**Detecting and measuring interventions effects in R**

Alexis R. Santos-Lozada1 and Mathew E. Hauer2

1. Pennsylvania State University

2. Florida State University

**Abstract**

We provide a brief overview of two R packages that can be used to detect and measure the causal effects of an intervention on a time series: *tsoutliers* and C*ausalImpact*. After an introduction of both packages, and the data used in this paper we discuss the advantages of these methods. During this explanation, we provide sample code and discuss usage and results from a population-level dataset. Finally, we highlight the similarities and differences of each package on modelling causal effects.

**Keywords:** Outlier detection, Causal Inference, R software, time series, causal methods

**INTRODUCTION**

The identification of causal effects (referred to as causal inference) is a powerful motivation for social scientists (Pearl 2009). Often, the identification of a causal mechanism relies on comparisons to counterfactual control groups -- the treatment group compared to a control group – and the difference between the two groups identifies the causal effect. “Real world data” rarely neatly bifurcate into treatment and control groups, especially with longitudinal time series data, but the creation of “synthetic” counterfactual control groups in possible with both inductive and abductive scientific approaches. Inductive analysis leverages knowledge of an intervention to produce a synthetic control group and thus a causal effect while abductive analysis identifies the potential effect via outlier analysis and produces a counterfactual control group to measure the size of the causal effect. This software review focuses on detecting and measuring intervention effects and evaluates two approaches available for conducting this analysis within the R software framework (R Core Team 2018).

Given the emergence and growing availability of intensive longitudinal data (IDL), ecological momentary assessments (EMA), and other sources of information such as social media activity or phone use, there is increasing need to analyze time series, and isolate effects within the phenomenon of interest (McNeish et al. 2021; Ram et al. 2020). This paper provides an overview of two packages in R that offer specific functions to detect and measure intervention effects using both inductive and abductive approaches: *CausalImpact* package (Brodersen and Hauser 2021) and *tsoutlier* package (López-de-Lacalle 2019). In the following software review, we first describe the illustrative data used in this paper, then provide sample code and explore some of the options available within both packages. We conclude this software review by comparing and discussing the results derived from both functions.

**ILLUSTRATIVE DATA**

We will use monthly death counts from the Puerto Rico Vital Statistics System to illustrate the functionality of the *tsoutliers* and *CausalImpact* packages. The data contain monthly aggregates of deaths for Puerto Rico between 2010 and 2018. These data have been used in previous …

/Head Print Table 1 Here/

**R PACKAGES FOR DETECTION AND MEASURING INTERVENTIONS**

and utilizes a classic, inductive approach to measuring causal intervention effects

***tso() in the tsoutliers package***

The tsoutliers package (López-de-Lacalle 2019) implements a mathematical approach for the automatic detection of outliers in a time series originally formulated by Chen and Liu in 1993. Time series are affected by exogenous factors and the effects are felt differently across the phenomenon of our interest. Aside from detecting outliers within out time series, the *tso()* function offers insights about effect being captured when the outlier is detected. By default, three types of outliers detected are:

1. Additive outliers (AO) - are isolated large or small values within the time series,

2) Level shifts (LS) - are a change in the average levels with the observations following the outlier shifting accordingly. This change may be due to seasonality, but has the distinctive feature of the change being permanent, and

3) Temporary or transient changes (TC) - are similar to LS but the effect of the outlier reduces over subsequent observations. Eventually, the values return to the levels observed prior to the outlier.

The two additional outliers featured in this package are:

1) Innovative outliers (IO) - are outliers that derive from innovation in the data generating process that affects all subsequent observations, and

2) Seasonal level shifts (SLS) - are similar to LS but they occur at some point and reoccur every year (time window) at the same season and its effect affects the subsequent seasons.

A detailed discussion of these outliers is available in extant literature (Asghar and Urooj 2017; Burman and Otto 1988; Chen and Liu 1993).

