

SPATIAL ANALYSIS OF PIPELINE ACCIDENTS IN KANSAS STATE USING R

2023-05-16

Ivan Innocent Sekibenga

1. Introduction

1.1 Study area

Pipelines are very crucial and important in the United States of America(USA) for the transportation of natural gas, crude oil, and other products over long distances across states(**Figure 1**). Like other USA states, the state of Kansas(**Figure 2**) has a wide network of pipelines(**Figure 3**) that traverse its 105 counties.

Major US Pipelines

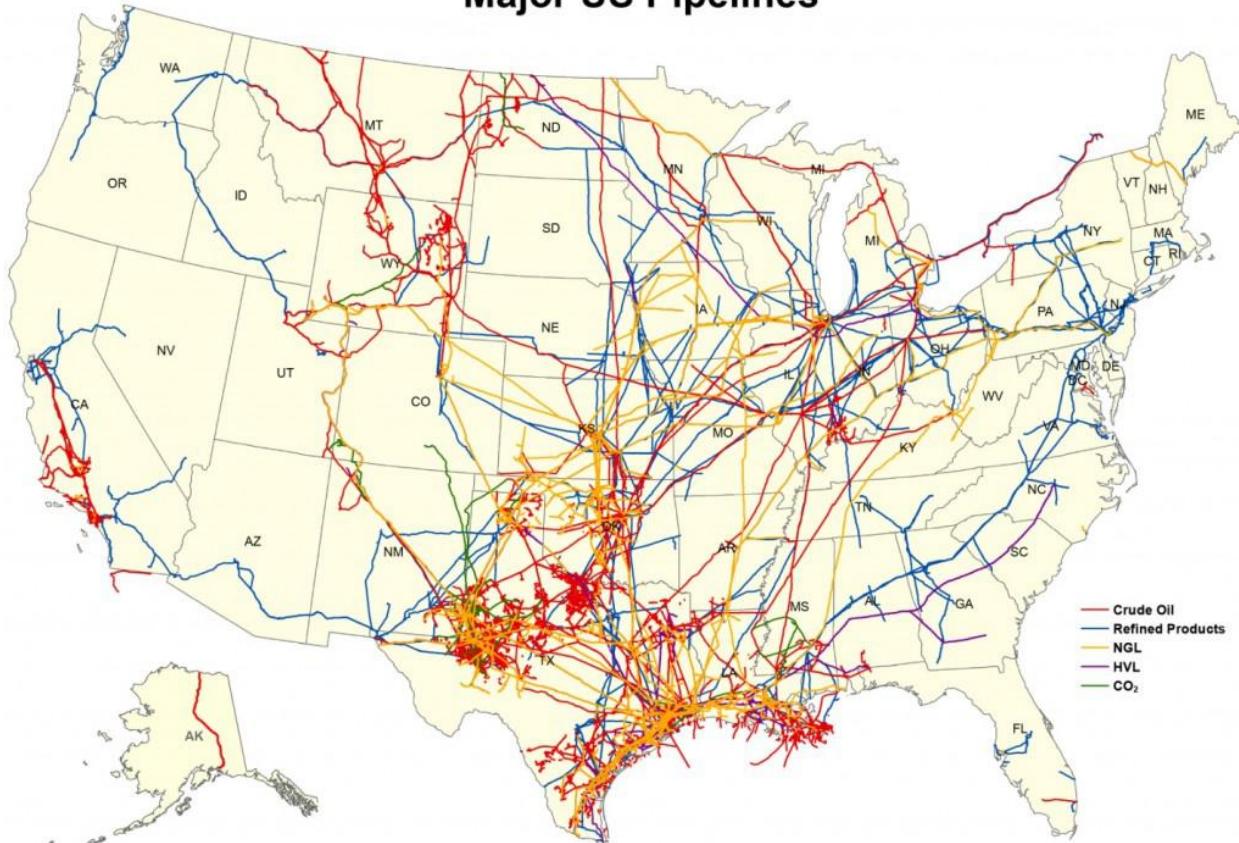


Figure 1: Major pipelines in the USA (Source: American Energy Mapping,2013)

The state of Kansas is bordered by the states of Nebraska, Missouri, Oklahoma, and Colorado to the North, East, South, and West respectively(**Figure 2**). The approximate center of Kansas is located at 38.5° North latitude and 98.0° West longitude. Its capital is Topeka with Wichita as its largest city.

