

Refining Historical Limits Method to Improve Disease Cluster Detection, New York City, New York, USA

Alison Levin-Rector,¹ Elisha L. Wilson,² Annie D. Fine, Sharon K. Greene

Since the early 2000s, the Bureau of Communicable Disease of the New York City Department of Health and Mental Hygiene has analyzed reportable infectious disease data weekly by using the historical limits method to detect unusual clusters that could represent outbreaks. This method typically produced too many signals for each to be investigated with available resources while possibly failing to signal during true disease outbreaks. We made method refinements that improved the consistency of case inclusion criteria and accounted for data lags and trends and aberrations in historical data. During a 12-week period in 2013, we prospectively assessed these refinements using actual surveillance data. The refined method yielded 74 signals, a 45% decrease from what the original method would have produced. Fewer and less biased signals included a true citywide increase in legionellosis and a localized campylobacteriosis cluster subsequently linked to live-poultry markets. Future evaluations using simulated data could complement this descriptive assessment.

Detecting aberrant clusters of reportable infectious disease quickly and accurately enough for meaningful action is a central goal of public health institutions (1–3). Clinicians' reports of suspected clusters of illness remain critical for surveillance (4), but the application of automated statistical techniques to detect possible outbreaks that might otherwise not be recognized has become more common (5). These techniques are particularly important in jurisdictions that serve large populations and receive a high volume of reports because manual review and investigation of all reports are not feasible.

Challenges such as lags in reporting and case classification and discontinuities in surveillance case definitions, reporting practices, and diagnostic methods are common across jurisdictions. These factors can impede the timely detection of disease clusters. Statistically and computationally simple methods, including historical limits (6), a log-linear regression model (7), and cumulative sums (8), each have strengths and weaknesses for prospective cluster

detection, but none adequately address these common data challenges. As technology advances, statistically and computationally intensive methods have been developed (2,3,5,9–12), and although these methods might successfully correct for biases, many lack the ease of implementation and interpretation desired by health departments.

Since 1989, the US Centers for Disease Control and Prevention has applied the historical limits method (HLM) to disease counts and displayed the results in Figure 1 of the Notifiable Diseases and Mortality Tables in the Morbidity and Mortality Weekly Report (13). Because the method relies on a straightforward comparison of the number of reported cases in the current 4-week period with comparable historical data from the preceding 5 years, its major strengths include simplicity, interpretability, and implicit accounting for seasonal disease patterns. These strengths make it a potentially very useful aberration-detection method for health departments (12,14–18). The Bureau of Communicable Disease (BCD) of the New York City (NYC) Department of Health and Mental Hygiene (DOHMH) implemented the HLM in the early 2000s (HLM_{original}) as a weekly analysis for all reportable diseases for which at least 5 years of historical data were available.

In HLM_{original}, 4 major causes of bias existed: 1) inconsistent case inclusion criteria between current and historical data; 2) lack of adjustment in historical data for gradual trends; 3) lack of adjustment in historical data for disease clusters or aberrations; and 4) no consideration of reporting delays and lags in data accrual. Our objectives were to develop refinements to the HLM (HLM_{refined}) that preserved the simplicity of the method's output and improved its validity and to characterize the performance of the refined method using actual reportable disease surveillance data. Although we describe the specific process for refining BCD's aberration-detection method, the issues presented are common across jurisdictions, and the principles and results are likely to be generalizable.

Author affiliation: New York City Department of Health and Mental Hygiene, Queens, New York, USA

DOI: <http://dx.doi.org/10.3201/eid2102.140098>

¹Current affiliation: RTI International, Research Triangle Park, North Carolina, USA.

²Current affiliation: Colorado Department of Public Health and Environment, Denver, Colorado, USA.

Methods

Overview of Disease Monitoring at BCD

BCD monitors ≈ 70 communicable diseases among NYC's 8.3 million residents (19). For passive surveillance, laboratories and providers are required to submit disease reports (20), and these reports flow into a database system (Maven, Consilience Software, Austin, TX, USA). Each case is classified into 1 of 12 case statuses (Table 1). Depending on the disease, cases initially might be assigned a transient pending status and, upon investigation, be reclassified as a case (confirmed, probable, or suspected) or "not a case." For each disease, a designated disease reviewer is responsible for reviewing cases.

HLM Overview

HLM compares the number of reported cases diagnosed in the past 4 weeks (X_0) with the number diagnosed within 15 prior periods (X_{1-15}) comprising the same 4-week period, the preceding 4-week period, and the subsequent 4-week period during the past 5 years (Figure 1). A 4-week temporal unit of analysis balances timeliness with stability (6,21). For any given disease, if the ratio of current counts to the mean of the fifteen 4-week totals is greater than historical limits, then the current period is considered aberrant (i.e., a signal is generated) (online Technical Appendix (<http://wwwnc.cdc.gov/EID/article/21/2/14-0098-Techapp1.pdf>)). In applying this method in NYC, only increases in case counts >2 SD above the historical mean are considered because artifactual decreases in case counts would be detected by separate quality-control measures.

HLM_{original} was run each Monday for the 4-week interval that included cases diagnosed through the most recent Saturday. Data on confirmed, probable, suspected, or pending cases (Table 1) were analyzed at 3 geographic

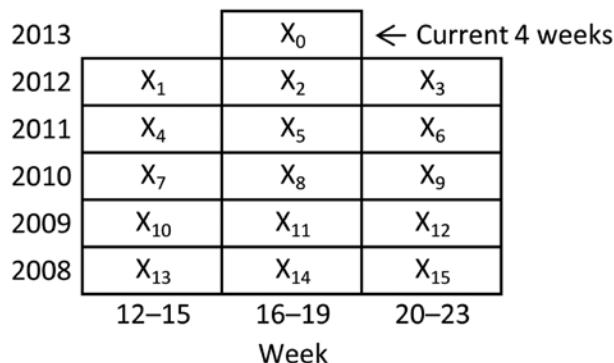


Figure 1. Following Stroup et al. (21), a schematic of the periods included in analyses using the historical limits method.

resolutions: citywide, borough (5 boroughs), and United Hospital Fund (UHF) neighborhood (42 neighborhoods). UHF neighborhoods are aggregations of contiguous ZIP codes used to define communities (22). Data were analyzed at the 2 subcity geographic resolutions to improve the signal-to-noise ratio for spatial clusters. For a signal to be generated, the current period was required to contain at least 3 cases, and the ratio of cases to the historical mean was required to be greater than historical limits. Disease reviewers were promptly notified of any signals and were provided with a corresponding case line list.

Refinements to Address Biases

Bias 1: Inconsistent Case Inclusion Criteria

The first limitation of HLM_{original} as applied in NYC was that case inclusion criteria caused current disease counts to be systematically higher than baseline disease counts for many diseases. Cases classified as confirmed, probable, suspected, or pending were analyzed, but some cases with an initial pending status were ultimately reclassified after investigation as "not a case." This reclassification process was complete for historical periods but ongoing for the current period.

The proportion of initially pending cases that were reclassified to confirmed, probable, or suspected (rather than "not a case") varied widely by disease (Figure 2). For diseases for which this confirmatory proportion was low, the disease counts in the current period included a high proportion of pending cases that would ultimately be reclassified as "not a case," leading to false signals (type I errors). A similar bias might apply for nationally notifiable data in that provisional and final case counts may be systematically different (23).

Refinement 1: Consistent Case Inclusion Criteria

HLM_{refined} included almost all reported cases in the analysis regardless of current status (Table 1). This simple modification led to a more valid comparison of total reporting

Table 1. Case statuses in current and baseline periods included in HLM_{original} and HLM_{refined}, New York City, New York, USA*

Case status	Included in HLM _{original}	Included in HLM _{refined}
Confirmed	Yes	Yes
Probable	Yes	Yes
Suspected	Yes	Yes
Pending†	Yes	Yes
Unresolved	No	Yes
"Not a case"	No	Yes
Chronic carrier	No	Yes
Asymptomatic infection	No	Yes
Seroconversion 1 y	No	Yes
Not applicable	No	Yes
Contact	No	No
Possible exposure	No	No

*HLM, historical limits method; HLM_{original}, method as originally applied in New York City before May 20, 2013; HLM_{refined}, refined method applied starting May 20, 2013.

†Pending is a transient status that in the normal course of case investigations can be assigned to a case in the current period but not in the baseline period.

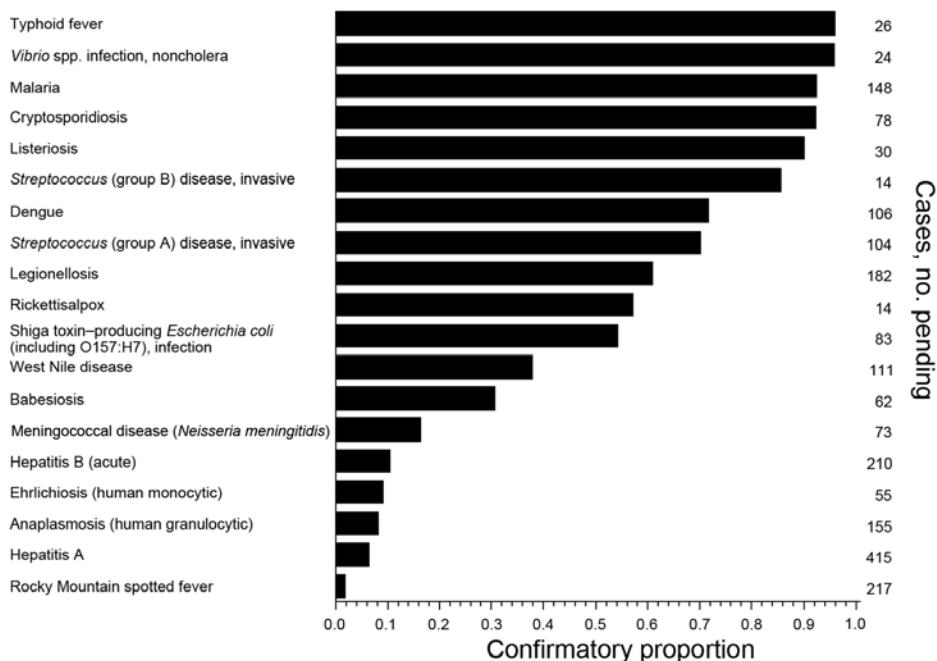


Figure 2. Confirmatory proportion of pending cases for diseases with any pending cases, New York City, New York, USA, July–December 2012. The confirmatory proportion was defined as the proportion of initially pending cases that were reclassified to confirmed, probable, or suspected (rather than to “not a case”). Diseases that are not routinely investigated, e.g., campylobacteriosis, enter the database with confirmed (not pending) case status and are not shown.

volume between current and historical periods, assuming that reporting is consistent over time, rather than biased estimates of the true level of disease. We maintained the requirement of the presence of at least 3 confirmed, probable, suspected, or pending cases to be considered a signal to prevent alerts driven by cases classified as “not a case.”

Bias 2: Gradual Trends in Historical Data

The second limitation of HLM_{original} was the existence of increasing or decreasing trends over time in historical data for many diseases. Whether these trends are true changes in disease incidence or artifacts of changing reporting or diagnostic practices, anything that causes disease counts in the baseline period to be systematically higher than current disease counts increases type II errors, and anything that causes baseline disease counts to be systematically lower than current disease counts increases type I errors.

Refinement 2: Adjusted Historical Data to Remove Gradual Trends

For HLM_{refined}, we identified and removed any significant linear trend in historical data. We accomplished this refinement by running a linear regression on weekly case counts for each disease at each geographic resolution and refitting the resulting residuals to a trend line with a slope of 0 and an intercept set to the most recent fitted value. Across diseases, linear trends were of relatively small magnitude; the greatest was for *Campylobacter*, for which the slope increased by ≈ 0.25 cases per week (Figure 3).

To minimize the influence of outliers on the overall trend, we excluded weekly counts >4 SD above or below

the average for the baseline period from the regression. However, these counts were added back after the model had been fitted.