The tso function provides an

Usage of tso() is shown below with all arguments set to the default:

tso(y, xreg = NULL, cval = NULL, delta = 0.7, types = c("AO", "LS", "TC"), maxit = 1, maxit.iloop = 4, maxit.oloop = 4, cval.reduce = 0.14286, discard.method = c("en-masse", "bottom-up"), discard.cval = NULL, remove.method, remove.cval, tsmethod = c("auto.arima", "arima"), args.tsmethod = NULL, logfile = NULL, check.rank = FALSE)

Most of the arguments will work well for basic exploration of outliers with the defaults options, unless such adjustments are needed for an operation other than detecting the outlier. Here, we use the *y*, *types* and *xreg* arguments to determine whether the number of deaths following Hurricane María is considered an outlier, and if so, what type of outlier it is. In addition, we explore whether the results change when we control for population size. First, the *y* argument is the time series where we are interested in detecting the outlier. Regularly, we read our data from a spreadsheet; the imported data is normally not read as a time series. Thus, some simple data processing is required to have the data in a format that is compatible to *tso()*. After having imported the data, we transform the information into a time series using the *tso()* function available through the *tsoutliers* library, as follows:

Deaths\_ts<-ts(Deaths0917$Deaths,frequency=1,start=Deaths0917$Number[1])

Popula\_ts<-ts(Deaths0917$Population\_Estimate,frequency=1,start=Deaths0917$Number[1])

The first line creates a time series for the monthly deaths from 2010 until 2018 and the second one creates a time series object for the corresponding monthly population estimates. This information was contained within the dataset we imported in the illustrative data section (see *Table 1*). This is all the data manipulation required to have the data in a format that is familiar to the *tso()* function.

The essential arguments for *tso()* are the time series where we want to detect the outlier, the type of outlier we want to detect, and potentially a control variable or variables (*xreg*). We start with a simple model that only considers the monthly death counts (*y*), specifying the detection for the three default types of outliers: AO, LS, and TC. If the interest is to detect only one type of outlier, it will suffice to exclude the others from the type argument. Similarly, if the interest is to detect one of the two other types, it will simply suffice to include it within the *types* argument. The models are estimated using the following code:

analysis <- tso(Deaths\_ts,types=c("AO","LS","TC"))

analysis2 <- tso(Deaths\_ts), xreg = Population\_Estimates\_ts, types=c("AO","LS","TC"))

The above code stores the output from tso() in two separate objects called analysis and analysis2. The difference between both analyses is the inclusion of population estimates to account for changes in population size. The resulting: (1) by looking at the output in table form (Table 2) or (2) through data visualization (Figure 1). To examine the output table one must simply write the name of the object where the results are stored in the console. The output includes the type of outlier detected, the observation id, the estimated excess and the t-statistic associated with the outlier.

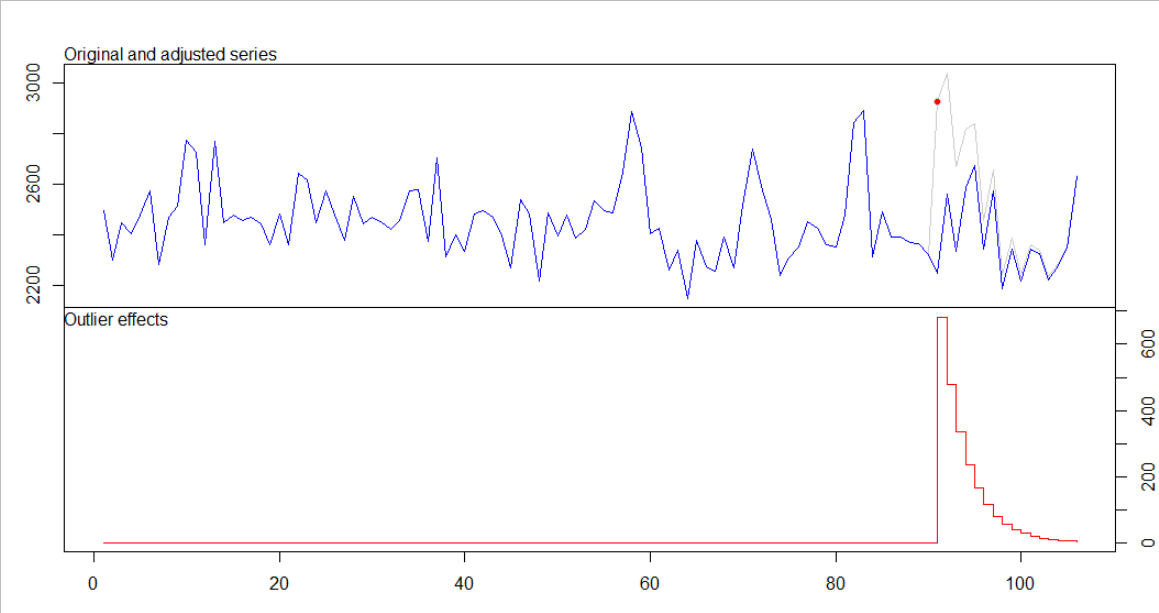
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 2** - Outlier detection for monthly deaths for Puerto Rico, 2010-2018 without and with controls for population size | | | | |
| **Simple Model - Default Settings** | | | | |
| Type | Ind | Time | coefhat | tstat |
| TC | 91 | 91 | 597.2 | 4.718 |
|  | | | | |
| **Controlling for Population Size - Default Settings** | | | | |
| Type | ind | time | coefhat | tstat |
| TC | 91 | 91 | 682.4 | 5.163 |
| Source: Puerto Rico Vital Statistics System (2009-2018) | | | | |

