Spatio-Temporal Analysis of Flash Flood Events in Texas

by

Wilkho, Rohan Singh[1], and Ogieriakhi, Macson[2]

[1] Ph.D. Student - Civil Engineering, MS Student - Computer Science

[2] Ph.D. Student - Agricultural Economics

1.0 Introduction and Research Problem

Increasing urbanization and a changing climate have resulted in a greater frequency of extreme weather events around the globe [1-3]. Despite advances in forecasting methods, the frequency and intensity of events like droughts, snowstorms, wildfires, hurricanes, floods, among others, continue to expose many communities to high risk [4, 5]. This continued exposure necessitates managing extreme weather events through preparedness and mitigation rather than response and recovery. One such weather event which puts many (if not most) communities at risk is flash flooding (FF).

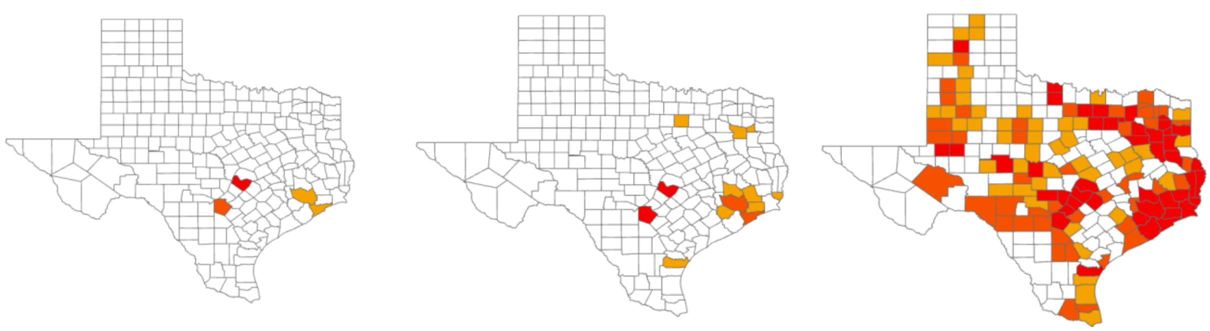
FF refers to flooding that begins within a short period (3 - 6 hours) of a triggering event (heavy rainfall, dam break) [6]. FFs are characterized by small spatial extents and high water flow velocities, making the FF phenomenon localized. The short onset and localized behavior leave communities with little to no lead time to respond to FF [7].

Furthermore, FFs are the leading cause of fatalities and injuries due to extreme weather events in the US [8]. From 2000 to 2019, FFs have accounted for approximately 72%, 71%, and 73% of all flood-related fatalities, injuries, and economic damages in the US [8]. Moreover, FF is the most frequent natural disaster in Texas [9].

Despite the unpredictability and menacing impacts of FFs, a thorough understanding of the FF phenomenon remains wanting. *Analysis of past events* is a prerequisite to the phenomenon's better understanding. Moreover, since FF remains localized and more frequent in some regions, *targeted intervention* is necessary. This project aims to better understand the FF phenomenon by performing a *Spatio-temporal analysis of past FF events* from 2010 to 2019. We have restricted our analysis to Texas.

1.1 Objective

This study aims to analyze the Spatio-temporal distribution of FF events in Texas between 2010-2019 to identify counties most susceptible to FF. However, we cannot achieve this by simply distributing FF events in Texas because different numerical distributions will identify different counties as the ones most susceptible to FF (Figure below).



Equal-Interval distribution　　Natural-Breaks distribution　　Quantile distribution

Fig 1: Distribution of FF events in 2019 in Texas Counties

Fig 1 shows the distribution of FF events in Texas counties in 2019. The darker the county's color, the higher is the FF event concentration in 2019. We can observe from the figure that different distributions highlight a varying number of counties as most susceptible to FF. For example, equal-interval and quantile distributions highlight four and nearly half the counties in

Texas as susceptible to FF, respectively. Moreover, this distribution does not include any statistical randomness test (whether the event distribution is random over space) or significance (the regions are statistically significant). Hence, this calls for performing spatial clustering of FF events in Texas counties - identifying counties where FF occurrence is not random and is significant.

Moreover, a particular county may receive significantly high FF only in a specific year (possibly due to some severe storm event) but not in other years. Such unusual patterns also necessitate performing a temporal clustering of FF events, where we can identify the counties that are frequently experiencing significantly high FF (most susceptible counties).