Bias 3: Inclusion of Past Clusters in Historical Data

The third major bias in HLM_{original} was the inclusion of past clusters or aberrations in historical data. This bias reduced the method’s ability to detect aberrations going forward, which increased type II errors.

Refinement 3: Exclusion of Past Clusters from Historical Data

To prevent this bias, after adjusting for gradual trends, we considered any 4-week period in which disease counts were >4 SD above the average to be an outlier and reset the count to the average number of cases in the remaining historical instances of that 4-week period. (We selected the threshold of 4 SD after manually reviewing case counts over time for all diseases.) For example, during 2007–2011, the number of dengue fever cases diagnosed during weeks 35–38 in 2010 was >4 SD above the average number of cases during those 5 years. Consequently, that 4-week period in 2010 was considered an outlier and reset to the average dengue fever count in weeks 35–38 in 2007, 2008, 2009, and 2011 (Figure 4). This technique can cause the case counts over time to appear jagged, but because our objective was to ensure a valid comparison between historical and current data, the smoothness of trends over time is irrelevant.

Bias 4: Delays in Data Accrual

Finally, data accrual delays can contribute to type II errors. This method is applied on Mondays for the 4-week period

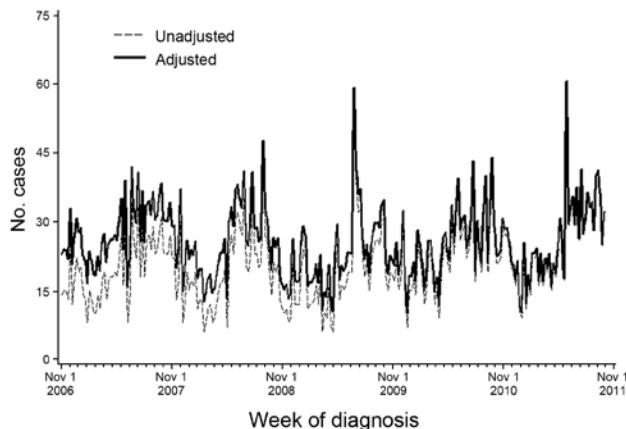


Figure 3. Unadjusted and adjusted weekly citywide counts of campylobacteriosis cases to illustrate adjustment for a linear trend in historical data, New York City, New York, USA, November 2006–October 2011.

that includes cases diagnosed through the most recent Saturday, so any lag between diagnosis and receipt by BCD of >2 days has the potential to deflate disease counts in the current period and reduce signal sensitivity. During July 18, 2012–August 28, 2013, the median lag between diagnosis and receipt by BCD was 5 days (range in median lag by disease 0–24 days).

Although DOHMH works with laboratories and providers to improve reporting practices, substantial reporting lags will continue for some diseases because of practices related to testing (e.g., time required for culturing and identifying *Salmonella* from a clinical sample) and surveillance (e.g., for some diseases, reports are held for delivery to the surveillance database until both a positive screening test and a confirmatory test are reported).

Refinement 4: Repeated Analyses to Accommodate Delays in Data Accrual

For diseases for which a delay of ≥ 1 week is not too long for a signal to be of public health value, we repeated the analysis for a given 4-week period over 4 consecutive weeks to allow for data accrual, thus improving signal sensitivity. In other words, we first analyzed cases diagnosed during a 4-week period on the following Monday. Updated data for the same 4-week period were re-analyzed on the subsequent 4 Mondays as data accrued to identify any signals that were initially missed because of incomplete case counts.

Customization by Disease

In HLM_{original} we conducted the same analysis for all diseases under surveillance, despite very different disease agents and epidemiologic profiles. We solicited comments from disease reviewers to ensure that the method was being applied meaningfully to all diseases and received

feedback that HLM_{original} produced an unmanageable number of signals, which led to their dismissal without investigation. We also suspect that on some occasions HLM_{original} did not detect true clusters because trends in disease counts decreased over the baseline period or because historical outbreaks masked new clusters. We responded by allowing for disease-specific analytic modifications, which included reducing the number of diseases monitored using this method, allowing for customized signaling thresholds, and accounting for sudden changes in reporting (Table 2).

We reduced the ≈ 70 diseases to which HLM_{original} had been applied to the 35 for which prospective and timely identification of clusters might result in public health action. For example, clusters of leprosy or Creutzfeldt-Jakob disease diagnoses within a 4-week period would not be informative because these diseases have long incubation periods, measured in years. We also excluded diseases that occur very infrequently or are nonexistent (defined as having an annual mean of <4 cases during 2008–2012). For example, we excluded tularemia and human rabies because any clusters of these diseases would be detected without automated analyses and because the underlying normality assumption of the method is violated for rare events.

Signals were most common at the neighborhood geographic level because of the increased noise resulting from small counts. Therefore, we also provided the option to reviewers to require >3 confirmed, probable, suspected, or pending cases to qualify as a signal at this geographic resolution.

Evaluation of HLM_{refined}

BCD implemented HLM_{refined} on May 20, 2013, including automatically generating reports for disease reviewers to summarize information about cases included in signals (online Technical Appendix). To determine the effects of the

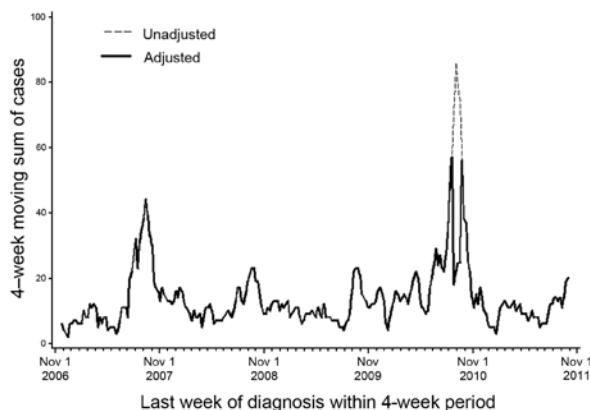


Figure 4. Unadjusted and adjusted 4-week moving sum of citywide dengue fever cases to illustrate adjustment for outliers in historical data, New York City, New York, USA, November 2006–October 2011.

above refinements, we compared signals detected during the 12 weeks after implementation with those that would have been detected had HLM_{original} still been in place. A signal was defined as any set of consecutive 4-week periods, permitting 1-week gaps, where the disease counts were above historical limits for either HLM_{original} or HLM_{refined}. Signals that were repeated in the same geographic area over multiple consecutive weeks were counted only once. Restricting analysis to a common set of 35 diseases (Table 2), we quantified the number of signals, determined the cause of any differences in signals between HLM_{original} and HLM_{refined}, and monitored the outcome of any public health investigations triggered by automated signals.

We describe our experience with these methods in a government setting to support applied public health practice.

In this setting, a complete list of true disease clusters and the resources to thoroughly investigate every statistical signal do not exist. We instead defined the set of true disease clusters as those identified using either method that could not be explained by any known systematic bias. We calculated type I and type II error rates using this set. Although artificial surveillance data generated through simulations have been created (24,25), those existing data do not reflect the dynamism and variability in actual reportable disease surveillance data, such as pending case reclassification (bias 1) and data accrual lags (bias 4). Accounting for this dynamism is essential for a valid comparison of HLM_{original} and HLM_{refined}. Thus, we chose a practical and descriptive approach to evaluating these methods rather than a quantitative simulation study.

Table 2. Diseases included in analyses using HLM_{refined} and details of customizations, New York City, New York, USA, May 20–August 5, 2013*

Disease	Minimum no. cases in UHF neighborhood to qualify for signal	Further customization
Amebiasis	5	
Anaplasmosis (human granulocytic)	3	
Babesiosis	3	
Campylobacteriosis	8	
Cholera	3	
Cryptosporidiosis	5	
Cyclosporiasis	3	
Dengue	3	
Ehrlichiosis (human monocytic)	3	
Giardiasis	5	
<i>Haemophilus influenzae</i> disease, invasive	3	
Hemolytic uremic syndrome	3	
Hepatitis A	5	
Hepatitis B (acute)	2†	
Hepatitis D	2†	
Hepatitis E	2†	
Legionellosis	5	
Listeriosis	3	
Malaria	3	
Meningitis, bacterial	4	
Meningitis, viral (aseptic)	3	
Meningococcal disease (<i>Neisseria meningitidis</i>)	3	
Paratyphoid fever	3	
Rickettsialpox	3	
Rocky Mountain spotted fever	3	Restrict analysis to confirmed, probable, and suspected cases and implement a 4-wk lag to allow for data accrual
Shiga toxin–producing <i>Escherichia coli</i> (including <i>E. coli</i> O157:H7) infection	3	
Shigellosis	10	
<i>Staphylococcus aureus</i> infection, vancomycin intermediate	3	
<i>Streptococcus</i> (group A) disease, invasive	5	Restrict analysis to confirmed, probable, suspected, and pending cases
<i>Streptococcus</i> (group B) disease, invasive	5	
<i>Streptococcus pneumoniae</i> disease, invasive	5	
Typhoid fever	3	
<i>Vibrio</i> spp. infection, noncholera (including <i>parahaemolyticus</i> and <i>vulnificus</i>)	3	
West Nile disease	3	
Yersiniosis	3	

* HLM_{refined}, refined method applied starting May 20, 2013; UHF, United Hospital Fund.

†These are the only diseases for which the signaling threshold was decreased below 3 cases.

Results

In the first 12 weekly analyses, HLM_{original} would have produced 134 signals, and HLM_{refined} produced 74 signals, a 45% decrease (Table 3). Of the HLM_{original} signals during this period, 47 (35%) would have been at the neighborhood geographic resolution with fewer cases than the reviewers' threshold for action; these signals were omitted from further evaluation. Of the remaining 107 signals across both methods, 54 (50%) were detected by both methods, 33 (31%) only by HLM_{original}, and 20 (19%) only by HLM_{refined}.

We classified each signal into 1 of 3 categories (Table 4): attributable to an uncorrected bias toward signaling, attributable to the correction of a bias against signaling, or not attributable to any known systematic bias. Of the signals detected by HLM_{original}, 2 campylobacteriosis signals and 1 invasive *Haemophilus influenzae* disease signal were attributable to a bias toward signaling caused by an increasing trend in historical data. HLM_{refined} missed 9 signals that were detected only by HLM_{original} because the confirmatory proportion was larger in current data than in historical data.

Two signals detected by HLM_{refined} were attributable to the removal of outliers from historical data; a legionellosis increase in the Bronx was masked by a prior increase in comparable weeks in 2009, and an amebiasis signal in a neighborhood was masked by a prior increase in comparable weeks in 2012. One signal detected by HLM_{refined} was attributable to the adjustment of a decreasing trend in baseline disease counts of viral meningitis. Seventeen signals detected only by HLM_{refined} were attributable to accounting for lags in data accrual (10 signals were first detectable after 1-week lag, 4 signals after 2 weeks, 2 signals after 3 weeks, and 1 signal after 4 weeks).

Overall, we identified 83 true clusters that could not be explained by any known systematic bias (i.e., 54 clusters identified by both HLM_{original} and HLM_{refined} and 29 clusters detected by only 1 of the methods and attributable to the correction of a bias against signaling). During the evaluation period, the percentage of all signals that did not correspond to these true clusters (type I error rate) for HLM_{original} was 28% (24 of 87 signals) and, for HLM_{refined}, 0% (0 of 74 signals). The percentage of all true clusters that were not detected (type II error rate) for HLM_{original} was 24% (20 of 83 true clusters) and, for HLM_{refined}, 11% (9 of 83 true clusters).

During these 12 weeks, 2 disease clusters occurred that we would have expected to detect using HLM. The first cluster of interest was a citywide increase in legionellosis in June 2013 (26). HLM_{refined} first detected this increase with a cluster in Queens on June 24, 2013. The next week, both HLM_{refined} and HLM_{original} detected the citywide increase. Although HLM_{refined} and HLM_{original} might detect similar disease clusters at slightly different times because

of differences in event inclusion criteria, the refinements do not directly affect timeliness.

