**Title: Spatiotemporal Modeling of Air Quality (PM10 Levels) and a Disease Case in Arizona Counties**

**Author:** Jiseon Yang

**Instructor:** Shiwei Lan

**Course: STP598,** Spatiotemporal Analysis

**Date:** November 29, 2024

Table of Contents

[**Abstract** 3](#_Toc183819500)

[**1. Introduction** 3](#_Toc183819501)

[**2. Data and Methods** 3](#_Toc183819502)

[**2.1 Data Sources** 3](#_Toc183819503)

[**2.2 Spatial Data** 4](#_Toc183819504)

[**2.3 Models and Statistical Framework** 4](#_Toc183819505)

[**2.3.1 Moran's I (Spatial Autocorrelation Test)** 4](#_Toc183819506)

[**2.3.2 Spatial Regression Models** 4](#_Toc183819507)

[**2.3.3 Spatiotemporal Model (INLA)** 5](#_Toc183819508)

[**2.3.4. Time Series Models: ARIMA with Seasonality (Seasonal ARIMA, SARIMA)** 5](#_Toc183819509)

[**2.3.5. Spatiotemporal Model (INLA) with Seasonality** 5](#_Toc183819510)

[**2.3.6. Seasonal Decomposition** 5](#_Toc183819511)

[**2.3.7. Seasonal Regression Models** 6](#_Toc183819512)

[**3. Results** 6](#_Toc183819513)

[**3.1 Descriptive Analysis** 6](#_Toc183819514)

[**3.2 Spatial Autocorrelation** 7](#_Toc183819515)

[**3.3 Spatial Regression Models** 7](#_Toc183819516)

[**3.4 Spatiotemporal Modeling (INLA)** 8](#_Toc183819517)

[**3.5. Time Series Models: ARIMA with Seasonality (Seasonal ARIMA, SARIMA)** 10](#_Toc183819518)

[**3.6. Spatiotemporal Model (INLA) with Seasonality** 11](#_Toc183819519)

[**3.6.1. Model Specifications:** 11](#_Toc183819520)

[**3.6.2. Analysis** 12](#_Toc183819521)

[**3.7. Comparison of modeling results** 14](#_Toc183819522)

[**4. Discussion** 15](#_Toc183819523)

[**5. Conclusion** 16](#_Toc183819524)

[**References** 16](#_Toc183819525)

[**Codes:** 17](#_Toc183819526)

[Step 1: Load data 17](#_Toc183819527)

[Step 2: Adjust Dataset for Spatial Analysis 17](#_Toc183819528)

[Step 3: Load and Prepare Shapefile 18](#_Toc183819529)

[Step 4: Visualize Data 18](#_Toc183819530)

[Step 5: Spatial Autocorrelation 19](#_Toc183819531)

[Step 6: Temporal Analysis 20](#_Toc183819532)

[Step 7: Regression Modeling 21](#_Toc183819533)

[Step 8: Spatial Regression (Lag and Error Models) 21](#_Toc183819534)

[Step 9: Spatiotemporal Modeling with INLA 23](#_Toc183819535)

# **Abstract**

This study investigates the spatiotemporal dynamics of air quality (PM10 levels) and its association with disease cases across Arizona counties from 2000 to 2022. Using advanced statistical techniques, including ARIMA, spatial regression models, and Bayesian spatiotemporal modeling via the Integrated Nested Laplace Approximation (INLA) framework, the research captures spatial heterogeneity, temporal trends, and seasonal effects. Key findings reveal significant seasonal and spatial patterns: PM10 levels and disease cases peak during specific months, influenced by environmental factors such as temperature and arid conditions. While spatial regression models highlight the significance of spatial autocorrelation and heterogeneity, ARIMA effectively captures seasonal cycles in PM10. The INLA model emerges as the most comprehensive approach, integrating spatial, temporal, and seasonal components to identify high-risk counties and periods. The findings underscore the critical role of seasonality and localized factors in disease propagation and air quality, offering actionable insights for targeted public health interventions and environmental management strategies. This work emphasizes the necessity of multi-dimensional modeling to address complex environmental health challenges.

# **1. Introduction**

The relationship between air quality and public health has been widely studied, with particulate matter (PM10) recognized as a key environmental hazard contributing to respiratory diseases or airborne infection. Arizona's unique climatic conditions, characterized by high temperatures and arid environments, make it an ideal case for studying PM10's impact on airborne diseases. This study addresses the following objectives:

1. To examine the impact of meteorological factors on PM10 concentrations.
2. To explore the association between PM10 levels and disease cases.
3. To model spatial and temporal dependencies using advanced statistical techniques.

# **2. Data and Methods**

## **2.1 Data Sources**

* **Air Quality and Meteorological Data**: Monthly averages for PM10, temperature, humidity, and wind speed were collected for 14 Arizona counties from 2000 to 2022.
* **Disease Data**: Monthly reported cases of a specific airborne disease.

## **2.2 Spatial Data**

The spatial structure of Arizona counties was obtained from a shapefile of U.S. counties. The adjacency matrix for spatial weights was derived using the queen contiguity method.

## **2.3 Models and Statistical Framework**

Three modeling approaches were implemented:

### **2.3.1 Moran's I (Spatial Autocorrelation Test)**

Spatial autocorrelation for PM10 levels was quantified using Moran's I:

A black text on a white background

Description automatically generated

Where:

* N: Number of observations (counties).
* W = Σi Σj wij ​: Sum of spatial weights.
* wij ​: Weight between counties i and j (1 if adjacent, 0 otherwise).
* xi : PM10 level for county i.
* xi\_bar: Mean PM10 level.

### **2.3.2 Spatial Regression Models**

1. **Spatial Lag Model**: captures spatial spillover effects

y = ρWy + Xβ + ε

where:

* y: PM10 levels.
* Wy: Spatially lagged dependent variable.
* ρ: Spatial lag coefficient.
* X: Covariate matrix (temperature, humidity, wind speed).
* β: Regression coefficients.
* ϵ∼N(0,σ2) : Error term.

1. **Error Model**:

y = Xβ + λWε + v

where:

* λ : Spatial autocorrelation coefficient in residuals.
* v ∼ N(0,σ2): Residual error.

### **2.3.3 Spatiotemporal Model (INLA)**

1. The Bayesian hierarchical model:

Casesit ​ ∼ Poisson(λit​),

log(λit​) = β0​ + β1​TEMPit ​+ β2​PM10\_meanit​ + ui + vt

where:

* Casesit ​: Disease cases in county i at time t
* **λit​** : Expected cases in county i at year t (Poisson mean)
* β0, β1, β2 : Fixed effects for intercept, temperature, and PM10 for county i at year t
* **ui​:** Spatial random effect for county i, modeled as IID. ui∼N (0, τu−1)
* **vt ​:** Temporal random walk effect for year t. vt ∼ N (vt−1,τv−1)

### **2.3.4. Time Series Models: ARIMA with Seasonality (Seasonal ARIMA, SARIMA)**

1. SARIMA: seasonal autoregressive (SAR) differencing, moving average (SMA)

Φp(B) Φp(Bs)(1-B)d(1-BS)Dyt = Θq​(B)ΘQ​(Bs)εt​

Where:

* B: Backshift operator.
* s: Seasonal period (e.g., s=12s = 12s=12 for monthly data).
* Φp(Bs), ΘQ​(Bs) : Seasonal AR and MA components.
* (1-BS)Dyt: Seasonal differencing.

### **2.3.5. Spatiotemporal Model (INLA) with Seasonality**

1. Adding a **cyclical component** for monthly seasonality

log(λit​) = β0​ + β1​TEMPit ​+ β2​PM10\_meanit​ + ui + vt + mt(model = cyclic)

Where:

* **mt** : cyclic random effect for month t, Captures monthly seasonality in a cyclic fashion. mt ~ N(mt−1​,τmonth−1​)
* mt−1: the cyclic random effect for the preceding month.
* τm−1 : the precision (inverse variance) of the monthly random effect.

### **2.3.6. Seasonal Decomposition**

Seasonal decomposition was performed on the time series data to identify trends, seasonal patterns, and residual variability. Using an additive model, the decomposition splits the observed time series (yt ​) into:

yt = Tt + St + Rt

* Tt​: Long-term trend component.
* St ​: Seasonal component.
* Rt ​: Random component.

This decomposition provides insights into the regular seasonal cycles in PM10 levels and disease cases, as well as any long-term changes in the data.

### **2.3.7. Seasonal Regression Models**

Seasonal regression models were developed to quantify the relationship between disease cases and environmental factors, while explicitly accounting for seasonal effects. The model includes a cyclical term for seasonality (monthly or yearly effects):

Casesit ​= β0 ​+ β1​TEMPit ​+ β2​PM10\_meanit ​+ ∑ ​γk​⋅Monthk​ (k=1..12) +ϵit​

# **3. Results**

## **3.1 Descriptive Analysis**

* **PM10 Levels**: Higher PM10 levels were observed in in Maricopa and Pima counties, urbanized counties.

A map of the state of arizona

Description automatically generated

**Figure 1. Spatial distribution of average PM10 levels across 14 Arizona counties.** PM10 levels (measured in µg/m³) are color-coded, with higher concentrations observed in urban counties such as Maricopa and Pima. Gray areas represent missing data or counties not includ

* **Disease Cases**: Disease cases varied significantly across years, with peaks during dry seasons.

