COUNCIL OF STATE AND TERRITORIAL EPIDEMIOLOGISTS

Anomaly Detection Methods for Overdose: Technical Guidance for Analytic Code

Prepared by: Surveillance and Outbreak Response Team, Johns Hopkins Bloomberg School of Public Health

July 2026

Recommended citation

Smith, Zachary[†]; Misra, Shilpi[†]; Perez, Alexander[‡]; Martin, Augustin[‡]; Shade, Becky; Truong, Ha; Gurley, Emily S.; Marx, Melissa A., 2026, 'Data associated with: Anomaly detection methods for overdose: A customizable approach to detecting anomalies in count data', https://doi.org/10.7281/T1JMRJDI, Johns Hopkins Research Data Repository, V1.

* = Equal contribution. † = Equal contribution.

Funding support

This project was supported by the Centers for Disease Control and Prevention of the U.S. Department of Health and Human Services (HHS) as part of a financial assistance award totaling $90,000 with 100 percent funded by CDC/HHS. The contents are those of the authors and do not necessarily represent the official views of, nor an endorsement, by CDC/HHS, or the U.S. Government.

Acknowledgments

The Surveillance and Outbreak Response Team specifically thank and acknowledge

- the state, Tribal, local, and territorial epidemiologists who volunteered their time and shared their insights on overdose anomaly detection during the CSTE Anomaly Detection Methods for Overdose Advisory Committee meetings, landscape review, and pilot testing;
- Dr. Michael Desjardins, Assistant Research Professor in the Department of Epidemiology at the Johns Hopkins Bloomberg School of Public Health, for providing expert review of the statistical methods and analytic code; and
- Kahuina Consulting, LLC for its work on the CSTE Overdose Anomaly Toolkit, review of the landscape analysis, and expert facilitation of Advisory Committee meetings.

Contact information

	Surveillance and Outbreak Response Team, Johns Hopkins Bloomberg School of Public Health	Council of State and Territorial Epidemiologists
Website	https://publichealth.jhu.edu/departments/epidemiology/programs/tracks/infectious-disease-epidemiology/surveillance-and-outbreak-response-team-sort	https://www.cste.org/
Email	SORT@jh.edu	overdose@cste.org

Table of Contents

Abbreviations

ADMO	Anomaly Detection Methods for Overdose
CDC	United States Centers for Disease Control and Prevention
CSTE	Council of State and Territorial Epidemiologists
ED	Emergency Department
EMS	Emergency Medical Services
ESSENCE	Electronic Surveillance System for the Early Notification of Community-based Epidemics
ICD-10-CM	International Classification of Diseases, Tenth Revision, Clinical Modification
MAE	Mean absolute error
NSSP	National Syndromic Surveillance Program
OD	Overdose
SNOMED CT	Systematized Nomenclature of Medicine Clinical Terms
SORT	Surveillance and Outbreak Response Team
STLT	State, Tribal, Local, and Territorial
WONDER	Wide-ranging ONline Data for Epidemiologic Research

1. Background

1.1. Purpose of this technical guidance

The Johns Hopkins Bloomberg School of Public Health Surveillance and Outbreak Response Team (SORT) was contracted by the Council of State and Territorial Epidemiologists (CSTE) to develop analytic code and accompanying technical guidance for state, Tribal, local, and territorial (STLT) epidemiologists to detect anomalies in the frequency of overdose (OD) in their jurisdictions. To inform the development of the code and technical guidance for OD anomaly detection, SORT conducted a landscape review of peer-reviewed and grey literature and held discussions with STLT epidemiologists to understand existing tools, methods, and data sources for OD anomaly detection. SORT then drafted and piloted the code and technical guidance with STLT epidemiologists and analysts across a range of geographic locations, public health governance structures, and existing capacities for OD anomaly detection. An advisory group of STLT epidemiologists and analysts provided additional feedback during the development of the code and technical guidance.

The purpose of the ***Anomaly Detection Methods for Overdose (ADMO): Technical Guidance for Analytic Code*** is to expand upon the methods and concepts used in the analytic code. Specifically, this guidance describes:

- the data sources, variables, and format required to use the code;
- the statistical rationale and analytic workflow of the code;
- instructions on customizing code, which is programmed in R; and
- guidance on the interpretation of results, statistical outputs, and visualizations.

The code and guidance provide two standardized options for STLT epidemiologists and analysts to detect anomalies in counts of OD in their jurisdiction: non-spatial methods and spatial methods. The analytic code works when it is run on standardized data, and this document includes guidance on preparing a dataset for use. The primary use of the code is to identify anomalies for additional investigation or public health action; however, STLTs may also choose to use the code for other purposes.

The code accommodates statistical and non-statistical approaches to defining "expected" (i.e., "usual," "baseline," "threshold") counts to which observed counts are compared. In this technical guidance, we describe statistical approaches to defining expected counts of OD but do not discuss considerations when defining expected counts non-statistically (e.g., based on public health response capacity). For more information about applied methods for non-statistically defining expected count thresholds, see the "Prepare: Early Planning Steps" page of the CSTE OD Anomaly Toolkit.

1.2. ADMO knowledge base

What is the ADMO approach and what is it designed to detect?

The ADMO approach is a set of standardized and customizable statistical methods that detect anomalies in count data. By using it, users can identify anomalies over time and space. It is *not used* to detect clusters. This distinction is important because not all anomalies are clusters and not all clusters are anomalies. Anomaly detection seeks to identify unusual events, meaning events that deviate from a pattern. Cluster detection aims to identify groups of similar events based on characteristics like location or time.

What data are required to use this approach?

The ADMO approach works with any data that have a date, case count, cause of the count (e.g., poisoning by fentanyl, poisoning by heroin), and geographic identifier. It was designed for use with OD-specific data exported from the Electronic Surveillance System for the Early Notification of Community-based Epidemics (ESSENCE) but can be used for counts of any health outcome regardless of source.

How does the approach determine what is "expected" or "usual"?

Anomalies are identified by comparing counts from the period of interest, also called the "observed period," to what is "expected" or "usual" in the period of interest. "Expected" or "usual" can be determined with a statistical model or according to a pre-set threshold. In the statistical application, historical data are used to train a statistical model. After it is trained, the statistical model is applied to the period of interest to determine what would be "expected" or "usual." In the threshold application, "expected" or "usual" is set as a constant number defined by the user. The threshold can be based on a jurisdiction's response capacity (e.g., how many staff can investigate anomalies), an established response plan threshold, or any other rationale from the health department.

How does the approach select the best statistical model?

The statistical model used to define "expected" or "usual" can be either zero-inflated or standard, and either a Poisson or negative binomial. The analytic code automatically determines which model best fits the data based on the proportion of zeros in the historical data and the identified model's dispersion statistic.

When should I use the non-spatial versus spatial approach?

Both approaches assume that counts vary over time. The spatial approach also assumes counts vary over space. The non-spatial approach may be more appropriate if counts are thought to vary only over time, whereas the spatial approach may be more appropriate if counts are thought to vary over space and time.

How can I customize the approach?

The analytic code includes various customization points. The customization points allow users to adapt the approach based on their expertise. The technical guidance details considerations to make when customizing the code.

How are anomaly flags determined?

Anomalies are identified by comparing counts in the observed period to expected counts during the observed period. Specifically, anomaly flags are generated based on whether the observed count differs from the expected count.

How should I interpret an anomaly flag?

Non-spatial and spatial anomaly flags can be interpreted as follows: *The actual count of [insert outcome] on [insert temporal unit] in [insert spatial unit] was flagged as a level [insert number 0-7 depending on anomaly flag categories] anomaly, on a scale of 0 (no anomaly) to 7 (most strict anomaly). This means the actual count was likely [different (level 1-7) / not different (level 0)] from expected.*

The following text fills in the blanks of the above interpretation: *The actual count of opioid ODs on January 1, 2023 in Anomaly County was flagged as a level 6 anomaly, on a scale of 0 (no anomaly) to 7 (most strict anomaly). This means the actual count was likely different from expected.*

What output does the ADMO approach produce?

The non-spatial and spatial approaches output formatted tables and time-series plots of anomaly flags. The non-spatial approach also generates stratum-specific plots in stratified analyses. The spatial approach also creates choropleth maps of anomalies.

What are the main strengths and limitations of the ADMO approach?

The main strengths of the ADMO approach are: (1) it is a standardized, customizable way to detect anomalies in count data from various sources; (2) it can detect temporal and spatiotemporal anomalies; (3) it explicitly and automatically accounts for excessive zeros; (4) the modeling approach can be adjusted based on computational capacity; and (5) it produces interpretable, reproducible outputs for operational use.

The main limitations of the ADMO approach are: (1) expected counts estimated statistically depend on the quality of data from the historical period; (2) anomalies are sensitive to appropriate selection of customization options; and (3) spatial analyses must be done on areal data, potentially introducing error associated with the modifiable areal unit problem.

2. Data sources and preparation

2.1. Data sources

STLT health departments use a wide range of data sources for OD anomaly detection. These data sources include national (e.g., CDC's State Unintentional Drug Overdose Reporting System, National Center for Health Statistics Vital Statistics), state-specific (e.g., ESSENCE, emergency medical services [EMS] data, medical examiner and coroner records for fatal ODs), and other data sources (e.g., toxicology, laboratory, prescription, treatment, law enforcement data).

The analytic code and technical guidance use ESSENCE data as an example. However, the analytic code and technical guidance are compatible with any count data, which is a common method of collecting and storing OD data. When choosing a source of OD data, STLT epidemiologists and analysts should consider verifying OD data before additional investigation and escalating a public health response.[1]

2.2. Data format and variables

2.2.1. Minimum variable requirements

The analytic code can be applied to any count data with the minimum variable requirements from the chosen data source (Table 1).

Table 1. Minimum variable requirements from ESSENCE or chosen data source

Name	Date	Case count*	Cause	Geographic identifier†
Format	Date (YYYY-MM-DD)	Numeric	Character	Character

* = Case count should either be a variable in the data or a variable that can be derived from the data. For example, line-list data, where each row represents one case, can be aggregated to get counts.
† = Geographic identifiers include patient ZIP code or county of residence or ZIP code or county of medical facility. The mock dataset includes information for patient and facility ZIP code and county. Users can specify all these, none of these, or other geographic identifiers depending on the chosen data source.

The analytic code transforms data containing the minimum variable requirements into a dataset containing eight variables required for analysis (Table 2). If data are available, we suggest including additional variables in the dataset to enable subgroup or stratified analyses (Table 2). Note, variable names, formats, and parameterizations are suggestions. Users can adapt the variable names, formats, and parametrizations to their preferences. However, such adaptations will require users to edit the code accordingly.

Table 2. Required and additional suggested variables for analysis

Variable name	Format	Parameterization	ESSENCE variable name
Required variables for analysis			
event_date	Date (%m/%d/%y)	Date of admission to emergency department	Date
od_cause*	Categorical	ICD CM codes	ICD_Diagnosis_Flat
od_flag*	Binary	0 = not OD event 1 = OD event	No corresponding variable
ZIP_patient	Character	Patient ZIP code of residence	Patient_ZIP
county_patient	Character	Patient county or county-equivalent of residence	C_Patient_County
ZIP_fac	Character	Facility ZIP code	HospitalZIP
county_fac	Character	Facility county or county-equivalent	No corresponding variable
case_count†	Numeric	Number of cases per stratification	No corresponding variable
Additional suggested variables for analysis‡			
age_grp	Character	"[15-24]" "[25-34]" "[35-44]" "[45-54]" "[55-64]" "[65+]"	Age
sex	Character	"F" = female "M" = male "U" = unknown "N" = not reported	Sex
race_grp	Character	User-defined racial identity groups	C_Race
eth_grp	Character	User-defined ethnicity groups	C_Ethnicity

* = The mock dataset uses CDC Wide-ranging Online Data for Epidemiologic Research (WONDER) cause-of-death codes. These codes do not appear in ESSENCE data, which uses International Classification of Diseases, Tenth Revision, Clinical Modification (ICD-10-CM) and Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) codes, among others.[2] Users can choose their definitions of interest for od_cause and od_flag depending on their data source. Consider identifying suspected nonfatal OD-related emergency department (ED) visits using the OD-related syndromes available on the National Syndromic Surveillance Program (NSSP) Knowledge Repository.[3]
† = Case count should either be a variable in the data or a variable that can be derived from the data. For example, line-list data, where each row represents one case, can be aggregated to get counts.
‡ = Additional variables beyond those suggested in this table can be incorporated in the analysis. Incorporating variables not include in this table will require modification of the code.

2.2.2. Converting data into "long" format

For use in this code, data must be a CSV or Excel file in "long" format (Table 3), where multiple rows exist per day, and each row corresponds to a specific stratification (e.g., sex, geography) and its associated count. The analytic code includes suggestions for transforming and aggregating individual-level data, where a row corresponds to one patient's observation, into the required long format. Specifically, the code includes a worked example showing how to transform a mock individual-level line-list into the long format (Table 9).

The mock dataset accompanying the analytic code and technical guidance was adapted from CDC WONDER data and modified to simulate small OD counts because the CDC WONDER platform does not publish statistics representing nine or fewer cases. See Appendix A for how the mock dataset was adapted from CDC WONDER data.

Table 3. Example "long" data format

Date	od_flag	age_grp	case_count
2026-01-12	0	[25-34]	7
2026-04-04	0	[35-44]	200
2026-05-30	1	[25-34]	1
2026-08-02	1	[35-44]	5

2.3. Necessary software to complete analysis

The analytic code and technical guidance were designed for use with R. R version 4.5.3 was used to develop the code and technical guidance. Different versions of R may result in impediments to using the analytic code. If encountering errors with older or newer versions, consider using version 4.5.3.

The analytic code installs various R packages that are used at different stages of the analysis to detect non-spatial and spatial anomalies of OD in count data (Table 4).

Table 4. Necessary R packages to complete analysis

Purpose	Packages	R markdown used in
Data import, cleaning, and wrangling	tidyverse	Data.Formatting, Data.Analyses
	lubridate	Data.Formatting, Data.Analyses
	sf	Data.Analyses
	tigris	Data.Analyses
	ZIPcodeR	Data.Formatting
	readxl	Data.Formatting
Figures and tables	viridis	Data.Analyses
	RColorBrewer	Data.Analyses
	tmap	Data.Analyses
	Knitr	Data.Analyses
	classInt	Data.Analyses
Statistical analysis	pscl	Data.Analyses
	MASS	Data.Analyses
	spdep	Data.Analyses
	gstat	Data.Analyses
	fixest	Data.Analyses
	glmmTMB	Data.Analyses

2.3.1. Adapting the code for other software and programming languages

Although the analytic code and technical guidance were designed for use with R, the analytic code can be adapted for other software languages (e.g., SAS, Python).

We recommend a stepwise approach using artificial intelligence to translate the code into other software and programming languages (**Table 5**). The process should be completed without entering any real data into the artificial intelligence interface. Only the R code and example prompts need to be entered into the artificial intelligence tool. Make sure to check the code provided by artificial intelligence to correct any mistakes that artificial intelligence makes. The stepwise approach in **Table 5** and code checks may help rectify logic mismatches between R and the other programming language.

Table 5. Stepwise approach to using artificial intelligence to translate R code to other programming languages

Step	Action
1	Identify an artificial intelligence tool to translate the code. Before using the artificial intelligence tool, ensure it is approved for use per guidance from the Information Technology team at your workplace.
2	Initiate the chat. Below is an example prompt: I would like to translate R code into [*insert name of programming language*]. I would like to proceed in a stepwise manner to translate the code. I will provide sections of the R code one at a time for translation. First, I will provide the list of R packages.
3	Provide the list of R packages from the analytic code. No specific prompt required: copy the list of R packages and paste into the artificial intelligence tool.
4	From the analytic code, copy and paste into the artificial intelligence tool: (1) the preamble to section 0 and the code specifying (2) the required packages (**Chunk 2**) and (3) the customization points (**Chunk 3**). Below is an example prompt: The purpose of this section is: "[*insert preamble directly beneath section header*]". The code for this section is immediately below. [*insert code chunks*]
5	Copy the output from the artificial intelligence tool and paste into a separate document.
6	Repeat steps 4-5 for the next code chunk.
7	Check translation of previous sections, based on the most recent section. Below is an example prompt: Based on the code chunk just translated, update the preceding chunks to ensure the code runs more smoothly.
8	Repeat steps 4, 5, and 7 for the next code chunk.
9	When all code chunks are translated, test the translated code using the mock data. Compare the visualizations in Section ***4.3.2. Interpretation of key model diagnostics and stakeholder visualizations*** with the equivalent visualizations from the translated code. If discrepancies exist, provide the artificial intelligence tool with the R and translated versions of the visualization to remedy. Below is an example prompt: The plot called [*insert name of plot*] from the translated code looked different than the corresponding R version. Both plots are attached. Both plots were created using the same data. Identify and remedy the source of the discrepancy.