On June 24, 2013, HLM_{original} would have generated 16 automated signals (including 3 for campylobacteriosis), and HLM_{refined} generated 5 signals (including 1 for campylobacteriosis); both methods detected a cluster of 11 campylobacteriosis cases in 1 neighborhood. After investigation, 8 of the cases were determined to be among children 0–5 years of age from Mandarin- or Cantonese-speaking families, 5 of whom had direct links to 1 of 2 local live-poultry markets. Consequently, pediatricians were educated about the association between live-poultry markets and campylobacteriosis, and health education materials about proper poultry preparation and hygiene were distributed to live-poultry markets.

Discussion

In refining the HLM to correct for major biases, we improved the ability to prospectively detect clusters of reportable infectious disease in NYC while preserving the simplicity of the output. Specifically, we addressed data challenges that are common to many jurisdictions, including improving consistency of case inclusion criteria, accounting for gradual trends and aberrations in historical data, and accounting for reporting delays.

HLM_{refined} found fewer signals overall than HLM_{original}, which, in practice, is perhaps the greatest improvement. Disease reviewers had become accustomed to a large number of signals that did not represent true outbreaks, which led to dismissal of many signals without investigation. Fewer, higher quality signals produced by HLM_{refined}, supported by improvements in the ad hoc type I and type II error rates, led to more careful inspection and a higher probability of identifying true clusters, e.g., the true campylobacteriosis cluster in a Brooklyn neighborhood.

Although we consider HLM_{refined} to be a substantial improvement upon HLM_{original}, we are aware that some limitations exist. In expanding case inclusion criteria to encompass all reports, we corrected a large bias but might have introduced a small bias. Because HLM_{refined} considers the overall volume of reported cases, the implicit assumption is that the confirmatory proportion is constant over time outside of seasonal patterns. If this assumption is violated, and the confirmatory proportion differs between historical and current data, HLM_{refined} can be biased. This bias is the reason that 9 signals detected by HLM_{original} were not also detected by HLM_{refined} during the evaluation period. Because these 9 signals might reflect disease clusters that would have been missed because of changes in the confirmatory proportion over time, we recommend implementing a lagged analysis that is restricted to confirmed, probable, and suspected cases. The signals produced by this lagged analysis can then be compared with signals produced in near real-time using all case statuses, and thus whether HLM_{refined} systematically

Table 3. Geographic resolution of signals produced by HLM_{original} and HLM_{refined} in 12 weekly analyses, New York City, New York, USA, May 20–August 5, 2013*

Geographic area	≥3 Cases required for signal: No. signals produced by HLM _{original}	No. signals produced by HLM _{original} †	No. signals produced by HLM _{refined} †
City	14	14	8
Borough	40	40	26
UHF neighborhood	80	33	40
Total	134	87	74

*HLM, historical limits method; HLM_{original}, method as originally applied in NYC prior to May 20, 2013; HLM_{refined}, refined method applied starting May 20, 2013; UHF, United Hospital Fund.
†At neighborhood level, reviewer could require >3 cases for signal.

fails to detect clusters can be assessed. Implementing this approach post hoc yielded 2 additional clusters that both HLM_{refined} and HLM_{original} missed. Also, as with any method that defines geographic location according to patient residence, HLM_{refined} can miss point source outbreaks when exposure occurs outside the residential area.

Next steps include addressing the arbitrary temporal and geographic units of analysis. HLM_{refined} is optimized to detect clusters of 4-week duration at citywide, borough, or neighborhood geographic resolution. This method is likely to fail to detect clusters of shorter or longer duration, at sub-neighborhood geographic resolution, and in locations that span borough or neighborhood borders. In February 2014, we began applying the prospective space–time permutation scan statistic so we could use flexible spatial and temporal windows (27). We plan to expand the application of HLM_{refined} to disease subspecies and serogroups within diseases (e.g., for salmonellosis) as this information becomes available in BCD's database system.

Health departments that receive a high volume of reports might consider adopting a method similar to HLM_{refined} to improve prospective outbreak detection and contribute to timely health interventions. Simulation studies using complex artificial data that adequately reflect the dynamic nature of real-time surveillance data across a wide range of reportable diseases with variable trends over time and historical outbreaks would be valuable.

Acknowledgments

We thank the members of the analytic team who work to detect disease clusters each week, including Ana Maria Fireteanu, Deborah Kapell, and Stanley Wang. We also thank Nimi Kadar who contributed substantially to the original SAS code for this method.

A.L.R., E.L.W., and S.K.G. were supported by the Public Health Emergency Preparedness Cooperative Agreement (grant 5U90TP221298-08) from the Centers for Disease Control and Prevention. A.D.F. was supported by New York City tax levy funds. The authors declare no conflict of interest.

Ms. Levin-Rector is a public health analyst within the Center for Justice, Safety and Resilience at RTI International. Her primary research interests are developing or improving upon existing statistical methods for analyzing public health data.

References

- Hutwagner L, Thompson W, Seaman GM, Treadwell T. The bioterrorism preparedness and response Early Aberration Reporting System (EARS). *J Urban Health*. 2003;80(Suppl 1):i89–96.
- Farrington P, Andrews N. Outbreak detection: application to infectious disease surveillance. In: Brookmeyer R, Stroup DF, editors. *Monitoring the health of populations*. New York: Oxford University Press; 2004. p. 203–31.
- Choi BY, Kim H, Go UY, Jeong J-H, Lee JW. Comparison of various statistical methods for detecting disease outbreaks. *Comput Stat*. 2010;25:603–17. <http://dx.doi.org/10.1007/s00180-010-0191-7>
- Schuman SH. When the community is the “patient”: clusters of illness. *environmental epidemiology for the busy clinician*. London: Taylor & Francis; 1997.

Table 4. Explanation of signals produced by HLM_{original} and HLM_{refined} in the 12 weekly analyses, New York City, New York, USA, May 20–August 5, 2013*

Explanation	No. signals produced by HLM _{original}	No. signals produced by HLM _{refined}
Attributable to an uncorrected bias toward signaling		
Neighborhood disease count threshold too low	47†	0
Pending cases in current period	21	0
Increasing trends in baseline period	3	0
Total signals attributable to an uncorrected bias toward signaling	71	0
Attributable to the correction of a bias against signaling		
Confirmatory proportion higher in current period than in baseline period	9	0
Accounted for data accrual lags	0	17
Deleted outliers in baseline period	0	2
Adjusted for decreasing trends in baseline period	0	1
Total signals attributable to the correction of a bias against signaling	9	20
Not attributable to any known systematic bias	54	54
Total signals	134	74

*HLM, historical limits method; HLM_{original}, method as originally applied in NYC prior to May 20, 2013; HLM_{refined}, refined method applied starting May 20, 2013.

†These were excluded from the calculation of type I and type II error rates.

5. Unkel S, Farrington CP, Garthwaite PH. Statistical methods for the prospective detection of infectious disease outbreaks: a review. *J R Stat Soc Ser A Stat Soc.* 2012;175:49–82. <http://dx.doi.org/10.1111/j.1467-985X.2011.00714.x>
6. Stroup DF, Williamson GD, Herndon JL, Karon JM. Detection of aberrations in the occurrence of notifiable diseases surveillance data. *Stat Med.* 1989;8:323–9. <http://dx.doi.org/10.1002/sim.4780080312>
7. Farrington CP, Andrews NJ, Beale D, Catchpole MA. A statistical algorithm for the early detection of outbreaks of infectious disease. *J R Stat Soc Ser A Stat Soc.* 1996;159:547–63. <http://dx.doi.org/10.2307/2983331>
8. Hutwagner LC, Maloney EK, Bean NH, Slutsker L, Martin SM. Using laboratory-based surveillance data for prevention: an algorithm for detecting *Salmonella* outbreaks. *Emerg Infect Dis.* 1997;3:395–400.
9. Strat YL. Overview of temporal surveillance. In: Lawson AB, Kleinman K, editors. *Spatial and syndromic surveillance for public health.* Chichester (UK): John Wiley & Sons; 2005. p. 13–29.
10. Serfling RE. Methods for current statistical analysis of excess pneumonia-influenza deaths. *Public Health Rep.* 1963;78:494–506. <http://dx.doi.org/10.2307/4591848>
11. Noufaily A, Enki DG, Farrington P, Garthwaite P, Andrews N, Charlett A. An improved algorithm for outbreak detection in multiple surveillance systems. *Stat Med.* 2013;32:1206–22. <http://dx.doi.org/10.1002/sim.5595>
12. Wharton M, Price W, Hoesly F, Woolard D, White K, Greene C, et al. Evaluation of a method for detecting outbreaks of diseases in six states. *Am J Prev Med.* 1993;9:45–9.
13. Centers for Disease Control and Prevention. Proposed changes in format for presentation of notifiable disease report data. *MMWR Morb Mortal Wkly Rep.* 1989;38:805–9.
14. Centers for Disease Control and Prevention. Notes from the field: *Yersinia enterocolitica* infections associated with pasteurized milk—southwestern Pennsylvania, March–August, 2011. *MMWR Morb Mortal Wkly Rep.* 2011;60:1428.
15. Rigau-Pérez JG, Millard PS, Walker DR, Deseda CC, Casta-Velez A. A deviation bar chart for detecting dengue outbreaks in Puerto Rico. *Am J Public Health.* 1999;89:374–8. <http://dx.doi.org/10.2105/AJPH.89.3.374>
16. Pervaiz F, Pervaiz M, Abdur Rehman N, Saif U. FluBreaks: early epidemic detection from Google flu trends. *J Med Internet Res.* 2012;14:e125. <http://dx.doi.org/10.2196/jmir.2102>
17. Winscott M, Betancourt A, Ereth R. The use of historical limits method of outbreak surveillance to retrospectively detect a syphilis outbreak among American Indians in Arizona. *Sex Transm Infect.* 2011;87:A165. <http://dx.doi.org/10.1136/sextrans-2011-050108.155>
18. Hutwagner L, Browne T, Seeman GM, Fleischauer AT. Comparing aberration detection methods with simulated data. *Emerg Infect Dis.* 2005;11:314–6. <http://dx.doi.org/10.3201/eid1102.040587>
19. New York City Department of Health and Mental Hygiene. Communicable disease surveillance data [cited 2013 Nov 15]. <http://www.nyc.gov/html/doh/html/data/cd-epiquery.shtml>
20. Nguyen TQ, Thorpe L, Makki HA, Mostashari F. Benefits and barriers to electronic laboratory results reporting for notifiable diseases: the New York City Department of Health and Mental Hygiene experience. *Am J Public Health.* 2007;97(Suppl 1):S142–5. <http://dx.doi.org/10.2105/AJPH.2006.098996>
21. Stroup DF, Wharton M, Kafadar K, Dean AG. Evaluation of a method for detecting aberrations in public health surveillance data. *Am J Epidemiol.* 1993;137:373–80.
22. United Hospital Fund. *Neighborhoods. New York City community health atlas: sources, methods and definitions.* New York: United Hospital Fund; 2002. p. 2–3.
23. Centers for Disease Control and Prevention. Comparison of provisional with final notifiable disease case counts—National Notifiable Diseases Surveillance System, 2009. *MMWR Morb Mortal Wkly Rep.* 2013;62:747–51.
24. Lotze T, Shmueli G, Yahav I. Simulating multivariate syndromic time series and outbreak signatures [cited 2014 Dec 3]. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=990020
25. Centers for Disease Control and Prevention. Simulation data sets for comparison of aberration detection methods. 2004 April 16, 2004 [cited 2013 Aug 30]. <http://www.bt.cdc.gov/surveillance/ears/datasets.asp>
26. Layton M. Increase in reported legionellosis cases. 2013 [cited 2013 Sep 18]. https://a816-health29ssl.nyc.gov/sites/NYCHAN/Lists/AlertUpdateAdvisoryDocuments/2013-07-03%20HAN_Legionella%20final2.pdf
27. Kulldorff M, Heffernan R, Hartman J, Assuncao R, Mostashari F. A space–time permutation scan statistic for disease outbreak detection. *PLoS Med.* 2005;2:e59. <http://dx.doi.org/10.1371/journal.pmed.0020059>

Address for correspondence: Alison Levin-Rector, New York City Department of Health and Mental Hygiene, 42-09 28th St, WS 6-145, Queens, NY 11101, USA; email: levinrec@gmail.com

Bat Flight and Zoonotic Viruses



Reginald Tucker
reads an abridged
version of the EID
perspective
**Bat Flight and
Zoonotic Viruses.**



<http://www2c.cdc.gov/podcasts/player.asp?f=8632573>

Refining Historical Limits Method to Improve Disease Cluster Detection, New York City, New York, USA

Technical Appendix

The Technical Appendix contains mathematical notation for the HLM, details on customized analyses implemented for 2 diseases with unique patterns of reporting and diagnosis over time, a technical note, sample output for summarizing and presenting signals, and sample SAS code (SAS v.9.2, SAS Institute, Cary, NC).