In both instances, the model identified September 2017 as a temporary change (TC). This tells us that the number of deaths detected in the month of Hurricane Maria exceeded the expected levels and that this effect was not constrained to that month, it continued affecting Puerto Rico for subsequent months until the point the number of deaths returned to expected levels. In this case, coefhat represents the excess deaths observed in this month. The simple model indicates that there were 597 deaths in excess of historical patterns and the model that accounts for population size indicates that 682 deaths occurred in excess of expected levels. Previous studies that employ a time series approach to estimate excess deaths in Puerto Rico have estimated the deaths in excess to be: 574 (95% C.I. 515-630), 449 (95% C.I. 377-527) and 459 (95% C.I. 425-293)(Rivera and Rolke 2019; Santos-Burgoa et al. 2018; Santos-Lozada and Howard 2018). The results derived from the using the *tso* function are consistent or close to these estimates. Consistent with these papers, the tso function identifies September 2017 as an outlier when compared to the expected counts (adjusted series).

This outlier was classified as a TC type of outlier; this means the effect of the hurricane is felt in some subsequent periods. To better understand the impact of the Hurricane and its diminishing effects, it is best to visualize the outlier in comparison to the time series and how this observation, and subsequent ones deviate from the expected pattern. The *tsoutlier* package is compatible with the basic plotting functions. To produce a visual representation of the time series and the outlier effects we use plot(). For purposes of brevity, we show the visualization of the model that controls for population size in Figure 1.

We simply ask R to plot the object where the output is stored:

plot(analysis2)



**Figure 1.** Plot of the tso function that considers population size through the xreg argument. Detecting one outlier in September 2017 (red dot) with a diminishing effect in the following months until the time series converges towards the expected levels based on pre-Hurricane María patterns represented in the outlier effects panel.

**Comparison of the tsoutlier and CausalImpact packages**

While the results of both functions are similar the process and formulation of the approach differed. In tso() we specify the time series of interest, the different types of outliers to be detected, and potentially include control variables. The result was the identification of the outlier and the length of the effect. This means that we do not need to know the point in which the intervention happened to detect it. For the *CausalImpact* function we need to know when the intervention happened in order to measure it. When the latter is specified we define a pre- and post-period something that is not required for the *tso* function. The *tso* function detected September 2017 as an outlier with an effect that diminished as we moved away from the time of the intervention (ind and time = 91 in Table 2). In the case of the *CausalImpact* model, we indicated that the intervention occurred in September 2017 or observation 91 (*post.period* argument) and the algorithm measured the intervention effects. Both functions have some form of assessing significance. In the case of tso() we have a t-statistic that corresponds to the outlier that was detected. On the other hand, the *CausalImpact* function provides a p-value that corresponds to a Bayesian-one sided tail area probability which allows us to determine whether the causal impact was significant or not. Finally, both functions provided a way of visually examining the time series and the resulting analysis. Both methods allow us to conclude that the Hurricane María caused deaths in excess of expected levels as established in the extant literature (Sandberg et al., 2019).