A graph of disease cases over time and months

Description automatically generated

**Figure 2. The temporal distribution of disease cases.** **Left:** **yearly trend** of disease cases aggregated over time, indicating fluctuations and peaks in specific years. **Right:** **monthly trend**, how cases vary throughout a typical year, with an apparent seasonal pattern.

## **3.2 Spatial Autocorrelation**

Moran's I for PM10, I=0.148 and p=0.053, Indicating weak positive spatial autocorrelation.

A computer screen shot of a math problem

Description automatically generated

## **3.3 Spatial Regression Models**

* **Spatial Lag Model**: y\_hat = 72241 − 54.20TEMP − 1411.67HUM −646.12WIND,

ρ=0.07 (p=0.84, insignificant).

A screenshot of a computer

Description automatically generated

* **Spatial Error Model**: Y\_hat = 82834 + 93.54TEMP − 1760.02HUM − 1541.99WIND,

λ=−1.28, p<0.05 🡺 Spatial autocorrelation in residuals was significant.

A screenshot of a computer

Description automatically generated

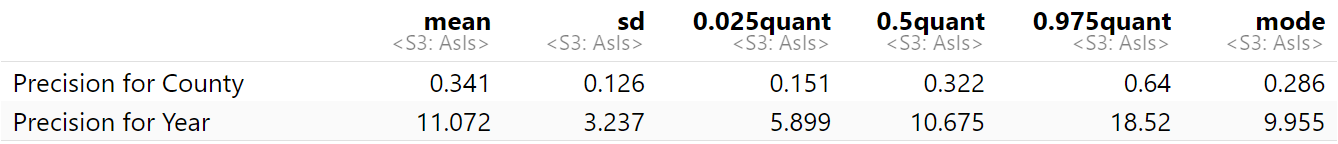
## **3.4 Spatiotemporal Modeling (INLA)**

log(λit​) = 4.176 − 0.004⋅TEMP\_AVGit ​ +0.000⋅PM10\_meanit​ + ui​+ vt​

* Spatial random effect for county I (**ui​)**, τu = 0.341
* Temporal random effect for year t (vt), RW1, τv = 11.072
* Marginal log-likelihood: −2650.91.

A screenshot of a computer

Description automatically generated



A map of the state of arizona

Description automatically generated

**Figure3:** Predicted Disease Cases by County in Arizona. The map visualizes the spatial distribution of predicted disease cases based on the spatiotemporal INLA model. Higher predicted cases are concentrated in urbanized counties such as Maricopa and Pima, reflecting potential environmental and demographic influences.

## **3.5. Time Series Models: ARIMA with Seasonality (Seasonal ARIMA, SARIMA)**

A screenshot of a computer code

Description automatically generatedA screenshot of a computer

Description automatically generated

* Model Parameters: Non-seasonal ARIMA (p, d, q) = (0, 1, 3) and Seasonal ARIMA (P, D, Q, S) = (1, 1, 1, 12) with a 12-month seasonal cycle.
* Key Findings: PM10 levels are significantly influenced by both short-term moving averages and seasonal patterns.
  + The **Seasonal Moving Average (SMA1)** parameter (-1.000, p < 0.0001) indicates significant smoothing of seasonal patterns.
  + Seasonal peaks, particularly in the summer months, align with environmental factors like dust storms and arid conditions.
* Model Performance: Residuals show low autocorrelation (ACF1 = -0.007), confirming that the model effectively captures the temporal structure.
* Seasonal Insights: Seasonal peaks in PM10 levels align with specific times of the year, which might be driven by environmental factors such as dust and weather conditions.
* Seasonality and Autoregression: While SMA1 significantly explains cyclic variations, the non-significance of SAR1 suggests weaker seasonal autoregressive effects in this dataset.
* Conclusion: The SARIMA model effectively captures and highlights the seasonality in PM10 levels, providing critical insights into temporal patterns of air pollution.

A graph showing a number of levels

Description automatically generated with medium confidence

**Figure 4: PM10 Levels in Maricopa County**. PM10 levels (in micrograms per cubic meter) over time in Maricopa County. Seasonal fluctuations and trends highlight periods of increased air pollution, providing insights into environmental factors influencing public health.

## **3.6. Spatiotemporal Model (INLA) with Seasonality**

The spatiotemporal analysis leverages the **Integrated Nested Laplace Approximation (INLA)** framework to examine disease cases across counties over time, integrating spatial, temporal, and seasonal effects.

### **3.6.1. Model Specifications:**

* **Model Equation**:

log(λit​) = 4.176 − 0.004⋅TEMPit ​+ 0.000⋅PM10\_meanit​ + ui​ + vt​ + f(Montht​,model = cyclic)

Where:

* λit ​: Expected number of disease cases in county i at time t.
* TEMPit ​: Average temperature (β1 =−0.004), a slight negative effect).
* PM10\_meanit ​: Average PM10 levels (β2=0.000, indicating no significant direct effect).
* County-specific spatial random effect (τu = 0.371): moderate spatial variability).
* Year-specific temporal random effect (τv=11.072): strong temporal precision).
* Cyclic effect for months (τm=115.269): substantial seasonal variations).

### **3.6.2. Analysis**

**1. Spatial and Temporal Dynamics**:

* Counties with higher PM10 levels (e.g., Maricopa, Yuma) exhibit significant seasonal variations in disease cases.
* Spatial variability (τu) reflects unobserved heterogeneity, potentially related to demographic or socio-economic factors.
* Temporal effects (τv) highlight year-to-year changes, emphasizing long-term environmental impacts.

2. **Seasonality**:

* The cyclic monthly component (τm=115.269) effectively captures seasonal trends in disease cases, with peaks in specific months correlating with environmental factors.

3. **Model Performance**:

* The **marginal log-likelihood** of the model is −14316.77, demonstrating robust fit and predictive capability.

A screenshot of a computer

Description automatically generated



* County variability (Precision=0.371) indicates moderate spatial heterogeneity.
* Year (11.065) and Month (115.269) show higher precision: temporal dynamics and the significant impact of seasonality.
* Marginal log-likelihood (−14316.77): a robust model fit.
* Variability across counties and seasons is effectively captured, supporting the hypothesis of spatial and seasonal impacts on case patterns.

A graph of a graph of a graph

Description automatically generated with medium confidence

**Figure5. Residual Diagnostics for SARIMA (1,1,1) × (1,1,1)₁₂ Model. (Top):** Standardized residuals over time. Displays the distribution and randomness of residuals, indicating whether the model adequately captures temporal dependencies. **(Bottom Left):** Autocorrelation Function (ACF) of residuals. Shows no significant autocorrelation, suggesting that residuals are random. **(Bottom Center):** Normal Q-Q plot of standardized residuals. Highlights whether the residuals follow a normal distribution, with most points aligning along the diagonal. **(Bottom Right):** Ljung-Box p-values over varying lags. P-values above the significance threshold indicate no significant serial correlation in residuals.

A graph of different types of lines

Description automatically generated with medium confidence

**Figure 6: Decomposition of Additive Time Series for PM10 Levels. (Top):** Observed PM10 time series. Represents raw monthly PM10 data for Maricopa County from 2000 to 2022. **(Second):** Trend component of the time series. Captures long-term changes in PM10 levels, indicating overall increasing trends until a plateau. **(Third):** Seasonal component. Reveals regular fluctuations within each year, showing peak and low PM10 levels corresponding to seasonal cycles. **(Bottom):** Random component. Represents residuals after removing the trend and seasonal effects, highlighting short-term variability and noise in the data.

A map of the state of arizona

Description automatically generated

**Figure 7: Predicted Cases with Seasonal Effect in Arizona.** Colors range from **purple** (low cases) to **yellow** (high cases), with Maricopa County having the highest predictions due to environmental and population factors. Maricopa County shows the highest predicted cases (highlighted in yellow/orange), reflecting its higher population density and environmental conditions influencing case numbers. Spatial variation highlights the interplay of environmental factors, including PM10 levels and temperature, across the state.

## **3.7. Comparison of modeling results**

Based on the modeling results, the **spatiotemporal INLA framework** demonstrates robust performance with precise estimates for spatial (τu=0.371) and temporal (τv=11.072) random effects, along with a strong cyclic effect for seasonality (τm=115.269). These metrics confirm the model's ability to effectively capture the interplay of spatial, temporal, and seasonal variations in disease cases across Arizona counties. The high marginal log-likelihood (−14316.77) further validates the model fit and predictive capability.

The **ARIMA model**, with a seasonal cycle of 12 months, highlights strong seasonal components such as the significant SMA1 parameter (−1.000, p<0.0001) and adequately captures short-term moving averages and trends. The low autocorrelation of residuals (ACF1 = −0.007) suggests that the ARIMA model efficiently captures temporal dependencies and can effectively isolate and explain seasonality in PM10 levels.

The **spatial regression models** provide valuable insights into spatial spillover effects and autocorrelation. However, they show limited utility in addressing temporal dependencies and complex seasonal cycles, as evidenced by the insignificant spatial lag coefficient (ρ=0.07, p=0.84) in the Spatial Lag Model and the significant spatial error coefficient (λ=−1.28, p<0.05) in the Spatial Error Model.

# **4. Discussion**

The spatiotemporal modeling results using the INLA framework provide valuable insights into the interplay between meteorological variables, PM10 levels, and disease cases in Arizona counties. The model’s predictions, visualized in **Figure 3**, reveal distinct spatial patterns, with higher predicted cases concentrated in urbanized and densely populated counties such as Maricopa and Pima. These findings align with previous research suggesting that urban areas, characterized by elevated PM10 concentrations and environmental stressors, tend to experience higher disease burdens.