Mathematical Notation for HLM

$$\frac{X_0}{\mu} > 1 + 2 * \left(\frac{\sigma_x}{\mu} \right) \quad (1)$$

where X_0 is the current total of cases in the most recent four-week interval, and μ and σ_x are the mean and standard deviation, respectively, of the 15 historical four-week periods (X_{1-15}).

Customized Analyses for Two Diseases

For Group A *Streptococcus*, new filtering rules in our surveillance database system were applied in July 2012 to screen out reports of specimens collected from noninvasive sources, resulting in an abrupt decrease in the number of reported cases. Ignoring this change would have biased against signaling. Therefore, we continued to consider only cases with confirmed, probable, suspected, or pending statuses. Because the confirmatory proportion is high for this disease (73%, Figure 2), including pending cases does not strongly bias toward signaling.

For Rocky Mountain Spotted Fever, the total number of reported cases increased beginning in summer 2011, while there was no corresponding increase in the number of confirmed, probable, and suspected cases. Ignoring this pattern would have biased toward false signaling. We therefore monitor only confirmed, probable, and suspected cases at a lag of four

weeks to allow for near complete data accrual. Signal quality was prioritized over timeliness for this disease, which has no immediate public health intervention.

Technical Note

Since these adjustments of the baseline data require a time series in which to identify outliers, calculate averages, and run regressions, baselines will need to be prospectively updated at regular intervals. It is not necessary to recalibrate historical data on a weekly basis because the adjustments will not be very different from week to week. However, the interval at which recalibrations are made should be sufficiently short such that recent disease trends are taken into account. The interval must be shorter than one year because data less than a year in the past is considered as the most recent historical data point in the method (X_3 in Figure 1), and we want at least 12 weeks to have passed between the end of the baseline period and the date of recalibration to allow for sufficient data accrual. We chose this 12 week cutoff based on the fact that in 2010 and 2011, all relevant diseases had at least 70% data accrual at 12 weeks post diagnosis, and all but two diseases (encephalitis and human granulocytic anaplasmosis) had at least 90% data accrual.

It is these considerations that led us to conclude that the baseline data should be recalibrated every 26 weeks (twice per year), e.g., on the 1st and 27th weeks of the year, and include historical data from the earliest week that is required for comparison by the method (X_{13} in Figure 1) through 12 weeks prior to the recalibration date. In other words, in the baseline period to be used for prospective surveillance during weeks 1 through 26 of year Y , the earliest week that is required is week 46 of year $Y-6$ (the 4-week period from week 46 through week 49 of year $Y-6$ constitutes time period X_{13} for week 1 of year Y). The latest week that is included is week 40 of year $Y-1$ (12 weeks prior to week 1 of year Y). Analogously, the baseline period for weeks 27 through 52 of year Y will include week 20 of year $Y-5$ through week 14 of year Y .

Sample SAS Code for Implementing Refinements 2 and 3

The following sample SAS code adjusts historical data to remove gradual trends and resets outliers that could indicate past clusters to the average number of cases in the remaining instances of that 4-week period in historical data.

Structure your historical dataset called `collapsed_events_city` in the following format. The variable “`fsatdiag`” refers to the Saturday ending the week of interest (i.e. 5/17/2008 refers to the week from Sunday, 5/11/2008, through Saturday, 5/17/2008, inclusive). Ensure that every Saturday is included in the dataset for each disease and geographic area, even if the number of events for that week is zero. If you are running an analysis at a smaller geographic area, include another variable to identify the disease count within each area.

Sample structure for input dataset, named “`collapsed_events_city`”

Disease_code	Disease	Events	fsatdiag
Dis1	Disease1	15	5/17/2008
Dis1	Disease1	4	5/24/2008
...
Dis2	Disease2	0	5/17/2008

```
*****;
*   PROGRAM NAME: Adjusting Baseline for HLM Refined
*   PROGRAMMER: Alison Levin-Rector
*****;

*-- assign libnames;
libname signals 'SPECIFY LOCATION TO SAVE ADJUSTED BASELINE DATA';

* 1. REMOVE GRADUAL TRENDS FROM HISTORICAL DATA;
* weekly event count is excluded from the regression model if it is more than
4 standard deviations greater or less than the mean of entire dataset, to
avoid biasing trend;
*-- define the baseline period depending on whether we are in the first or
second half of the calendar year;
data _null_;
    if week(date()) <= 26 then firstday = nwksdom(1,7,1,year(date())-
        6)+45*7;
    if week(date()) <= 26 then lastday = nwksdom(1,7,1,year(date())-1)+39*7;
    if week(date()) > 26 then firstday = nwksdom(1,7,1,year(date())-5)+19*7;
    if week(date()) > 26 then lastday = nwksdom(1,7,1,year(date()))+13*7;
    call symputx ('firstday_BL',firstday);
    call symputx ('lastday_BL',lastday);
run;

*-- exclude weeks with event counts that are more than 4 standard deviations
above or below the mean for that disease from the regression;
proc sql;
    create table collapsed_events_outliers as
select *, mean(events) + 4*std(events) as cutoff1, mean(events) -
    4*std(events) as cutoff2
    from collapsed_events_city
where fsatdiag <= &lastday_BL
    group by disease_code
    order by disease_code, fsatdiag;
quit;
```

```

data collapsed_events_outliers;
  set collapsed_events_outliers;
  if events <= cutoff1 & events >= cutoff2 then events_model = events;
run;
*-- run regression;
proc reg data = collapsed_events_outliers;
  by disease_code;
  model events_model = fsatdiag;
  output out = reg_output r = res p = pred;
  ods output ParameterEstimates = params;
run;
*-- save the number of events predicted by the linear regression at the most
recent week for each disease;
proc sql;
  create table newline as
  select disease_code, pred as new_intercept
  from reg_output
  having max(fsatdiag) = fsatdiag
  order by disease_code;
quit;
*-- save the p-values from slope term to determine whether the regression
found a significant trend for each disease;
proc sql;
  create table slopes as
  select disease_code, probt
  from params
  where strip(variable) = "fsatdiag"
  order by disease_code;
quit;
*-- if the trend had a significant slope, then adjust event counts so that
the trend over time is flat and if not, preserve the original event counts;
data adjusted_events_sig;
  merge reg_output slopes newline;
  by disease_code;
  res2 = events-pred;
  events_adj = new_intercept + res2;
  if probt > .05 | probt = . then events_adj = events;
run;

* 2. EXCLUDE PAST CLUSTERS FROM HISTORICAL DATA;
* This step is applied to 4-week periods rather than weekly counts, so first
we create a moving 4-week sum of events over our entire baseline period;
data adjusted_events_sig_outliers;
  set adjusted_events_sig;
  by disease_code;
  retain num_sum num_sum_adj 0;
  if first.disease_code then do;
    count=0;
    num_sum=0;
    num_sum_adj=0;
  end;
  count+1;
  last4=lag4(events);
  if count gt 4 then num_sum=sum(num_sum,events,-last4);
  else num_sum=sum(num_sum,events);
  last4adj=lag4(events_adj);
  if count gt 4 then num_sum_adj=sum(num_sum_adj,events_adj,-last4adj);

```

```

        else num_sum_adj=sum(num_sum_adj,events_adj);
        drop last4 last4adj;
run;
proc sql;
    create table adjusted_events_sig_outliers as
    select disease_code, disease, fsatdiag, num_sum, num_sum_adj,
           significant, mean(num_sum_adj) + 4*std(num_sum_adj) as cutoff
    from adjusted_events_sig_outliers
    where count >= 4
    group by disease_code
    order by disease_code, fsatdiag;
quit;
*-- require more than 2 events to be considered an outlier;
data adjusted_events_sig_outliers;
    set adjusted_events_sig_outliers;
    num_sum_outliers = num_sum_adj;
    if num_sum_adj >= cutoff & num_sum_adj > 2 then num_sum_adj = .;
    week = week(fsatdiag) + 1;
run;
*-- fill in dropped outliers with average of the same week from other years;
proc sql;
    create table averages as
    select disease_code, week, avg(num_sum_adj) as num_sum_adj_avg
    from adjusted_events_sig_outliers
    group by disease_code, week
    order by disease_code, week;
quit;
proc sort data = adjusted_events_sig_outliers; by disease_code week; run;
data adjusted_events_sig_fill;
    merge adjusted_events_sig_outliers averages;
    by disease_code week;
    if num_sum_adj = . then num_sum_adj = num_sum_adj_avg; /* this is where
        we fill them in*/
    drop week num_sum_adj_avg;
run;
*-- save the adjusted historical data to memory;
proc sort data = adjusted_events_sig_fill out =
signals.adjusted_baseline_city;
by disease_code fsatdiag;
run;

```

Sample Output for Summarizing and Presenting Signals

The following sample output is an example of the presentation of a signal for one disease and geographic resolution. SAS code used to produce this output is provided in the subsequent section. This output is automatically generated and placed in a secured folder. The location of this output is then sent by e-mail to the appropriate disease reviewer for each signal. Not

included in this sample output is a summary of all signals produced each week that is distributed to the entire Bureau of Communicable Disease.

Campylobacteriosis

UHF Signal in Neighborhood X

Disease	Unit of geography	Date of interest	Total dx past 4 weeks: 26MAY13 - 22JUN13	Signal Strength (# of SDs above mean)	new signal since last week?	if not new, how many new events in signal?	Rate per 100,000 in signal area past 4 weeks	Citywide rate per 100,000 past 4 weeks
Campylobacteriosis	Neighborhood X	Diagnosis date	11	5.70	yes	N/A	8.6	1.55