**DISCUSSION**

There is a growing need to assess the effects of interventions (ref). The *tsoutliers* and *CausalImpact* packages provide avenues to do this with some level of consistency. The consistency in the results is formidable as the first relies on a frequentist approach, while the second does it through a probabilistic one. There are some who argue that probabilistic approaches are more powerful and would recommend them over frequentist ones, our results show that both approach … . However, it is possible that researchers are interested in detecting outliers within a time series without knowing the specific timing of an intervention as required by *CausalImpact*. Thus, *tsoutliers* can be useful as a method for detection of intervention effects for instances where the timing of the intervention is unknown. Because of this, we evaluated the functionality of both packages using the same illustrative data.

The tso() function in the *tsoutliers* package and CausalImpact() in the package with the same name have similar requirements. For both *tso*() and *CausalImpact*(), the first requirement is a time series for which we want to detect or measure an intervention effect. For tso() we need to provide the type of outlier to be detected, or else the function will default to detecting “AO”, “LS” and “TC”. In the case of *CausalImpact*() we need to provide a pre- and post-intervention period. Simply said, while the initial formulation and coding structure are very similar, the information we need to provide to the function differs significantly. This is a direct result of the objective of each function, while one is more powerful as a detection tool; the second one is better suited to measure an intervention effect when we know the specific timing of such intervention. In both instances, we can provide additional variables, which serve as controls. For *tso()* we include them by specifying the *xreg* argument. To incorporate controls within the *CausalImpact* approach we need to include them in the data matrix. This matrix should be structured in a way that the first column corresponds to the time series of interest, and the subsequent columns are the controls. Despite these differences in both mathematical formulation and empirical approach, both methods produce similar results. These results are comparable to those produce using other approaches as summarized in previous systematic review of this issue (Sandberg et al. 2019).

The biggest difference between both packages is the method through which we determine when to start measuring the intervention effect. The *tso()* function detects outliers provided that they fit the specific type of outlier requested in the type argument. While it is always possible to include the five types of outliers, determining the root cause of every outlier may be a time-consuming task and some can be the result of natural fluctuations in the time series. Thus, the detection of an outlier and its effect are better evaluated in a case-by-case basis. This is complicated when, for example, the analysis consists of more than one unit of analysis with longer time series. In the case of the *CausalImpact*() function, we define the pre- and post-intervention period and researchers may miss any outliers of significant shifts in the time series that occur in the pre-intervention period. Thus, we recommend using the functions contained within *tsoutliers* to detect intervention effects, which can be analyzed based on what type of outlier it is determined to be. Further, we recommend using the functions contained within the *CausalImpact* package to measure an intervention effect when the precise timing of the intervention is known. Both functions may be used in combination if/when necessary.

Although the *tso()* function does not offer an alternative to calculating the cumulative effect, this can be remedied by using basic arithmetical operations. To measure the cumulative effect, we subtract the expected values from the observed ones after the outlier is detected. The summation of these differences is the cumulative effect of the intervention. While simple in coding, the researcher needs to be careful not to include effects that correspond to another outlier into the total. In our case, only one outlier was detected so this is not an issue. However, the situation becomes more complex when multiple outliers are detected and for instances where these outliers may have an effect in subsequent periods or when we observe a shift in the time series. Thus, estimating cumulative effects is a process that should be performed with caution and after a careful examination of the results.