Despite the fact that the transportation of energy resources by pipelines fosters the growth of the economy, development, and stability of society, pipelines suffer from incidents of accidents leading to destruction and pollution of the environment, economic loss, and in some cases, human casualties(Hongfang et al.,2023). Oil spills often happen with accidents involving oil pipelines and such spillage flows to nearby water bodies and soil causing unquantifiable damage. According to Hongfang et al.(2023), hazardous liquid pipeline accidents are mainly caused by corrosion of pipes, equipment failure, welding failure and poor operation. Other natural causes of such accidents are meteorological disasters, climatic and geological disasters such as earthquakes, tornados and winter.

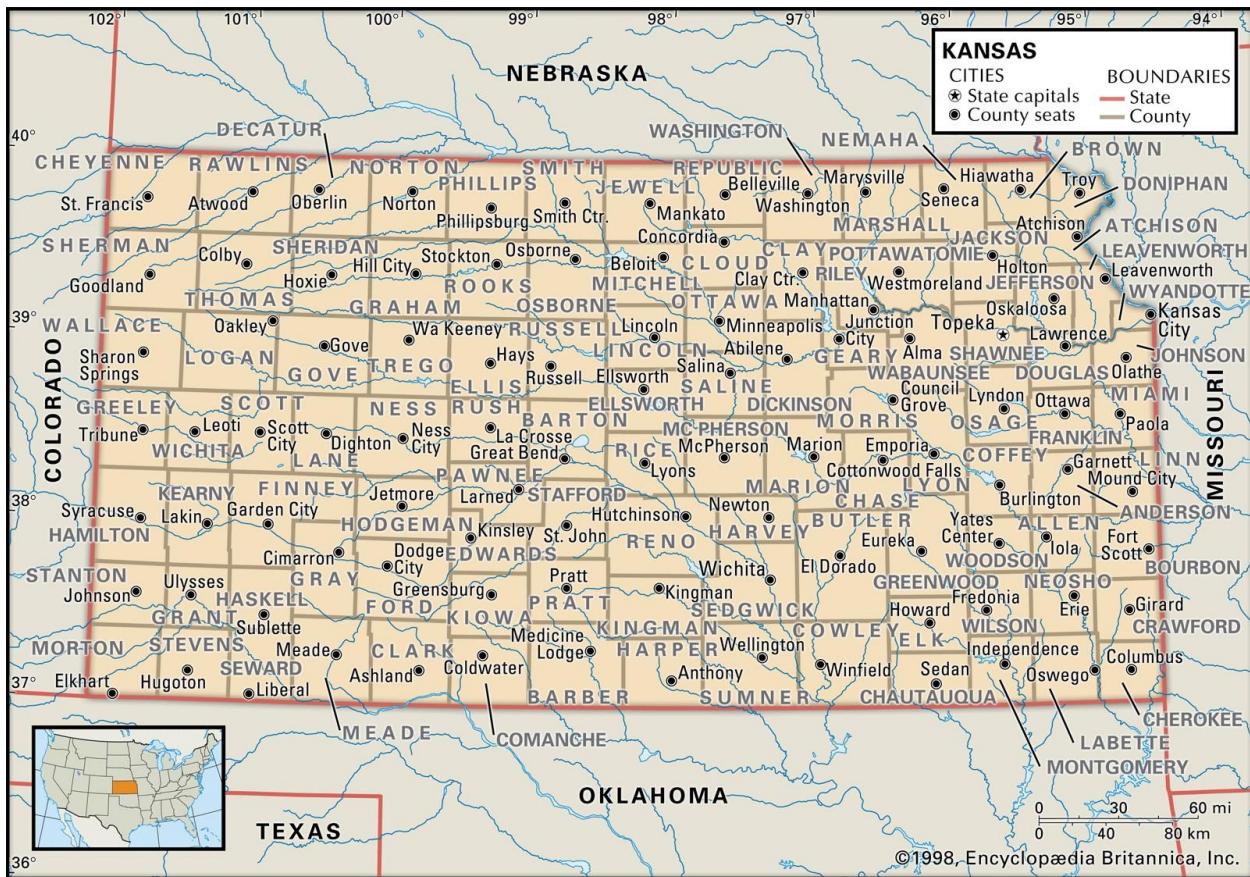


Figure 2: Counties and cities in the state of Kansas (source: Encyclopedia Britannica,1998)

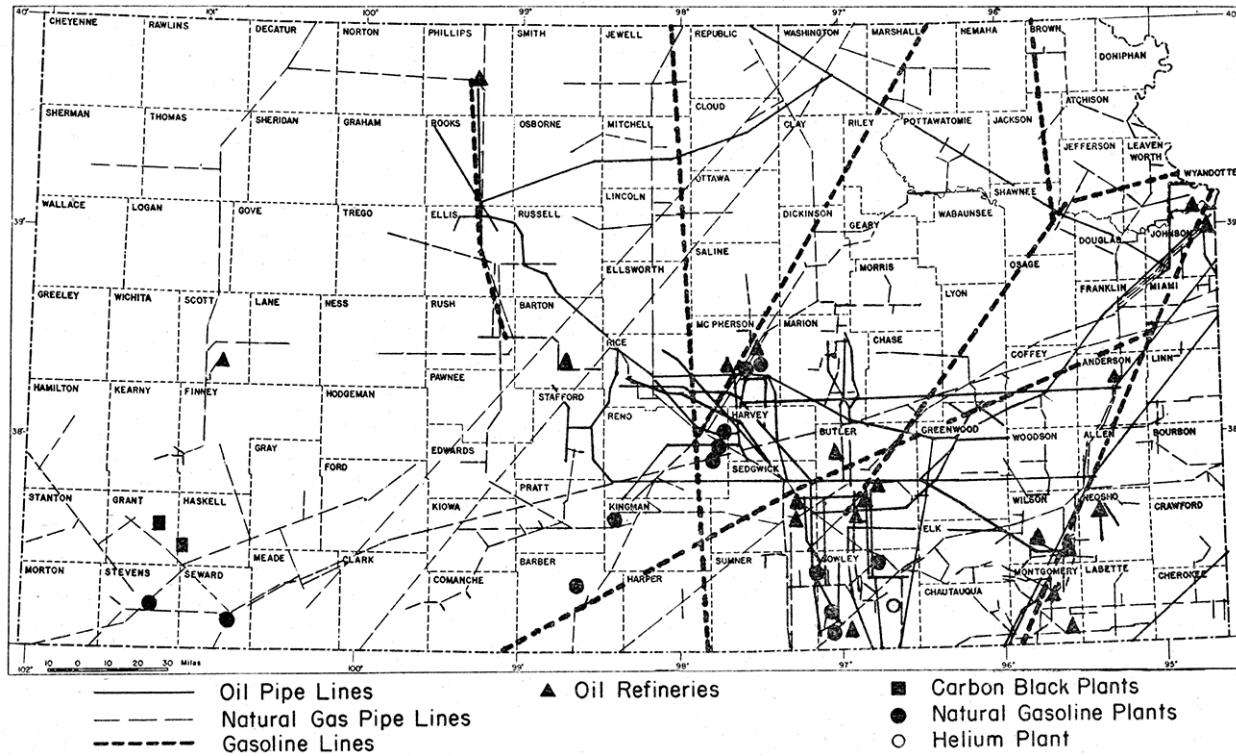


Figure 3: Pipelines in the state of Kansas (source: University of Kansas)