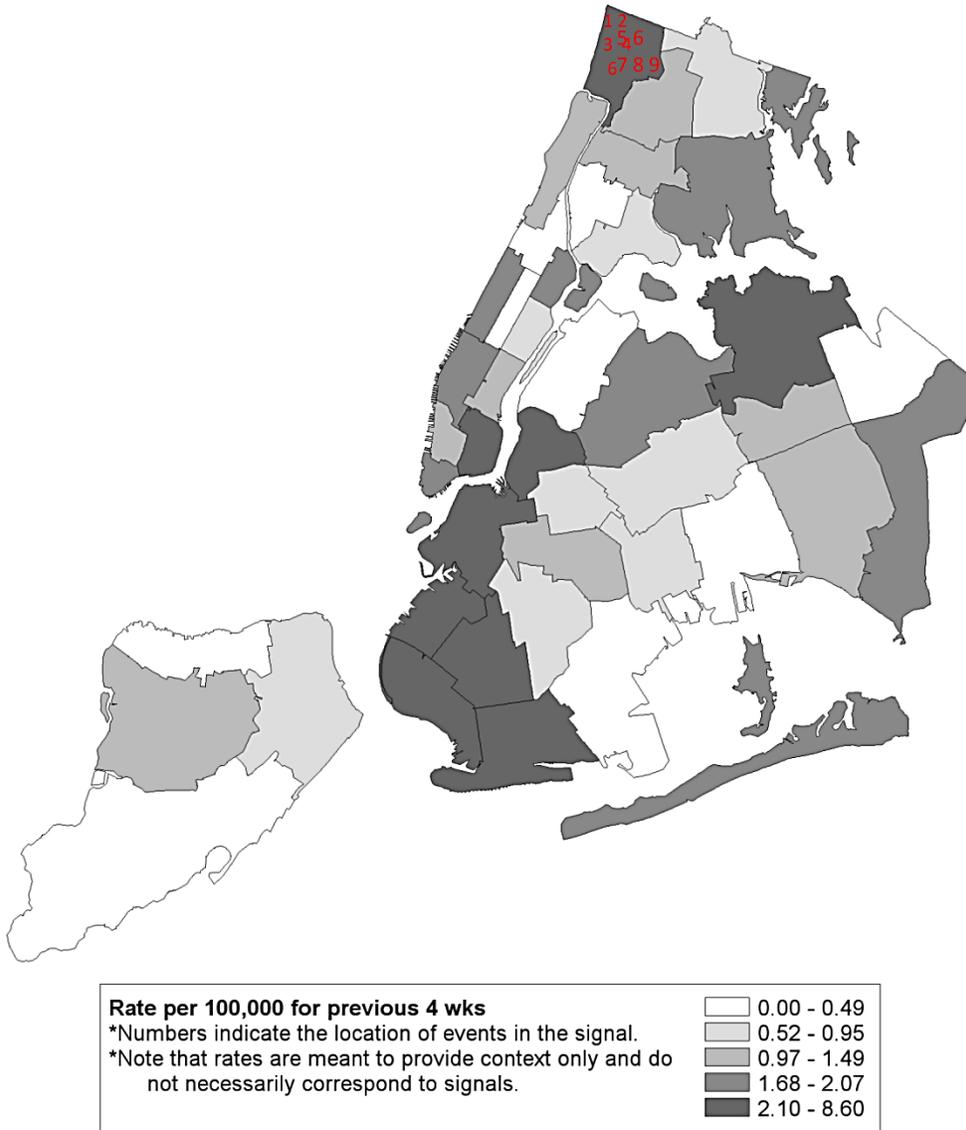
Total dx past 4 weeks includes only confirmed, probable, suspect and pending case statuses
 When # of SDs above mean > 2, current period is considered a signal
 A missing signal strength value indicates an SD = 0 in the baseline period

Year	Most recent week (Sun-Sat)	2 weeks ago	3 weeks ago	4 weeks ago	Total
2013 Pending	0	0	0	0	0
2013 Conf/Prob/Susp	4	2	3	2	11
2012 Conf/Prob/Susp	3	1	1	0	5
2011 Conf/Prob/Susp	1	1	1	0	3
2010 Conf/Prob/Susp	0	0	3	0	3
2009 Conf/Prob/Susp	1	0	0	1	2
2008 Conf/Prob/Susp	0	1	1	0	2

Unadjusted counts of cases by year, week and case status

Campylobacteriosis

UHF Signal in Neighborhood X

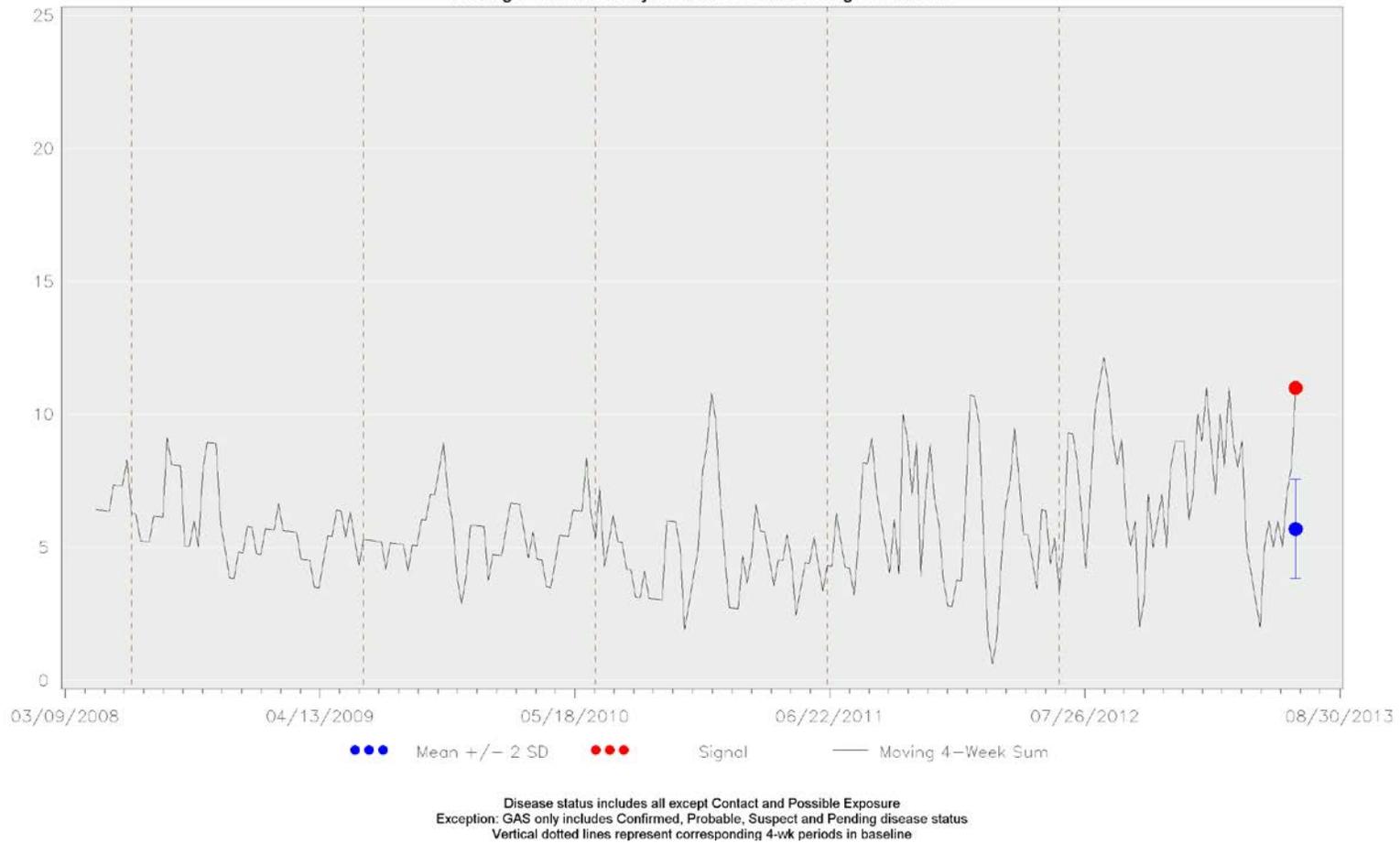


Technical Appendix Figure 1. Spatial distribution of the address at time of report for cases included in the signal and rates of disease by UHF neighborhood in the previous 4 weeks.

Case locations have been moved to a different neighborhood to protect confidentiality.

#	Event ID	Pat. init.	Disease status	Investigation status	Diagnosis date	Gender	Age	Address	Boro	Zip	UHF	Geocode
1	XXXXXX	XX	CONFIRMED	NOT_NEEDED	05/28/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	yes
2	XXXXXX	XX	CONFIRMED	NOT_NEEDED	05/29/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	yes
3	XXXXXX	XX	CONFIRMED	NOT_NEEDED	06/04/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	yes
4	XXXXXX	XX	CONFIRMED	NOT_NEEDED	06/05/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	yes
5	XXXXXX	XX	CONFIRMED	NOT_NEEDED	06/06/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	yes
6	XXXXXX	XX	CONFIRMED	NOT_NEEDED	06/12/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	yes
7	XXXXXX	XX	CONFIRMED	NOT_NEEDED	06/13/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	yes
8	XXXXXX	XX	CONFIRMED	NOT_NEEDED	06/17/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	yes
9	XXXXXX	XX	CONFIRMED	NOT_NEEDED	06/17/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	yes
10	XXXXXX	XX	CONFIRMED	NOT_NEEDED	06/18/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	no
11	XXXXXX	XX	CONFIRMED	NOT_NEEDED	06/21/2013	XXXXX	N/A	XXXXXX	XX	XXXXX	Neighborhood X	no

Identifying information is suppressed to protect confidentiality.



Technical Appendix Figure 2. Moving 4-week sum of adjusted case counts compared with historical mean \pm 2 SD.

Sample SAS Code for Summarizing and Presenting Signals

The following sample SAS code applies the HLM method and creates an output document with a summary of all signals as well as a detailed linelist report for each signal detected (see sample output above). Also included is code that automates the emailing of signal details to reviewers.

The input dataset should be event-level data in the following structure. The variable “fsatdiag” refers to the Saturday following the diagnosis date for each event. The following code is only for a citywide analysis. If you are running an analysis at multiple geographic resolutions, then include another variable to indicate the geographic unit. The “confirmatory” variable is an indicator variable for disease status that is set to 1 if the case is confirmed, probable, suspected, or pending and 0 if it is not. The input dataset should also include variables such as patient initials, disease status, diagnosis date, gender, and age for display in linelists, and X and Y coordinates for mapping.

Sample structure for input dataset, named “event_level_input”