The spatial random effect (f(County)) highlights unobserved heterogeneity among counties, indicating that factors not explicitly included in the model—such as socioeconomic disparities, healthcare accessibility, or additional pollutants—may influence disease patterns. Similarly, the temporal random effect (f(Year)) captures smooth year-to-year variability, reflecting broader trends in environmental conditions and public health outcomes over time.

The fixed effects further reveal that temperature has a slight but significant negative association with disease cases (β1=−0.004), while PM10 levels have an inconclusive direct effect (β2=0.000). This suggests that while PM10 might contribute to disease propagation, its impact may be confounded by other meteorological or demographic factors, such as humidity or advancements in healthcare infrastructure. Further exploration is warranted to disentangle these effects and better understand the complex interactions driving disease cases.

The observed spatial autocorrelation in the residuals of the spatial error model (λ=−1.28, p<0.05) underscores the importance of including spatial dependencies in the INLA framework. This autocorrelation suggests that disease spread is influenced not only by local conditions but also by regional spillover effects, such as population mobility or shared environmental exposures.

A key highlight of this study is the strong seasonal patterns confirmed by both ARIMA and INLA models. The SARIMA decomposition and INLA's f(Month, model = cyclic) component effectively capture these patterns, revealing seasonal peaks in disease cases and PM10 levels that align with environmental and climatic changes. These findings emphasize the critical role of seasonality in disease modeling and underscore the need for public health interventions to account for temporal cycles. Moreover, the spatiotemporal heterogeneity observed in the INLA model suggests that localized strategies tailored to specific county-level characteristics would be most effective.

# **5. Conclusion**

This study offers a detailed spatiotemporal analysis of PM10 levels and disease cases in Arizona counties, leveraging advanced statistical techniques to uncover critical insights into their complex relationship. By utilizing Bayesian hierarchical modeling with the INLA framework, the analysis highlights significant spatial heterogeneity and temporal trends, revealing localized and temporal dynamics that influence disease propagation. Spatial variability underscores the importance of county-specific factors, such as demographic and socioeconomic conditions, while temporal trends indicate stable long-term patterns useful for public health planning.

Seasonal effects, identified using ARIMA and INLA’s cyclic model, emphasize the influence of environmental factors, such as temperature variations and dust cycles, on both PM10 levels and disease cases. While PM10’s direct effect on disease cases was inconclusive, the study highlights the necessity of incorporating confounding factors, such as humidity and healthcare access, in future analyses. These findings provide a robust framework for policymakers and public health officials to implement targeted, seasonally informed interventions aimed at mitigating environmental health risks in high-burden areas like Maricopa and Pima counties.

# **References**

1. Banerjee, S., Carlin, B. P., & Gelfand, A. E. (2014). *Hierarchical Modeling and Analysis for Spatial Data*. CRC Press.
2. Robert H. Shumway and David S. Stoffer. *Time Series Analysis and its applications, with examples in R (4th Edition).* Springer
3. Rue, H., Martino, S., & Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models using INLA. *Journal of the Royal Statistical Society*, 71(2), 319-392.

# **Codes:**

Final\_mydata\_14counties

Jiseon Yang

2024-11-29

# Step 1: Load data

# Load libraries  
library(sf)

## Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf\_use\_s2() is TRUE

library(spdep)

## Loading required package: spData

library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(viridis)

## Warning: package 'viridis' was built under R version 4.4.2

## Loading required package: viridisLite

library(tmap)

## Breaking News: tmap 3.x is retiring. Please test v4, e.g. with  
## remotes::install\_github('r-tmap/tmap')

# Load your dataset  
final\_data <- read.csv("C:/Users/jyang/OneDrive - Arizona State University/10 Classes\_OneDrive/2024 8F\_STP598\_Spatiotemporal Analysis/Final Project/Final\_data.csv")  
  
# Preview the dataset structure  
str(final\_data)

## 'data.frame': 3864 obs. of 8 variables:  
## $ Year : int 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ...  
## $ Month : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ County : chr "Apache" "Apache" "Apache" "Apache" ...  
## $ TEMP\_AVG: num NA NA NA NA NA NA NA NA NA NA ...  
## $ HUM\_AVG : num NA NA NA NA NA NA NA NA NA NA ...  
## $ WIND\_AVG: num NA NA NA NA NA NA NA NA NA NA ...  
## $ PM10 : int NA NA NA 704 568 588 609 635 565 913 ...  
## $ Cases : int 1 0 0 0 0 1 0 0 0 1 ...

### Step 2: Adjust Dataset for Spatial Analysis

# Filter counties (excluding "Greenlee")  
az\_counties <- c(  
 "Apache", "Cochise", "Coconino", "Gila", "Graham",  
 "La Paz", "Maricopa", "Mohave", "Navajo", "Pima",   
 "Pinal", "Santa Cruz", "Yavapai", "Yuma"  
)  
  
# Verify the dataset includes only these counties  
final\_data <- final\_data %>% filter(County %in% az\_counties)  
  
# Aggregate data (if needed, by Year and Month)  
agg\_data <- final\_data %>%  
 group\_by(County, Year, Month) %>%  
 summarize(  
 TEMP\_AVG = mean(TEMP\_AVG, na.rm = TRUE),  
 HUM\_AVG = mean(HUM\_AVG, na.rm = TRUE),  
 WIND\_AVG = mean(WIND\_AVG, na.rm = TRUE),  
 PM10 = mean(PM10, na.rm = TRUE),  
 Cases = sum(Cases, na.rm = TRUE),  
 .groups = "drop"  
 )

### Step 3: Load and Prepare Shapefile

# Load Arizona counties shapefile  
shapefile\_path <- "C:/Users/jyang/OneDrive - Arizona State University/10 Classes\_OneDrive/2024 8F\_STP598\_Spatiotemporal Analysis/Final Project/tl\_2024\_us\_county/az\_counties.shp"  
az\_counties\_sf <- st\_read(shapefile\_path)

## Reading layer `az\_counties' from data source   
## `C:\Users\jyang\OneDrive - Arizona State University\10 Classes\_OneDrive\2024 8F\_STP598\_Spatiotemporal Analysis\Final Project\tl\_2024\_us\_county\az\_counties.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 15 features and 18 fields  
## Geometry type: POLYGON  
## Dimension: XY  
## Bounding box: xmin: -114.8163 ymin: 31.33234 xmax: -109.0452 ymax: 37.00373  
## Geodetic CRS: NAD83

# Filter shapefile to match dataset counties (excluding "Greenlee")  
az\_counties\_sf <- az\_counties\_sf %>% filter(NAME %in% az\_counties)  
  
# Join aggregated data to shapefile  
az\_combined <- az\_counties\_sf %>%  
 left\_join(agg\_data, by = c("NAME" = "County"))  
  
# Check for mismatches in county names  
setdiff(az\_counties\_sf$NAME, unique(agg\_data$County)) # Counties in shapefile but not in agg\_data

## character(0)

setdiff(unique(agg\_data$County), az\_counties\_sf$NAME) # Counties in agg\_data but not in shapefile

## character(0)

### Step 4: Visualize Data

# Plot PM10 levels  
ggplot(data = az\_combined) +  
 geom\_sf(aes(fill = PM10), color = "white") +  
 scale\_fill\_viridis\_c(option = "C", na.value = "grey50") +  
 labs(title = "PM10 Levels by County",  
 fill = "PM10 (µg/m³)") +  
 theme\_minimal()

A map of state with different colors

Description automatically generated

### Step 5: Spatial Autocorrelation

# Create neighbors list and weights matrix  
nb <- poly2nb(az\_counties\_sf, queen = TRUE)  
lw <- nb2listw(nb, style = "W")  
  
# Moran's I for PM10  
county\_pm10 <- agg\_data %>%  
 group\_by(County) %>%  
 summarize(  
 PM10\_mean = mean(PM10, na.rm = TRUE),  
 .groups = "drop"  
 )  
  
# Ensure order matches shapefile  
county\_pm10 <- county\_pm10 %>%  
 arrange(match(County, az\_counties\_sf$NAME))  
  
# Run Moran's I  
moran\_test <- moran.test(county\_pm10$PM10\_mean, lw)  
print(moran\_test)

##   
## Moran I test under randomisation  
##   
## data: county\_pm10$PM10\_mean   
## weights: lw   
##   
## Moran I statistic standard deviate = 1.6114, p-value = 0.05355  
## alternative hypothesis: greater  
## sample estimates:  
## Moran I statistic Expectation Variance   
## 0.14825315 -0.07692308 0.01952756

### Step 6: Temporal Analysis

# Aggregate cases over time  
time\_series <- final\_data %>%  
 group\_by(Year) %>%  
 summarize(Cases = sum(Cases))  
  
# Plot time series  
ggplot(time\_series, aes(x = Year, y = Cases)) +  
 geom\_line() +  
 geom\_point() +  
 labs(title = "Disease Cases Over Time",  
 x = "Year", y = "Cases")

A graph showing the disease cases over time

Description automatically generated

# Aggregate cases over time  
time\_series <- final\_data %>%  
 group\_by(Month) %>%  
 summarize(Cases = sum(Cases))  
  
# Plot time series  
ggplot(time\_series, aes(x = Month, y = Cases)) +  
 geom\_line() +  
 geom\_point() +  
 labs(title = "Disease Cases Over Time",  
 x = "Month", y = "Cases")