3. Statistical analysis

This section details the statistical methods unpinning the ADMO approach. We recommend reviewing this section to understand how the approach works.

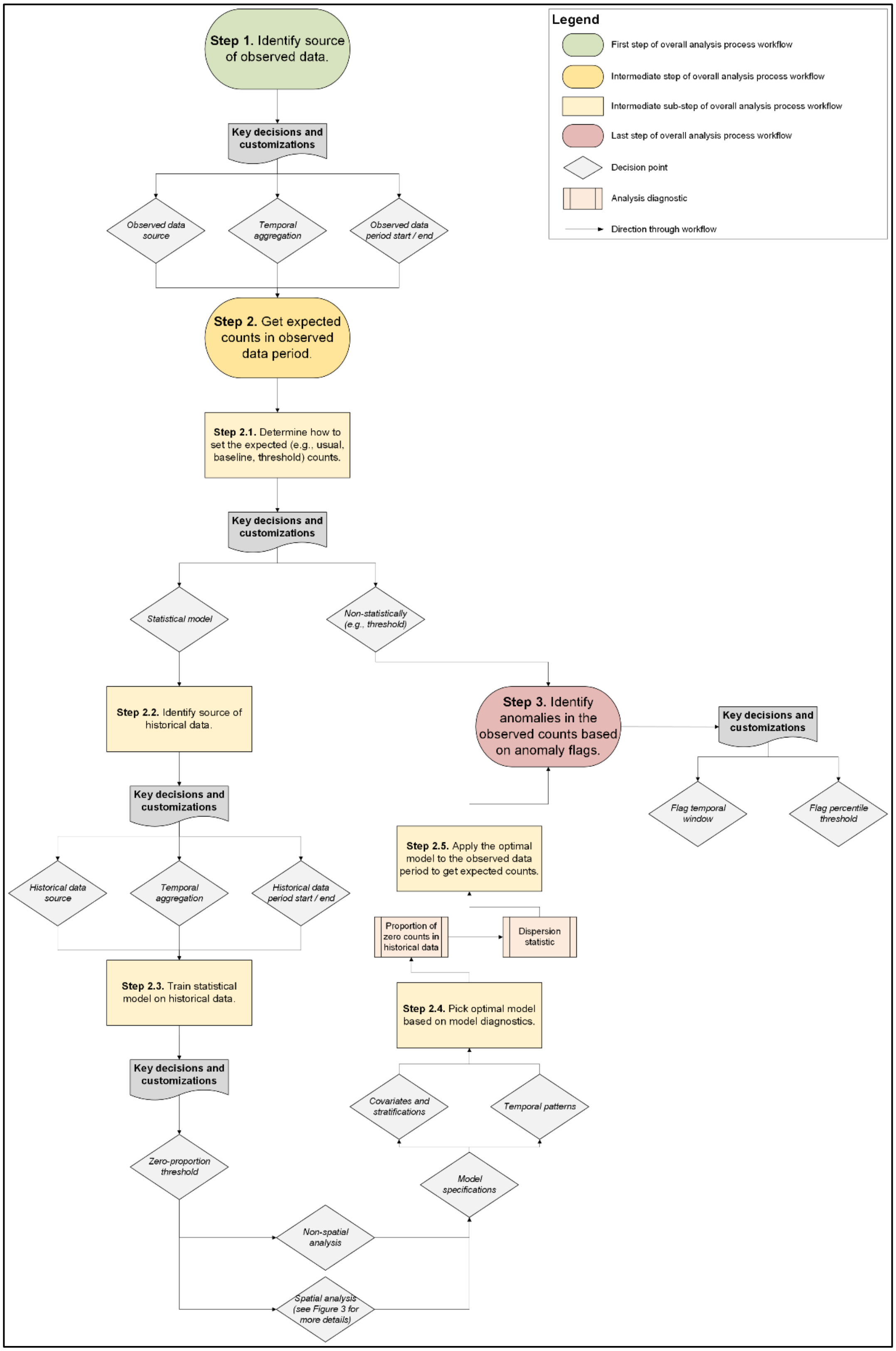
3.1. Analysis process workflow

ESSENCE provides deidentified individual-level data from EDs across jurisdictions and can aggregate individual-level data into counts by different variable stratifications. ESSENCE and its associated R package (*Rnssp)* allow users to apply baseline-comparison algorithms, including regression-based or Serfling-type methods, to compare expected and observed counts and flag anomalies.[4]

Expanding the existing methods applied to ESSENCE data, the ADMO approach uses Poisson or negative binomial models to detect anomalies in counts of OD, including using spatiotemporal methods (Figure 1). Review Section ***3.1.2.3. Step 2.3: Train statistical model on historical data*** to decide whether a spatial analysis may be appropriate to accompany a non-spatial analysis. The specific regression model used by the ADMO approach in a given analysis depends on the dispersion and proportion of zero counts in the historical data.

The analytic code and technical guidance focus on anomaly detection, not cluster detection. Anomaly detection seeks to identify unusual events, or events that deviate from a pattern. Cluster detection aims to identify groups of similar events based on characteristics like location or time. This distinction is important because not all anomalies are clusters and not all clusters are anomalies. Both are commonly used methods to analyze spatiotemporal patterns in public health surveillance data. Section ***3.2.1. Spatial analyses*** includes more information about the differences between anomaly and cluster detection with spatial data.

Figure 1. Analysis process workflow



3.1.1. Step 1: Identify source of observed data

STLT health departments should identify a source of data for the observed counts of OD. Observed counts are the OD events being evaluated for anomalies. As previously noted, the analytic code and technical guidance were designed for use with data exported from ESSENCE. However, the methods apply to any count data meeting the minimum variable requirements (**Table 1**).

3.1.2. Step 2: Get expected counts in the observed period

The purpose of step 2 is to set the expected counts in the observed period to which observed counts in the observed period are compared. Step 2 includes five sequential sub-steps (**Figure 1**), which are detailed below.

3.1.2.1. Step 2.1: Determine how to set the expected counts

As previously noted, the code accommodates statistical and non-statistical approaches to defining expected counts in the observed period to which observed counts are compared. STLT health departments must decide whether to define the expected counts statistically using a statistical model trained on historical data and applied to the observed period or non-statistically using a constant threshold that is set by the STLT health department. Reasons for this choice will vary across jurisdictions. Some STLT health departments will define the constants based on the number of ODs required to activate their OD response plan or other capacity-based thresholds. See the "Prepare: Early Planning Steps" page of the CSTE OD Anomaly Toolkit for more considerations when picking constant, non-statistical thresholds. The approach used to define the expected counts will impact the frequency of anomaly flags (**Appendix B**).

If defining counts statistically, proceed to Sections ***3.1.2.2. Step 2.2: Identify source of historical data***, ***3.1.2.3. Step 2.3: Train statistical model on historical data***, ***3.1.2.4. Step 2.4: Pick optimal model based on model diagnostics***, and ***3.1.2.5. Step 2.5: Apply the optimal model to the observed data period to get expected counts***. If defining expected counts non-statistically, users will pick a constant to represent the expected counts and then skip to Section ***3.1.3. Step 3: Identify anomalies in the observed counts based on anomaly flags***.

3.1.2.2. Step 2.2: Identify source of historical data

Users should identify historical data before proceeding with subsequent steps in the analytic process (**Figure 1**). Considerations for identifying historical data include the data source and amount of available data, spatiotemporal resolution, and available covariates. The data should reflect a period the STLT health department feels represents “usual” or “standard” counts for comparison to observed counts. For example, if State A is analyzing fentanyl-specific nonfatal OD, and fentanyl was not found in the drug supply in State A until 2018, it may be preferable to estimate the expected counts using historical data starting from 2018. The historical data should comprise counts from at least two temporal periods (e.g., two daily counts, two weekly counts) in order to develop the model used to estimate the expected counts in the observed period. If historical data are sparse, users may have difficulty with model convergence.

Further consideration should be directed to OD surges in the baseline period. If the baseline period includes known OD surges (e.g., fentanyl waves), the statistical model will treat those events as part of the expected pattern. Including known surges in the historical period used to estimate the expected counts could reduce sensitivity to similar future anomalies. Users should carefully assess whether known surges are appropriate to include or should be excluded from generating the statistical model used to estimate the expected counts. Incorporating such temporal discontinuities for OD surges or other reasons may impact the model used to estimate expected counts.

If attempting to identify anomalies associated with new substances, consider selecting a baseline period before the emergence of the new substance. Any observed counts during the period when the new substance is detected would represent deviations from the period before introduction of the new substance. STLT health departments could identify periods prior to the emergence of the new substance by reviewing toxicological data or communicating with community partners, among other methods.

3.1.2.3. Step 2.3: Train statistical model on historical data

Historical data identified in Section ***3.1.2.2. Step 2.2: Identify source of historical data*** are now used to train a statistical model to estimate expected counts in the historical period. The statistical model trained in this step will be applied to the observed period in Section ***3.1.2.5. Step 2.5: Apply the optimal model to the observed data period to get expected counts*** to get the expected counts in the observed period. Training the statistical model on the historical data comprises several steps (**Figure 1**).

Users first specify a zero-proportion threshold. The zero-proportion threshold is the proportion of zero counts in the highest level of stratification in the historical data. It is used to determine whether zero-inflated or standard regression models are optimal to train the statistical model. The default zero-proportion threshold is 25 percent.[5] Changing the zero-proportion threshold from the default could impact the chance the optimal statistical model will be standard or zero-inflated. Inappropriately selecting a standard model when a zero-inflated model is warranted, or vice versa, may result in model misspecification, potentially leading to bias in the expected counts.[6] See Section ***3.1.2.4. Step 2.4: Pick optimal model based on model diagnostics*** for more information about how the analytic code automatically uses the zero-proportion threshold to select the optimal statistical model to fit on the historical data.

Next, users select whether a non-spatial or spatial analysis is preferred. The hypothesized source of variation in counts and anomalies informs which approaches to consider for a given dataset. Non-spatial analyses assume counts and anomalies only vary over time. Spatial analyses assume counts and anomalies vary over time and space. If a spatial analysis is desired, review Section ***3.2.1. Spatial analyses*** for information about the spatial methodology.

Finally, users select their preferred specification for the statistical model trained on the historical data. Model specification refers to the covariates included in the functional form of the regression model. The analytic code includes various model specification options (**Table 8**). Generally, the appropriate model takes the functional form:

$$\log(\mu_i) = \beta_0 + \beta_1 * event\ date_i + \beta_2 * temporality_i + \beta_3 * covariates_i + \beta_4 * (covariates_i * event\ date_i) + \varepsilon_i$$

where $log(\mu_i)$ is the natural logarithm of the expected count for observation i, β_0 is the intercept, β_1 captures the expected change in log counts for each day of the year, and ϵ_i is the error term. The model may include terms β_2, a vector of temporal terms; β_3, a vector of covariates that may impact the expected counts; and β_4, an interaction term between event date and covariates of interest, which enables stratification of the results. For spatial analyses, β_3 includes spatial lag terms. If stratifying models by covariates (e.g., **Table 2**), those covariates should be included as main effects in β_3, which assumes the covariates have different baseline effects but the same temporal patterns. Only include β_4 in non-spatial models if it is hypothesized that event-date effects differ by covariate. See the last paragraph of this section for considerations when stratifying in non-spatial analyses and approaches to stratifying spatial analyses.

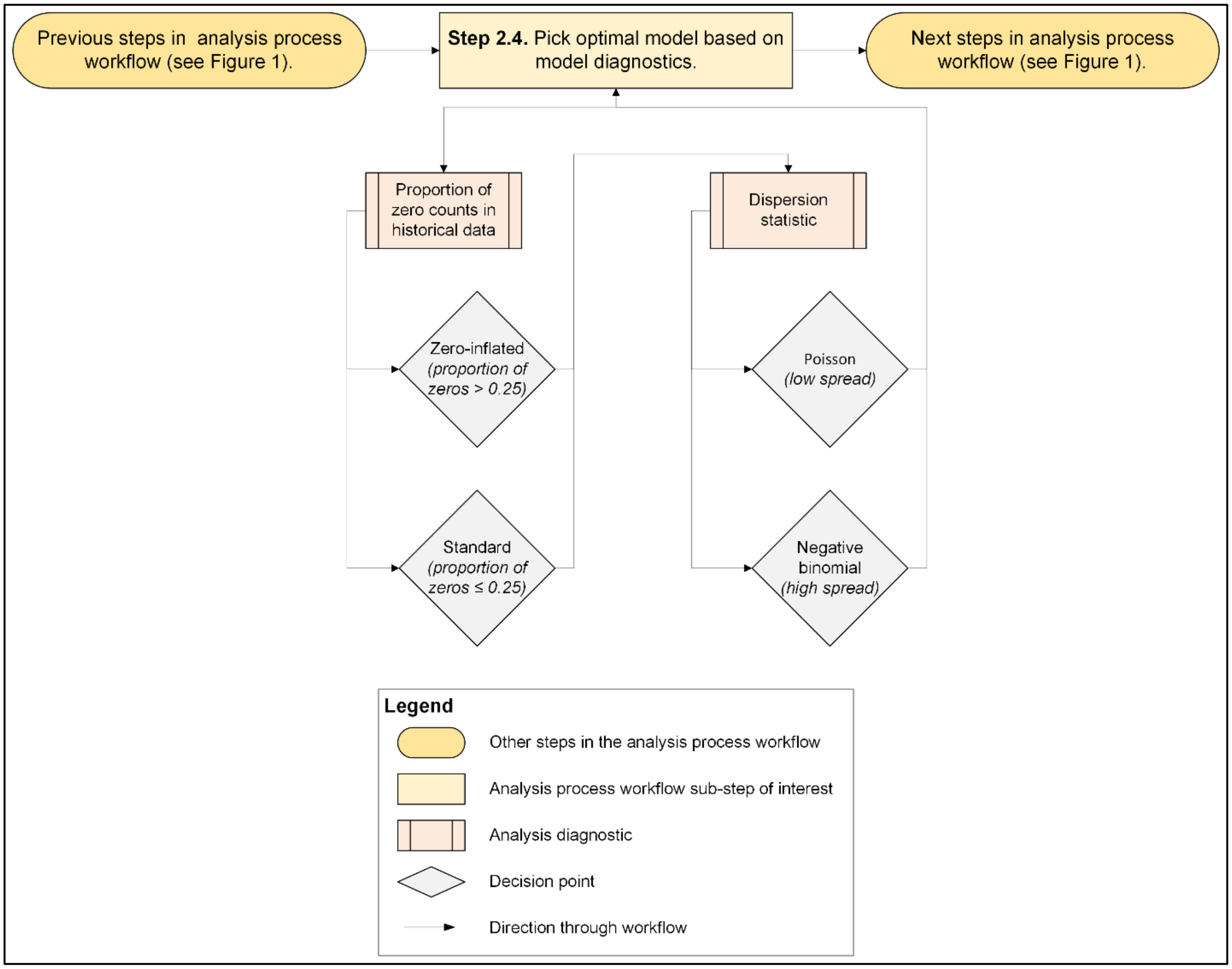
The vector of temporal terms, β_2, may include different parameterizations of time. Potential parameterizations of time included in the analytic code include Fourier transformations to account for seasonality (include only when at least one year of historical data are available), point or moving average temporal lags (e.g., count one day before today, average count of two days before today), and nominal categorical variables to account for other temporal effects (e.g., day-of-week, month-of-year, holidays). Temporal effects to include in the model depend on the overall aggregation of the counts. For example, if data are aggregated weekly, a week-of-year term may be included, but a day-of-week term cannot.

Increasing the number of covariates used to stratify the data leads to smaller cell counts of other variables and may adversely impact the uncertainty in model estimates and model convergence. If conducting a spatial analysis, stratifying results by covariates (e.g., age group) could provide insight into factors driving spillover between geographic units. To stratify the spatial analysis by covariates, run separate models per level of the stratification variable or add interaction terms between the stratification variable and all other terms in the model, including temporal and spatial lags. Both approaches allow relationships, including temporal and spatial autocorrelation, to vary by stratum. Although the approaches are equivalent, the dispersion parameter may vary between the approaches because error variance is pooled when using interaction terms and distinct in each stratum when running stratified models. The ultimate impact on the dispersion parameter depends on the sample size and underlying dispersion within each stratum. Running stratified models would be the simplest approach to implement in the analytic code because it does not require alterations to the formula specifications. Critically, the potential for strata to have small sample sizes means stratifying spatial analyses using either approach could adversely impact model performance.