Therefore, the objective is to perform a ***spatio-temporal clustering of past FF events*** (for each year between 2010-2019) in Texas. This spatio-temporal clustering aids in:

- Identifying counties that are most susceptible to FF
- Identifying counties and underlying reasons for a particular region's significantly high FF.

2.0 Data

We sourced the data for this project from the public-domain NOAA Storm Events (SE) database. The SE database contains information on reported weather/storm events (floods, hurricanes, tornadoes, among others) from January 1950 to the present. The SE database is available publicly, and NOAA's National Weather Service (NWS) maintains it. The database is updated monthly; however, the published data lags 90–120 days behind the current date. NOAA curates the information in the SE database from multiple sources: NWS, media, law enforcement, government agencies, emergency managers, private businesses, and individuals [8]. Essential information recorded in the SE database about these events is as follows:

a) Event date and approximate location (beginning and ending coordinates) of the affected area

b) Impact: human injuries, human fatalities, and cost estimates of damage to crops and property

c) Episode narrative: a brief description of the event meteorology

d) Event narrative: a brief description of the event's impact and the surrounding conditions, such as road conditions.

3.0 R-package and Model:

Package:

The statistical package used in this project was the rflexscan package [10]. It provides functions and classes to analyze spatial count data using the flexible spatial scan statistic developed by Tango and Takahashi [11]. The term 'flexible' indicates that the scanning window in the statistic is not circular but rather is flexible. We can use the package for :

- evaluating spatial clusters, i.e., to see if they are statistically significant.
- To test if the event's distribution is random over space.
- To detect areas of significantly high rates.

The flexible scanning window has a limitation, i.e., it cannot perform an optimal analysis for a vast number of areas (counties in our case).

Model:

The steps followed while creating the model are:

- *Aggregate the FF records at the county level*: Each row in the SE database corresponds to a FF event. So, the first data processing step was to aggregate these records at the county level for each year between 2010-2019.
- *Normalize the FF count with the county area*: The variable of interest for our analysis was the *number of FF events per unit area* in a county. Since counties have variable areas, a county having a larger area may receive more FFs. So, to negate that, we normalized the FF count in a county with the county area.
- *Define Neighborhood structure*: The next step was to define a neighborhood structure. We can define the neighborhood structure based on adjacency or distance. We had defined the neighborhood structure based on adjacency.
- *Perform spatial clustering using the 'rflexscan' function of the rflexscan package*: This function analyzes the spatial count data (number of FF/county area in this case) using the flexible scan statistic. It accepts the *coordinates of the county centroid*, the *observed variable count*, the *expected variable count*, the *neighborhood structure* (nb), and the *cluster size* (cluster size) as inputs. It then returns an rflexscan object which contains analysis and specified parameters. The expected variable count for a particular year is calculated simply by averaging the FF event density across all counties for that year. The cluster size denotes the maximum number of counties in a cluster.
- *Select significant clusters*: The rflexscan object returned by the 'rflexscan' function contains a list of all the clusters identified. Each of these clusters has a *p-value*, which indicates the significance level of the cluster. Since the objective of this study is to identify significant clusters/counties, we only select those clusters with a p-value < 0.05.

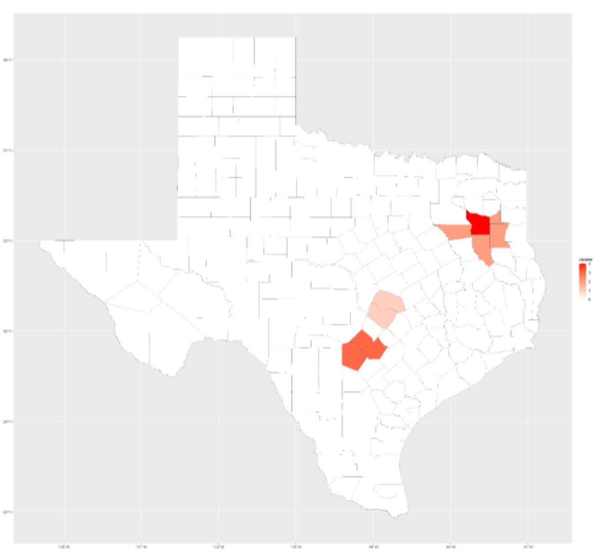
- *Visualize the results*: The final step was to visualize the results. We visualized the results in maps for each of the ten years. We show and discuss them in the following section.