1.2 Literature Review

Huanxin et al.(2022) carried out point density analysis, buffer analysis and cluster analysis on global maritime accidents in ArcGIS. The accident data used stretched from 2010 to 2019. Cluster analysis included hot spot analysis via Getis-Ord statistic and outlier analysis via the Local Moran's I statistic. Their spatial analysis focused on the severity and frequency of these accidents. High severity accident clusters were observed in coastal waters of Philippines, China and South Korea. They compared the results of density analysis to those of cluster analysis and found that cluster analysis avails more information on spatial attributes. However, density analysis was found to be easy to implement and saved computation time.

Prasannakumar et al.(2011) conducted a temporal and spatial analysis of road accidents in the Indian state of Kerala using accident data for 2008. ArcGIS was used for analysis. Hot spot analysis was utilized via the Getis-Ord statistic to identify hot and cold spots and kernel density estimation followed. Spatial auto correlation was implemented via Moran's I method. Results showed that generally there was a clustering pattern in the road accidents especially around educational institutions and places of worship. These were hot-spots for road accidents.

From the literature above, ArcGIS software was used by the authors. Our study will not use ArcGIS but will seek to use R programming language to implement the methods used by those studies. In addition, the literature above explored road and maritime accidents. This study will instead apply the methods used by the authors to conduct spatial analysis of pipeline accidents in Kansas.

1.3 Objective

To mitigate risks due to these accidents and implement effective safety measures, the need to understand the spatial distribution and characteristics of these accidents is very important. This report conducts spatial analysis of pipeline accidents within the state of Kansas. It utilises the R programming language to conduct hot spot analysis, kernel density estimation, spatial auto correlation analysis using Moran's I, buffer analysis

and overlays to conduct the spatial analysis. Maps are also plotted to show spatial distributions and patterns in accident incidents or locations. The overall aim is to discover patterns, trends, and potential areas of concern, thereby providing a doorway for developing targeted strategies for pipeline safety and accident prevention. This endeavor can play a crucial role in fostering a safer, more efficient, and sustainable energy transportation landscape in the state of Kansas.

2. Methods

The methodology used includes spatial autocorrelation, hot spot analysis, point density analysis and buffer analysis with overlays.

2.1 Spatial autocorrelation: Moran's I method

The Spatial autocorrelation (Moran's I method) measures the degree to which nearby locations have similar values or characteristics. It works on both point locations and feature values simultaneously (Prasannakumar et al., 2011). Given a set of points and a corresponding attribute, it investigates whether the presented pattern is dispersed, clustered or random. Moran's I compares the value of the variable at any one location with the value at all other locations and according to Huanxin et al. (2022) can be calculated as:

$$I_i = \frac{(n-1)(x_i - \bar{X}) \sum_{j=1, j \neq i}^n (w_{ij}(x_j - \bar{X}))}{\sum_{j=1, j \neq i}^n (x_j - \bar{X})^2}$$

where I_i is the local Moran's value for point i , n is the count of all data points, x_i and x_j are attribute values (such as accident damages or severity) of points (accident locations) i and j , \bar{X} is the mean of all attribute values and w_{ij} is the spatial weight or reciprocal of the length between these points (Huanxin et al., 2022; Prasannakumar et al., 2011; Aghajan et al., 2017).

According to Prasannakumar et al. (2011), the null hypothesis is that there is absence of spatial clustering and generally, Moran's values close to +1 imply clustering while that close to -1 implies outliers or dispersion. Huanxin et al. (2022) assert that a positive value of I shows that the point is part of either high-high or low-low clustering of values while a negative I value depicts the point as an outlier.