3.1.2.4. Step 2.4: Pick optimal model based on model diagnostics

The analytic code automatically decides which model best fits the historical data based on the proportion of zeros in the highest level of stratification in the historical data and the dispersion statistic (**Figure 2**). If the proportion of zeros is large (>25%),[5] counts are modeled using zero-inflated Poisson and zero-inflated negative binomial model; if not (≤25%), standard Poisson and negative binomial models are used. In each case, the model with a dispersion statistic closest to one is selected as the optimal model.

Figure 2. Automatic selection of the optimal statistical model fit on the historical data



Zero-inflated models are preferred to non-zero-inflated models when the data contain many zeros that cannot be described by standard distributions.[6,7] We recommend zero-inflated (i.e., mixture) models over zero-altered (i.e., hurdle) models because they decompose zeros into “structural” (zero-mass) and “sampling” (count process) components. Structural zeros are assumed to occur due to the disease-generating process over space and time (i.e., sometimes there are truly zero counts), whereas sampling zeros occur by chance (i.e., sometimes zero counts randomly occur). In comparison, zero-altered models have a more rigid assumption that no zeros occur by chance.[6,7] Both Poisson and negative binomial distributions should be evaluated for zero-inflated data, as zero inflation can contribute to overdispersion; however, accounting for zero inflation alone may not eliminate overdispersion.[7]

As noted above, only the proportion of zeros and dispersion statistic are used to automatically select the optimal model for the historical data. Users may consider other diagnostics to contextualize model performance, like the mean absolute error (MAE), which the analytic code calculates for the model that best fits the historical data. The MAE is the average of the absolute differences between predicted and observed values. It indicates, on average, how far off the estimated counts are in the historical data. The MAE ranges from zero to infinity, with values nearer to zero indicating better model fit. For example, a MAE of 2.2 means that, on average, the model's estimated counts in the historical period were off by 2.2 counts. The MAE should be interpreted in the context of the average observed counts per temporal or spatiotemporal unit in the historical period. For example, a MAE of 2.2 when the average daily observed count was 10 is more concerning than when the average daily observed count was 50. Two potential options to address a concerning MAE include (1) lowering the temporal window and percentile threshold of the anomaly flags—which compensates for, but does not fix, poor model fit by increasing the sensitivity of the anomaly flags—and (2) adding covariates to the optimal statistical model, which may improve model fit. We do not use the MAE, or other model fit diagnostics, to automatically select the optimal model because a "good" diagnostic value depends on the specific analysis. Further, the analytic code estimates the MAE in lieu of other model diagnostics, like the root mean squared error, because MAE is robust to outliers, aligning the ADMO approach's goal of identifying anomalies. Users can modify the model training function in the analytical code if they wish to estimate other model diagnostics when fitting the optimal model to the historical data.

3.1.2.5. Step 2.5: Apply the optimal model to the observed data period to get expected counts

The statistical model that best fit the historical data, considering the specifications defined in Sections ***3.1.2.2. Step 2: Identify source of historical data***, ***3.1.2.3. Step 3: Train statistical model on historical data***, and ***3.1.2.4. Step 4: Pick optimal model based on model diagnostics***, is applied to the observed data to get the expected counts in the observed period. The expected counts generated after fitting the optimal model on the data in the observed period represent the "usual" counts we expect at a given temporal (non-spatial analysis) or spatiotemporal (spatial analysis) unit in the observed period.

3.1.3. Step 3: Identify anomalies in the observed counts based on anomaly flags

Expected counts in the observed period, defined either statistically or non-statistically (e.g., capacity-based thresholds), can be compared to the observed counts in the observed period to determine whether an anomaly has occurred. In the ADMO approach, anomaly flags are based on whether the observed count over X consecutive days in the observed period exceeds the Y^{th} percentile of the expected counts in the observed period.

Both the temporal window (X) and percentile threshold (Y) are customizable. Stricter criteria, defined by increasing the temporal windows and setting the percentile cutoffs closer to 0.00 or 1.00, will result in fewer identified anomalies (i.e., decreased sensitivity and increased specificity) than less strict criteria. Conversely, shorter temporal windows and percentile cutoffs further from 0.00 and 1.00 will identify more anomalies.

Default anomaly flags definitions, increasing in strictness, are provided in **Table 6**. For example, a level 7 flag identifies an anomaly count when the smaller observed count across two consecutive days is greater than or equal to the 99.5^{th} percentile of the expected count. Consecutive days are identified using temporal lag0 and lag1. For example, when assessing whether Y on April 30, 2026 is anomalous, flag levels 1-3 look only at Y on April 30 (lag0), and flag levels 4-7 look at Y on April 29 (lag1) and April 30 (lag0). Similarly, to determine whether Y on May 1, 2026 is anomalous, flags levels 1-3 look at Y on May 1 (lag0), and flag levels 4-7 look at Y on April 30 (lag1) and May 1 (lag0). It is important to note that these default thresholds are intentionally strict and may yield few anomaly flags. STLT epidemiologists and analysts should investigate the identified anomalies in accordance with their standard operating procedures. The temporal window and percentile threshold may be changed based on STLT health department capacity to investigate, tolerance for missing real anomalies, and other practical factors.

The analytic code uses the anomaly flags in **Table 6** for the non-spatial analysis. The spatial analysis also uses the anomaly flags from **Table 6** but replaces the non-strict inequalities with strict inequalities (i.e., > replaced ≥; < replaced ≤). This adjustment was made to avoid incorrectly flagging observed zero counts as upper or higher-than-expected anomalies. In other words, when the expected count was near zero, observing zero cases was statistically expected, which caused the zero count to be flagged as an anomaly. When the expected counts are very close to zero in the non-spatial or spatial analyses, consider (1) replacing non-strict inequalities with strict inequalities (default for the spatial analysis), (2) increasing the percentile thresholds to reduce sensitivity, (3) using a non-statistical threshold (i.e., constant) to define the expected counts, and (4) aggregating to a higher temporal or spatial scale (e.g., daily to weekly, ZIP code to county).

Table 6. Default anomaly flags for daily count data

Strictness	Flag level (hex code)	Temporal window*	Percentile threshold based expected counts	Flag side?
NA	Level 0 (#252525)	1 or 2	Y does not meet percentile range for any flag	NA
Least	Level 1 (#fee5d9)	1	$0.95 < Y \leq 0.99$	Upper
			$Y \leq 0.05$	Lower
↓	Level 2 (#fcbba1)	1	$0.99 < Y < 0.995$	Upper
↓	Level 3 (#fc9272)	1	$Y \geq 0.995$	Upper
↓	Level 4 (#fb6a4a)	2	$0.95 < Y_1, Y_2 \leq 0.99$	Upper
			$\max(Y_1, Y_2) \leq 0.05$	Lower
↓	Level 5 (#ef3b2c)	2	$0.95 < \min(Y_1, Y_2) \leq 0.99$, $0.99 < \max(Y_1, Y_2) \leq 0.995$	Upper
↓	Level 6 (#cb181d)	2	$\min(Y_1, Y_2) \geq 0.99$, $\max(Y_1, Y_2) < 0.995$	Upper
Most	Level 7 (#99000d)	2	$\min(Y_1, Y_2) \geq 0.995$	Upper

Abbreviations: NA = not applicable; Y = observed count on day X.
* = The unit of the temporal window is informed by the temporal aggregation of the counts. For example, if analyzing daily count data, the unit of the temporality column is days. If analyzing weekly count data, the unit of the temporality column is weeks.

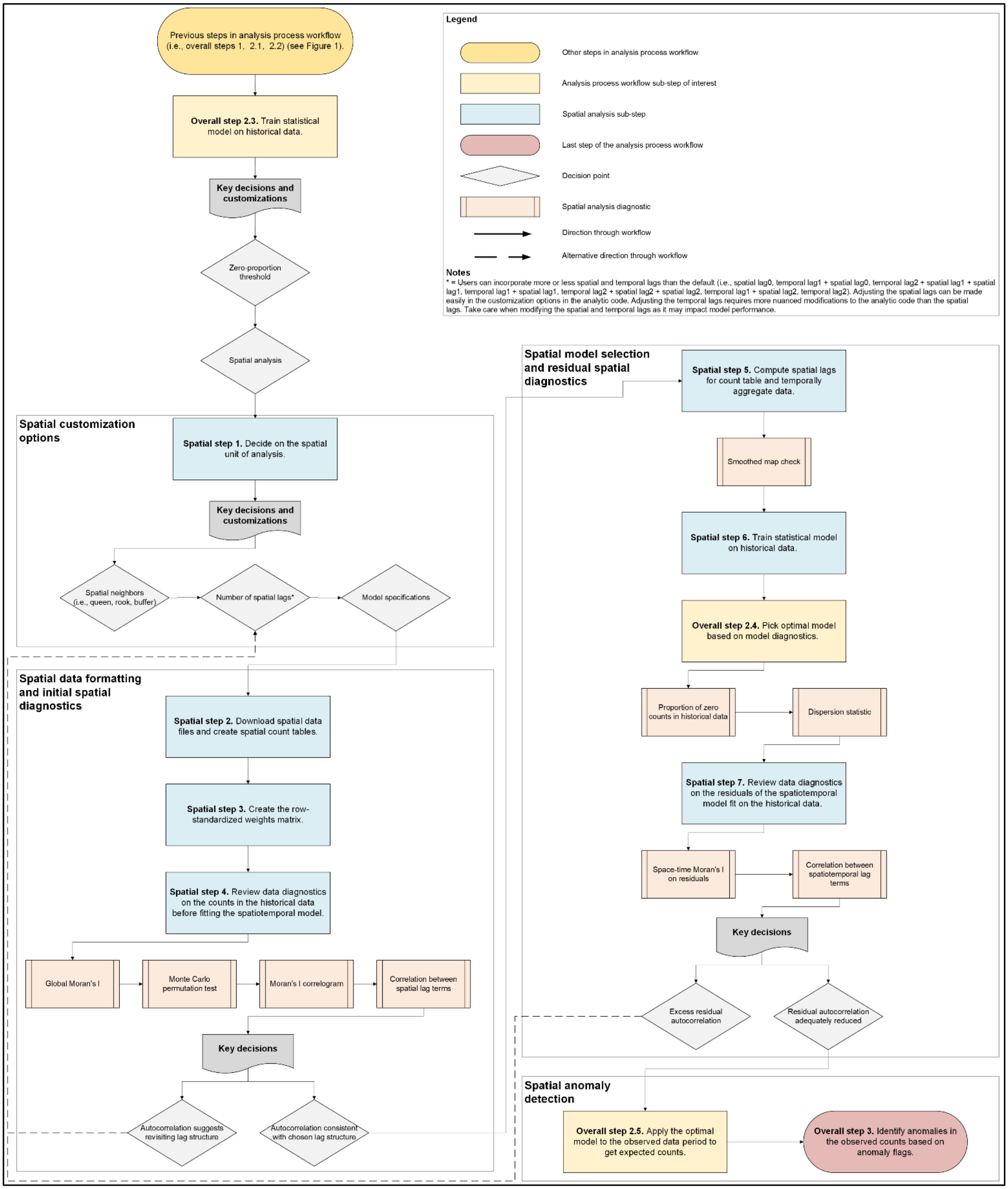
3.2. Special considerations

3.2.1. Spatial analyses

Spatial analyses may be considered if counts and anomalies of OD are hypothesized to vary over space and time. Spatial analyses follow the general flow of the overall process workflow (Figure 1). Spatial-specific customizations and activities are detailed below and outlined in the spatial process workflow (Figure 3).

The spatial analysis in the analytic code uses spatial lag models to identify spatiotemporal anomalies. Spatial lag models explicitly account for spatial dependence, also known as spatial autocorrelation, in the outcome (e.g., counts of OD) by incorporating spatially lagged terms as predictors. These spatial lag terms codify the assumption that the count in one spatial unit influences the count in neighboring spatial units (i.e., spillover or diffusion). In contrast, spatial error models codify spatial dependence in the error structure and assume that this dependence is driven by unobserved factors rather than direct spillover effects. Both approaches are appropriate to address spatial autocorrelation; however, spatial lag models are used here because of the hypothesized spillover of OD counts across neighboring spatial units and processes that drive them (e.g., drug distribution networks).

Figure 3. Spatial analysis process workflow



In this spatial analysis, users must first determine the spatial scale they want to use for the analysis. The analytic code supports spatial analyses at the ZIP code and county levels. With modifications, it is possible to conduct the analyses at different spatial scales (e.g., Census tracts, Combined Statistical Areas). The choice of spatial scale should be informed by the analytic question and purpose of the analysis, data availability, and jurisdiction-specific technical guidelines (e.g., public reporting requirements).

Next, users must select the preferred neighbor definition, which determines how spatial lags are estimated. Neighbors can be defined based on adjacency (i.e., contiguity) or buffers (i.e., distance). Adjacency neighbors are spatial units that directly touch the area of interest and are defined as either queen (i.e., units sharing at least one boundary point) or rook (i.e., units sharing at least one common boundary edge). Buffer neighbors are spatial units within a specified distance of the area of interest; this distance should realistically reflect the underlying spatial factors associated with the outcome of interest. The choice of neighbor definition should be informed by the underlying spatial process of the distribution of counts, local context about that spatial process, and the spatial scale of the analysis (e.g., ZIP codes, counties).

Once neighbors are defined, a row-standardized spatial weights matrix is estimated on the historical counts. Row-standardizing the weights matrix enables comparison across spatial units, regardless of the number of neighbors, which is particularly beneficial in analyses involving many spatial units. Global Moran's I is then calculated to assess spatial autocorrelation in the historical counts, and a Moran's I correlogram is constructed to examine spatial dependence across spatial lags. Based on these diagnostics, the weights matrix may be refined.

After selecting the preferred weights matrix, historical counts are aggregated across the weights matrix and used to train a statistical model to estimate counts in the historical data (see Sections ***3.1.2.3. Step 2.3: Train statistical model on historical data***, ***3.1.2.4. Step 2.4: Pick optimal model based on model diagnostics***, ***3.1.2.5. Step 2.5: Apply the optimal model to the observed data period to get expected counts***). As noted in Section ***3.1.2.4. Step 2.4: Pick optimal model based on model diagnostics***, the optimal model has a dispersion statistic closer to one than the other candidate model.

The ADMO spatial anomaly detection approach differs from spatial cluster detection methods, such as spatial-only or spatiotemporal analyses using SaTScan (**Table 7**). Although both approaches use clustering statistics, they are designed to answer different questions. Essentially, spatial anomaly detection is confirmatory, assessing whether the count in a given spatial unit deviates from its expected value. In contrast, spatial cluster detection (e.g., SaTScan) is exploratory, aiming to identify spatially contiguous groups of spatial units with similar counts relative to the surrounding area.

Table 7. Comparing spatial anomaly detection and spatial cluster detection

Characteristic	Spatial anomaly detection	Spatial cluster detection
Question answered	In my jurisdiction, is the case count in spatial unit X higher or lower than expected?	Where in my jurisdiction are there spatial units with similar case counts?
Analytic goal	*Confirmatory*, because, within the jurisdiction, the spatial unit of interest is specified before conducting the analysis.	*Exploratory*, because the geographic location of clusters with the jurisdiction is unknown prior to conducting the analysis.
How it works	Identifies areas of dissimilarity from the expectation using statistical measures of clustering to inform what the expected count in each spatial unit should be, then compares what was observed to what was expected.	Focuses on identifying collections of adjacent spatial units whose case counts are more similar to each other than to spatial units further away (i.e., clusters) – that is, it identifies groups of spatial units that form unusual patterns.

3.2.2. Underestimation of overdoses

Event reporting to surveillance systems varies across STLT health departments and may contribute to underestimation of counts. Underestimation of counts, or the combination of under-ascertainment of counts at the community level and underreporting of counts at the healthcare level,[8] can misrepresent the true public health burden across space and time. When underestimation is a concern, a lower percentile threshold for anomaly flags may be warranted to increase sensitivity.

3.2.3. Population offset and modeling rates

The modeling approach does not adjust for population size by default, as the population is assumed to remain approximately stable over the study period. Under this assumption, changes in population size are considered negligible and unlikely to materially affect the expected counts. If substantial population changes occur (e.g., substantial in- or out-migration), the study period may be shortened to focus on a timeframe in which the population remains relatively stable, thereby improving the reliability of expected counts.

If this is not feasible (i.e., there is a substantial change to the population during the time period of interest), an alternative is to incorporate a population offset into the model. The *tidycensus* package in R can be used to obtain population estimates. Including a population offset would model rates rather than counts, accounting for differences in population size enabling more appropriate comparison of OD rates across spatial units.

Modeling rates is recommended when STLT health departments aim to directly compare the magnitude of anomalies across spatial units. However, incorporating a population offset may not be appropriate if (1) separate models are fit for each spatial unit and analyzed independently or (2) models aggregate counts across multiple spatial units.