Note: We tried different cluster sizes (5, 10, and 20) for this study. We finally selected 20 because none of the clusters returned (when cluster size was 20) had a size greater than 11 counties.

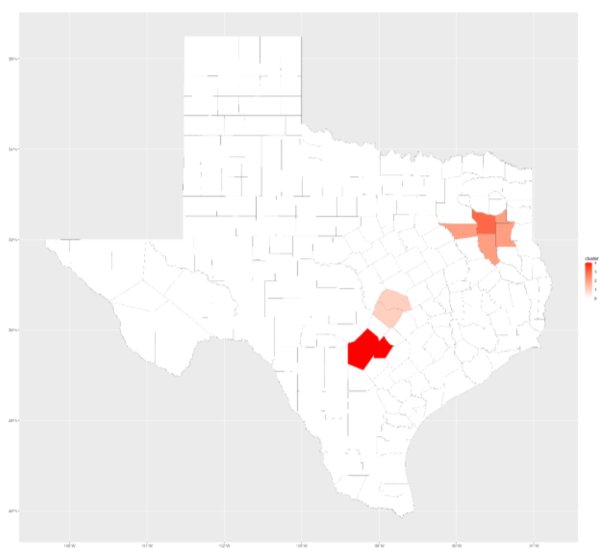
4.0 Results and Discussion

Fig 2 shows the clusters of Texas counties receiving significant FF from 2010-2019. Darker color represents greater susceptibility to FF in that particular year.

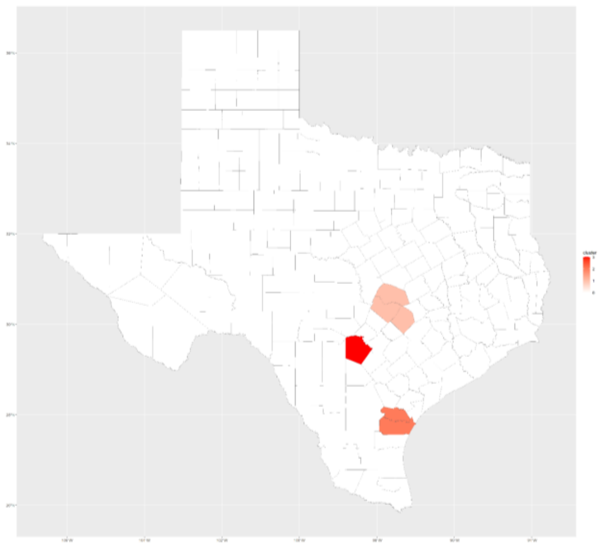
We summarize the results of these maps in Table 1 below (the counties at the same bullet point indicate the same level of significance).



