubNet. http://ccl.northwestern.edu/netlogo/hubnet.html. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

World Health Organization (WHO). 2017. "WHO's commitment to air quality: from the 1950s to today". Environmental and Health. News. http://www.euro.who.int/en/health-topics/environment-and-health/pages/news/news/2017/02/whos-commitment-to-air-quality-from-the-1950s-to-today. Last Accessed: December 7, 2017.