Disease_code	Disease	Event_ID	fsatdiag	confirmatory
Dis1	Disease1	XXXXX1	5/17/2008	1
Dis1	Disease1	XXXXX2	5/24/2008	0
...
Dis2	Disease2	XXXXX4	5/17/2008	1

```
*****;  
*      PROGRAM NAME: Analysis and Output for HLM Refined  
*      PROGRAMMERS: Alison Levin-Rector  
*                      Elisha Wilson  
*                      Deborah Kapell  
*****;  
  
* create macro variables for the most recent Saturday in the dataset and  
today's date;  
proc sql noprint;  
    select max(fsatDiag)  
    into :maxFSATDiag  
    from event_level_input;  
quit;  
  
data _null_;  
call symput ('fileweek',put(today(),date9.));  
run;
```

```
***** CITYWIDE *****;
* pull in recent data (since the end of the baseline period);
proc sql;
    create table current_data as
    select disease_code, disease, fsatdiag, count(event_id) as events_adj,
           count(confirmatory) as confirmatory
    from event_level_input
    where fsatdiag > (&lastday_BL - 28)
    group by disease_code, disease, fsatdiag
    order by disease_code, disease, fsatdiag;
quit;
* merge current data with baseline data and categorize weeks into relevant
time periods for analysis;
data trends1;
    set signals.adjusted_baseline_city (rename=(num_sum_adj = events_adj))
        current_data;
    if fsatdiag >= (&maxfsatdiag-22) & fsatdiag <= &maxfsatdiag then
        period='current';
    if abs(fsatdiag - (&maxfsatdiag-365-28)) <= 3 then period='p1';
    if abs(fsatdiag - (&maxfsatdiag-365)) <= 3 then period='c1';
    if abs(fsatdiag - (&maxfsatdiag-365+28)) <= 3 then period='s1';
    if abs(fsatdiag - (&maxfsatdiag-365*2-28)) <= 3 then period='p2';
    if abs(fsatdiag - (&maxfsatdiag-365*2)) <= 3 then period='c2';
    if abs(fsatdiag - (&maxfsatdiag-365*2+28)) <= 3 then period='s2';
    if abs(fsatdiag - (&maxfsatdiag-365*3-28)) <= 3 then period='p3';
    if abs(fsatdiag - (&maxfsatdiag-365*3)) <= 3 then period='c3';
    if abs(fsatdiag - (&maxfsatdiag-365*3+28)) <= 3 then period='s3';
    if abs(fsatdiag - (&maxfsatdiag-365*4-28)) <= 3 then period='p4';
    if abs(fsatdiag - (&maxfsatdiag-365*4)) <= 3 then period='c4';
    if abs(fsatdiag - (&maxfsatdiag-365*4+28)) <= 3 then period='s4';
    if abs(fsatdiag - (&maxfsatdiag-365*5-28)) <= 3 then period='p5';
    if abs(fsatdiag - (&maxfsatdiag-365*5)) <= 3 then period='c5';
    if abs(fsatdiag - (&maxfsatdiag-365*5+28)) <= 3 then period='s5';
run;
*-- carry out HLM analysis at citywide level;
proc sql;
    create table City1 as
    select disease_code, disease, period, sum(events_adj) as count
    from trends1
    where period ^= ''
    group by disease_code, disease, period;
quit;
* count the number of confirmed/probable/suspect/pending cases;
proc sql;
    create table confirmatory1 as
    select disease_code, disease, sum(confirmatory) as confirmatory
    from trends1
    where period = "current"
    group by disease_code, disease, period;
quit;
proc transpose data=City1 out=City2;
    by disease_code disease;
    id period;
    var count;
run;
data City2;
```

```
merge City2 confirmatory1;
by disease_code disease;

run;
data City3;
set City2;
array xx current p1 c1 s1 p2 c2 s2 p3 c3 s3 p4 c4 s4 p5 c5 s5;
do over xx;
    if xx=. then xx=0;
end;
if current>0 then do;
    mean=mean(p1,c1,s1,p2,c2,s2,p3,c3,s3,p4,c4,s4,p5,c5,s5);
    sd= std(p1,c1,s1,p2,c2,s2,p3,c3,s3,p4,c4,s4,p5,c5,s5);
    if sd>0 then ratio= (current-mean)/sd;
    if current >=mean+2*(sd) then significant=1;
    else significant = 0;
end;
format mean 5.1;
format sd 5.2;
format ratio 5.2;
length geography $20.;
geography= 'City';
geoUnit='City';
metric='Diagnosis date';

run;

** The equivalent analysis above is carried out at all geographic resolutions
(in our case at Borough and UHF neighborhood);

* merge significant signals at all geographic resolutions;
data AllSignificant;
set City3 Boro3 UHF3;
/* only keep signals that are significant and that have at least 3
confirmed, probable, suspected, or pending events */
if (confirmatory>2 & significant=1);
fsatDiag=&maxfsatDiag;
rundate=today();
format fsatDiag rundate mmddyy10.;

run;
proc sort data=allsignificant;
by disease_code geography;

run;

* delete saved signals if the analysis is run multiple times on the same day;
data signals.trends_signals;
set signals.trends_signals;
where rundate ~= date();

run;
* create macro variable with the last time the analysis was run;
proc sql noprint;
select max(rundate,date9.)
into :lastweek
from signals.trends_signals;
quit;

* save signal details datasets;
proc append base=signals.trends_signals data=allsignificant;
```

```
run;

* compare this week's signals with last week's signals in order to flag new
signals;
proc sort data = signals.trends_signals out = lastweek (keep=disease_code
geography);
    where rundate = input("&lastweek",date9.);
    by disease_code geography;
run;
data Allsignificant_compare;
    merge allsignificant (in=a) lastweek (in=b);
    by disease_code geography;
    if a;
    if ~b then new = "*";
run;

*-- Output the signal summary document;
data health2;
confidential="Please do not Distribute";
run;
ods noresults;
ods rtf file= "...\&fileweek\Weekly trends &fileweek..doc";
title1 'TO: STAFF';
title2 'SUBJECT: WEEKLY TRENDS';
title3 '
';
title4 'As always, comments and feedback much appreciated.';
title5 '
';
title6 'Thanks !';
proc print data=health2 noobs;
var confidential;
run;

options orientation=landscape;
proc print data=allsignificant_compare noobs label ;
var disease geography confirmatory ratio new;
label geography ='unit of geography';
label confirmatory ='Total dx past 4 weeks';
label ratio = "Signal Strength (# of SDs above mean)";
label new='* indicates new signal since last week';
title1 "Trends Report based on diagnosis date";
title2 ;
title3 'The trends report compares the count of all cases (except contacts
and possible exposures)';
title4 'diagnosed in the past 4 weeks to the mean and standard deviation of
cases diagnosed';
title5 'during similar time periods in the past 5 years.';
title6 ;
title7 "This report was created
%sysfunc(left(%qsysfunc(date(),worddate18.)))";
footnote1 font='Arial' height=1 "Total dx past 4 weeks includes only
confirmed, probable, suspect and pending case statuses";
footnote2 font='Arial' height=1 "When # of SD's above mean > 2, current
period is considered a signal";
```

footnote3 font='Arial' height=1 "A missing signal strength value indicates an SD = 0 in the baseline period";

run;

```
*****;
* The following code preps data for graphs of signals
*****;

***** CITYWIDE *****;
* fill in missing Saturdays with zeroes for all diseases;
proc transpose data = current_data(rename=(events_adj=events)) out = trans;
  by disease_code;
  id fsatdiag;
  var events;
  format fsatdiag 8.;
run;
proc transpose data = trans out = trans2;
  by disease_code;
run;
data collapsed_events_fillin;
  set trans2;
  if events = . then events = 0;
  fsatdiag = input(substr(_name_,2),8.);
  format fsatdiag mmddy10.;
  drop _name_;
  if fsatdiag < date();
run;
proc sort data = collapsed_events_fillin; by disease_code fsatdiag; run;
* create a moving sum of events for the previous four weeks;
data events_4wk_moving_sum;
  set collapsed_events_fillin;
  by disease_code;
  retain num_sum 0;
  if first.disease_code then do;
    count=0;
    num_sum=0;
  end;
  count+1;
  last4=lag4(events);
  if count gt 4 then num_sum=sum(num_sum,events,-last4);
  else num_sum=sum(num_sum,events);
  drop count last4;
run;
* append to adjusted 4-wk sum of event counts;
data events_4wk_moving_sum;
set events_4wk_moving_sum(where=(fsatdiag>&lastday_BL) drop=events)
  signals.adjusted_baseline_city(rename=(num_sum_adj=num_sum)
  keep=disease_code fsatdiag num_sum_adj);
run;
proc sort data = events_4wk_moving_sum; by disease_code fsatdiag; run;
* pull signals for this week;
proc sort data = AllSignificant out = trends_signals;
  by disease_code fsatdiag;
  where geography = "City";
run;
```

```
* merge signal data with event counts;
data events_and_signals;
    merge trends_signals events_4wk_moving_sum;
    by disease_code fsatdiag;
run;
proc sort data = events_and_signals;
    by disease_code fsatdiag rundeate;
run;
* create a record for the current mean and low and high interval that the
current count is being compared to;
data reshape;
    set events_and_signals;
    where fsatdiag = &maxfsatdiag & mean ~= .;
    yvar = mean; num_sum = current; output;
    yvar = mean - sd; num_sum = .d; output;
    yvar = mean + sd; num_sum = .d; output;
run;
* output the rest of the dataset (minus the current information);
data therest;
    set events_and_signals;
    where (fsatdiag ~= &maxfsatdiag | mean = .) & fsatdiag >= (date()-
        365*5-60);
    yvar = .;
    mean = .;
    current = .;
run;
* append the current data to the rest of the data;
proc append base = reshape data = therest;
run;
data reshape_city;
    set reshape;
    if &maxfsatDiag-365-7 <= fsatdiag <= &maxfsatDiag-365 then
        period1=fsatdiag;
    if &maxfsatDiag-730-7 <= fsatdiag <= &maxfsatDiag-730 then
        period2=fsatdiag;
    if &maxfsatDiag-365*3-7 <= fsatdiag <= &maxfsatDiag-365*3 then
        period3=fsatdiag;
    if &maxfsatDiag-365*4-7 <= fsatdiag <= &maxfsatDiag-365*4 then
        period4=fsatdiag;
    if &maxfsatDiag-365*5-7 <= fsatdiag <= &maxfsatDiag-365*5 then
        period5=fsatdiag;
    geounit = "City";
    geog = "NYC";
run;
* sort for graphing;
proc sort data = reshape_city(keep=disease_code geounit geog fsatdiag num_sum
    current yvar mean period:);
    by disease_code fsatdiag;
run;

** The equivalent analysis above is carried out at all geographic resolutions
(in our case at Borough and UHF neighborhood);

* append all geographic levels;
data reshape_all;
    length geog $50.;
```

```
set reshape_city reshape_boro reshape_uhf;
run;

*****;
* The following code preps data for table 2 in the linelist reports;
*****;

***** CITYWIDE *****;
* pull in raw data going back 5 years;
proc sql;
    create table raw_data as
    select disease_code, disease, fsatdiag, count(event_id) as events
    from event_level_input
    where disease_status in ("CONFIRMED", "PROBABLE", "SUSPECT")
    group by disease_code, disease, fsatdiag
    order by disease_code, disease, fsatdiag;
quit;
* reshape to fill in all dates for all diseases;
proc transpose data = raw_data out = trans;
    by disease_code disease;
    id fsatdiag;
    var events;
    format fsatdiag 8.;
run;
proc transpose data = trans out = trans2;
    by disease_code disease;
run;
data raw_data_fillin;
    set trans2;
    if events = . then events = 0;
    fsatdiag = input(substr(_name_,2),8.);
    format fsatdiag mmddyy10.;
    drop _name_;
run;
proc sort data = raw_data_fillin; by disease_code disease fsatdiag; run;
* save current week's dates for labels in output document;
proc sql noprint;
    select put(max(fsatdiag),date8.), put(max(fsatdiag)-27,date8.)
    into :weekmax, :weekmin
    from raw_data_fillin;
quit;
* merge current data with baseline data and categorize weeks into relevant
time periods for analysis;
data raw_city;
    set raw_data;
    length week $50.;
    if events = . then events = 0;
    year = cat("year",put(year(fsatdiag),$4.));
    if (fsatdiag >= (date()-27) & fsatdiag <= (date()-21)) |
        (fsatdiag >= (date()-365-27) & fsatdiag <= (date()-365-21)) |
        (fsatdiag >= (date()-365*2-27) & fsatdiag <= (date()-365*2-21)) |
        (fsatdiag >= (date()-365*3-27) & fsatdiag <= (date()-365*3-21)) |
        (fsatdiag >= (date()-365*4-27) & fsatdiag <= (date()-365*4-21)) |
        (fsatdiag >= (date()-365*5-27) & fsatdiag <= (date()-365*5-21))
    then week='3 weeks ago';
```

```
if (fsatdiag >= (date()-20) & fsatdiag <= (date()-14)) |
  (fsatdiag >= (date()-365-20) & fsatdiag <= (date()-365-14)) |
  (fsatdiag >= (date()-365*2-20) & fsatdiag <= (date()-365*2-14)) |
  (fsatdiag >= (date()-365*3-20) & fsatdiag <= (date()-365*3-14)) |
  (fsatdiag >= (date()-365*4-20) & fsatdiag <= (date()-365*4-14)) |
  (fsatdiag >= (date()-365*5-20) & fsatdiag <= (date()-365*5-14))
  then week='2 weeks ago';
if (fsatdiag >= (date()-13) & fsatdiag <= (date()-7)) |
  (fsatdiag >= (date()-365-13) & fsatdiag <= (date()-365-7)) |
  (fsatdiag >= (date()-365*2-13) & fsatdiag <= (date()-365*2-7)) |
  (fsatdiag >= (date()-365*3-13) & fsatdiag <= (date()-365*3-7)) |
  (fsatdiag >= (date()-365*4-13) & fsatdiag <= (date()-365*4-7)) |
  (fsatdiag >= (date()-365*5-13) & fsatdiag <= (date()-365*5-7))
  then week='1 week ago';
if (fsatdiag >= (date()-6) & fsatdiag <= (date())) |
  (fsatdiag >= (date()-365-6) & fsatdiag <= (date()-365)) |
  (fsatdiag >= (date()-365*2-6) & fsatdiag <= (date()-365*2)) |
  (fsatdiag >= (date()-365*3-6) & fsatdiag <= (date()-365*3)) |
  (fsatdiag >= (date()-365*4-6) & fsatdiag <= (date()-365*4)) |
  (fsatdiag >= (date()-365*5-6) & fsatdiag <= (date()-365*5))
  then week='Most recent week';
if week ~= "";
run;
proc sort data = raw_city; by disease_code disease week year; run;
proc transpose data = raw_city out = trans;
  by disease_code disease week;
  var events;
  id year;
run;
proc transpose data = trans out = trans2 (rename=(name_=year1));
  by disease_code disease week;
run;
proc sort data = trans2; by disease_code disease year1; run;
proc transpose data = trans2 out = final_city (drop=name_);
  by disease_code disease year1;
  var events;
  id week;
run;
data final_city2;
  retain disease_code disease geog year _3_weeks_ago _2_weeks_ago
    _1_week_ago Most_recent_week;
  set final_city;
  array weeks _3_weeks_ago _2_weeks_ago _1_week_ago Most_recent_week;
  do over weeks;
    if weeks = . then weeks = 0;
  end;
  length year $30.;
  year = cat(substr(year1,5), " Conf/Prob/Susp");
  geog = "NYC";
  label _3_weeks_ago = "4 weeks ago";
  label _2_weeks_ago = "3 weeks ago";
  label _1_week_ago = "2 weeks ago";
  label Most_recent_week = "Most recent week (Sun-Sat)";
  drop year1;
run;
proc sql;
```

```

create table alldiseases as select distinct disease_code, disease,
    max(substr(year,1,4)) as year2
from final_city2 group by disease_code order by disease_code;
quit;

* count pending separately;
proc sql;
    create table raw_pending as
    select disease_code, disease, fsatdiag, count(event_id) as pending
    from event_level_input
    where disease_status = "PENDING" & year(fsatdiag) = year(date())
    group by disease_code, disease, fsatdiag
    order by disease_code, disease, fsatdiag;
quit;

* reshape to fill in all dates for all diseases;
proc transpose data = raw_pending out = trans;
    by disease_code disease;
    id fsatdiag;
    var pending;
    format fsatdiag 8.;
run;

proc transpose data = trans out = trans2;
    by disease_code disease;
run;

data raw_pending_fillin;
    set trans2;
    if pending = . then pending = 0;
    fsatdiag = input(substr(_name_,2),8.);
    format fsatdiag mmddyy10.;
    drop _name_;
run;

proc sort data = raw_pending_fillin; by disease_code disease fsatdiag; run;
* merge current data with baseline data and categorize weeks into relevant
time periods for analysis;
data raw_city_pending;
    set raw_pending_fillin;
    length week $50.;
    if pending = . then pending = 0;
    year = cat("year",put(year(fsatdiag),$4.));
    if (fsatdiag >= (date()-27) & fsatdiag <= (date()-21)) |
        (fsatdiag >= (date()-365-27) & fsatdiag <= (date()-365-21)) |
        (fsatdiag >= (date()-365*2-27) & fsatdiag <= (date()-365*2-21)) |
        (fsatdiag >= (date()-365*3-27) & fsatdiag <= (date()-365*3-21)) |
        (fsatdiag >= (date()-365*4-27) & fsatdiag <= (date()-365*4-21)) |
        (fsatdiag >= (date()-365*5-27) & fsatdiag <= (date()-365*5-21))
        then week='3 weeks ago';
    if (fsatdiag >= (date()-20) & fsatdiag <= (date()-14)) |
        (fsatdiag >= (date()-365-20) & fsatdiag <= (date()-365-14)) |
        (fsatdiag >= (date()-365*2-20) & fsatdiag <= (date()-365*2-14)) |
        (fsatdiag >= (date()-365*3-20) & fsatdiag <= (date()-365*3-14)) |
        (fsatdiag >= (date()-365*4-20) & fsatdiag <= (date()-365*4-14)) |
        (fsatdiag >= (date()-365*5-20) & fsatdiag <= (date()-365*5-14))
        then week='2 weeks ago';
    if (fsatdiag >= (date()-13) & fsatdiag <= (date()-7)) |
        (fsatdiag >= (date()-365-13) & fsatdiag <= (date()-365-7)) |
        (fsatdiag >= (date()-365*2-13) & fsatdiag <= (date()-365*2-7)) |