A graph with a line graph and numbers

Description automatically generated

### Step 7: Regression Modeling

# Linear model: PM10 and temperature predicting cases  
lm\_model <- lm(Cases ~ PM10 + TEMP\_AVG + HUM\_AVG + WIND\_AVG, data = agg\_data)  
summary(lm\_model)

##   
## Call:  
## lm(formula = Cases ~ PM10 + TEMP\_AVG + HUM\_AVG + WIND\_AVG, data = agg\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -534.85 -20.31 -3.12 9.33 970.57   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.4910744 21.8642193 0.022 0.982   
## PM10 0.0182837 0.0004511 40.533 <2e-16 \*\*\*  
## TEMP\_AVG -0.3782118 0.2078975 -1.819 0.069 .   
## HUM\_AVG 0.4137455 0.2638547 1.568 0.117   
## WIND\_AVG 0.0142269 0.0760729 0.187 0.852   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 132.8 on 2262 degrees of freedom  
## (1597 observations deleted due to missingness)  
## Multiple R-squared: 0.4321, Adjusted R-squared: 0.4311   
## F-statistic: 430.2 on 4 and 2262 DF, p-value: < 2.2e-16

### Step 8: Spatial Regression (Lag and Error Models)

# Summarize PM10 and include other variables  
county\_pm10 <- agg\_data %>%  
 group\_by(County) %>%  
 summarize(  
 PM10\_mean = mean(PM10, na.rm = TRUE),  
 TEMP\_AVG = mean(TEMP\_AVG, na.rm = TRUE),  
 HUM\_AVG = mean(HUM\_AVG, na.rm = TRUE),  
 WIND\_AVG = mean(WIND\_AVG, na.rm = TRUE),  
 .groups = "drop"  
 )  
  
# Ensure county order matches the spatial weights matrix  
county\_pm10 <- county\_pm10 %>%  
 arrange(match(County, az\_counties\_sf$NAME))  
  
  
library(spatialreg)

## Warning: package 'spatialreg' was built under R version 4.4.2

## Loading required package: Matrix

##   
## Attaching package: 'spatialreg'

## The following objects are masked from 'package:spdep':  
##   
## get.ClusterOption, get.coresOption, get.mcOption,  
## get.VerboseOption, get.ZeroPolicyOption, set.ClusterOption,  
## set.coresOption, set.mcOption, set.VerboseOption,  
## set.ZeroPolicyOption

# Spatial Lag Model  
lag\_model <- lagsarlm(PM10\_mean ~ TEMP\_AVG + HUM\_AVG + WIND\_AVG, data = county\_pm10, listw = lw)  
  
# Summarize the model  
summary(lag\_model)

##   
## Call:lagsarlm(formula = PM10\_mean ~ TEMP\_AVG + HUM\_AVG + WIND\_AVG,   
## data = county\_pm10, listw = lw)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4885.95 -2453.03 -941.55 2430.69 6606.62   
##   
## Type: lag   
## Coefficients: (asymptotic standard errors)   
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 72241.121 35971.117 2.0083 0.04461  
## TEMP\_AVG -54.198 215.928 -0.2510 0.80182  
## HUM\_AVG -1411.665 552.222 -2.5563 0.01058  
## WIND\_AVG -646.116 418.304 -1.5446 0.12244  
##   
## Rho: 0.069234, LR test value: 0.025255, p-value: 0.87373  
## Asymptotic standard error: 0.35041  
## z-value: 0.19758, p-value: 0.84337  
## Wald statistic: 0.039038, p-value: 0.84337  
##   
## Log likelihood: -105.6228 for lag model  
## ML residual variance (sigma squared): 12801000, (sigma: 3577.9)  
## Number of observations: 11   
## Number of parameters estimated: 6   
## AIC: 223.25, (AIC for lm: 221.27)  
## LM test for residual autocorrelation  
## test value: 2.6383, p-value: 0.10432

# Spatial Error Model  
error\_model <- errorsarlm(PM10\_mean ~ TEMP\_AVG + HUM\_AVG + WIND\_AVG, data = county\_pm10, listw = lw)  
summary(error\_model)

##   
## Call:errorsarlm(formula = PM10\_mean ~ TEMP\_AVG + HUM\_AVG + WIND\_AVG,   
## data = county\_pm10, listw = lw)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3334.00 -1715.54 -375.74 1442.64 4317.47   
##   
## Type: error   
## Coefficients: (asymptotic standard errors)   
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 82834.241 25890.133 3.1995 0.001377  
## TEMP\_AVG 93.539 145.554 0.6426 0.520458  
## HUM\_AVG -1760.017 404.812 -4.3477 1.375e-05  
## WIND\_AVG -1541.986 302.934 -5.0902 3.578e-07  
##   
## Lambda: -1.2837, LR test value: 3.9256, p-value: 0.047557  
## Asymptotic standard error: 0.26753  
## z-value: -4.7984, p-value: 1.5994e-06  
## Wald statistic: 23.025, p-value: 1.5994e-06  
##   
## Log likelihood: -103.6727 for error model  
## ML residual variance (sigma squared): 5831600, (sigma: 2414.9)  
## Number of observations: 11   
## Number of parameters estimated: 6   
## AIC: 219.35, (AIC for lm: 221.27)

# Compare AIC values  
AIC(lag\_model, error\_model)

## df AIC  
## lag\_model 6 223.2457  
## error\_model 6 219.3454

### Step 9: Spatiotemporal Modeling with INLA

library(INLA)

## Loading required package: sp

## This is INLA\_24.06.27 built 2024-06-27 02:36:04 UTC.  
## - See www.r-inla.org/contact-us for how to get help.  
## - List available models/likelihoods/etc with inla.list.models()  
## - Use inla.doc(<NAME>) to access documentation

# Create county-level aggregated data  
obs <- agg\_data %>%  
 group\_by(County, Year) %>%  
 summarize(  
 PM10\_mean = mean(PM10, na.rm = TRUE),  
 TEMP\_AVG = mean(TEMP\_AVG, na.rm = TRUE),  
 Cases = sum(Cases, na.rm = TRUE),  
 .groups = "drop"  
 )  
  
  
# Create adjacency graph with correct order  
nb <- poly2nb(az\_counties\_sf, queen = TRUE)  
adj\_matrix <- nb2mat(nb, style = "B", zero.policy = TRUE)  
rownames(adj\_matrix) <- az\_counties\_sf$NAME  
graph <- inla.read.graph(adj\_matrix)  
  
  
# Convert County to a factor, ensuring levels match the adjacency matrix  
obs$County <- factor(obs$County, levels = rownames(adj\_matrix))  
  
  
# Check for mismatches  
setdiff(levels(obs$County), rownames(adj\_matrix)) # Counties in data but not in adjacency matrix

## character(0)

setdiff(rownames(adj\_matrix), levels(obs$County)) # Counties in adjacency matrix but not in data

## character(0)

# Ensure the graph's nodes match County levels  
stopifnot(all(levels(obs$County) %in% rownames(adj\_matrix)))  
  
# Fit INLA model  
# formula <- Cases ~ TEMP\_AVG + PM10\_mean +  
# f(County, model = "bym", graph = graph) +  
# f(Year, model = "rw1")  
  
formula <- Cases ~ TEMP\_AVG + PM10\_mean +  
 f(County, model = "iid") + # Temporarily use iid for spatial effect  
 f(Year, model = "rw1") # Retain temporal random walk  
  
  
fit\_inla <- inla(  
 formula,  
 family = "poisson",  
 data = obs,  
 control.predictor = list(compute = TRUE)  
)  
  
  
# Summarize results  
summary(fit\_inla)

## Time used:  
## Pre = 0.484, Running = 0.371, Post = 0.0671, Total = 0.921   
## Fixed effects:  
## mean sd 0.025quant 0.5quant 0.975quant mode kld  
## (Intercept) 4.176 0.488 3.210 4.175 5.142 4.175 0  
## TEMP\_AVG -0.004 0.001 -0.006 -0.004 -0.003 -0.004 0  
## PM10\_mean 0.000 0.000 0.000 0.000 0.000 0.000 0  
##   
## Random effects:  
## Name Model  
## County IID model  
## Year RW1 model  
##   
## Model hyperparameters:  
## mean sd 0.025quant 0.5quant 0.975quant mode  
## Precision for County 0.341 0.126 0.151 0.322 0.64 0.286  
## Precision for Year 11.072 3.237 5.899 10.675 18.52 9.955  
##   
## Marginal log-Likelihood: -2650.91   
## is computed   
## Posterior summaries for the linear predictor and the fitted values are computed  
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')

#### 

print(adj\_matrix)

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]  
## Yuma 0 0 0 1 1 0 0 0 0 0 0 1 0  
## Pinal 0 0 0 1 1 0 0 1 1 0 0 0 0  
## Navajo 0 0 0 0 0 0 1 1 1 0 0 0 0  
## Maricopa 1 1 0 0 1 0 0 1 0 0 1 1 0  
## Pima 1 1 0 1 0 1 0 0 1 0 0 0 1  
## Cochise 0 0 0 0 1 0 0 0 1 0 0 0 1  
## Coconino 0 0 1 0 0 0 0 1 0 1 1 0 0  
## Gila 0 1 1 1 0 0 1 0 1 0 1 0 0  
## Graham 0 1 1 0 1 1 0 1 0 0 0 0 0  
## Mohave 0 0 0 0 0 0 1 0 0 0 1 1 0  
## Yavapai 0 0 0 1 0 0 1 1 0 1 0 1 0  
## La Paz 1 0 0 1 0 0 0 0 0 1 1 0 0  
## Santa Cruz 0 0 0 0 1 1 0 0 0 0 0 0 0  
## Apache 0 0 1 0 0 0 0 0 1 0 0 0 0  
## [,14]  
## Yuma 0  
## Pinal 0  
## Navajo 1  
## Maricopa 0  
## Pima 0  
## Cochise 0  
## Coconino 0  
## Gila 0  
## Graham 1  
## Mohave 0  
## Yavapai 0  
## La Paz 0  
## Santa Cruz 0  
## Apache 0  
## attr(,"call")  
## nb2mat(neighbours = nb, style = "B", zero.policy = TRUE)

class(az\_counties) # Should include "sf"