3.2.4. Temporal aggregation

We anticipate that STLT health departments will aggregate counts to their preferred temporal unit. When selecting the temporal aggregation, consider how the analyses will inform decision making, data suppression and reporting requirements, and the precision of expected counts. Note, more granular temporal units may yield more precise expected counts. SORT developed a heuristic to guide selection of temporal and spatial aggregations levels (**Appendix B**), as well as adjustments to anomaly flag parameters (**Table 6**).

4. Step-by-step instructions for using the code

4.1. Mock data disclaimer and details

The mock dataset accompanying the code and technical guidance was adapted from CDC WONDER data and modified to include simulated small OD counts, because CDC WONDER does not publish statistics representing nine or fewer cases. **These data are provided for instructional purposes only and should not be used for analysis.** For additional details on how the mock dataset was adapted from CDC WONDER data to look like data from ESSENCE, see **Appendix A**.

4.2. Detailed instructions for using the analytic code

The analytic code comprises two R Markdown files organized into sections that align with the analysis workflow (**Figure 1**).

(1) The first R Markdown file, ADMO_Data.Formatting.Rmd, transforms an ESSENCE-style line-list into a line-list of the format required for non-spatial and spatial analyses. It uses the mock dataset described in Section ***4.1. Mock data disclaimer and details*** and **Appendix A** as an example.
(2) The second R Markdown file, ADMO_Data.Analyses.Rmd, performs non-spatial and spatial anomaly detection based on user-defined customizations. Customization points are described in Section ***4.2.2. Customization points in the analytic code***.

Section ***4.2.3. Purpose of code sections, chunks, functions, and objects,*** explains the purpose of each code chunk, function, and object, organized by sections, across the two R Markdown files.

4.2.1. Errors and warning messages in R

Errors in R prevent code from running, whereas warnings in R indicate potential issues that do not halt execution. Although warnings do not halt execution, they can flag important issues that impact interpretation of the results. Users of the analytic code may encounter warning messages when estimating expected counts, particularly in the spatial models. Warnings in this section are likely due to model convergence issues. If a model fails to converge, consider using a simpler model to estimate the expected counts (e.g., by removing covariates). The customization points in **Table 8** provide examples of alternative spatial models to evaluate if convergence warnings appear.

Other warnings and errors are likely to be analysis-specific. We recommend consulting an artificial intelligence tool to troubleshoot these issues. Before doing so, ensure that the tool is approved for use in accordance with your organization's Information Technology team and policies.

4.2.2. Customization points in the analytic code

Section **0b (Chunk 2: Customization points)** in the non-spatial and spatial analysis R markdown (i.e., ADMO_Data.Analyses.Rmd) includes a list of all analytic customization options (**Table 8**). Users specify in **Chunk 2** their preferred value of the customization options. The specified customizations are then carried through the remainder of the code. If using the mock dataset accompanying the code and technical guidance, the only updates users must make to the code are specifying the preferred value of the customization options.

Users should set their preferred customizations based on review of **Table 8** and Sections ***2. Data sources and preparation*** and ***3. Statistical analysis*** of the technical guidance.

Table 8. Customization options for non-spatial and spatial anomaly detection (code section 0b, chunk 2: customization points)

Analysis	Object name	Description	Parameters	Considerations
All (non-spatial and spatial)	hist_start	Start of the historical period used to define the expected counts model.	Date in %Y%m%d format	• Historical period should not overlap with the observed period to avoid biasing the expected counts. • See Section ***3.1.2.2. Step 2.2: Identify source of historical data*** for more information about selecting the optimal historical period.
	hist_end	End of the historical period used to define the expected counts model.	Date in %Y%m%d format	
	obs_start	Start of the observed period, which comprises the data that will be compared to expected counts to detect anomalies.	Date in %Y%m%d format	
	obs_end	End of the observed period, which comprises the data that will be compared to expected counts to detect anomalies.	Date in %Y%m%d format	
	all_days	Generates sequence of days for the entire study period (historical + observed).	NA	• Automatically generates sequence of days across the entire study period based on hist_start and obs_end.
	state_fp	State-level FIPS code comprising jurisdiction of interest.	Character taking value from “01” to “56”	• See here to determine the appropriate state-level FIPS code.

Analysis	Object name	Description	Parameters	Considerations
	prop_threshold	Proportion of zero counts in the historical data used to determine whether zero-inflated models are used.	Numeric taking values from 0 to 1	• The default is 0.25; a higher proportion increases the likelihood that zero-inflated models are selected as optimal. • See Section ***3.1.2.3. Step 2.3: Train statistical model on historical data*** for more information about selecting the optimal proportion.
	geo_var	Spatial unit of interest for analysis.	One of: ZIP_patient, ZIP_fac, county_patient, county_fac	• See Section ***3.2.1. Spatial analyses*** for more information about selecting the optimal spatial unit.
	non.statistical_threshold	Non-statistical threshold to use in lieu of modeling expected counts.	NULL or numeric taking values from 0 to ∞	• Define this parameter if wanting to use a number that was not modeled to determine anomalies (e.g., based on public health response capacity). • The code sets the same non-statistical threshold across all stratifications if defined with a number; amendments to the code are necessary if wanting different non-statistical thresholds for different stratifications (e.g., different thresholds by age).
Non-spatial only	strat_options	Whether the non-spatial analysis should be stratified by geo_id, age_grp, sex, race_grp, or eth_grp.	TRUE, FALSE	• Stratification by too many covariates and covariates with too many levels may lead to sparse data, unstable estimates, and anomalies that are difficult to interpret. • Select TRUE if non-spatial analysis should be stratified by the variable or FALSE is non-spatial

Analysis	Object name	Description	Parameters	Considerations
				analysis should not be stratified by the variable. • Stratify the non-spatial analysis by geography (geo_id = TRUE) if you wish to identify anomalies in independent spatial units. This may be most relevant if the outcome data included nested geographies. For example, a user has data for an entire state and wants to identify whether there are anomalies in each independent county. The equivalent would be filtering and analyzing data from only the county of interest. Specifying geo_id = TRUE does not conduct a spatial analysis because the expected counts are estimated solely from the spatial unit of interest (i.e., there is no spatial smoothing).
	strat_terms	Terms to add to the statistical model based on strat_options.	NA	• This object is automatically calculated based on strat_options; run even if all options in strat_options = FALSE.
	formula_lightweight	Model formula with only event_date and strat_terms.	NA	• Run all formula options regardless of which you plan to use. • In order of increasing computational intensity and temporal complexity, the models go formula_lightweight, formula_default, and formula_advanced.
	formula_default	Model formula with event_date, temporal lag1, temporal lag2, day_of_week, month, and strat_terms.	NA	

Analysis	Object name	Description	Parameters	Considerations
	formula_advanced	Model formula with event_date, temporal lag1, temporal lag2, day_of_week, month, Fourier transformation, and strat_terms.	NA	• "formula_default" is the default formula for the non-spatial analysis and appropriate for most users; it accounts for temporal variation using temporal lag1 and lag2, day of week, and month terms. • "formula_lightweight" only estimates expected counts using date. • "formula_advanced" includes the same temporal terms as "formula_default" plus Fourier transformations, which may increase the model's ability to detect seasonal trends.
	seasonality_option	Selects which non-spatial model to run.	One of: none, categorical, fourier	• Select one of "none" (formula_lightweight, no seasonal structure), "categorical" (formula_default, minimal seasonal structure), or "fourier" (formula_fourier, maximal seasonal structure). • Use "fourier" only if there is least one year's worth of historical data.
	model_formula_ns	Non-spatial model that will be used to fit expected counts.	NA	• Automatically populated based on seasonality_option.
Spatial only	neighbor_method	Select the appropriate neighbor definition.	One of: queen, rook, or buffer	• Queen = neighbors share at least one point; this definition is considered the default. • Rook = neighbors share only an edge; this definition may be more appropriate for rural jurisdictions because contiguous spatial units may still be far apart.

Analysis	Object name	Description	Parameters	Considerations
				• Buffer = boundaries of neighbors fall within a certain distance. • If your jurisdiction is mostly urban, queen or rook contiguity are likely appropriate because neighbors are physically close.
	mindist_m	If neighbor_method = "buffer", input the minimum distance between centroids for inclusion as neighbors.	Numeric	• If neighbor_method = "queen" or "rook", input mindist_m = 0. • If neighbor_method = "queen" or "rook", input mindist_m = 4500000; the United States is ~3000 miles (4,500,000 meters wide), so this setting will ensure all neighbors in your jurisdictions will be estimates.
	maxdist_m	If neighbor_method = "buffer", input the maximum distance between centroids for inclusion as neighbors.	Numeric	
	max_lag	Input the number of spatial lags to include.	Numeric	• The default is 2 lags and is appropriate for most analyses; only increase to 3 lags if the Moran's I correlogram shows strong autocorrelation at lag 3—beyond lag 3 is rarely justified as some autocorrelation at higher lags is expected but does not meaningfully improve model fit. • Fewer lags are typically needed for large spatial units (e.g., counties) since neighbors are already physically distant.
	formula_lightweight_spatial	One of the choices for model formula for detecting higher-	NA	

Analysis	Object name	Description	Parameters	Considerations
		than-expected anomalies.		• "formula_default_spatial" is a balanced option suitable for most users. • "formula_lightweight_spatial" replaces the 7-level day-of-week term with a binary weekday / weekend distinction; use if the "default" formula is slow or fails to converge. • "formula_advanced_spatial" adds a curved long-term trend (poly) and h seasonal cycle (Fourier terms) instead of months; use if counts changed substantially over the historical period—this model formula is slower and more likely to have convergence issues. • If max_lag > 2, add the additional lag terms to the formula manually. • Add other covariates of interest (e.g., age, sex) to the formula manually; however, including too many variables can lead to sparse data, unstable estimates, and anomalies that are difficult to interpret.
	formula_default_spatial	One of the choices for model formula for detecting higher-than-expected anomalies.	NA	
	formula_advanced_spatial	One of the choices for model formula for detecting higher-than-expected anomalies.	NA	
	formula_lower_default_spatial	One of the choices for model formula for detecting lower-than-expected anomalies.	NA	• **Note:** These spatial models, which are used to identify lower-than-expected anomalies, aggregate counts weekly. The other spatial models, which are used to identify higher-than-expected anomalies, aggregate counts daily.
	formula_lower_advanced_spatial	One of the choices for model formula for detecting lower-	NA	

Analysis	Object name	Description	Parameters	Considerations
		than-expected anomalies.		• "formula_lower_default_spatial" is a balanced option suitable for most users. • "formula_lower_advanced_spatial" adds a curved long-term trend (poly) and h seasonal cycle (Fourier terms) instead of months; use if counts changed substantially over the historical period—this model formula is slower and more likely to have convergence issues. • Add other covariates of interest (e.g., age, sex) to the formula manually; however, including too many variables can lead to sparse data, unstable estimates, and anomalies that are difficult to interpret.
	model_formula	Choose the model formula to use for higher-than-expected anomalies.	One of: formula_default_spatial, formula_lightweight_spatial, or formula_advanced_spatial	• See considerations for formula_default_spatial, formula_lightweight_spatial, or formula_advanced_spatial.
	model_formula_lower	Choose the model formula to use for lower-than-expected anomalies.	One of: formula_lower_default_spatial or formula_lower_advanced_spatial	• See considerations for formula_lower_default_spatial or formula_lower_advanced_spatial.
	selected_geo	Select the spatial unit to review for time-series plots.	If ZIP code: Numeric If county: Character	• For ZIP: enter a 5-digit ZIP code (e.g., "80202"). • For county: enter a county name (e.g., "Denver County").
	map_date	Choose a date to visualize higher-than-expected	Date in %Y%m%d format	• Use the time-series plots to focus on days where anomalies occurred.

Analysis	Object name	Description	Parameters	Considerations
		anomalies on the anomaly map.		
	map_week	Choose a week to visualize lower-than-expected anomalies on the anomaly map.	Date in %Y%m%d format	• Input the date that represents the first day of the week of interest.

4.2.3. Purpose of code sections, chunks, functions, and objects

Each R markdown file includes various code chunks, functions, and objects (**Table 9**). Understanding the purpose of the chunks, functions, and objects will help users interpret the outputs and visualizations of the non-spatial and spatial analyses (Section ***4.3. Interpreting outputs and visualizations***).

Table 9. Purpose of code sections, chunks, functions, and objects

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
Data.Formatting	NA	NA	1: Packages	Install and load packages needed to clean ESSENCE line-list to appropriate line-list format.	pkgs	List of packages to install and load.
			2: Import line-list data	Read in the ESSENCE line-list.	df	Dataframe with the ESSENCE line-list used for subsequent analysis.
					metadata	Extracts metadata from line-list.
			3: Create the count table shell	Generate a shell for the appropriately formatted line-list.	n	Specify the number of rows the shell should be.
					shell	Dataframe that includes the required variables outlined in Section ***2.2.1. Minimum variable requirements*** of the technical guidance.
			4: Populate the variables	Populate shell with values from the ESSENCE line-list (df).	od_codes	List of OD-specific codes of interest; the codes in the code are for use with the mock data, see the footnotes for **Table 2** of the technical guidance for recommendations for other codes to use.
					shell	Updated version of shell from **Chunk 3** with the corresponding values from df.
			5: Exporting data	Export the correctly formatted line-list for use with the non-spatial and spatial analyses.	shell_export	Adds metadata to the correctly formatted line-list.

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
Data.Analyses	0	Installs and loads packages and allows users to specify customization points.	1: Installing and loading packages	Install and load packages needed to conduct non-spatial and spatial anomaly detection.	pkgs	List of packages to install and load.
			2: Customization options	Read in the properly formatted line-list data and specify preferred parameterization of the customization options.	data	Properly formatted line-list data.
					metadata	Extracts metadata from line-list.
					hist_start	See **Table 8** in section ***4.2.2. Customization points in the analytic code***.
					hist_end	
					obs_start	
					obs_end	
					all_days	
					state_fp	
					prop_threshold	
					geo_var	
					non.statistical_threshold	
					strat_options	
					strat_terms	
					formula_lightweight	
					formula_default	
					formula_advanced	
					seasonality_option	
					model_formula_ns	
					neighbor_method	
					mindist_m	
					maxdist_m	
					max_lag	
					formula_lightweight_spatial	
					formula_default_spatial	
					formula_advanced_spatial	
					formula_lower_default_spatial	

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
					formula_lower_advanced_spatial	
					model_formula	
					model_formula_lower	
					selected_geo	
					map_date	
					map_week	
	1	Prepares and validates data and conducts exploratory spatial analyses.	3: Non-spatial and spatial: data cleaning and quality checks	Formats key columns to appropriate parameterization, checks structure of data, assesses zeros and NAs for the non-spatial analysis.	data2	Creates working copy of line-list data for subsequent manipulation (e.g., variable formatting).
			4: Spatial: downloading the spatial shapefile	Downloads sub-state spatial unit geographic boundaries (e.g., ZIP code, count) and crops to the state of analysis.	geo_sf	Dataframe with geometry for the sub-state spatial unit specified in geo_var.
					states_sf	Dataframe with geometry for all states in the United States.
					states_geom	states_sf filtered to the state specified in state_fp.
					geo_sf_state	geo_sf filtered to the state specified in state_fp.
			5: Spatial: aggregate line-list to count table (spatial)	Generate a count table from the line-list for the spatial analysis.	df_geo_day	Aggregate line-list to count table for spatial analysis (i.e., case counts by day and geography).
			6: Spatial: add zero count days to count table	Add days with zero counts to the count table for the spatial analysis.	all_geos	List of sub-state spatial units used when generating the complete time-series for the count table.
					df_geo_day_complete	Count table with all days, including zero days, for spatial analysis.

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
			7: Spatial: join geometry to count table	Construct final spatial count table by appending geometries to the count table created in **Chunk 6.**	df_geo_day_sf	Count table with all days, including zero days, and geometries, for spatial analysis.
			8: Spatial: create the spatial neighbors list from the polygons	Make list of spatial neighbors, which will be used to assess spatial autocorrelation.	geo_sf_nb	Dataframe with only one row per spatial unit for spatial neighbor and weight calculations.
					nb	List of spatial neighbors for each spatial unit specified in geo_var.
			9: Spatial: create the row-standardized weights matrix	Create the weights matrix, which quantifies the relationship between spatial units.	nb_lags	Row-standardized weights object.
					lw_list	Spatial lag weights object.
					lw1	Spatial lag weights object at lag1.
					lw2	Spatial lag weights object at lag2.
					hist_geo	df_geo_day_complete count table with only data from the historical period.
					y	Stores case counts from hist_geo and reorders to align with the order of the spatial unit polygons, which is needed for the Moran's I calculations.
			10: Spatial: Morans I and correlogram	Assess spatial autocorrelation to determine appropriate	mi	Analytic test for spatial autocorrelation.
					mi_mc	Monte Carlo test for spatial autocorrelation.