```

```
(fsatdiag >= (date()-365*3-13) & fsatdiag <= (date()-365*3-7)) |
(fsatdiag >= (date()-365*4-13) & fsatdiag <= (date()-365*4-7)) |
(fsatdiag >= (date()-365*5-13) & fsatdiag <= (date()-365*5-7))
then week='1 week ago';
if (fsatdiag >= (date()-6) & fsatdiag <= (date())) |
(fsatdiag >= (date()-365-6) & fsatdiag <= (date()-365)) |
(fsatdiag >= (date()-365*2-6) & fsatdiag <= (date()-365*2)) |
(fsatdiag >= (date()-365*3-6) & fsatdiag <= (date()-365*3)) |
(fsatdiag >= (date()-365*4-6) & fsatdiag <= (date()-365*4)) |
(fsatdiag >= (date()-365*5-6) & fsatdiag <= (date()-365*5))
then week='Most recent week';
if week ~= "" ;run;
proc sort data = raw_city_pending; by disease_code disease week year; run;
proc transpose data = raw_city_pending out = trans;
  by disease_code disease week;
  var pending;
  id year;
run;
proc transpose data = trans out = trans2 (rename=(name=year1));
  by disease_code disease week;
run;
proc sort data = trans2; by disease_code disease year1; run;
proc transpose data = trans2 out = final_city_pending (drop=name);
  by disease_code disease year1;
  var pending;
  id week;
run;
data final_city_pending2;
  retain disease_code disease geog year _3_weeks_ago _2_weeks_ago
    _1_week_ago Most_recent_week;
  merge final_city_pending alldiseases;
  by disease_code disease;
  array weeks _3_weeks_ago _2_weeks_ago _1_week_ago Most_recent_week;
  do over weeks;
    if weeks = . then weeks = 0;
  end;
  length year $30.;
  if year1 ~= "" then year = cat(substr(year1,5), " Pending");
  else year = cat(strip(year2), " Pending");
  geog = "NYC";
  drop year1 year2;
  if ~(_3_weeks_ago=0 & _2_weeks_ago=0 & _1_week_ago=0 &
    Most_recent_week=0);
run;
data final_city;
  set final_city2 final_city_pending2;
run;
proc sort data = final_city; by disease_code disease geog descending year;
run;

** The equivalent code above is carried out at all geographic resolutions (in
our case at Borough and UHF neighborhood);

data final_freqs;
  retain disease_code disease geog year _3_weeks_ago _2_weeks_ago
    _1_week_ago Most_recent_week;
```

```
length geog $50.;
set final_city final_boro final_uhf;
Total = _3_weeks_ago + _2_weeks_ago + _1_week_ago + Most_recent_week;
run;

*****;
* This code creates output reports with signal details;
*****;

* create folders for output to be saved;
options noxwait;
x "cd ...\\&fileweek\\";
x "md Linelist";
x "cd ...\\&fileweek\\Linelist";
x "md Lab_Results";

*-- signals by city, boro and uhf;
%macro rollup(level=,merge=);
proc sql noprint;
    create table signals_&level as
    select *
    from allsignificant
    where geounit="&level";
quit;

*-- linelist by city, boro and uhf;
* while the analysis is based on all disease statuses, only confirmatory
cases are printed in the linelist;
proc sort data = signals_&level (keep=&merge geography metric confirmatory
    mean sd ratio rate); by &merge; run;
proc sort data = event_level_input out = event_level_input2
    (keep=event_id disease_code disease patinit disease_status
    investigation_status diagnosis_date gender
    age_years street_1 boro zip uhfname x_coord y_coord);
    by &merge;
    where fsatdiag >= (&maxfsatdiag-22) & fsatdiag <= &maxfsatdiag &
    disease_status in("CONFIRMED", "PROBABLE", "SUSPECT", "PENDING");
run;
data linelist_&level;
    merge signals_&level (in=insignals) event_level_input2;
    by &merge;
    if insignals;
    level = "&level";
run;
%mend rollup;
%rollup(level=City,merge=disease_code);
%rollup(level=Boro,merge=disease_code boro);
%rollup(level=UHF,merge=disease_code uhfname);
* save events included in signals to permanent file for future references;
data signals.linelist_&fileweek;
    set linelist_city linelist_boro linelist_uhf;
    keep disease_code event_id geography;
run;

* annotation for map;
```

```
data signals;
  length function style color $ 8 text $ 20 geocode $ 3;
  retain xsys ysys '2' hsys '3' when 'a';
  set linelist_boro linelist_city linelist_uhf;
  x = input(x_coord,12.);
  y = input(y_coord,12.);
  function='label'; style='arial'; text='+'; size=1.5; color = "red";
  if level = "UHF" then geog = uhfname;
  if level = "Boro" then geog = boro;
  if level = "City" then geog = "NYC";
  if x = . & y = . then geocode = "no";
  else geocode = "yes";

run;
proc sort data = signals; by disease_code geography diagnosis_date event_id;
run;
* add a count variable to number each case within each signal;
data signals;
  set signals;
  count + 1;
  by disease_code geography;
  if first.disease_code | first.geography then count = 1;
  text = strip(put(count,$3.));

run;
* merge this week's linelist with last week's linelist to identify new cases
in signals that existed in the prior week;
proc sort data = signals.linelist_&lastweek out = lastweek; by disease_code
  geography event_id; run;
proc sort data = signals; by disease_code geography event_id; run;
data signals;
  merge lastweek (in=a) signals (in=b);
  by disease_code geography event_id;
  if b;
  if b & ~a then new = "*";
  if b & ~a then newnum = 1;
  else newnum = 0;

run;
proc sort data = signals; by disease_code geography diagnosis_date; run;

* create macro variables to facilitate looping through signals below;
proc sort data = allsignificant_compare (keep=disease_code disease geounit
  boro uhfname new) out = mapsignificant; by disease_code disease geounit boro
  uhfname; run;
data _null_;
  set mapsignificant;
  length geog $42.;
  by disease_code disease geounit boro uhfname;
  geog = boro;
  if geog = "" then geog = uhfname;
  if geounit="City" then geog = "NYC";
  where ~(geounit="Boro" & boro in("UNKNOWN" " "));
  if first.disease | first.geounit | first.boro | first.uhfname then do;
    i+1;
    ii=left(put(i,20.));
    call symputx ('disease' || ii,strip(disease));
    call symputx ('diseasecode' || ii,strip(disease_code));
    call symputx ('geounit' || ii,strip(geounit));
```

```
call symputx ('geog' || ii, strip(geog));
call symputx ('geog2' || ii, substr(strip(geog), 1, 3));
call symputx ('total', put(ii, 20.));
call symputx ('new' || ii, strip(new));
end;
run;

*-- create summary dataset for choropleth map;
* read in shapefile;
proc mapimport datafile=.../Maps/zip_code_areas_w_uhf.shp' out = uhfmap;
run;
data uhfpop;
    set /* read in a dataset with each geographic unit in map and the
        corresponding population */;
run;
* calculate the number of events in the last four weeks without missing zip
code by disease and geographic unit;
proc sql;
    create table disease_summ as
    select disease_code, disease, input(uhfcode, 3.) as uhfcode, count(*) as
        reports
    from event_level_input2
    where (disease_status in("PENDING", "CONFIRMED", "PROBABLE", "SUSPECT") &
        year(fsatdiag) = year(date()) & week(fsatdiag) < week(date()) &
        week(fsatdiag) >= week(date()) - 4 & zip ~= "")
    group by uhfcode, disease_code, disease
    order by uhfcode, disease_code, disease;
quit;
* calculate rates by geographic unit for mapping;
data disease_summ_rates;
    merge disease_summ(in=indisease) uhfpop(in=inuhfpop);
    by uhfcode;
    if indisease & inuhfpop & census_pop ~= 0;
    rate = round((reports / census_pop) * 100000, .01);
run;

* loop through signals to create a report for each one;
%macro linelist;
%do i=1 %to &total;
* subset datasets to the relevant information for each signal;
data &&diseasecode&i;
    set disease_summ_rates;
    where disease = "&&disease&i";
run;
data signals&i;
    set signals;
    where disease = "&&disease&i" & level = "&&geounit&i" & geog =
        "&&geog&i";
run;
data final_freqs_&i;
    set final_freqs;
    where disease = "&&disease&i" & geog = "&&geog&i";
run;
* summary data for first page;
proc sql;
    create table signals2_&i as
```

```
select distinct disease, geography, metric, confirmatory, mean,
      sd, ratio, rate, cityrate, count(new) as new_count,
      min(newnum) as new_signal
from signals&i
group by disease, geography, metric;
quit;
proc format; value na .="N/A"; run;
data signals2_&i;
  length new_signal2 $3.;
  set signals2_&i;
  if new_signal = 1 then new_signal2 = "yes";
  else new_signal2 = "no";
  if new_signal2 = "yes" then new_count = .;
  format new_count na.;
run;

ods noresults;
options orientation=landscape;
* output results to Linelist folder;
ods rtf
file="...\&fileweek\Linelist\weeklylinelist_&&disease&i...&&geog&i...rtf" bodytitle;

* PAGE 1 - summary of signal;
title "&&disease&i";
title2 "&&geounit&i Signal in &&geog&i";
footnotel font='Arial' height=1 "Total dx past 4 weeks includes only
  confirmed, probable, suspect and pending case statuses";
footnote2 font='Arial' height=1 "When # of SDs above mean > 2, current
  period is considered a signal";
footnote3 font='Arial' height=1 "A missing signal strength value
  indicates an SD = 0 in the baseline period";
ods proclabel="&&geounit&i Signal in &&geog&i";
proc report data=signals2_&i nowd style(report)={outputwidth=7in
  font_size=10pt};
  columns disease geography metric confirmatory ratio new_signal2
    new_count rate cityrate;
  define disease/display "Disease" width=20;
  define geography/display "Unit of geography" width=20;