## [1] "character"

colnames(az\_counties) # Should include "NAME" for county names

## NULL

az\_counties <- st\_read("C:/Users/jyang/OneDrive - Arizona State University/10 Classes\_OneDrive/2024 8F\_STP598\_Spatiotemporal Analysis/Final Project/tl\_2024\_us\_county/az\_counties.shp")

## Reading layer `az\_counties' from data source   
## `C:\Users\jyang\OneDrive - Arizona State University\10 Classes\_OneDrive\2024 8F\_STP598\_Spatiotemporal Analysis\Final Project\tl\_2024\_us\_county\az\_counties.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 15 features and 18 fields  
## Geometry type: POLYGON  
## Dimension: XY  
## Bounding box: xmin: -114.8163 ymin: 31.33234 xmax: -109.0452 ymax: 37.00373  
## Geodetic CRS: NAD83

library(ggplot2)  
library(sf)  
  
# Ensure az\_counties is an sf object with a column "NAME" for county names  
ggplot(data = az\_counties) +  
 geom\_sf(fill = "lightblue", color = "white") + # Draw the map  
 geom\_sf\_text(aes(label = NAME), size = 3, color = "black") + # Add county names  
 labs(title = "Arizona Counties",  
 subtitle = "County names labeled on the map") +  
 theme\_minimal()

## Warning in st\_point\_on\_surface.sfc(sf::st\_zm(x)): st\_point\_on\_surface may not  
## give correct results for longitude/latitude data

A map of the state of arizona

Description automatically generated

# Load required libraries  
library(sf)  
library(dplyr)  
library(ggplot2)  
library(viridis)  
library(tmap)  
library(INLA)  
library(fields)

## Loading required package: spam

## Spam version 2.11-0 (2024-10-03) is loaded.  
## Type 'help( Spam)' or 'demo( spam)' for a short introduction   
## and overview of this package.  
## Help for individual functions is also obtained by adding the  
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##   
## Attaching package: 'spam'

## The following object is masked from 'package:INLA':  
##   
## Oral

## The following object is masked from 'package:Matrix':  
##   
## det

## The following objects are masked from 'package:base':  
##   
## backsolve, forwardsolve

##   
## Try help(fields) to get started.

library(spdep)  
  
# Load your dataset  
final\_data <- read.csv("C:/Users/jyang/OneDrive - Arizona State University/10 Classes\_OneDrive/2024 8F\_STP598\_Spatiotemporal Analysis/Final Project/Final\_data.csv")  
  
# Preview the dataset structure  
str(final\_data)

## 'data.frame': 3864 obs. of 8 variables:  
## $ Year : int 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ...  
## $ Month : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ County : chr "Apache" "Apache" "Apache" "Apache" ...  
## $ TEMP\_AVG: num NA NA NA NA NA NA NA NA NA NA ...  
## $ HUM\_AVG : num NA NA NA NA NA NA NA NA NA NA ...  
## $ WIND\_AVG: num NA NA NA NA NA NA NA NA NA NA ...  
## $ PM10 : int NA NA NA 704 568 588 609 635 565 913 ...  
## $ Cases : int 1 0 0 0 0 1 0 0 0 1 ...

# Define the list of Arizona counties  
az\_counties <- c(  
 "Apache", "Cochise", "Coconino", "Gila", "Graham",  
 "La Paz", "Maricopa", "Mohave", "Navajo", "Pima",  
 "Pinal", "Santa Cruz", "Yavapai", "Yuma"  
)  
  
# Filter the dataset to include only these counties  
final\_data <- final\_data %>% filter(County %in% az\_counties)  
  
# Aggregate the data by Year and Month  
agg\_data <- final\_data %>%  
 group\_by(County, Year, Month) %>%  
 summarize(  
 TEMP\_AVG = mean(TEMP\_AVG, na.rm = TRUE),  
 HUM\_AVG = mean(HUM\_AVG, na.rm = TRUE),  
 WIND\_AVG = mean(WIND\_AVG, na.rm = TRUE),  
 PM10 = mean(PM10, na.rm = TRUE),  
 Cases = sum(Cases, na.rm = TRUE),  
 .groups = "drop"  
 )  
  
# Load Arizona counties shapefile  
shapefile\_path <- "C:/Users/jyang/OneDrive - Arizona State University/10 Classes\_OneDrive/2024 8F\_STP598\_Spatiotemporal Analysis/Final Project/tl\_2024\_us\_county/az\_counties.shp"  
az\_counties\_sf <- st\_read(shapefile\_path)

## Reading layer `az\_counties' from data source   
## `C:\Users\jyang\OneDrive - Arizona State University\10 Classes\_OneDrive\2024 8F\_STP598\_Spatiotemporal Analysis\Final Project\tl\_2024\_us\_county\az\_counties.shp'   
## using driver `ESRI Shapefile'  
## Simple feature collection with 15 features and 18 fields  
## Geometry type: POLYGON  
## Dimension: XY  
## Bounding box: xmin: -114.8163 ymin: 31.33234 xmax: -109.0452 ymax: 37.00373  
## Geodetic CRS: NAD83

# Filter shapefile to match the dataset counties  
az\_counties\_sf <- az\_counties\_sf %>% filter(NAME %in% az\_counties)  
  
# Join aggregated data with the shapefile  
az\_combined <- az\_counties\_sf %>%  
 left\_join(agg\_data, by = c("NAME" = "County"))  
  
# Ensure there are no mismatches in the join  
setdiff(az\_counties\_sf$NAME, unique(agg\_data$County)) # Should return character(0)

## character(0)

setdiff(unique(agg\_data$County), az\_counties\_sf$NAME) # Should return character(0)

## character(0)

# Create neighbors list and spatial weights matrix  
nb <- poly2nb(az\_counties\_sf, queen = TRUE)  
lw <- nb2listw(nb, style = "W")  
  
# Plot PM10 levels across counties  
ggplot(data = az\_combined) +  
 geom\_sf(aes(fill = PM10), color = "white") +  
 scale\_fill\_viridis\_c(option = "C", na.value = "grey50") +  
 labs(title = "PM10 Levels by County",  
 fill = "PM10 (µg/m³)") +  
 theme\_minimal()

A map of state with different colors

Description automatically generated

# Temporal trend plot for disease cases  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

# Plot for Yearly Cases  
time\_series <- final\_data %>%  
 group\_by(Year) %>%  
 summarize(Cases = sum(Cases))  
  
  
plot\_year <- ggplot(time\_series, aes(x = Year, y = Cases)) +  
 geom\_line() +  
 geom\_point() +  
 labs(title = "Disease Cases Over Time",  
 x = "Yearh", y = "Cases")  
  
# Plot for Monthly Cases  
time\_series <- final\_data %>%  
 group\_by(Month) %>%  
 summarize(Cases = sum(Cases))  
  
plot\_month <- ggplot(time\_series, aes(x = Month, y = Cases)) +  
 geom\_line() +  
 geom\_point() +  
 scale\_x\_continuous(breaks = seq(1, 12, by = 1)) + # Ensures every month is displayed  
 labs(title = "Disease Cases Over Time",  
 x = "Month", y = "Cases")  
  
grid.arrange(plot\_year, plot\_month, ncol = 2)

A graph of disease cases

Description automatically generated

# Prepare for Spatiotemporal Modeling with INLA  
obs <- agg\_data %>%  
 group\_by(County, Year) %>%  
 summarize(  
 PM10\_mean = mean(PM10, na.rm = TRUE),  
 TEMP\_AVG = mean(TEMP\_AVG, na.rm = TRUE),  
 Cases = sum(Cases, na.rm = TRUE),  
 .groups = "drop"  
 )  
  
# Convert County to a factor and ensure levels match adjacency matrix  
obs$County <- factor(obs$County, levels = az\_counties\_sf$NAME)  
  
# Ensure adjacency matrix matches the data  
adj\_matrix <- nb2mat(nb, style = "B", zero.policy = TRUE)  
rownames(adj\_matrix) <- az\_counties\_sf$NAME  
graph <- inla.read.graph(adj\_matrix)  
  
# Fit INLA spatiotemporal model  
formula <- Cases ~ TEMP\_AVG + PM10\_mean +  
 f(County, model = "iid") + # Temporarily use iid for spatial effect  
 f(Year, model = "rw1") # Retain temporal random walk  
  
fit\_inla <- inla(  
 formula,  
 family = "poisson",  
 data = obs,  
 control.predictor = list(compute = TRUE)  
)  
  