Visualizations with both packages are easy to produce and interpret. These are produced using the plot() function which is part of the original or *base* graphics functions included with R. When plotting the results from *tso*() it produced a two-panel graph where the upper panel presents the time series and highlights any outliers detected. The lower panel provides insights as to how far the outlier is from the expected value and for how long this effect lasted. How long did the effect last? The *tso()* can detect isolated points, concentrated effects, or significant shifts in the time series. Thus, it provides a simple, yet elegant visualization that allows following the effect across time. In the case of the *CausalImpact()* visualization, the plot function produces three panels, something that can be modified with the code. The first panel, labeled *original*, is similar to the upper panel produced *with tso()* and it shows the time series and the expected values. The second panel is similar to the lower panel produced with *tso()* and it shows the effect estimates for each post-intervention observation. The third panel is unique to *CausalImpact()* and consists of the cumulative effect of the intervention. The key distinction between both visualizations is that *CausalImpact()* includes a vertical dashed-vertical line representing the start of the post-intervention period. In our code, we also provide examples that combine the initial plots with functions from the *ggplot2* and *ggpubr* packages to put titles and axes titles in both figures, and to join them into one data visualization. Although this last comment deals with the aesthetics of visualization, and not with the results, we highlight the possibility of modifying the visualization, which can be useful when researchers are interested in producing publication-ready figures.

In closing, both packages are similar in that they produce consistent estimates of excess mortality following the intervention. However, they offer different insights into the intervention that users should be aware of. The *tsoutliers* package is better suited for analyses where we are interested in detecting and understanding intervention effects within a time series. On the other hand, the *CausalImpact* package is better suited for analyses where the timing of the intervention is known. While other approaches exist, we see a fruitful and promising avenue for the use of both functions to detect and measure intervention effects when time series are available. Both packages have their strengths and weaknesses, but they can be used to perform both inductive and deductive assessments of interventions.

**Acknowledgments**

This research supported by the Social Sciences Research Institute (SSRI) and the Population Research Institute (PRI) at the Pennsylvania State University. Alexis R. Santos is a Social Disparities Cluster faculty at SSRI. PRI is supported by a grant from the Eunice Kennedy Shriver National Institute of Child Health and Human Development (P2CHD041025) and by the Pennsylvania State University and by SSRI. Mathew E. Hauer is supported by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**References**

Asghar, Z., & Urooj, A. (2017). Analysis of seasonal level shift (SLS) detection in SARIMA models. *Communications in Statistics: Simulation and Computation*. https://doi.org/10.1080/03610918.2016.1236952

Brodersen, K. H., Gallusser, F., Koehler, J., Remy, N., & Scott, S. L. (2015). Inferring causal impact using bayesian structural time-series models. *Annals of Applied Statistics*, *9*(1), 247–274. https://doi.org/10.1214/14-AOAS788

Burman, J. P., & Otto, M. C. (1988). Outliers in Time Series. In *Statistical Research Division Report Series*.

Chen, C., & Liu, L.-M. (1993). Forecasting Time Series with Outliers. *Journal of Forecasting*, *12*, 13–35. https://doi.org/10.1016/0304-4076(93)90010-3

Rivera, R., & Rolke, W. (2018). Estimating the death toll of Hurricane Maria. *Significance*, *15*(1), 08–09.

Sandberg, J., Santos-Burgoa, C., Roess, A., Goldman-Hawes, A., Pérez, C. M., Garcia-Meza, A., & Goldman, L. R. (2019). All over the place? Differences in and consistency of excess mortality estimates in Puerto Rico after hurricane Maria. *Epidemiology*, *30*(4), 549–552. https://doi.org/10.1097/EDE.0000000000000970

Santos-Burgoa, C., Sandberg, J., Suárez, E., Goldman-Hawes, A., Zeger, S., Garcia-Meza, A., Pérez, C. M., Estrada-Merly, N., Colón-Ramos, U., Nazario, C. M., Andrade, E., Roess, A., & Goldman, L. (2018). Differential and persistent risk of excess mortality from Hurricane Maria in Puerto Rico: a time-series analysis. *The Lancet Planetary Health*, *2*(November 2018), e478–e488. https://doi.org/10.1016/S2542-5196(18)30209-2

Santos-Lozada, A. R., & Howard, J. T. (2018). Use of Death Counts from Vital Statistics to Calculate Excess Deaths in Puerto Rico Following Hurricane Maria. *JAMA*, *320*(14), 1491–1493. https://doi.org/doi:10.1001/jama.2018.10929