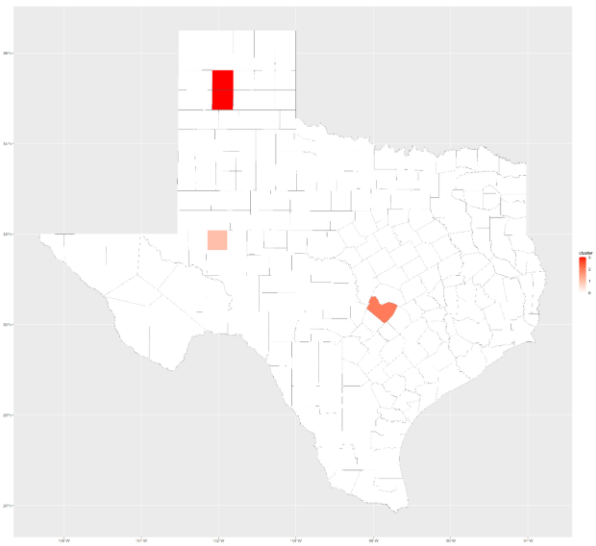
2010



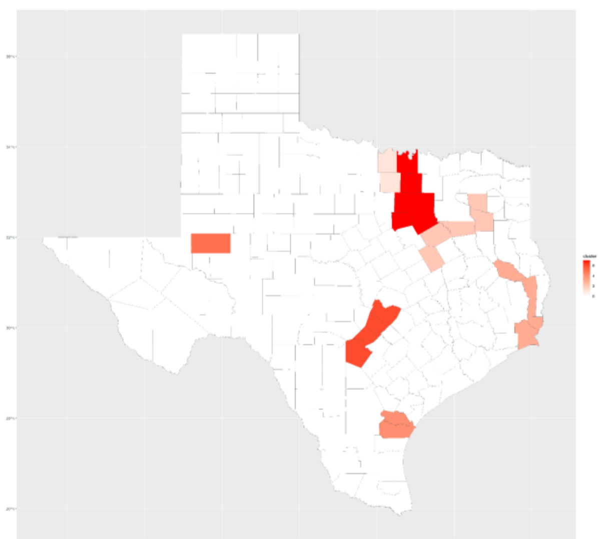
2011



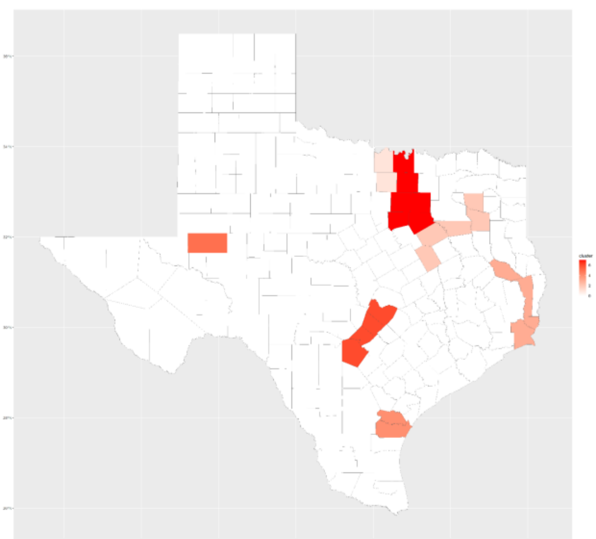
2012



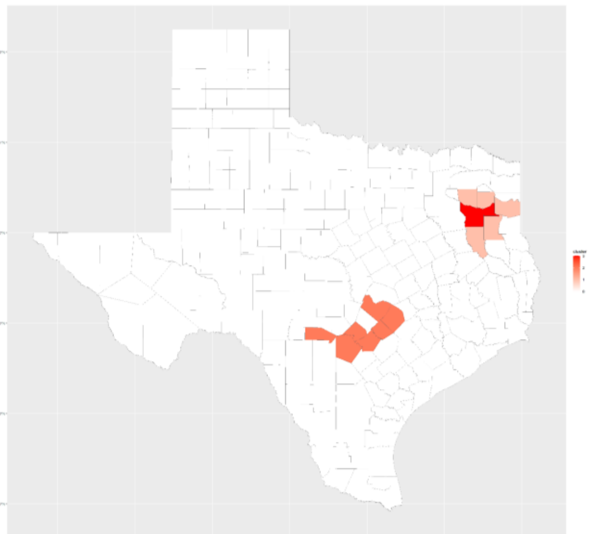
2013



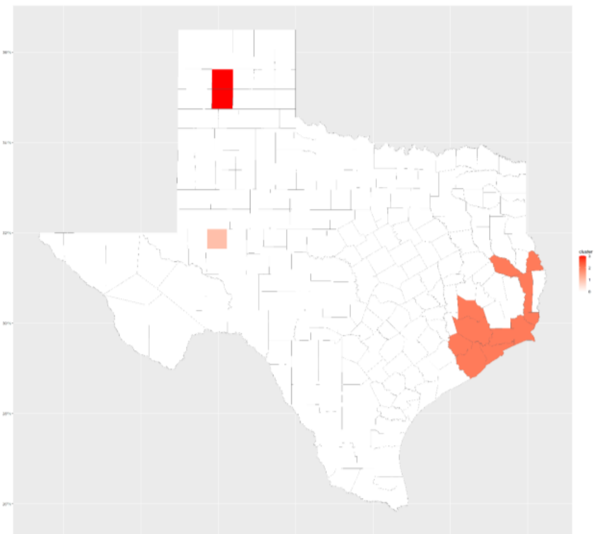
2014



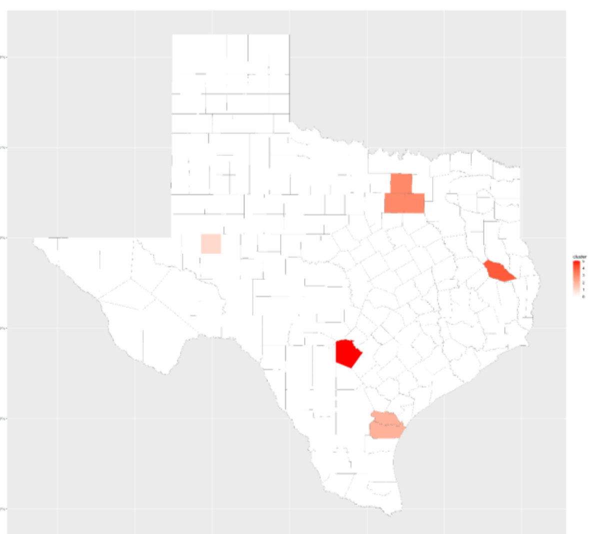
2015



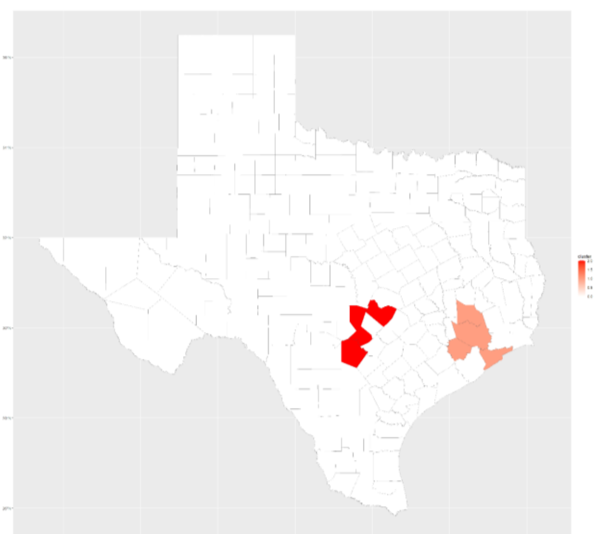
2016



2017



2018



2019

Fig 2: Spatio-temporal clustering results.

Table 1: County Clusters for Flash Flood Events in Texas from 2010 to 2019

Year	**Susceptible counties** (Ranked in decreasing order of statistical significance, all of them have p-value < 0.05)
2010 (4 significant clusters)	• Smith • Bexar, Comal, Guadalupe • Gregg, Rusk, Cherokee, Henderson • Travis, Williamson
2011 (4 significant clusters)	• Bexar, Comal, Guadalupe • Smith • Gregg, Rusk, Cherokee, Henderson • Travis, Williamson
2012 (3 significant clusters)	• Bexar • San Patricio, Nueces • Travis, Williamson, Bastrop
2013 (3 significant clusters)	• Potter, Randall • Travis • Midland
2014 (8 significant clusters)	• Denton, Tarrant, Dallas, Johnson, Ellis • Bexar, Comal, Hays, Travis • Ector, Midland • Cooke, Grayson, Wise • San Patricio, Nueces • Angelina, Jasper, Orange, Jefferson • Navarro, Limestone • Harrison, Gregg, Smith, Wood