# Summarize INLA results  
summary(fit\_inla)

## Time used:  
## Pre = 0.244, Running = 0.366, Post = 0.0474, Total = 0.658   
## Fixed effects:  
## mean sd 0.025quant 0.5quant 0.975quant mode kld  
## (Intercept) 4.176 0.488 3.210 4.175 5.142 4.175 0  
## TEMP\_AVG -0.004 0.001 -0.006 -0.004 -0.003 -0.004 0  
## PM10\_mean 0.000 0.000 0.000 0.000 0.000 0.000 0  
##   
## Random effects:  
## Name Model  
## County IID model  
## Year RW1 model  
##   
## Model hyperparameters:  
## mean sd 0.025quant 0.5quant 0.975quant mode  
## Precision for County 0.341 0.126 0.151 0.322 0.64 0.286  
## Precision for Year 11.072 3.237 5.899 10.675 18.52 9.955  
##   
## Marginal log-Likelihood: -2650.91   
## is computed   
## Posterior summaries for the linear predictor and the fitted values are computed  
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')

# Add predicted cases to the dataset for visualization  
obs$Cases\_pred <- fit\_inla$summary.fitted.values$mean  
  
# Join predicted cases back to the spatial dataset  
az\_combined <- az\_combined %>%  
 left\_join(obs %>% select(County, Cases\_pred), by = c("NAME" = "County"))

## Warning in sf\_column %in% names(g): Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 1 of `x` matches multiple rows in `y`.  
## ℹ Row 300 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

# Plot predicted cases  
ggplot(data = az\_combined) +  
 geom\_sf(aes(fill = Cases\_pred), color = "white") +  
 scale\_fill\_viridis\_c(option = "C", na.value = "grey50") +  
 labs(title = "Predicted Cases by County",  
 fill = "Predicted Cases") +  
 theme\_minimal()

A map of the state of arizona

Description automatically generated

#Impact of Seasonality on Research Modeling

1. Time Series Models :ARIMA with Seasonality

# install.packages("forecast")  
library(forecast)

## Warning: package 'forecast' was built under R version 4.4.2

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

maricopa\_pm10 <- ts(final\_data$PM10[final\_data$County == "Maricopa"], frequency = 12)  
sarima\_model <- auto.arima(maricopa\_pm10, seasonal = TRUE)  
summary(sarima\_model)

## Series: maricopa\_pm10   
## ARIMA(1,1,1)(2,0,1)[12]   
##   
## Coefficients:  
## ar1 ma1 sar1 sar2 sma1  
## 0.3125 -0.9773 -0.7022 0.0759 0.5548  
## s.e. 0.0872 0.0123 0.2175 0.1144 0.2149  
##   
## sigma^2 = 32132016: log likelihood = -2766.76  
## AIC=5545.52 AICc=5545.84 BIC=5567.22  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 541.7634 5606.558 4459.248 -17.09702 40.87713 0.4789699 -0.041938

library(astsa)

##   
## Attaching package: 'astsa'

## The following object is masked from 'package:forecast':  
##   
## gas

## The following object is masked from 'package:viridis':  
##   
## unemp

# Plot and decompose  
plot(maricopa\_pm10, main = "PM10 Levels in Maricopa")

A graph of a time line

Description automatically generated with medium confidence

decompose(maricopa\_pm10)

## $x  
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1 8105 5175 5293 5042 4370 8842 22301 12366 10725 11247 11473 13558  
## 2 19541 16447 25295 12080 14948 9969 23626 13225 14256 21414 16048 4954  
## 3 3210 5173 3465 4332 6055 21421 10553 10256 8914 7757 11788 17299  
## 4 10499 21646 13104 17395 10595 18277 8220 13428 16164 16200 3831 4164  
## 5 4288 2442 5111 8545 12340 13903 15410 13842 9364 14913 18859 17942  
## 6 22302 12690 23912 15245 15718 13632 9744 16748 18487 4879 4416 5905  
## 7 3401 4651 12717 15231 16301 16720 11383 12429 13707 18823 22862 22455  
## 8 15993 18615 17802 27245 15538 14289 19199 23868 5121 5311 5637 7575  
## 9 5334 12301 15756 16927 14791 11813 14118 15973 22426 24782 27637 13451  
## 10 17030 17332 23417 14420 19051 20169 22781 3942 4333 5921 4956 5684  
## 11 13682 17401 19079 18898 10571 13620 18362 28474 25109 22530 16203 21683  
## 12 20978 20672 17464 19441 20804 21110 5126 3447 5870 8310 5239 13731  
## 13 13302 21335 12256 16776 12739 30466 17615 21169 24474 13488 19655 19368  
## 14 29585 19715 19305 20750 20420 7364 3834 5173 4690 7087 9180 11216  
## 15 19044 11833 13868 10854 29000 20002 21308 14609 16561 14242 15920 29491  
## 16 17704 28329 15778 14866 5076 5198 5546 4546 5685 13552 12556 18383  
## 17 12901 11867 14425 26793 18531 17292 16510 13928 18499 19885 19793 14989  
## 18 24187 16114 15255 3670 7135 5188 6354 11050 14136 13467 25188 18868  
## 19 20606 13064 21042 23084 23141 18267 13306 18931 29369 9566 21138 31101  
## 20 23109 15138 4655 5667 5263 4002 7830 17462 21868 28808 18903 23143  
## 21 14157 18053 27190 17386 20526 16442 14701 28840 20110 18295 24903 28510  
## 22 17387 6889 4438 4866 5236 10479 21404 19342 13341 10349 16188 15230  
## 23 13223 20047 16194 13406 19918 15397 27522 15978 10667 21201 17581 15664  
##   
## $seasonal  
## Jan Feb Mar Apr May Jun Jul  
## 1 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 2 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 3 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 4 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 5 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 6 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 7 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 8 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 9 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 10 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 11 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 12 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 13 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 14 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 15 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 16 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 17 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 18 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 19 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 20 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 21 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 22 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## 23 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472 -712.56487  
## Aug Sep Oct Nov Dec  
## 1 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 2 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 3 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 4 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 5 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 6 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 7 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 8 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 9 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 10 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 11 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 12 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 13 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 14 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 15 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 16 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 17 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 18 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 19 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 20 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 21 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 22 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
## 23 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
##   
## $trend  
## Jan Feb Mar Apr May Jun Jul  
## 1 NA NA NA NA NA NA 10351.250  
## 2 15051.042 15142.042 15324.958 15895.708 16509.958 16342.083 15303.125  
## 3 10886.875 10218.458 9872.167 9080.542 8334.000 8670.875 9488.958  
## 4 13076.375 13111.333 13545.583 14199.458 14219.708 13340.875 12534.792  
## 5 9352.583 9669.417 9403.333 9066.375 9638.917 10839.167 12163.833  
## 6 15916.333 15801.333 16302.542 16264.583 15244.708 14141.375 12852.292  
## 7 10834.958 10723.292 10344.167 10726.000 12075.583 13533.750 14748.000  
## 8 17920.750 18723.042 18841.917 17921.167 16640.458 15302.750 14238.625  
## 9 11757.708 11217.042 11609.125 13141.458 14869.417 16030.917 16763.083  
## 10 19511.458 19371.125 18115.958 16576.208 14845.292 13576.625 13113.500  
## 11 11554.875 12392.917 14280.750 15838.458 16999.125 18134.375 19105.000  
## 12 20517.667 18923.375 17078.958 15684.833 14635.500 13847.333 13196.167  
## 13 12903.458 14162.250 15675.833 16666.750 17483.167 18318.708 19232.042  
## 14 18834.792 17594.083 16103.250 15012.208 14309.042 13532.917 12754.042  
## 15 12876.500 13997.750 14885.542 15678.292 16257.250 17299.542 18005.167  
## 16 15933.417 14857.375 13984.917 13503.000 13334.083 12731.083 12068.125  
## 17 13963.250 14811.000 15735.833 16533.625 17099.042 17259.167 17588.000  
## 18 14172.917 13629.833 13328.125 12878.917 12836.292 13222.708 13235.125  
## 19 17645.250 18263.292 19226.375 19698.542 19367.250 19708.208 20322.208  
## 20 14875.583 14586.208 14212.458 14701.667 15410.292 14985.583 14281.000  
## 21 19600.292 20360.667 20761.500 20250.208 20062.167 20535.792 20894.000  
## 22 15667.125 15550.667 14872.875 14259.750 13565.542 12649.083 11922.250  
## 23 16424.833 16539.583 16288.000 16628.750 17138.958 17215.083 NA  
## Aug Sep Oct Nov Dec  
## 1 11297.417 12600.500 13727.167 14461.167 14948.875  
## 2 14152.917 12773.583 11541.167 10847.792 10954.417  
## 3 10479.042 11567.042 12512.958 13246.417 13304.583  
## 4 11475.833 10342.625 9640.833 9344.792 9235.250  
## 5 13341.417 14551.792 15614.333 16034.250 16163.708  
## 6 11729.792 10928.375 10461.333 10485.042 10638.000  
## 7 15854.500 16648.208 17360.667 17829.458 17696.375  
## 8 13531.417 13183.083 12667.917 12206.875 12072.583  
## 9 17460.042 17988.875 18203.625 18276.667 18802.333  
## 10 12976.875 12799.000 12804.833 12638.083 12011.875  
## 11 19545.292 19614.292 19569.625 20018.625 20757.083  
## 12 12903.958 12714.583 12386.542 11939.458 11993.250  
## 13 19843.000 20069.208 20528.500 21014.125 20371.583  
## 14 11986.417 11431.458 10792.583 10737.750 11621.833  
## 15 18636.667 19403.583 19650.333 18820.667 17207.000  
## 16 11182.083 10439.792 10880.375 11937.958 13002.500  
## 17 18235.208 18446.750 17517.875 16079.583 15100.417  
## 18 12958.833 13072.875 14122.917 15598.750 16810.625  
## 19 20512.917 19916.542 18508.042 17037.417 15698.125  
## 20 14029.458 15089.875 16517.125 17641.375 18795.667  
## 21 20563.417 19150.250 17680.583 16521.833 15636.292  
## 22 12297.000 13335.083 14180.750 15148.333 15965.000  
## 23 NA NA NA NA NA  
##   
## $random  
## Jan Feb Mar Apr May  
## 1 NA NA NA NA NA  
## 2 3970.04403 1243.09138 9470.63305 -3675.87453 -1201.70407  
## 3 -8196.78930 -5107.32528 -6906.57528 -4608.70786 -1918.74574  
## 4 -3097.28930 8472.79972 -940.99195 3335.37547 -3264.45407  
## 5 -5584.49763 -7289.28362 -4791.74195 -381.54119 3061.33759  
## 6 5865.75237 -3173.20028 7110.04972 -879.74953 833.54593  
## 7 -7953.87263 -6134.15862 1873.42472 4644.83381 4585.67093  
## 8 -2447.66430 -169.90862 -1539.32528 9463.66714 -742.20407  
## 9 -6943.62263 1022.09138 3647.46638 3925.37547 281.83759  
## 10 -3001.37263 -2100.99195 4801.63305 -2016.37453 4565.96259  
## 11 1607.21070 4946.21638 4298.84138 3199.37547 -6067.87074  
## 12 -59.58097 1686.75805 -114.36695 3896.00047 6528.75426  
## 13 -121.37263 7110.88305 -3919.24195 249.08381 -4383.91241  
## 14 10230.29403 2059.04972 2702.34138 5877.62547 6471.21259  
## 15 5647.58570 -2226.61695 -1516.95028 -4684.45786 13103.00426  
## 16 1250.66903 13409.75805 1293.67472 1502.83381 -7897.82907  
## 17 -1582.16430 -3005.86695 -1810.24195 10399.20881 1792.21259  
## 18 9494.16903 2422.29972 1427.46638 -9069.08286 -5341.03741  
## 19 2440.83570 -5261.15862 1316.21638 3525.29214 4134.00426  
## 20 7713.50237 489.92472 -10056.86695 -8894.83286 -9787.03741  
## 21 -5963.20597 -2369.53362 5929.09138 -2724.37453 824.08759  
## 22 1199.96070 -8723.53362 -10934.28362 -9253.91619 -7969.28741  
## 23 -3721.74763 3445.54972 -593.40862 -3082.91619 3139.29593  
## Jun Jul Aug Sep Oct  
## 1 NA 12662.31487 1190.96259 -1711.62642 -1716.94271  
## 2 -6057.40862 9035.43987 -805.53741 1646.29025 10636.05729  
## 3 13065.79972 1776.60653 -100.66241 -2489.16809 -3992.73438  
## 4 5251.79972 -3602.22680 2074.54593 5985.24858 7322.39063  
## 5 3379.50805 3958.73153 622.96259 -5023.91809 61.89062  
## 6 -193.70028 -2395.72680 5140.58759 7722.49858 -4819.10938  
## 7 3501.92472 -2652.43513 -3303.12074 -2777.33475 2225.55729  
## 8 -698.07528 5672.93987 10458.96259 -7898.20975 -6593.69271  
## 9 -3902.24195 -1932.51847 -1364.66241 4600.99858 7341.59896  
## 10 6908.04972 10380.06487 -8912.49574 -8302.12642 -6120.60938  
## 11 -4198.70028 -30.43513 9051.08759 5658.58191 3723.59896  
## 12 7578.34138 -7357.60180 -9334.57907 -6680.70975 -3313.31771  
## 13 12462.96638 -904.47680 1448.37926 4568.66525 -6277.27604  
## 14 -5853.24195 -8207.47680 -6691.03741 -6577.58475 -2942.35937  
## 15 3018.13305 4015.39820 -3905.28741 -2678.70975 -4645.10937  
## 16 -7217.40862 -5809.56013 -6513.70407 -4590.91809 3434.84896  
## 17 348.50805 -365.43513 -4184.82907 216.12358 3130.34896  
## 18 -7719.03362 -6168.56013 -1786.45407 1226.99858 107.30729  
## 19 -1125.53362 -6303.64347 -1459.53741 9616.33191 -8178.81771  
## 20 -10667.90862 -5738.43513 3554.92093 6941.99858 13054.09896  
## 21 -3778.11695 -5480.43513 8398.96259 1123.62358 1377.64062  
## 22 -1854.40862 10194.31487 7167.37926 169.79025 -3068.52604  
## 23 -1502.40862 NA NA NA NA  
## Nov Dec  
## 1 -3159.06392 -2716.59233  
## 2 5029.31108 -7326.13400  
## 3 -1629.31392 2668.69934  
## 4 -5684.68892 -6396.96733  
## 5 2653.85275 452.57434  
## 6 -6239.93892 -6058.71733  
## 7 4861.64441 3432.90767  
## 8 -6740.77225 -5823.30066  
## 9 9189.43608 -6677.05066  
## 10 -7852.98059 -7653.59233  
## 11 -3986.52225 -399.80066  
## 12 -6871.35559 412.03267  
## 13 -1530.02225 -2329.30066  
## 14 -1728.64725 -1731.55066  
## 15 -3071.56392 10958.28267  
## 16 447.14441 4054.78267  
## 17 3542.51941 -1437.13400  
## 18 9418.35275 731.65767  
## 19 3929.68608 14077.15767  
## 20 1090.72775 3021.61600  
## 21 8210.26941 11547.99100  
## 22 868.76941 -2060.71733  
## 23 NA NA  
##   
## $figure  
## [1] 519.91430 61.86695 499.40862 -139.83381 -360.25426 -315.67472  
## [7] -712.56487 -122.37926 -163.87358 -763.22396 170.89725 1325.71733  
##   
## $type  
## [1] "additive"  
##   
## attr(,"class")  
## [1] "decomposed.ts"