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
				number of lags to include in the historical counts model.	max_order	Maximum number of lag orders to compute.
					cor	Moran's I correlogram.
			11: Spatial: compute spatial lags for count table	Determine the weighted average count by spatial lag and temporally aggregate the counts from daily to weekly.	geo_order	Order spatial units in geo_sf_nb to match spatial weights object.
					df_geo_day_lag	Count table of all data with weighted average count by spatial lag.
					lag_cors	Pairwise Pearson correlation matrix across spatiotemporal lags.
					high_cor	lag_cors reduced to only include correlations where correlation coefficient > 0.7.
					df_geo_week	Aggregate the df_geo_day_lag count table to weekly counts from daily counts.
			12: Spatial: confirm geometries linked to count tables	Ensure count tables are linked to the geometry layer.	df_past_month	Aggregate counts from last 30 days of historical period for data quality check.
					raw_count_map	Map visualizing df_past_month.
			13: Spatial: choropleth map of smoothed data	Assess validity of chosen spatial lag structure.	df_past_month_smooth	Count table of spatially smoothed counts to assess validity of the chosen max_lag.
	2	Prepares the correctly formatted line-list data to a count table of the	14: Non-spatial: model historical data to get expected counts	Convert the correctly formatted line-list to a count table of historical data for	group_vars	List of variables used to stratify the historical counts table, based on event_date, geo_id, age_grp, sex, race_grp, and eth_grp.

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
		historical data, expanded by the various stratifications specified in the customization points.		the non-spatial analysis.	historical_data	Count table of historical data stratified by the covariates specified in group_vars.
					historical_summary	Summary statistics for the count table of historical data.
			15: Spatial: model historical data to get expected counts	Preparing the historical training dataset for a spatiotemporal statistical model and then calculating how sparse the overdose counts are.	df_train	Generates historical count table for higher-than-expected spatial anomalies.
					df_train_week	Generates historical count table for lower-than-expected spatial anomalies.
					geo_zero_prop	Calculates proportion of zeros in historical count table for higher-than-expected anomalies.
					geo_zero_prop_lower	Calculates proportion of zeros in historical count table for lower-than-expected anomalies.
	3	For non-spatial and spatial, fits an appropriate statistical model for computing expected counts based on the proportion of zeros and dispersion statistic in the historical data, and, for spatial, diagnoses whether the	16: Non-spatial: Identify the preferred non-spatial model	Build a non-spatial model to compute expected counts using the appropriate model specification based on the proportion of zero counts in the data and get model fit.	nonspatial_expectedcounts	Function to determine which statistical model to use when getting the expected counts for the non-spatial analysis.
					nonspatial_fit	Statistical model used to estimate expected counts for the non-spatial analysis, along with key diagnostics, including the proportion of zeros in the historical data, selected model type, stratification variables, MAE, and the historical data used to train the model. Use the data frame of the historical

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
		chosen spatial lag structure adequately accounts for spatial autocorrelation.				data used to train the model to contextualize the MAE.
			17: Spatial: Identify the preferred spatial model (1)	Pick the preferred spatiotemporal statistical model for upper and lower-than-expected anomalies based on the proportion of zeros.	geo_zero_prop	Summary of the proportions of zeros across spatial units with preferred model type, higher-than-expected anomalies.
					geo_zero_prop_lower	Summary of the proportions of zeros across spatial units with preferred model type, lower-than-expected anomalies.
					overall_model	Identifies preferred model for higher-than-expected anomalies.
					overall_model_lower	Identifies preferred model for lower-than-expected anomalies.
			18: Spatial: Identify the preferred spatial model (2)	Build a spatiotemporal statistical model to compute expected counts using the appropriate model specification based on the proportion of zero counts in the data.	spatial_expectedcounts	Function to determine which statistical model to use when getting the expected counts for the spatial analysis.
			19: Spatial: Identify the preferred spatial model (3)	Get spatial model fit from model identified in **Chunk 18**.	model_results	Results of function for preferred higher-than-expected anomaly model.
					model_results_lower	Results of function for preferred lower-than-expected anomaly model.

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
					best_model_type	Specifies the best model for higher-than-expected anomalies.
					best_model	Summary statistics for the best model for higher-than-expected anomalies.
					disp_model1	Dispersion statistic for one model option for higher-than-expected anomalies.
					disp_model2	Dispersion statistic for other model option for higher-than-expected anomalies.
					fit_model1	Summary statistics for one model option for higher-than-expected anomalies.
					fit_model2	Summary statistics for second model option for higher-than-expected anomalies.
					best_model_type_lower	Specifies the best model for lower-than-expected anomalies.
					best_model_lower	Summary statistics for the best model for lower-than-expected anomalies.
					disp_model1_lower	Dispersion statistic for one model option for lower-than-expected anomalies.
					disp_model2_lower	Dispersion statistic for other model option for lower-than-expected anomalies.
					fit_model1_lower	Summary statistics for one model option for lower-than-expected anomalies.

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
					fit_model2_lower	Summary statistics for second model option for lower-than-expected anomalies.
			20: Spatial: Identify the preferred spatial model (4)	Diagnose whether the chosen spatial lag structure sufficiently accounted for spatial autocorrelation in the data.	df_train_diag	Fit best_model to the historical data to estimate residuals for autocorrelation assessment.
					res_geo	Average Pearson residuals in df_train_diag by spatial unit across all days in historical data.
					resid_wide	Pivots Pearson residuals from df_train_diag to wide matrix, where columns are the temporal unit during the historical period (e.g., days, weeks) and columns are the geography of interest (e.g., ZIP).
					cross_moran	Function to calculate the space-time Moran's I.
					max_spatial_order	Maximum number of spatial lags for the space-time Moran's I calculation.
					max_tlag	Maximum number of temporal lags for the space-time Moran's I calculation.
					st_correlogram	Space-time Moran's I calculation based on Pearson residuals from resid_wide, using the max_spatial_order and max_tlag.
					st_sd	Standard deviation of the space-time Moran's I from

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
						st_correlogram, which is used in the space-time Moran's I heatmap below (st_plot).
					scale_lim	Data-driven scale used by the Moran's I heatmap below (st_plot).
					st_plot	Space-time Moran's I heatmap across the max_spatial_order and max_tlag combinations.
					lag_cors_st	Pairwise Pearson correlation matrix across spatiotemporal lags from df_train.
					high_cor_st	lag_cors_st reduced to only include correlations where correlation coefficient > 0.7.
	4	Estimate the expected counts in the observation period and determine which are anomalous.	21: Non-spatial: generate the expected counts in the observed period	Determine the expected counts in the observation period using the preferred non-spatial statistical model of the non-statistical threshold.	df_observed_nonspatial	Estimate the expected counts for the non-spatial data during the observed period using the preferred approach.
			22: Spatial: generate the expected counts in the observed period	Determine the expected counts in the observation period using the preferred spatial statistical model of the non-statistical threshold.	df_observed	Estimate the expected counts for the spatial data during the observed period using the preferred approach, higher-than-expected anomalies.
					df_observed_week	Estimate the expected counts for the spatial data during the observed period using the preferred approach, lower-than-expected anomalies.

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
			23: Non-spatial: create anomaly flags	Determine which observed counts are anomalies for the non-spatial analysis.	anomaly_flags	Function to determine which observed counts in the non-spatial data are anomalies.
					results_plot	Identifies which counts are anomalies and the type of anomaly.
			24: Spatial: create anomaly flags (1)	Determine which observed counts are anomalies, based on the spatial analysis.	df_observed	Update df_observed with percentiles.
					df_observed_weekly	Update df_observed_weekly with percentiles.
			25: Spatial: create anomaly flags (2)	Add anomaly flag levels to the observed counts for the spatial analysis.	df_observed	Update df_observed with anomaly flags.
					df_observed_weekly	Updated df_observed_weekly with anomaly flags.
	5	Visualize the anomalies.	26: Non-spatial: time-series plot with flags	Visualize non-spatial anomalies on a time-series plot for the full time series with stratifications.	timeseries_nonspatial	Time-series plot of anomalies based on non-spatial analysis.
			27: Non-spatial: alternative time-series plot with flags	Visualize non-spatial anomalies on a time-series plot faceted by any selected stratification variables.	strat_vars	Vector of active stratification variables selected in strat_options.
					create_nonspatial_plots	Function to generate time-series plot of anomalies based on non-spatial analysis, stratified by levels of strat_vars.
					timeseries_nonspatial_plots	List of time-series plots from the non-spatial analysis stratified by levels of strat_vars.

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
			28: Spatial: output table	Generates output table for plotting anomalies from the spatial analysis.	results_spatiotemporal	Simplified dataframe for spatial analysis plotting.
					summarise_flags	Function to examine distribution of anomaly flags.
					combined_table_upper	Higher-than-expected anomaly flags summarized in a table.
					combined_table_lower	Lower-than-expected anomaly flags summarized in a table.
			29: Spatial: time-series plot with flags (upper)	Visualize higher-than-expected spatial anomalies on a time-series plot.	df_geo_plot	Filter results_spatiotemporal by the spatial unit of interest specifed in selected_geo to make plotting more straightforward.
					plot_data_single	Spatial time series for geography of interested aggregated to weekly counts, no anomaly flags.
					plot_data_monthly	Dataframe for monthly time-series across spatial units.
					plot_data_weekly_geo	Dataframe for weekly time-series across five spatial units with most observed cases.
					top_geos1	List of five spatial units with most observed cases.
					plot_data_top1	plot_data_weekly_geo filtered to top_geos1 for plotting.
					plot_data_monthly_geo	Dataframe for monthly time-series across five spatial units with most observed cases.
					top_geos2	List of five spatial units with most observed cases.

R markdown	Section	Section purpose	Chunk	Chunk purpose	Functions / objects	Functions / objects purpose
					plot_data_top2	plot_data_monthly_geo filtered to top_geos2 for plotting.
			30: Spatial: map with flags (upper)	Visualize higher-than-expected anomalies from the spatial analysis on a map.	map_data	Dataframe for mapping, filtered to specific date based on map_date.
					spatial_higher_maps	Static map of anomalies, filtered to specific date based on map_date.
			31: Spatial: map with flags (lower)	Visualize lower-than-expected anomalies from the spatial analysis on a map.	map_week_data	Dataframe for weekly mapping, filtered to specific date based on map_week.

4.3. Interpreting outputs and visualizations

This section provides guidance on interpreting key analytic outputs, as well as model diagnostics and visualizations. Interpretations of key analytic outputs are also included in the analytic code for quick reference. If OD anomalies are detected, STLT epidemiologists should investigate further and cross-check findings with other data sources (e.g., EMS data; medical examiner and coroner records for fatal ODs; and toxicology, laboratory, prescription, and law enforcement data) to validate these anomalies.

4.3.1. Core interpretation

A true anomaly may exist when the observed count for a given temporal and spatial unit is flagged as anomalous according to the anomaly flag criteria (**Table 6**). Observed counts that exceed expected counts indicate more cases than anticipated (i.e., higher-than-expected or upper anomaly), whereas observed counts below expected counts indicate fewer cases than anticipated (i.e., lower-than-expected or lower anomaly). Detected anomalies may arise from various factors, including underreporting, outcome misclassification, or true anomalies.

In other words, non-spatial and spatial anomaly flags can be interpreted as follows: *The actual count of [insert outcome] on [insert temporal unit] in [insert spatial unit] was flagged as a level [insert number 0-7 depending on anomaly flag categories] anomaly, on a scale of 0 (no anomaly) to 7 (most strict anomaly). This means the actual count was likely [different (level 1-7) / not different (level 0)] from expected.*

Assumptions of both non-spatial and spatial models include:

(1) Estimated expected counts represent what would have been observed in the absence of an anomalous event;
(2) No unmeasured confounding factors influence the counts beyond those adjusted for in the model; and
(3) The assumptions of the selected statistical model used to estimate the expected counts (i.e., standard Poisson, standard negative binomial, zero-inflated Poisson, zero-inflated negative binomial).

Additional assumptions specific to spatial models include:

(1) The conclusions may change if the data were aggregated to different spatial units (i.e., modifiable spatial unit problem); and
(2) Residuals are assumed to be independent, which means the standard error in the spatial models may be underestimated because the model may not fully account for spatial autocorrelation.

4.3.2. Interpretation of key model diagnostics and stakeholder visualizations

The following section reviews the key diagnostics and visualizations users should consistently examine across analyses as part of the standard interpretation process. The diagnostics and visualizations below were generated using an example scenario based on the mock dataset. Specifically, non-spatial and spatial analyses were performed using daily historical data from 2018-2022 to identify anomalies in 2023, with default parameters for all customization options except for stratifying the non-spatial analysis by age group and analyzing the spatial data using the patient's county of residence. The values and patterns shown below may differ from other analyses, based on the user-selected data and customizations. The interpretations below represent a guiding example rather than a direct interpretation of all possible analyses.

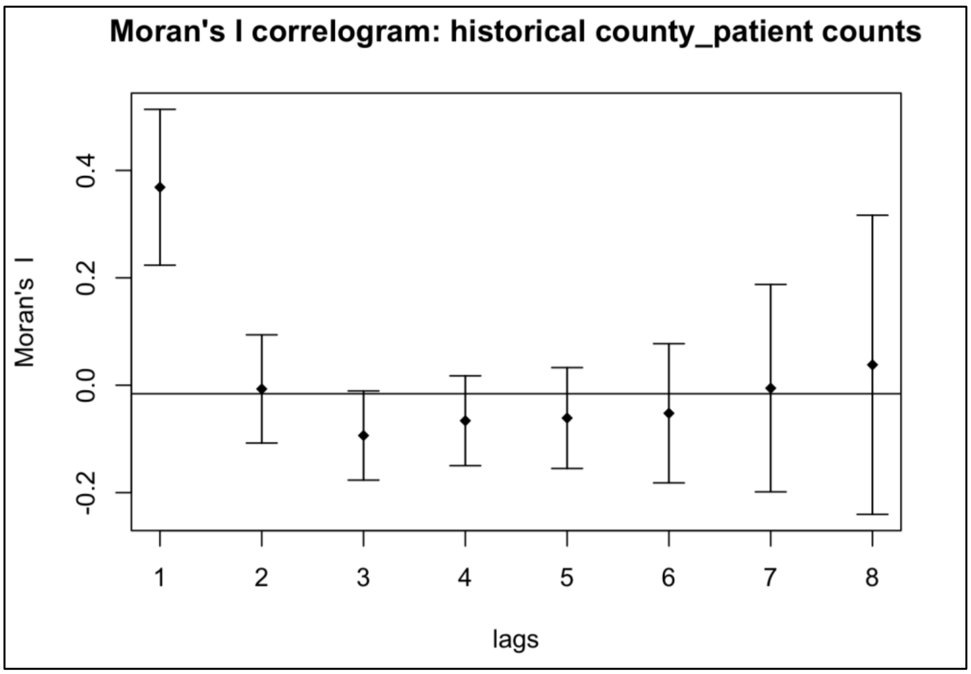
Section 1. Chunk 10: Spatial: Moran's I and correlogram

Chunk 10 calculates and visualizes spatial autocorrelation in historical case counts using global Moran's I, a measure of spatial clustering. First, it computes Moran's I using two approaches: analytically and via a Monte Carlo permutation test. The Monte Carlo permutation test may be more statistically robust than the analytic approach. Next, it generates a Moran's I correlogram, which illustrates how spatial autocorrelation changes across increasing spatial lag orders, indicating whether similarity in case counts varies with distance. The results help assess the degree of spatial clustering and inform the selection of an appropriate spatial lag order for subsequent analyses. Only spatial autocorrelation, and not spatial and temporal autocorrelation, are considered at this stage because the analytic code only allows users to easily change the spatial lag structure in spatial analyses. As noted in Figure 3, adjusting the temporal lags from the default (i.e., two temporal lags) in the spatial analysis requires more nuanced modifications to the analytic code than adjusting the spatial lags. Information about jointly assessing spatial and temporal autocorrelation is provided in **Chunk 20**.

```
	Moran I test under randomisation

data:  y
weights: lw1

Moran I statistic standard deviate = 5.2998, p-value = 5.795e-08
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.368651345      -0.015873016       0.005264074
```

```
        Monte-Carlo simulation of Moran I

data:  y
weights: lw1
number of simulations + 1: 1000

statistic = 0.36865, observed rank = 1000, p-value = 0.001
alternative hypothesis: greater
```

Interpretation: The p-values from both the analytic test for Moran's I (top) and Monte Carlo simulation of Moran's I (bottom) were statistically significant ($p < 0.05$), indicating that historical case counts were spatially clustered rather than randomly distributed.



Interpretation: Moran's I was highest and positive at spatial lag1, indicating positive spatial autocorrelation in OD counts by patient county among immediate neighbors (i.e., spatial lag1). Moran's I decreased as lag order increased, and was close to zero for lags 2-8, suggesting spatial dependence between counties weakened with increasing distance. Overall, spatial effects were strongest among immediate neighbors (i.e., spatial lag1) and declined with increasing separation. This means one spatial lag might be sufficient to account for spatial autocorrelation in this scenario.