2015 (8 significant clusters)	• Denton, Tarrant, Dallas, Johnson, Ellis • Bexar, Comal, Hays, Travis • Ector, Midland • Cooke, Grayson, Wise • San Patricio, Nueces • Angelina, Jasper, Orange, Jefferson • Navarro, Limestone • Harrison, Gregg, Smith, Wood
2016 (5 significant clusters)	• Gregg, Smith • Bexar, Comal, Guadalupe, Caldwell, Travis • Bandera, Uvalde • Upshur, Harrison, Rusk, Cherokee • Bastrop, Fayette
2017 (4 significant clusters)	• Potter, Randall • Galveston, Chambers, Harris, Brazoria, Fort Bend, Montgomery • Sabine • Midland
2018 (6 significant clusters)	• Bexar

	• Angelina • Denton, Tarrant, Dallas • San Patricio, Nueces • Midland • Tom, Green
2019 (2 significant clusters)	• Bexar, Comal, Blanco, Travis • Montgomery, Harris, Galveston, Fort Bend

Table 1 indicates that the number of clusters varies from year to year over the ten years. For example, 2014 and 2015 have the highest number of clusters, while 2019 has the least number of clusters. The table also helps us identify counties that have been significantly receiving high FF over the ten years, i.e., the ones most susceptible to FF. Table 2 contains a list of the most susceptible counties (along with the frequency of being in a significant cluster in the ten years).