1. Spatiotemporal Models (e.g., INLA)

obs <- agg\_data %>%  
 group\_by(County, Year, Month) %>% # Include Month in the grouping  
 summarize(  
 PM10\_mean = mean(PM10, na.rm = TRUE),  
 TEMP\_AVG = mean(TEMP\_AVG, na.rm = TRUE),  
 Cases = sum(Cases, na.rm = TRUE),  
 .groups = "drop"  
 )  
  
obs$Month <- as.numeric(obs$Month) # Ensure Month is numeric  
obs$Month <- factor(obs$Month, levels = 1:12) # Convert to factor if required  
  
formula <- Cases ~ TEMP\_AVG + PM10\_mean +  
 f(County, model = "iid") + # Spatial effect  
 f(Year, model = "rw1") + # Long-term trend  
 f(Month, model = "rw1", cyclic = TRUE) # Seasonal effect  
  
fit\_inla <- inla(  
 formula,  
 family = "poisson",  
 data = obs,  
 control.predictor = list(compute = TRUE)  
)  
  
summary(fit\_inla)

## Time used:  
## Pre = 0.323, Running = 0.671, Post = 0.0742, Total = 1.07   
## Fixed effects:  
## mean sd 0.025quant 0.5quant 0.975quant mode kld  
## (Intercept) 1.621 0.465 0.696 1.621 2.547 1.621 0  
## TEMP\_AVG -0.003 0.001 -0.004 -0.003 -0.002 -0.003 0  
## PM10\_mean 0.000 0.000 0.000 0.000 0.000 0.000 0  
##   
## Random effects:  
## Name Model  
## County IID model  
## Year RW1 model  
## Month RW1 model  
##   
## Model hyperparameters:  
## mean sd 0.025quant 0.5quant 0.975quant mode  
## Precision for County 0.371 0.137 0.164 0.35 0.697 0.311  
## Precision for Year 11.065 3.234 5.895 10.67 18.509 9.949  
## Precision for Month 115.269 46.629 47.155 107.58 227.480 93.341  
##   
## Marginal log-Likelihood: -14316.77   
## is computed   
## Posterior summaries for the linear predictor and the fitted values are computed  
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')

Visualize Predicted Cases

obs$Cases\_pred <- fit\_inla$summary.fitted.values$mean  
  
# Join predictions back to spatial data  
az\_combined <- az\_counties\_sf %>%  
 left\_join(obs, by = c("NAME" = "County"))  
  
# Plot predicted cases  
ggplot(data = az\_combined) +  
 geom\_sf(aes(fill = Cases\_pred), color = "white") +  
 scale\_fill\_viridis\_c(option = "C", na.value = "grey50") +  
 labs(title = "Predicted Cases with Seasonal Effect",  
 fill = "Predicted Cases") +  
 theme\_minimal()

A map of the state of arizona

Description automatically generated

1. ARIMA Model for Temporal Patterns in PM10

# Subset PM10 time series for Maricopa County  
maricopa\_pm10 <- ts(obs$PM10\_mean[obs$County == "Maricopa"], frequency = 12, start = c(2000, 1))  
  