2.2 Hot spot analysis: Getis-Ord statistic

Hot spots are locations in a boundary with a high density or concentration of incidents such as accidents. Getis-Ord statistic and kernel density estimation are the commonly used tools to identify hot spots (Prasannakumar et al., 2011; Aghajan et al., 2017). Getis-Ord statistic categorizes clusters into hot spots having high values and cold spots with low values. Hot spots are identified when a high z-score is obtained together with a small p-value. Cold spots exhibit low negative z-score with a small p-value (Prasannakumar et al., 2011). The stronger the clustering, the higher the z-score but a z-score close to zero shows the absence of clustering (Huanxin et al., 2022). According to Huanxin et al. (2022), the Getis-Ord statistic is calculated using the formula below;

$$G_i^* = \left[\sum_{j=1}^n (w_{ij}x_j) - \bar{X} \sum_{j=1}^n w_{ij} \right] \cdot \left[\sqrt{\frac{\sum_{j=1}^n x_j^2}{n} - \bar{X}^2} \cdot \sqrt{\frac{n \sum_{j=1}^n w_{ij}^2 - \left(\sum_{j=1}^n w_{ij} \right)^2}{n-1}} \right]^{-1}$$

2.3 Point density analysis: Kernel density estimation

This analysis computes the count of data points such as accidents per unit area. It utilizes a neighborhood such as a circle or a rectangle placed at each grid's center. The density of a point element is the quotient

between the sum of counts of all points such as accidents in the neighborhood and the neighborhood's area. From Huanxin et al(2022), if the neighborhood is a circle of radius ρ and centred at k , enclosing N_k accidents, then the accident density, D_k is obtained as

$$D_k^{accident} = N_k(\rho) / (\pi\rho^2)$$

The density for every cell is computed and a distribution of accident density is obtained.

2.4 Buffer analysis

Buffer analysis is for proximity and impact analysis to identify areas within a specific distance around a geographical feature of interest such as an accident point. Huanxin et al.(2022) define a buffer zone as

$$Buffer = \{x | d(x, O) \leq r\}$$

where O is an object, x can be any accident point inside the boundary of the buffer zone. $d(x, O)$ is the smallest length between point x and object O and r is the radius of the neighbourhood. points whose distances between x and O are less or equal to r are said to be inside the buffer.

3. Results and Discussions

This section presents the implementation of the methods using R code to generate plots and maps and other results. The results are interpreted and discussed simultaneously as they are presented.

Loading Libraries

```
library(dplyr) # Data manipulation functions making data analysis faster and easier
library(RColorBrewer) # colour palette library by Cindy Brewer
library(mapsf) # useful map making library
library(maps) # useful map making library
library(sf) # the core simple feature library
library(mapview) # helper library for viewing spatial data
library(tidyverse) # general set of useful libraries
library(systemfonts) # required by mapview
library(leaflet.providers) # required by mapview
library(rgdal) # library to access various CRS amongst other things
library(gridExtra) # Additional functions for arranging 'grid' graphics
library(grid) # Base R plotting system for creating/manipulating grid graphics
library(tidycensus) # Interface to U.S. Census Bureau data
library(ggplot2) # Versatile graphics package for creating complex plots
library(sp) # Classes/methods for spatial data
library(rgdal) # Reading/writing to geographic data formats
library(spatstat) # Analysis of spatial point patterns
library(raster) # Handling/processing/analysis of raster data
library(magrittr) # Forward-pipe operator for cleaner code
library(leaflet) # Creation of interactive web maps
library(spdep) # Classes/methods for spatial dependence and regression
library(ggspatial) # Annotate 'ggplot2' plots with spatial data and models
```