Table 2: Most Susceptible Counties and Frequency of their occurrence in a significant cluster (only those counties which have occurred in at least 5 clusters)

County	**Frequency of occurrence in a significant cluster** (out of 10)
Bexar	8
Travis	8
Comal	6
Midland	5
Smith	5
Gregg	5

Table 2 reveals that Bexar and Travis counties are most susceptible to FF in Texas. Comal, Midland, Smith, and Gregg counties come in next.

To understand more recent trends, we have also analyzed the clustering for the later five years, i.e., 2015-2019. The county frequency for that period is available in Table 3.

Table 3: Frequency of Cluster Appearance in the last 5-Year Period (only those counties which have occurred in at least 2 clusters)

County	**Frequency of occurrence in a significant cluster** (out of 5)
Bexar	4
Travis	3
Comal	3
Midland	3
Smith	2
Gregg	2
San Patricio	2
Nueces	2
Harrison	2
Dallas	2
Tarrant	2

Montgomery	2
Denton	2
Fort Bend	2
Angelina	2
Galveston	2

From Tables 2 and 3, we gather that Bexar, Travis, Comal, and Midland counties are most susceptible to FF in Texas. Hence, these counties should be the prime focus to mitigate FF risk in Texas.

Also, from Fig 2, we observe that the Greater Houston Area is one of the significant clusters in 2017, whereas it didn't show up as one in the previous years. A possible reason for this could be Hurricane Harvey, which caused significant destruction in and around Houston. Therefore, this kind of analysis also helps us observe unusual occurrences, leading to further investigation.

5.0 Conclusion

This study has proposed a methodology for spatio-temporal analysis of FF events. This methodology can help one:

- Identify regions most susceptible to FF - which can, in turn, lead to further analyses like causal, risk, among others.,
- Recognize and investigate unusual patterns,
- Identify neighboring regions with contrasting FF behavior - which can, in turn, lead to a comparative study to adopt mitigation strategies.

In today's day and age, where budgets are concerned, a study like this can also assist in effective budget allocation for mitigation purposes, i.e., focus on regions most susceptible to FF. This can help alleviate the risks of FF in those areas.

We can extend this study to:

- The census tract/census block levels (offering a more targeted approach),
- Considering variables like fatalities and injuries due to FFs, economic losses due to FFs to focus on the dangerous and impacted areas (due to FF).

The R package and the SE database pose limitations to this study. The R package doesn't return optimal results for many areas. And this analysis will not consider any FF event not recorded in the SE database. Moreover, we did not consider the intensity of flash floods, which can be a helpful addition.

References

[1] Milly, P. C. D., Wetherald, R. T., Dunne, K. A., & Delworth, T. L. (2002). Increasing risk of great floods in a changing climate. Nature, 415(6871), 514–517. https://doi.org/10.1038/415514a

[2] Yang, L., Smith, J. A., Wright, D. B., Baeck, M. L., Villarini, G., Tian, F., & Hu, H. (2013). Urbanization and climate change: An examination of nonstationarities in urban flooding. Journal of Hydrometeorology, 14(6), 1791–1809. https://doi.org/10.1175/JHM-D-12-095.1

[3] Wobus, C., Lawson, M., Jones, R., Smith, J., & Martinich, J. (2014). Estimating monetary damages from flooding in the United States under a changing climate. Journal of Flood Risk Management, 7(3), 217–229. https://doi.org/10.1111/jfr3.12043

[4] National Academies of Sciences, Engineering, and Medicine. 2019. Environmental Engineering for the 21st Century: Addressing Grand Challenges. Washington, DC: The National Academies Press. https://doi.org/10.17226/25121

[5] Javier, J. R. N., Smith, J. A., Meierdiercks, K. L., Baeck, M. L., & Miller, A. J. (2007). Flash flood forecasting for small urban watersheds in the Baltimore metropolitan region.