Section 1. Chunk 11: Spatial: compute spatial lags for count table

Chunk 11 creates spatiotemporal lags and assesses correlation between the lag terms. We assess correlation between the spatiotemporal lag terms in the spatial model formula because including spatiotemporal lag terms that are highly correlated may result in unstable coefficient estimates. Users should consider whether to remove from the model formula spatiotemporal lags terms with high correlation (correlation > 0.7).

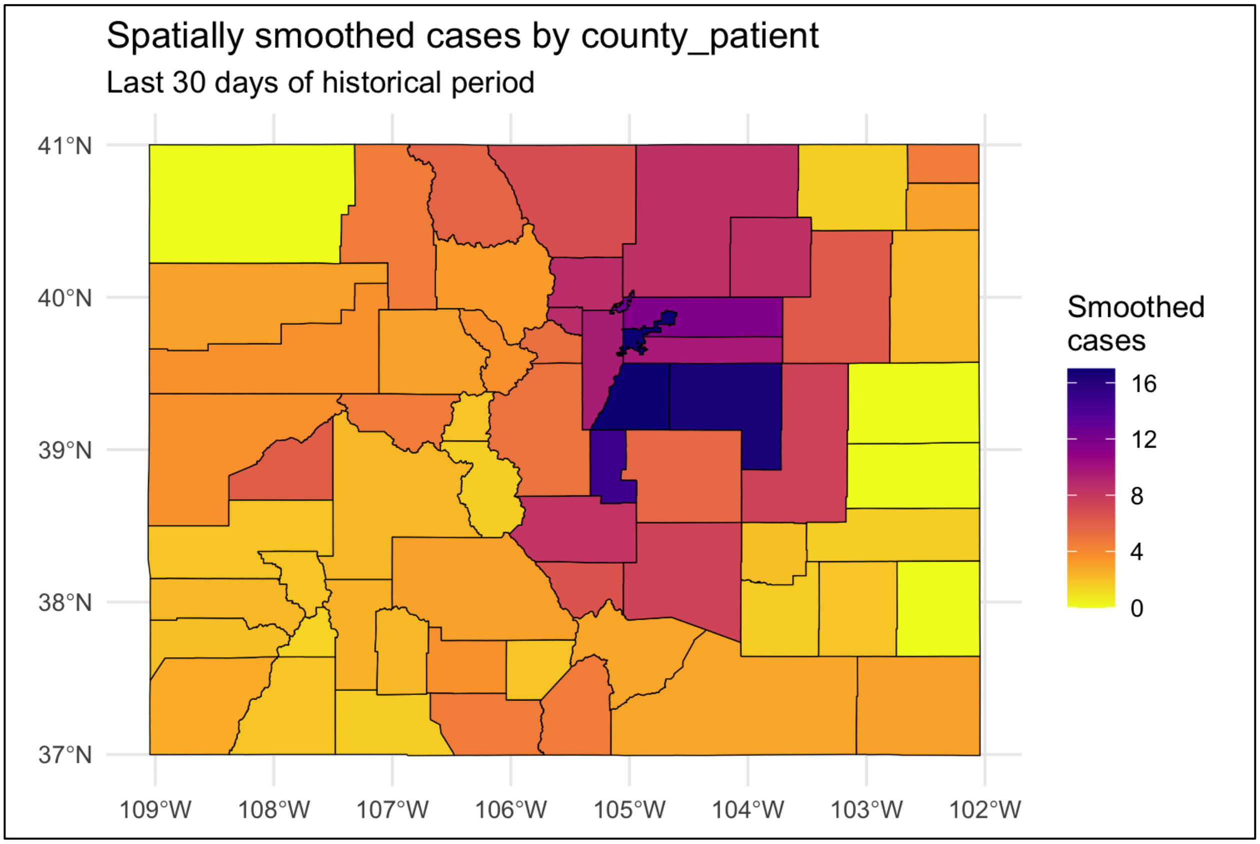
	lag1	lag2	lag1_d1	lag1_d2	lag2_d1	lag2_d2	count_d1	count_d2
lag1	1.00	0.11	0.31	0.31	0.10	0.11	0.13	0.13
lag2	0.11	1.00	0.10	0.11	0.26	0.27	0.00	0.00
lag1_d1	0.31	0.10	1.00	0.31	0.11	0.10	0.13	0.13
lag1_d2	0.31	0.11	0.31	1.00	0.10	0.11	0.13	0.13
lag2_d1	0.10	0.26	0.11	0.10	1.00	0.26	0.01	0.00
lag2_d2	0.11	0.27	0.10	0.11	0.26	1.00	0.00	0.01
count_d1	0.13	0.00	0.13	0.13	0.01	0.00	1.00	0.21
count_d2	0.13	0.00	0.13	0.13	0.00	0.01	0.21	1.00

Interpretation: No spatial (i.e., lag1, lag2), temporal (i.e., count_d1, count_d2), or spatiotemporal (i.e., lag1_d1, lag1_d2, lag2_d1, lag2_d2) lag terms were highly correlated. We do not have multicollinearity concerns with these spatial, temporal, or spatiotemporal lags. The lag terms included in the spatial model may not destabilize the regression coefficients. Importantly, if spatial, temporal, or spatiotemporal lags are added or removed from the statistical model, the diagnostics in **Chunk 11** should be re-run and re-evaluated accordingly.

Section 1. Chunk 13: Spatial: choropleth map of smoothed data

Chunk 13 generates a spatial map of smoothed OD case counts for each spatial unit over the final 30 days of the historical period. It aggregates case counts and their spatially smoothed values, joins them with spatial geometry data, and format the results for mapping. The resulting map visualizes the smoothed counts.

Smoothing reduces random noise and helps highlight spatial patterns or clustering across neighboring areas. This visualization supports identification of regions with persistent high or low case counts and allows evaluation of how well the smoothing captures the underlying spatial structure.



Interpretation: Higher patient case counts cluster in a few distinct regions (dark purple). Most counties had lower counts (yellow). Spatial smoothing revealed underlying trends and regional clusters that may be obscured in the raw counts, highlighting areas for further investigation.

Section 3. Chunk 16: Non-spatial: Identify the preferred non-spatial model

Chunk 16 generates the optimal non-spatial statistical model and outputs key model diagnostics. The four key diagnostics automatically output by the non-spatial model are: the proportion of zeros in the historical data, the regression type, the stratification variables, and the MAE. The proportion of zeros justifies whether the regression model is standard or zero inflated, the stratification variables output verifies whether the model appropriately stratified by the options selected in **Chunk 2**, and the MAE can be used to assess model performance. Users can contextualize the MAE by determining the average number of cases per unit time in the historical data.

```
> 100 * nonspatial_fit$proportion_zeros
[1] 41.92223
> nonspatial_fit$type
[1] "zinb"
> nonspatial_fit$strat_vars
[1] "age_grp"
> nonspatial_fit$mae
[1] 0.8980294
> mean(nonspatial_fit$training_data$case_count)
[1] 1.158178
```

Interpretation: The proportion of zeros in the historical data was 0.419 (41.9%), which was larger than the default zero-proportion threshold (25%). The optimal non-spatial model based on the historical data was a zero-inflated negative binomial (nonspatial-fit$type = “zinb”) because the proportion of zeros was above the zero-proportion threshold and the dispersion statistic on the zero-inflated negative binomial was closer to 1.0 than the zero-inflated Poisson. The model appropriately stratified by age group. The MAE was 0.898, meaning that, on average, the model’s estimated counts in the historical period were off by 0.898 counts per day. Considering the average number of counts per day across the historical period was 1.158, the MAE is large. We may want to (1) lower the temporal window and percentile threshold of the anomaly flags or (2) add covariates to the optimal statistical model, which may improve model fit. See Section *3.1.2.4. Step 2.4: Pick optimal model based on model diagnostics* for more information.

Section 3. Chunk 19: Spatial: Identify the preferred spatial model (3)

Chunk 19 generates the optimal spatial statistical model and outputs key model diagnostics. The two key diagnostics automatically output by the spatial model are: the regression type and the MAE. The MAE can be used to assess model performance. Users can contextualize the MAE by determining the average number of cases per spatiotemporal unit in the historical data.

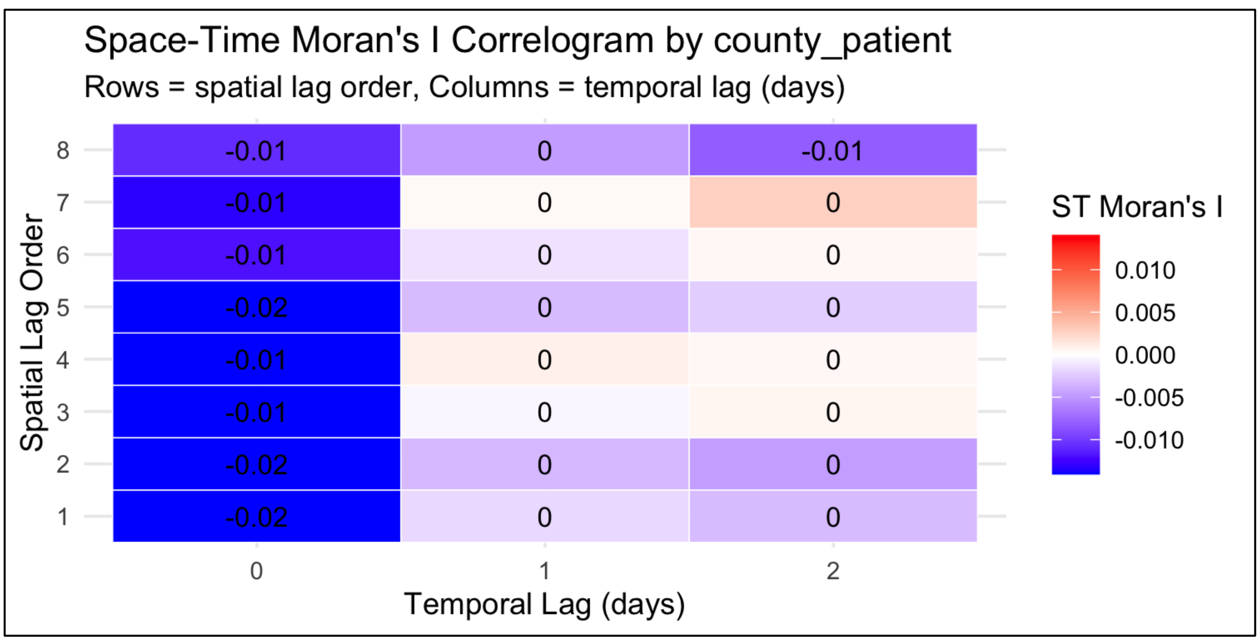
```
> best_model_type;model_results$mae_spatial;mean(df_train$case_count)
[1] "Zero-Inflated Negative Binomial"
[1] 0.1657975
[1] 0.1085792
```

Interpretation: The optimal spatial model was a zero-inflated negative binomial, meaning the proportion of zeros was larger than 25% and the dispersion statistic of the zero-inflated negative binomial was closer to 1.0 than the zero-inflated Poisson. The

MAE was 0.166, meaning that, on average, the model's estimated counts in the historical period were off by 0.166 counts per day. Considering the average number of counts per day across the historical period was 0.109, the MAE is large. We may want to (1) lower the temporal window and percentile threshold of the anomaly flags or (2) add covariates to the optimal statistical model, which may improve model fit. See Section *3.1.2.4. Step 2.4: Pick optimal model based on model diagnostics* for more information.

Section 3. Chunk 20: Spatial: Identify the preferred spatial model (4)

Chunk 20 evaluates whether a spatial model adequately accounts for spatial and temporal dependence by examining residual spatiotemporal autocorrelation. It first computes Pearson residuals for each spatial unit, averages them, and aligns them to spatial geometry. It then estimates the space-time global Moran's I and visualizes whether spatiotemporal dependence persists across spatiotemporal lags. Essentially, it assesses if the spatiotemporal lag structure appropriately accounted for autocorrelation. Each cell in the heatmap represents the Moran's I at the specific spatial and temporal lag. Temporal lag0 is the standard global Moran's I. Values are typically small for well-specified data and increase as data become more dependent. Focus interpretation of the heatmap on the values in each cell, not just the color scale—the color scale is relative to the data used to fit the model. Values near zero suggest little remaining spatiotemporal dependence. Finally, the code evaluates correlations among spatial and spatiotemporal lag terms to identify potential multicollinearity.



	lag1	lag2	lag1_d1	lag1_d2	lag2_d1	lag2_d2
lag1	1.00	0.11	0.28	0.28	0.09	0.09
lag2	0.11	1.00	0.09	0.10	0.24	0.24
lag1_d1	0.28	0.09	1.00	0.28	0.11	0.09
lag1_d2	0.28	0.10	0.28	1.00	0.09	0.11
lag2_d1	0.09	0.24	0.11	0.09	1.00	0.24
lag2_d2	0.09	0.24	0.09	0.11	0.24	1.00

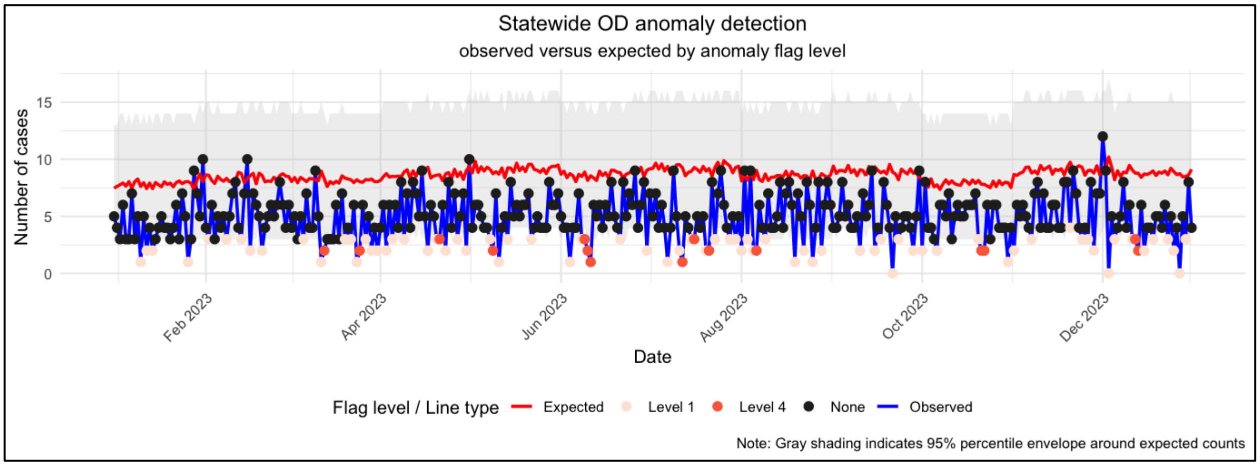
Interpretation: Space-time Moran's I values were consistently near zero across temporal and spatial lags, suggesting that the spatial model captured spatiotemporal dependence (i.e., two spatiotemporal lags).

The correlation between the selected spatial (i.e., lag1, lag2) and spatiotemporal (i.e., lag1_d1, lag1_d2, lag2_d1, lag2_d2) lag terms was less than 0.7, suggesting the lag terms included in the spatial model may not destabilize the regression coefficients. Importantly, if spatial or spatiotemporal lags are added or removed from the statistical model, the diagnostics in **Chunk 20** should be re-run and re-evaluated accordingly.

Section 5. Chunk 26: Non-spatial: time-series plot with flags

Chunk 26 generates a time-series plot comparing observed and expected OD case counts across the analytic domain, highlighting anomaly detection results. The plot uses colored ribbons to show percentiles, lines to display actual and expected counts, and points to indicate severity levels of anomalies based on statistical flags (Table 6). Together, these elements provide a visual and quantitative summary of periods with significant deviations from expected case counts for further investigation.

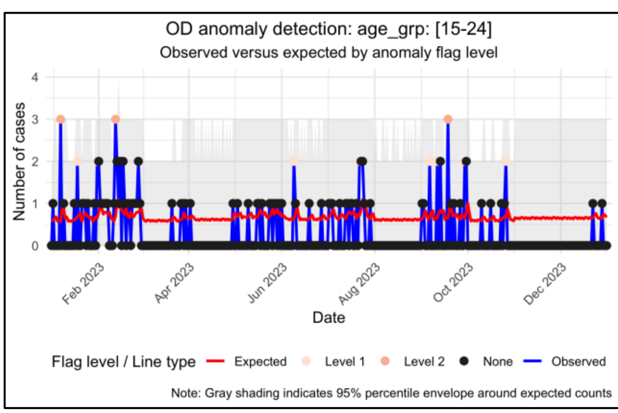
Note, if stratification options are specified in **Chunk 2**, the plot in **Chunk 26** will show all stratifications on the same plot. For example, if stratifying the non-spatial model by age, the plot in **Chunk 26** would have six points per day. Consider first running the non-spatial analysis code without stratification options, but all other preferred customizations, and generating the plot in **Chunk 26**. Then, specify stratification options, re-run the code, and compare the plots in **Chunk 27** with the plot from **Chunk 26** before stratifications were specified.