# Fit a SARIMA model  
sarima\_model <- astsa::sarima(maricopa\_pm10, p = 1, d = 1, q = 1, P = 1, D = 1, Q = 1, S = 12)

## initial value 8.630185   
## iter 2 value 8.383700  
## iter 3 value 8.329881  
## iter 4 value 8.303998  
## iter 5 value 8.265742  
## iter 6 value 8.262294  
## iter 7 value 8.255751  
## iter 8 value 8.246517  
## iter 9 value 8.235747  
## iter 10 value 8.229411  
## iter 11 value 8.226473  
## iter 12 value 8.218453  
## iter 13 value 8.217292  
## iter 14 value 8.213814  
## iter 15 value 8.212623  
## iter 16 value 8.212459  
## iter 17 value 8.212367  
## iter 18 value 8.212363  
## iter 19 value 8.212363  
## iter 19 value 8.212363  
## final value 8.212363   
## converged  
## initial value 8.219932   
## iter 2 value 8.215996  
## iter 3 value 8.215417  
## iter 4 value 8.215016  
## iter 5 value 8.214856  
## iter 6 value 8.214779  
## iter 7 value 8.214772  
## iter 8 value 8.214772  
## iter 8 value 8.214772  
## iter 8 value 8.214772  
## final value 8.214772   
## converged  
## <><><><><><><><><><><><><><>  
##   
## Coefficients:   
## Estimate SE t.value p.value  
## ar1 0.5151 0.0881 5.8441 0.0000  
## ma1 -0.8807 0.0554 -15.9057 0.0000  
## sar1 -0.0480 0.0634 -0.7575 0.4494  
## sma1 -1.0000 0.0685 -14.5934 0.0000  
##   
## sigma^2 estimated as 11738046 on 259 degrees of freedom   
##   
## AIC = 19.30544 AICc = 19.30603 BIC = 19.37336   
##

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

# View results  
print(sarima\_model)

## $fit  
##   
## Call:  
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),   
## include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace = trc,   
## REPORT = 1, reltol = tol))  
##   
## Coefficients:  
## ar1 ma1 sar1 sma1  
## 0.5151 -0.8807 -0.0480 -1.0000  
## s.e. 0.0881 0.0554 0.0634 0.0685  
##   
## sigma^2 estimated as 11738046: log likelihood = -2533.67, aic = 5077.33  
##   
## $degrees\_of\_freedom  
## [1] 259  
##   
## $ttable  
## Estimate SE t.value p.value  
## ar1 0.5151 0.0881 5.8441 0.0000  
## ma1 -0.8807 0.0554 -15.9057 0.0000  
## sar1 -0.0480 0.0634 -0.7575 0.4494  
## sma1 -1.0000 0.0685 -14.5934 0.0000  
##   
## $ICs  
## AIC AICc BIC   
## 19.30544 19.30603 19.37336

##########  
  
library(forecast)  
  
  
# Fit SARIMA model  
sarima\_model <- auto.arima(maricopa\_pm10, seasonal = TRUE)  
summary(sarima\_model)

## Series: maricopa\_pm10   
## ARIMA(0,1,3)   
##   
## Coefficients:  
## ma1 ma2 ma3  
## -0.2904 -0.3038 -0.1669  
## s.e. 0.0613 0.0566 0.0622  
##   
## sigma^2 = 13296965: log likelihood = -2644.48  
## AIC=5296.96 AICc=5297.11 BIC=5311.43  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 168.2646 3619.98 2679.405 -3.509521 19.37895 0.5771518 -0.00699456

# Decompose and visualize  
decomposed <- decompose(maricopa\_pm10)  
plot(decomposed)

A graph of different types of time

Description automatically generated with medium confidence

1. Spatiotemporal Analysis with the SpatioTemporal Package

# install.packages("SpatioTemporal")  
library(SpatioTemporal)  
  
str(obs)

## tibble [3,864 × 7] (S3: tbl\_df/tbl/data.frame)  
## $ County : chr [1:3864] "Apache" "Apache" "Apache" "Apache" ...  
## $ Year : int [1:3864] 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...  
## $ Month : Factor w/ 12 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ PM10\_mean : num [1:3864] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ...  
## $ TEMP\_AVG : num [1:3864] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN ...  
## $ Cases : int [1:3864] 1 0 0 0 0 0 0 0 0 0 ...  
## $ Cases\_pred: num [1:3864] 0.4 0.352 0.358 0.339 0.361 ...

library(dplyr) # For dplyr functions, including `%>%`  
library(tidyverse) # For the entire tidyverse, which includes dplyr

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.1  
## ✔ readr 2.1.5   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ gridExtra::combine() masks dplyr::combine()  
## ✖ tidyr::expand() masks Matrix::expand()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ tidyr::pack() masks Matrix::pack()  
## ✖ tidyr::unpack() masks Matrix::unpack()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Check missing data percentage  
sapply(obs, function(x) sum(is.na(x) | is.nan(x)) / length(x) \* 100)

## County Year Month PM10\_mean TEMP\_AVG Cases Cases\_pred   
## 0.00000 0.00000 0.00000 12.65528 34.73085 0.00000 0.00000

# # Option 1: Remove rows with NaN values  
# obs <- obs %>%  
# filter(!is.nan(PM10\_mean) & !is.nan(TEMP\_AVG))  
  
# Option 2: Impute missing values (e.g., with mean or interpolation)  
library(zoo)

##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

obs <- obs %>%  
 mutate(  
 PM10\_mean = na.approx(PM10\_mean, na.rm = FALSE),  
 TEMP\_AVG = na.approx(TEMP\_AVG, na.rm = FALSE)  
 )  
obs <- obs %>%  
 mutate(Date = as.Date(paste(Year, Month, "01", sep = "-")))  
obs$PM10\_mean <- as.numeric(obs$PM10\_mean)  
obs$TEMP\_AVG <- as.numeric(obs$TEMP\_AVG)  
  
# If not already created, create combined\_data  
combined\_data <- obs %>%  
 group\_by(County) %>%  
 summarize(  
 TEMP\_AVG = mean(TEMP\_AVG, na.rm = TRUE),  
 PM10\_mean = mean(PM10\_mean, na.rm = TRUE),  
 Cases = sum(Cases, na.rm = TRUE),  
 .groups = "drop"  
 )  
  
# If combined\_data does not yet have spatial geometry, ensure az\_counties\_sf is available  
library(sf)  
combined\_data <- az\_counties\_sf %>%  
 left\_join(combined\_data, by = c("NAME" = "County"))  
  
# Convert to Spatial object  
library(sp)  
combined\_data\_sp <- as(combined\_data, "Spatial")  
  
# Ensure the column `ID` exists in `combined\_data\_sp`  
combined\_data\_sp$ID <- combined\_data\_sp$NAME # Copy county names as IDs if necessary  
setdiff(obs$County, combined\_data\_sp$ID)

## character(0)

# Check mismatches  
setdiff(obs$County, combined\_data\_sp$ID) # Or whatever column identifies locations in `combined\_data\_sp`

## character(0)

# Ensure consistent IDs  
obs$County <- factor(obs$County, levels = unique(combined\_data\_sp$ID))  
  
str(obs)

## tibble [3,864 × 8] (S3: tbl\_df/tbl/data.frame)  
## $ County : Factor w/ 14 levels "Yuma","Pinal",..: 14 14 14 14 14 14 14 14 14 14 ...  
## $ Year : int [1:3864] 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...  
## $ Month : Factor w/ 12 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ PM10\_mean : num [1:3864] NA NA NA NA NA NA NA NA NA NA ...  
## $ TEMP\_AVG : num [1:3864] NA NA NA NA NA NA NA NA NA NA ...  
## $ Cases : int [1:3864] 1 0 0 0 0 0 0 0 0 0 ...  
## $ Cases\_pred: num [1:3864] 0.4 0.352 0.358 0.339 0.361 ...  
## $ Date : Date[1:3864], format: "2000-01-01" "2000-02-01" ...

obs <- obs %>%  
 rename(  
 obs = PM10\_mean, # Observed values (PM10\_mean for this example)  
 date = Date, # Ensure Date column is named `date`  
 ID = County # Ensure County column is named `ID`  
 )  
  
combined\_data\_sp$ID <- combined\_data\_sp$NAME # Replace NAME with the spatial identifier column  
  
covars <- obs %>%  
 group\_by(ID) %>%  
 summarize(  
 TEMP\_AVG = mean(TEMP\_AVG, na.rm = TRUE),  
 Cases = sum(Cases, na.rm = TRUE),  
 .groups = "drop"  
 )  
  
  
data\_st <- createSTdata(  
 obs = obs,  
 spLocations = combined\_data\_sp,  
 Time = unique(obs$date),  
 Variables = c("TEMP\_AVG", "Cases"),  
 covars = covars  
)  
  
  
  
summary(data\_st)

## Warning: Unknown or uninitialised column: `type`.

## Summary of observations:  
## obs date   
## Min. : 0.0 Min. :2000-01-01   
## 1st Qu.: 171.8 1st Qu.:2005-11-01   
## Median : 550.5 Median :2011-08-01   
## Mean : 2731.7 Mean :2011-07-21   
## 3rd Qu.: 1869.2 3rd Qu.:2017-04-01   
## Max. :31101.0 Max. :2022-12-01   
##   
## Summary of covariates:  
## ID TEMP\_AVG Cases   
## Length:14 Min. :44.14 Min. : 309.0   
## Class :character 1st Qu.:53.50 1st Qu.: 451.5   
## Mode :character Median :62.53 Median : 750.5   
## Mean :60.44 Mean : 11875.6   
## 3rd Qu.:69.74 3rd Qu.: 1621.2   
## Max. :71.96 Max. :124051.0   
## NA's :1   
##   
## Summary of smooth trends:  
## date   
## Min. :2000-01-01   
## 1st Qu.:2005-09-23   
## Median :2011-06-16   
## Mean :2011-06-16   
## 3rd Qu.:2017-03-08   
## Max. :2022-12-01   
##   
## No spatio-temporal covariates.