Weather and Forecasting, 22(6), 1331–1344. https://doi.org/10.1175/2007WAF2006036.1

[6] Sweeney, T. L. (1992). Modernized Areal Flash Flood Guidance. NOAA Technical Memorandum NWS Hydro 44. Silver Spring , MD : National Weather Service

[7] Wang, N., Lombardo, L., Tonini, M., Cheng, W., Guo, L., and Xiong, J.: Spatiotemporal clustering of flash floods in a changing climate (China, 1950–2015), Nat. Hazards Earth Syst. Sci., 21, 2109–2124, https://doi.org/10.5194/nhess-21-2109-2021, 2021

[8] NOAA Storm events database. National Climatic Data Center, 2021. Retrieved from http://www.ncdc.noaa.gov/stormevents/. Accessed Dec 04, 2021.

[9] Texas Department of State Health Services, (2021). https://texasready.gov/be-informed/natural-disasters/floods.html. Accessed Dec 04, 2021.

[10] Package, T., The, T., Spatial, F., Statistic, S., Otani, A. T., Takahashi, K., Rcpp, I., & Rcpp, L. (2021). Package ‘ rflexscan .’ https://doi.org/10.18637/jss.v099.i13>.URL

[11] Tango, T., & Takahashi, K. (2005). A flexibly shaped spatial scan statistic for detecting clusters. International Journal of Health Geographics, 4, 1–15. https://doi.org/10.1186/1476-072X-4-11

APPENDIX

R Code: *(Code only shown for 2019, for other years just the input file needs to be different)*

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

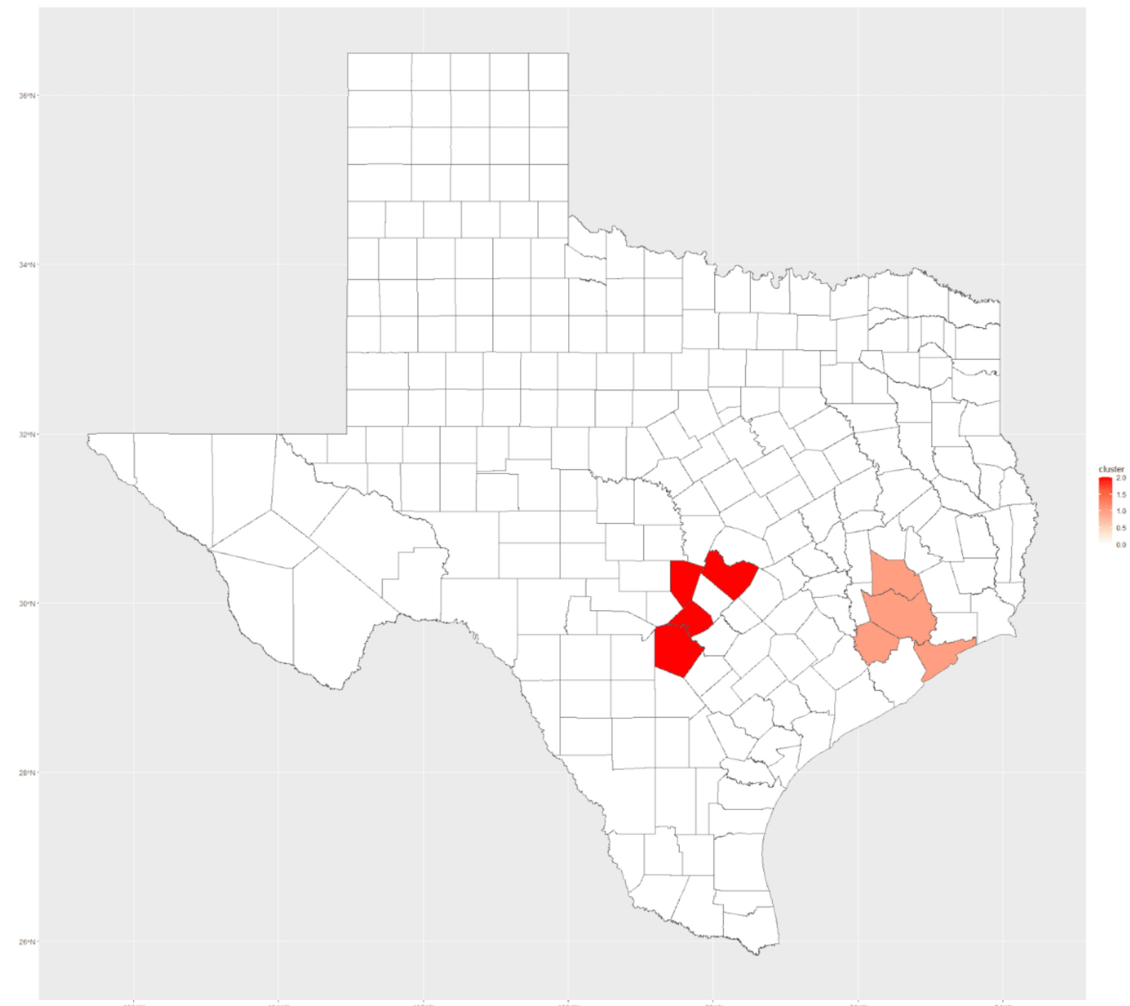
Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
rm(list = ls())
library(spdep)