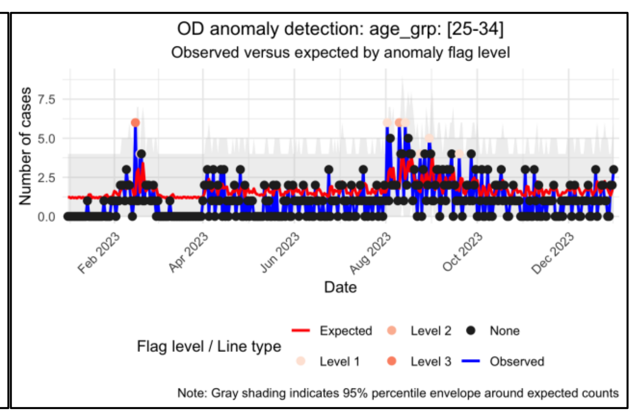


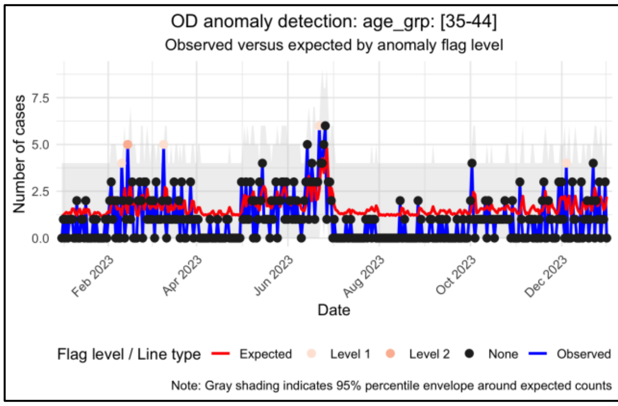
Interpretation: The red line represents the expected daily number of cases, modeled statistically in the example above, while the blue line shows the daily observed counts. The observed period includes numerous lower-than-expected anomaly flags, including both lower-than-expected anomaly flag levels (i.e., levels 1 and 4), suggesting potential lower anomalies. See Section *__4.3.1. Core interpretation__* for the formal interpretation of any specific anomaly. Consider zooming in on specific sections of observation period (e.g., August 2023) to better visualize anomalies. Alternatively, consider stratifying the non-spatial analysis by stratification variables (**Chunk 27**) or conducting a spatial analysis (**Chunks 29-31**) to see if anomalies vary across groups.

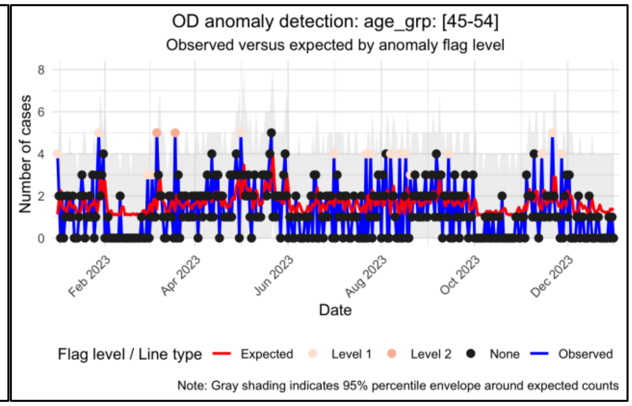
Section 5: Chunk 27: Non-spatial: alternative time-series plot with flags

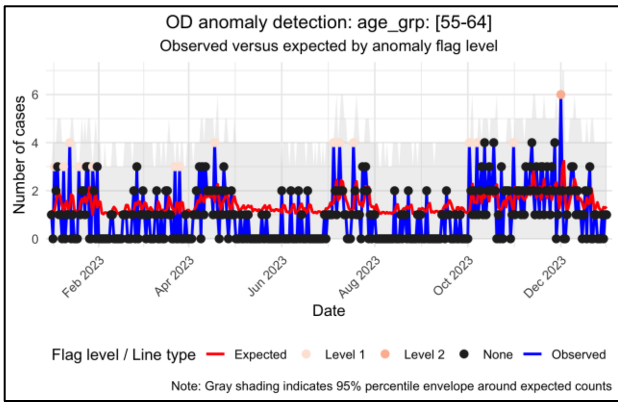
Chunk 27 generates a time-series plot, stratified by levels of the stratification variables, comparing observed and expected OD case counts across the analytic domain. The structure of the interpretation is the same as **Chunk 26**. Note the level of anomalies presented in each chart, indicated in the legends.

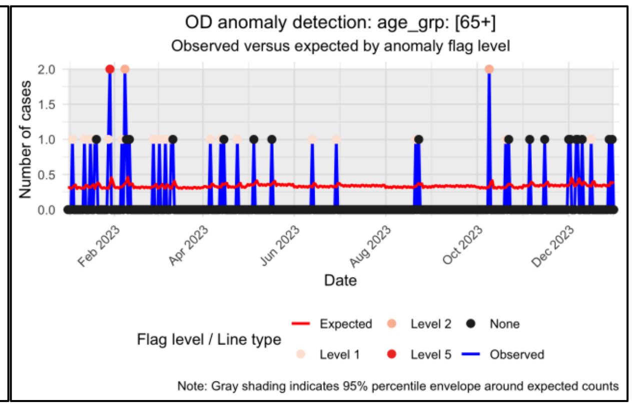








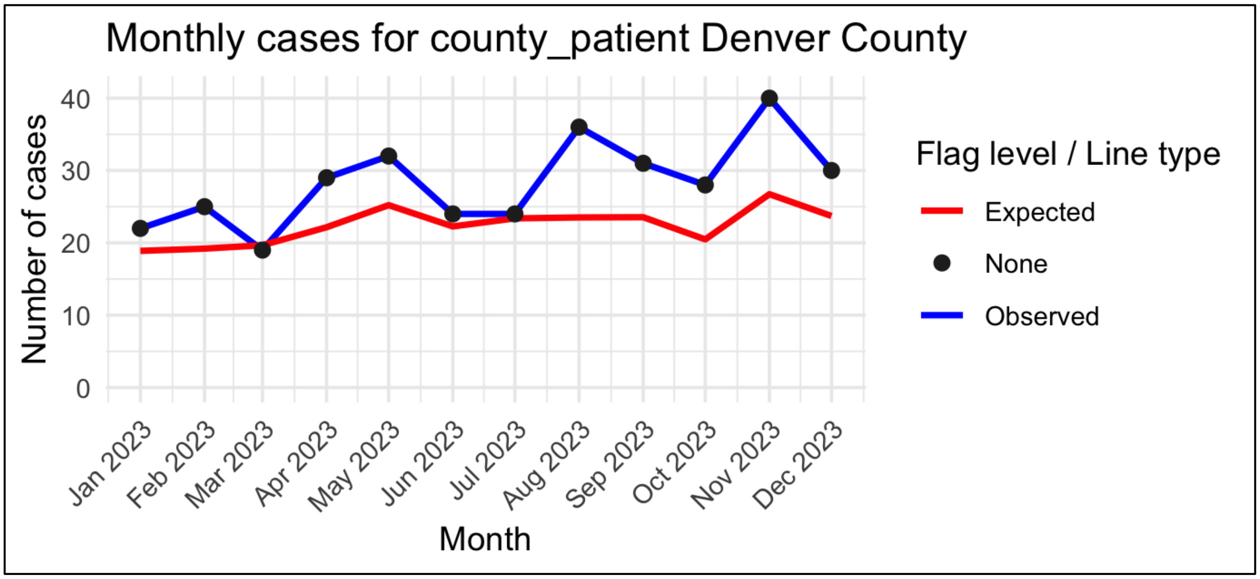




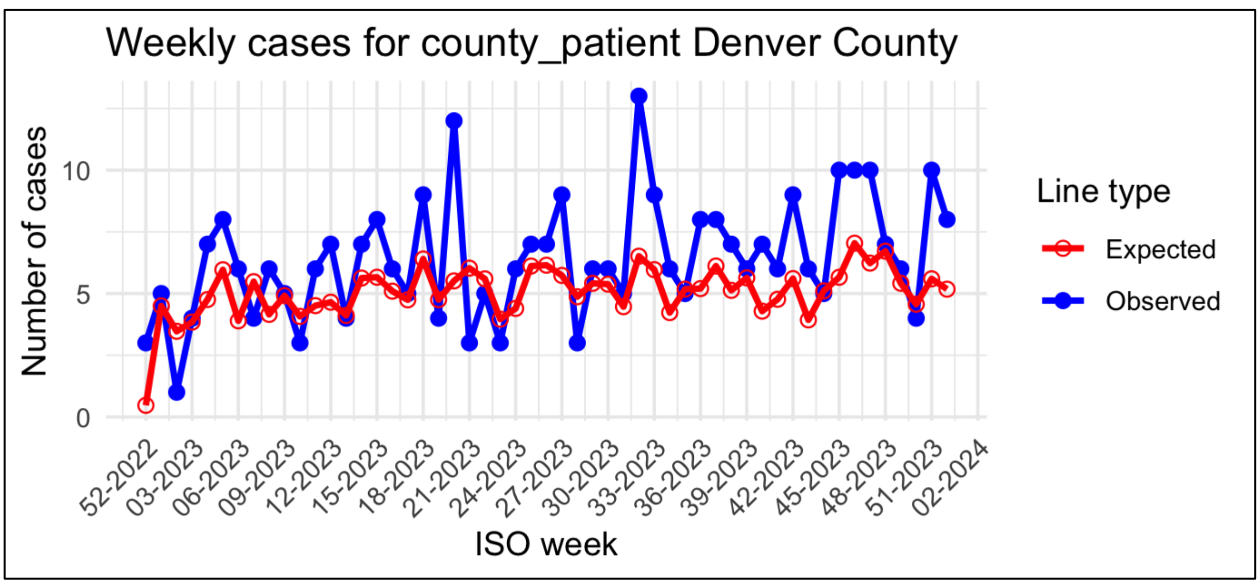
Interpretation: The red line represents the expected daily number of cases, modeled statistically in the example above, while the blue line shows the daily observed cases. The observed period includes numerous anomaly flags across age stratifications, including levels 1, 2, 3, and 5. There were no level 6 or 7 anomalies, as defined using the default anomaly flags (**Table 6**). See Section ***4.3.1. Core interpretation*** for the formal interpretation of any specific anomaly. Compared to the overall plot in **Chunk 26**—which identified only lower-than-expected anomalies when looking across the full, unstratified analytic domain—stratifying the data by age group revealed differential trends in expected counts and anomalies by age, including no lower-than-expected anomalies. For example, the 65+ age group had the lowest expected counts (red line) but the only level 5 anomaly. Meanwhile, the 35-44 age group had higher expected counts (red line) than the 65+ age group, but fewer total anomaly flags.

Section 5. Chunk 29: Spatial: time-series plot with flags (upper)

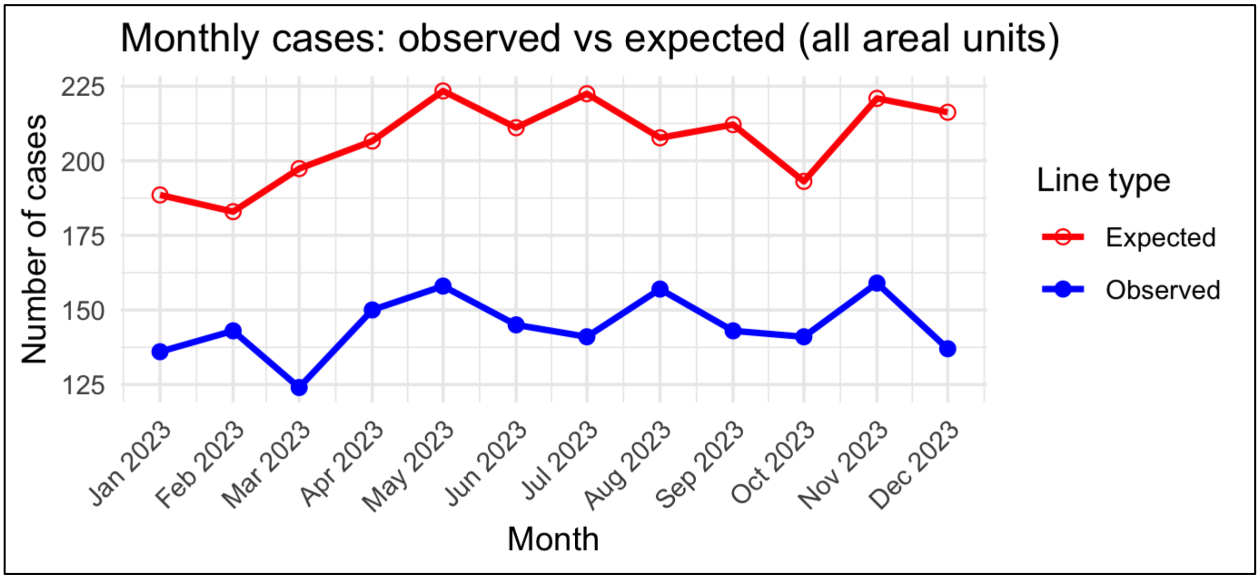
Chunk 29 produces multiple time-series visualizations of observed and expected OD case counts across different temporal and spatial aggregation schemes.



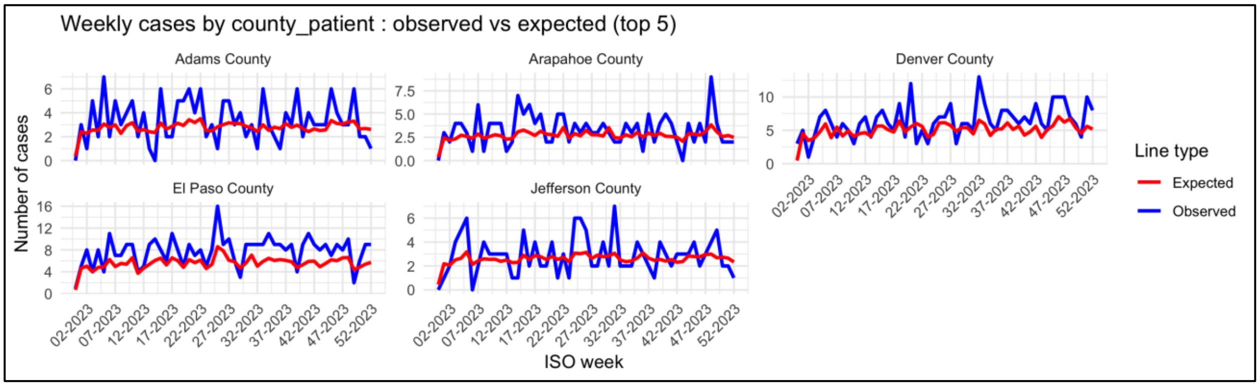
Interpretation: Observed and expected case counts fluctuated between 20 and 40 each month in the selected county (i.e., Denver County). No months were flagged as anomalous. Further exploration of these data at higher temporality may be warranted, particularly in months where the observed line was greater than the expected line (e.g., the map in **Chunk 30**). Note, the plot visualizes the maximum flag level among all days in a given month—the statistical model used to estimate the expected counts was based on daily data, and those daily data were aggregated to monthly counts for this plot. Results may differ if counts were aggregated to monthly prior to fitting the statistical model on the historical data and applying that model to the observation period.