  define metric/display "Date of interest" width=20;
  define confirmatory/display "Total dx past 4 weeks: &weekmin -
    &weekmax" width=20;
  define ratio/display "Signal Strength (# of SDs above mean)"
    width=10;
  define new_signal2/display "new signal since last week?"
    width=10;
  define new_count/display "if not new, how many new events in
    signal?" width=15;
run;

* PAGE 2 - raw counts;
footnotel font='Arial' height=1 "Unadjusted counts of cases by year,
  week and case status";
footnote2;
proc report data=final_freqs_&i nowd style(report)={outputwidth=7in};
```

```
run;

* PAGE 3 - map;
* subset signals dataset to those that are not missing x and y
  coordinates for mapping;
data nomiss_signals&i;
  set signals&i;
  where x ~= . & y ~= .;
run;
footnotel;
pattern1 v=s c=grayff;
pattern2 v=s c=graydd;
pattern3 v=s c=graybb;
pattern4 v=s c=gray88;
pattern5 v=s c=gray66;
goptions reset=goptions device=png300 target=png300 ftext='Arial'
  htext=1 fttitle='Arial/bold' htitle=1.5 xmax=9 in ymax=7 in;
legend1 label= (j=1 font='Arial/bold' 'Rate per 100,000 for previous 4
  wks'
  j=1 font='Arial' '*Numbers indicate the location of events in the
  signal.'
  j=1 '*Note that rates are meant to provide context only and do'
  j=1 '      not necessarily correspond to signals.'
  position=(top left)) across=1 down=5 frame position=(bottom outside);
proc gmap data = &&diseasecode&i map = uhfmap anno=nomiss_signals&i;
  id uhfcode;
  choro rate / levels=5 coutline=black legend=legend1 cdefault=white;
run;
quit;

* PAGE 4 - line list;
ods proclabel="Line List";
* if the signal is not new, include an extra column to indicate which
  events are newly added to the repeated signal;
%if "&&new&i" = "" %then %do;
proc report data=signals&i nowd;
  columns new text event_id patinit disease_status
    investigation_status diagnosis_date gender age_years
    street_1 boro zip uhfname geocode;
  define new/ display "New" width=1;
  define text/ display "#" width=2;
  define event_id/ display "Event ID" width=10;
  define patinit/display "Pat. Init." width=2;
  define disease_status/display "Disease Status" width=4;
  define investigation_status/display "Investigation Status"
    width=4;
  define diagnosis_date/display "Diagnosis Date" width=10;
  define gender/display "Gender" width=1;
  define age_years/display "Age" width=3;
  define street_1/display "Address" width=10;
  define boro/display "Boro" width=12;
  define zip/display "Zip" width=5;
  define uhfname/display "UHF" width=10;
  define geocode/display "Geocode" width=3;
```

```
run;
%end;
%if "&&new&i" = "*" %then %do;
proc report data=signals&i nowd;
    columns text event_id patinit disease_status investigation_status
           diagnosis_date gender age_years street_1 boro zip uhfname
           geocode;
    define text/ display "#" width=2;
    define event_id/ display "Event ID" width=10;
    define patinit/display "Pat. Init." width=2;
    define disease_status/display "Disease Status" width=4;
    define investigation_status/display "Investigation Status"
           width=4;
    define diagnosis_date/display "Diagnosis Date" width=10;
    define gender/display "Gender" width=1;
    define age_years/display "Age" width=3;
    define street_1/display "Address" width=10;
    define boro/display "Boro" width=12;
    define zip/display "Zip" width=5;
    define uhfname/display "UHF" width=15;
    define geocode/display "Geocode" width=3;
run;
%end;

* PAGE 6 - graph;
proc sql;
    create table graphit as
    select *
    from reshape_all
    where disease_code="&&diseasecode&i" and geounit="&&geounit&i"
           and geog="&&geog&i";
* allow axes to be flexible depending on counts;
proc sql noprint;
    select max(num_sum), min(min(yvar,num_sum)), max(fsatdiag)
           format=mmddy10., max(period1), max(period2), max(period3),
           max(period4), max(period5)
    into :maxsignal, :minsignal, :maxweek, :period1, :period2,
           :period3, :period4, :period5
    from graphit;
quit;
data _null_;
    if &maxsignal.<=10 then do; maxaxis=10; intaxis=1; end;
    else if 10<&maxsignal.<=25 then do; maxaxis=25; intaxis=5; end;
    else if 25<&maxsignal.<=50 then do; maxaxis=50; intaxis=5; end;
    else if 50<&maxsignal.<=100 then do; maxaxis=100; intaxis=10;
        end;
    else if 100<&maxsignal.<=500 then do; maxaxis=500; intaxis=50;
        end;
    else if 500<&maxsignal.<=1000 then do; maxaxis=1000;
        intaxis=100; end;
    else if 1000<&maxsignal.<=1500 then do; maxaxis=1500;
        intaxis=150; end;
    else if 1500<&maxsignal.<=2000 then do; maxaxis=2000;
        intaxis=200; end;
    else if 2000<&maxsignal.<=2500 then do; maxaxis=2500;
        intaxis=250; end;
```

```
else if 2500<&maxsignal.<=3000 then do; maxaxis=3000;
  intaxis=300; end;
else if 3000<&maxsignal.<=3500 then do; maxaxis=3500;
  intaxis=350; end;
else if 3500<&maxsignal.<=4000 then do; maxaxis=4000;
  intaxis=400; end;
else if 4000<&maxsignal.<=4500 then do; maxaxis=4500;
  intaxis=450; end;
else if 4500<&maxsignal.<=5000 then do; maxaxis=5000;
  intaxis=500; end;
if &minsignal.<0 & &minsignal>-1 then do; minaxis=-1; end;
else if &minsignal.<-1 & &minsignal>-5 then do; minaxis=-5;
  end;
else minaxis = 0;
%global maxaxis intaxis;
  call symput('minaxis',minaxis);
  call symput('maxaxis',maxaxis);
  call symput('intaxis',intaxis);
run;
symbol1 i=hiloctj color=blue line=2;
symbol2 i=none color=blue value=dot height=1.5;
symbol3 i=1 color=black value=none height=1.5;
symbol4 i=none color=red value=dot height=1.5;
legend1 label=none order=("mean" "current" "num_sum")
  value=(h=2 pct f=simplex j=c c=black 'Mean +/- 2 SD' 'Signal'
  'Moving 4-Week Sum');
axis1 label=none color=black
  value=(h=2 pct f=simplex j=c c=black)
  order=(&minaxis. to &maxaxis. by &intaxis.)
  rellabel=(color=black)
  width=1
  length=75 pct
  major=none minor=none;
axis2 label=none color=black
  value=(h=2 pct f=simplex j=c c=black)
  rellabel=(color=black)
  width=1
  major=(number=6) minor=(number=12);
goptions reset=goptions device=png300 target=png300 ftext='Arial'
  htext=1 ftitle='Arial/bold' httitle=1.5 xmax=10 in ymax=6.5 in;
proc gplot data=graphit;
  title "&&disease&i";
  title2 "&&geounit&i Signal in &&geog&i";
  title3 height=1.05 "Moving 4-wk sum of adjusted case counts
  ending &maxweek";
  plot (yvar mean num_sum current)*fsatdiag / cframe=GWH autovref
  cvref=wh overlay skipmiss vaxis=axis1 haxis=axis2
  legend=legend1 lhref=2 chref=stro href=&period1 &period2
  &period3 &period4 &period5;
  footnote1 height=0.75 "Disease status includes all except Contact
  and Possible Exposure";
  footnote3 height=0.75 "Vertical dotted lines represent
  corresponding 4-wk periods in baseline";
run;
quit;
ods rtf close;
```

```
%end;
%mend;
%linelist;

*****;
* The following code emails signal information to reviewers.
*****;

* read in list of email recipients by disease;
proc import datafile = "...\Data\reviewers.xls" out = reviewers replace dbms =
    excel; run;
* merge with signal information;
proc sql noprint;
create table signals_reviewer as
select distinct s.disease_code
    ,s.disease
    ,propcase(s.geog) as geography
    ,count(*) as cases_in_signal
    ,case (min(s.newnum)) when 0 then 'No' else 'Yes' end as new_signal
    ,case (min(s.newnum)) when 0 then count(s.new) else . end as new_cases
    ,p.notes as REVIEWER label=''
from signals as s left join reviewers as p on s.disease_code=p.code
where s.suppress = 'no'
group by s.disease_code, s.geography
order by p.notes;
quit;

* create macro to send emails;
%macro email;
data _null_;
set signals_reviewer;
by reviewer;
if first.reviewer then do;
    i+1;
    call symputx('reviewer' || left(put(i,2.)),reviewer);
    call symputx('end',left(put(i,2.)));
end;
run;
* loop through reviewers to send all signal information in one email;
%do i=1 %to &end;
    data reviewer&i;
        set signals_reviewer;
        where reviewer = "&&reviewer&i";
    run;
* &name macro set at beginning of code by whoever is running it;
FILENAME outbox EMAIL
    from= "&name@health.nyc.gov"
        to=(&&reviewer&i)
        cc=("&name@health.nyc.gov")
        subject="Maven: AOW Signals for &fileweek"
    type='text/html'
    CT='text/html';
ods html body=outbox style=minimal;
ods escapechar='^';
```

```
ods text = "^{style [just=1]AOW Signals for &fileweek}";
* print summary table of signals in body of email;
proc report data=reviewer&i nowd nocenter spacing=5;
  columns disease geography cases_in_signal new_signal new_cases;
  define disease/display "Disease" ;
  define geography/display "Geography" ;
  define cases_in_signal/display "total dx past 4 weeks" ;
  define new_signal/display "new signal since last week?" ;
  define new_cases/display "if not new, how many new events in signal?" ;
  title;
  footnote;
run;
* print link to line lists in body of email;
ods text = "^{style [just=1]Signal details and the linelists are here: }";
ods html text = "^{style [just=1] ...\&fileweek\Linelist\ }";
ods text = "^{style [just=1] If you have questions, notify the analyst.}";
ods html close;
%end;
%mend;

%email;
```