## Loading required package: sp

## Loading required package: spData

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`

## Loading required package: sf

## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1
```

```
library(rflexscan)
library(sf)
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(readxl)
library(fields)

## Loading required package: spam

## Loading required package: dotCall64

## Loading required package: grid

## Spam version 2.7-0 (2021-06-25) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

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

## Loading required package: viridis

## Loading required package: viridisLite

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

data <- st_read('C://Users//19795//OneDrive//Desktop//STAT 647//STAT 647
Project//Shapefile//2019_Cty.shp')

## Reading layer `2019_Cty' from data source
##   `C:\Users\19795\OneDrive\Desktop\STAT 647\STAT 647
Project\Shapefile\2019_Cty.shp'
##   using driver `ESRI Shapefile'
```

```
## Simple feature collection with 254 features and 7 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: -106.6456 ymin: 25.83716 xmax: -93.50804 ymax:
36.5007
## Geodetic CRS:  NAD83

data$num_FF = data$COUNT_COUN
data = subset(data, select = -c(COUNT_COUN))
data$FF_Area = data$num_FF/(data$SUM_ALAND1+data$SUM_AWATER)*100000000 #
USing a factor of 100000000

# Neighborhood based on Adjacency
A <- st_touches(data)
A <- as.matrix(A)
adj <- apply(A==1,1,which)


expected <- sum(data$FF_Area) / nrow(data)

fls <- rflexscan(x = as.numeric(data$Long), y = as.numeric(data$Lat),
  observed = as.numeric(data$FF_Area),
  expected = as.numeric(expected),
  name = data$COUNTYFP10,
  clustersize = 20,
  nb = adj)

sum <- summary(fls)
sum$cluster

##   NumArea   MaxDist Case  Expected        RR      Stats     P
## 1       4 1.1529388   44 1.1496063 38.273973 134.058467 0.001
## 2       4 1.1211008   16 1.1496063 13.917808  28.933222 0.001
## 3       2 0.4643589    4 0.5748031  6.958904   4.417193 0.994
## 4       1 0.0000000    3 0.2874016 10.438356   4.375102 0.994
## 5       4 0.9188121    4 1.1496063  3.479452   2.194408 1.000

# Saving only significant clusters
significant.clusters <- as.data.frame(list())

all.clusters.data.frame <- as.data.frame(sum$cluster)

for (row in 1:nrow(all.clusters.data.frame)){
  if(all.clusters.data.frame[row, ]$P < 0.05){
    significant.clusters <- rbind(significant.clusters,
all.clusters.data.frame[row, ])
  }
}

data$cluster = 0 # To store the cluster number for the rows
# Printing the different clusters
```

```r
num.clusters <- nrow(significant.clusters)

for (row in 1:nrow(significant.clusters)){
  for (ct in fls$cluster[[row]]$name){
    ind <- which(data$COUNTYFP10 == ct)
    data[ind, ]$cluster <- num.clusters - row + 1 # most significant cluster
has higher value
  }
}

# Just adjust the plot now.

ggplot(data=data)+geom_sf(size = 0.05, aes(fill=cluster)) +
scale_fill_gradient(high = "red", low = "white", na.value=NA)
```


36°N
34°N
32°N
30°N
28°N
26°N
106°W
104°W
102°W
100°W
98°W
96°W
94°W
cluster
2.0
1.5
1.0
0.5
0.0

```
ggsave("2019_CT_cluster_size_5.png")

## Saving 20 x 30 in image
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.