Reading in and pre-processing pipeline accidents data for Kansas state

```
# Reading in pipeline accidents data for Kansas state
accidents <- read_csv("pipeline.csv")

## checking object class of accidents data
class(accidents)

## [1] "spec_tbl_df" "tbl_df"      "tbl"          "data.frame"
# since object is a dataframe,
# create a sf object from coordinates of accidents dataframe
accidents_sf <- accidents %>%
  st_as_sf(coords = c("Longitude", "Latitude"))

# checking object class of accidents_sf object that has been created
class(accidents_sf)

## [1] "sf"          "tbl_df"      "tbl"          "data.frame"
# accidents_sf is a simple feature,
# checking if accidents_sf sf object has coordinate reference system
accidents_sf %>%
  st_crs()

## Coordinate Reference System: NA
accidents_sf %>%
  st_is_longlat()

## [1] NA
# since there is no CRS, we assign WGS84 Geographic Coordinate System
accidents_sf <- accidents_sf %>%
  st_set_crs("OGC:CRS84")

# checking again if accidents_sf sf object has coordinate reference system now
accidents_sf %>%
  st_crs()

## Coordinate Reference System:
##   User input: OGC:CRS84
##   wkt:
##   GEOGCRS["WGS 84",
##             DATUM["World Geodetic System 1984",
##                   ELLIPSOID["WGS 84",6378137,298.257223563,
##                             LENGTHUNIT["metre",1]],
##                   ID["EPSG",6326]],
##             PRIMEM["Greenwich",0,
##                   ANGLEUNIT["degree",0.0174532925199433],
##                   ID["EPSG",8901]],
##             CS[ellipsoidal,2],
##               AXIS["longitude",east,
##                     ORDER[1],
##                     ANGLEUNIT["degree",0.0174532925199433,
##                               ID["EPSG",9122]],
##               AXIS["latitude",north,
##                     ORDER[2],
##                     ANGLEUNIT["degree",0.0174532925199433,
```

```

##           ID["EPSG",9122]]]
accidents_sf %>%
  st_is_longlat()

## [1] TRUE
# object has WGS84 crs.
# extracting accidents for Kansas state that have recorded coordinates
kansas_accidents <- accidents_sf %>%
  subset(State=="KS") %>% subset(`Recorded Long/Lat`=="YES")

# transforming from WGS84 CRS to Kansas Lambert Conformal Conic (NAD83) with
# projected CRS EPSG code 6923
kansas_accidents_p <- kansas_accidents %>%
  st_transform(6923)
kansas_accidents_p %>%
  st_crs()

## Coordinate Reference System:
##   User input: EPSG:6923
##   wkt:
##   PROJCRS["NAD83 / Kansas LCC (ftUS)",
##             BASEGEOGCRS["NAD83",
##                         DATUM["North American Datum 1983",
##                                ELLIPSOID["GRS 1980",6378137,298.257222101,
##                                          LENGTHUNIT["metre",1]]],
##                         PRIMEM["Greenwich",0,
##                                ANGLEUNIT["degree",0.0174532925199433]],
##                         ID["EPSG",4269]],
##             CONVERSION["Kansas DOT Lambert (US Survey feet)",
##                         METHOD["Lambert Conic Conformal (2SP)",
##                                ID["EPSG",9802],
##                                PARAMETER["Latitude of false origin",36,
##                                          ANGLEUNIT["degree",0.0174532925199433],
##                                          ID["EPSG",8821]],
##                                PARAMETER["Longitude of false origin",-98.25,
##                                          ANGLEUNIT["degree",0.0174532925199433],
##                                          ID["EPSG",8822]],
##                                PARAMETER["Latitude of 1st standard parallel",39.5,
##                                          ANGLEUNIT["degree",0.0174532925199433],
##                                          ID["EPSG",8823]],
##                                PARAMETER["Latitude of 2nd standard parallel",37.5,
##                                          ANGLEUNIT["degree",0.0174532925199433],
##                                          ID["EPSG",8824]],
##                                PARAMETER["Easting at false origin",1312333.3333,
##                                          LENGTHUNIT["US survey foot",0.304800609601219],
##                                          ID["EPSG",8826]],
##                                PARAMETER["Northing at false origin",0,
##                                          LENGTHUNIT["US survey foot",0.304800609601219],
##                                          ID["EPSG",8827]],
##             CS[Cartesian,2],
##             AXIS["easting (X)",east,
##                  ORDER[1],
##                  LENGTHUNIT["US survey foot",0.304800609601219]],
##             AXIS["northing (Y)",north,