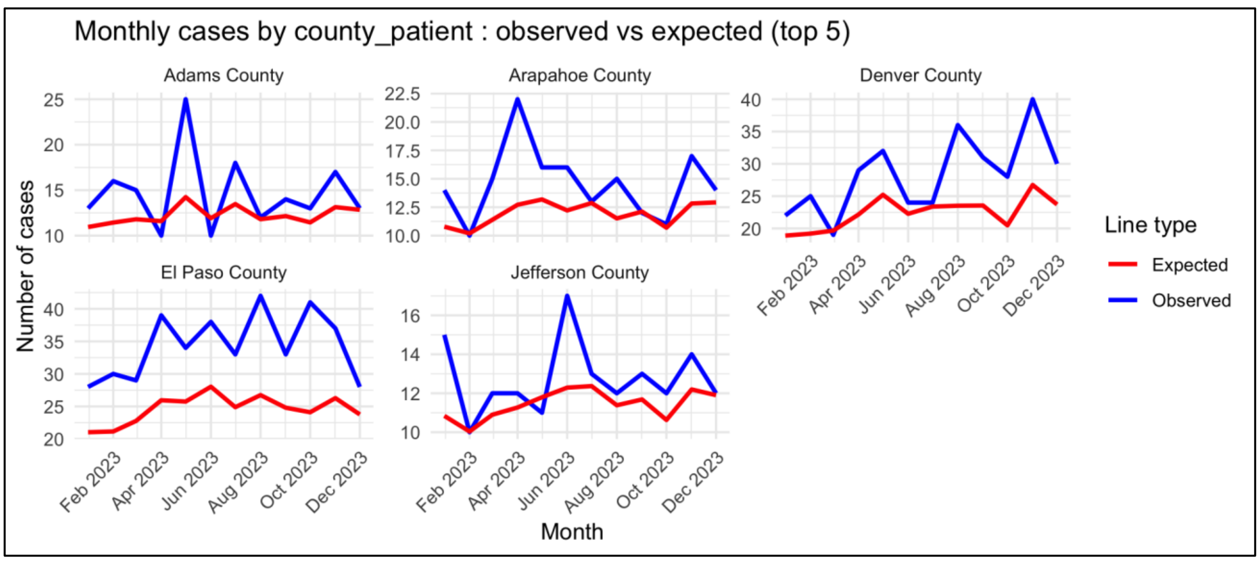
Interpretation: Observed and expected case counts fluctuated between one and 15 each week in the selected county (i.e., Denver County). The expected weekly counts are around five each week. In many weeks, the observed count in the selected county was above the expected count. Note, the plot visualizes expected and observed counts aggregated by week—the statistical model used to estimate the expected counts was based on daily data, and those daily data were summed to weekly counts for this plot. Results may differ if counts were aggregated to weekly prior to fitting the statistical model on the historical data and applying that model to the observation period.



Interpretation: The expected counts were consistently higher than observed counts each month, across all areal units. Two interpretations of this plot include: (1) the model may have overestimated the expected number of cases, suggesting a need adjust the statistical model estimating the expected counts; or (2) there were truly fewer cases observed than expected in this time period. Together, these interpretations suggest further exploration. Note, the plot visualizes expected and observed counts aggregated by month—the statistical model used to estimate the expected counts was based on daily data, and those daily data were summed to monthly counts for this plot. Results may differ if counts were aggregated to monthly prior to fitting the statistical model on the historical data and applying that model to the observation period.



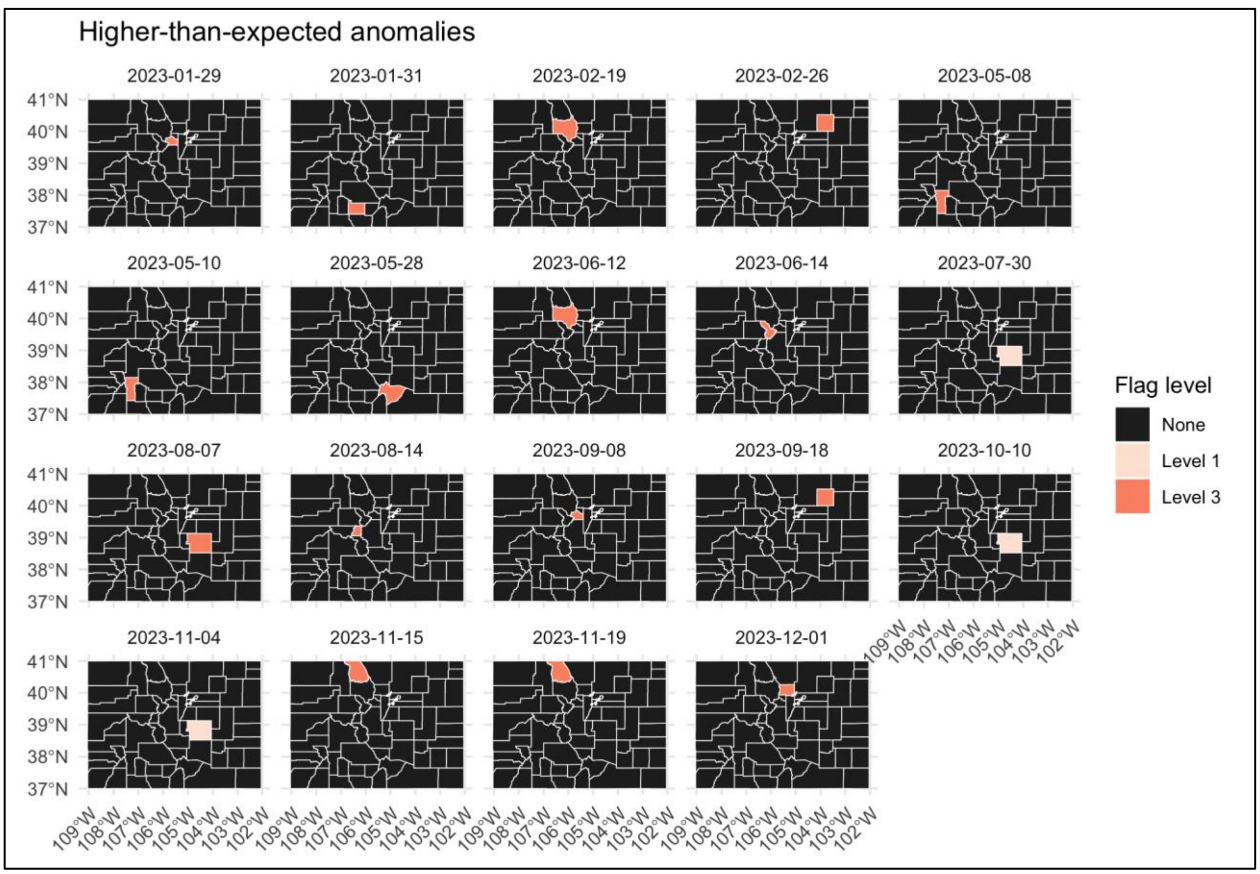
Interpretation: The output shows observed versus expected weekly counts for the five counties with the most observed cases across the observed period. It is important to note that the range in the number of cases varied across counties. Some weeks had more observed than expected cases, especially in El Paso County. Further exploration of weeks with many more observed than expected cases may be warranted. Note, the plot visualizes expected and observed counts aggregated by week—the statistical model used to estimate the expected counts was based on daily data, and those daily data were summed to weekly counts for this plot. Results may differ if counts were aggregated to weekly prior to fitting the statistical model on the historical data and applying that model to the observation period.



Interpretation: The output shows observed versus expected monthly counts for the five counties with the most cases across the observed period. It is important to note that the range in the number of cases varied across counties. Many months had more observed than expected cases. In El Paso County, all months had more observed cases than expected cases. Further exploration of months with many more observed than expected cases may be warranted. Note, the plot visualizes expected and observed counts aggregated by month—the statistical model used to estimate the expected counts was based on daily data, and those daily data were summed to monthly counts for this plot. Results may differ if counts were aggregated to monthly prior to fitting the statistical model on the historical data and applying that model to the observation period.

Section 5. Chunk 30: Spatial: map with flags (upper)

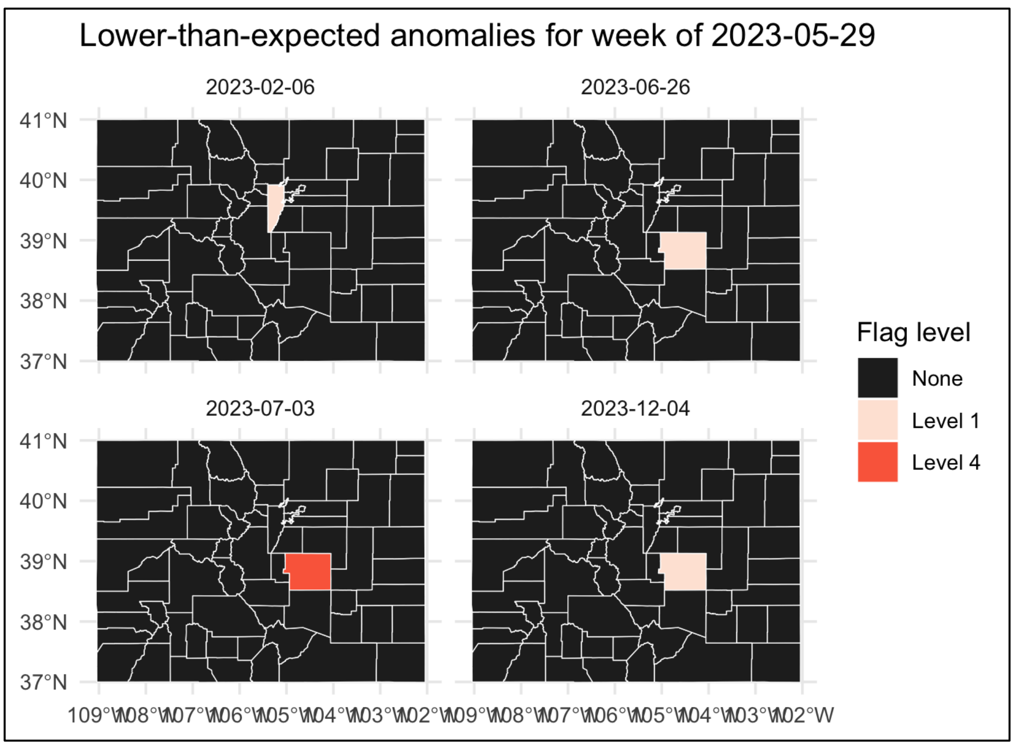
This code visualizes spatial anomalies for a selected date by merging spatial boundary data with anomaly detection results for each unit. It creates a color-coded map where polygons represent spatial units and their severity level of anomalies (**Table 6**). The legend describes the meaning of each severity level, allowing quick identification of units experiencing unusual events. Additionally, it generates a table listing spatial units flagged as anomalous, including their observed count, expected count, and severity, providing a summary of the locations and intensity of anomalies for the specified date.



Interpretation: Various counties experienced higher-than-expected anomalies across the observed period. Across the observed period, higher-than-expected anomalies occurred on 19 days. Interestingly, among days with anomalies, only one county per day was flagged as a level 1 or level 3 and no day had two or more counties with anomalies. Counties flagged as anomalous may warrant additional investigation of the identified anomaly. Note, it may be beneficial to increase the spatial resolution of the data (e.g., ZIP code) to identify areas within counties that drive county-level anomaly flag.

Section 5. Chunk 31: Spatial: map with flags (lower)

This code detects and visualizes spatial units with lower-than-expected overdose counts during a specific week, which may indicate gaps in data reporting or other unusual decreases. It merges weekly anomaly detection results with spatial boundary data and produces a map where polygons are color-coded by severity level of the negative anomaly ([Table 6](#)). Additionally, it outputs a table listing spatial units flagged with anomalies, showing their observed and expected counts and severity, to facilitate further investigation of the flagged locations.



Interpretation: Two counties were flagged as lower-than-expected anomalies across four different weeks in the observation period. It may be beneficial to review lower-than-expected anomalies by facility county or ZIP code, which could inform whether underestimation contributed to the lower-than-expected anomalies, especially in El Paso County. Note also the plot visualizes anomaly flags aggregated by week—the statistical model used to estimate the expected counts was based on daily data, and those daily data were summed to weekly counts for this plot. Results may differ if counts were aggregated to weekly prior to fitting the statistical model on the historical data and applying that model to the observation period.

5. Appendices

Appendix A. Adaptation of the mock dataset from CDC WONDER data

The process of generating the mock data included several steps. First, we downloaded from CDC WONDER counts of death for causes X40-X44, X60-X64, X85, and Y10-Y14 stratified by county and month in Colorado from 2018 to 2023. Second, we downloaded from CDC WONDER counts of death for causes X40-X44, X60-X64, X85, and Y10-Y14 in Colorado from 2018 to 2023 with the following stratifications: county-month-race, county-month-single-year-age, county-month-sex. Third, we downloaded from CDC WONDER counts of death stratified by county and month in Colorado from 2018 to 2023 individually for each specific cause (i.e., X40, X41, X42, X43, X44, X60, X61, X62, X63, X64, X85, Y10, Y11, Y12, Y13, Y14). Fourth, when death counts for a stratification were between one and nine, data were suppressed by CDC WONDER, and we assigned a value between one and nine to those stratifications based on the total sums for each stratification. Lastly, we generated a line-list from the aggregated counts and ensured the proportions of death by stratification in the line-list aligned with the proportion of deaths by stratification in the aggregated data from CDC WONDER. The variable names and formats for the key variables—date, cause, age, sex, racial identity, ethnicity, patient ZIP code of residence, patient county of residence, facility ZIP code—were coded to align with the NSSP Data Dictionary. Other variables that appear when exporting daily line-lists from ESSENCE were coded as NA because they were not required for this analysis.

Appendix B. Heuristic for decision points in the analysis based on anomaly flag frequency

The analytic code requires setting various parameters. We developed a heuristic to support users in setting some of the parameters based on the preferred frequency of anomaly flags.

The impact of decisions on the frequency of anomaly flags is based on statistical theory and may differ in real-world data. Further, the heuristic considers the independent effects of a decision on the frequency of anomaly flags (e.g., only using capacity-based thresholds). Users should investigate how different combinations of decisions affect their data.

Decision points (step in Figure 1)	**Considerations**	**Decision options**		**I want the frequency of anomaly flags to be…**	
				Rare	***Common***
Data aggregation (step 1)	It is important to identify anomalies based on…	different temporal units (e.g., daily, weekly, monthly)	Temporal	Aggregate data to larger temporal units	Keep data at finest temporal resolution available
Expected counts (step 2.1)	It is important to identify anomalies based on…	what my STLT health department is prepared to respond to	Capacity-based	↑ capacity-based threshold	↓ capacity-based threshold
		statistical cutoffs estimated from historical data	Model-based	Historical data used to get estimated counts are similar to observed counts	Historical data used to get estimated counts are NOT similar to observed counts
Expected counts (step 2.3)	It is important the expected counts are based on…	historical data from neighboring areas	Spatial analysis*	Holding the observed counts constant, including spatial neighbors impacts the expected counts by…	
				↑ the expected counts compared to not including spatial neighbors	↓ the expected counts compared to not including spatial neighbors
Flag strictness (step 3)	It is important the anomaly flags look at…	patterns observed across consecutive days	Temporality	↑ consecutive days	↓ consecutive days
		statistical deviation from expected counts	Percentiles	↑ percentile closer to 1.00 and ↓ closer to 0.00	↓ percentile closer to 1.00 and ↑ closer to 0.00

* = Example: Consider we had historical data for areas A, B, and C. We want to see if there are anomalies in the observed counts in area A. We will compare the observed counts in area A to expected counts modeled using the historical data from areas A, B, and C. Compared to only modeling the expected counts based on the historical data from area A, anomaly flags will be rare if the expected counts modeled using areas A, B, and C increase and anomaly flags will be common if the expected counts modeled using areas A, B, and C decrease.

6. References

1. Idaikkadar N, Bodin E, Cholli P, et al. Advancing Ethical Considerations for Data Science in Injury and Violence Prevention. *Public Health Reports®*. 2025;140(4):405-411. doi:10.1177/00333549241312055

2. Centers for Disease Control and Prevention. Query Development: Community and Technical Resources. Published online November 2023. Accessed April 27, 2026. https://www.cdc.gov/nssp/documents/guides/query-development-job-aid-508-fcx.pdf

3. Centers for Disease Control and Prevention. DOSE-SYS Dashboard: Nonfatal Overdose Syndromic Surveillance Data. Overdose Prevention. February 25, 2026. Accessed March 5, 2026. https://www.cdc.gov/overdose-prevention/data-research/facts-stats/dose-dashboard-nonfatal-surveillance-data.html

4. Azondekon GR, Sheppard M. Rnssp: A Signature R package for the National Syndromic Surveillance Program (NSSP). Published online August 2025. doi:10.5281/zenodo.16929440

5. Russell M. *Statistics in Natural Resources: Applications with R*. Chapman and Hall/CRC; 2022. doi:10.1201/9781003285809

6. Hu MC, Pavlicova M, Nunes EV. Zero-Inflated and Hurdle Models of Count Data with Extra Zeros: Examples from an HIV-Risk Reduction Intervention Trial. *Am J Drug Alcohol Abuse*. 2011;37(5):367-375. doi:10.3109/00952990.2011.597280

7. Zuur AF, Ieno EN, Walker N, Saveliev AA, Smith GM. *Mixed Effects Models and Extensions in Ecology with R*. Springer New York; 2009. doi:10.1007/978-0-387-87458-6

8. Gibbons CL, Mangen MJJ, Plass D, et al. Measuring underreporting and under-ascertainment in infectious disease datasets: a comparison of methods. *BMC Public Health*. 2014;14(1):147. doi:10.1186/1471-2458-14-147