```

```

##           ORDER[2],
##           LENGTHUNIT["US survey foot",0.304800609601219]],
## USAGE[
##   SCOPE["Topographic mapping (small scale)."],
##   AREA["United States (USA) - Kansas."],
##   BBOX[36.99,-102.06,40.01,-94.58]],
##   ID["EPSG",6923]

```

Reading in and pre-processing Kansas state counties shape file data

```

# reading in shape file of counties in Kansas state
kansas_counties <- st_read("Tiger_2020_Counties.shp")

## Reading layer `Tiger_2020_Counties` from data source
##   `C:\Users\ivan innocent\Desktop\GeospatialDataAnalytics\practical_1\Tiger_2020_Counties.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 105 features and 20 fields
## Geometry type: POLYGON
## Dimension:      XY
## Bounding box:   xmin: -102.0518 ymin: 36.99302 xmax: -94.58839 ymax: 40.00317
## Geodetic CRS:   WGS 84

# transforming Kansas_counties shape file from WGS84 to Kansas LCC/NAD83
# EPSG code is 6923
kansas_counties_p <- kansas_counties %>%
  st_transform(6923)
kansas_counties_p %>%
  st_crs()

## Coordinate Reference System:
##   User input: EPSG:6923
##   wkt:
##   PROJCRS["NAD83 / Kansas LCC (ftUS)",
##     BASEGEOGCRS["NAD83",
##       DATUM["North American Datum 1983",
##         ELLIPSOID["GRS 1980",6378137,298.257222101,
##           LENGTHUNIT["metre",1]]],
##       PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433]],
##       ID["EPSG",4269]],
##     CONVERSION["Kansas DOT Lambert (US Survey feet)",
##       METHOD["Lambert Conic Conformal (2SP)",
##         ID["EPSG",9802]],
##       PARAMETER["Latitude of false origin",36,
##         ANGLEUNIT["degree",0.0174532925199433],
##         ID["EPSG",8821]],
##       PARAMETER["Longitude of false origin",-98.25,
##         ANGLEUNIT["degree",0.0174532925199433],
##         ID["EPSG",8822]],
##       PARAMETER["Latitude of 1st standard parallel",39.5,
##         ANGLEUNIT["degree",0.0174532925199433],
##         ID["EPSG",8823]],
##       PARAMETER["Latitude of 2nd standard parallel",37.5,
##         ANGLEUNIT["degree",0.0174532925199433],
##         ID["EPSG",8824]],
##       ANGLEUNIT["degree",0.0174532925199433],
##       ID["EPSG",8825]]]

```

```

##      PARAMETER["Easting at false origin",1312333.3333,
##      LENGTHUNIT["US survey foot",0.304800609601219],
##      ID["EPSG",8826]],
##      PARAMETER["Northing at false origin",0,
##      LENGTHUNIT["US survey foot",0.304800609601219],
##      ID["EPSG",8827]],
##      CS[Cartesian,2],
##      AXIS["easting (X)",east,
##      ORDER[1],
##      LENGTHUNIT["US survey foot",0.304800609601219]],
##      AXIS["northing (Y)",north,
##      ORDER[2],
##      LENGTHUNIT["US survey foot",0.304800609601219]],
##      USAGE[
##      SCOPE["Topographic mapping (small scale)."],
##      AREA["United States (USA) - Kansas."],
##      BBOX[36.99,-102.06,40.01,-94.58],
##      ID["EPSG",6923]]

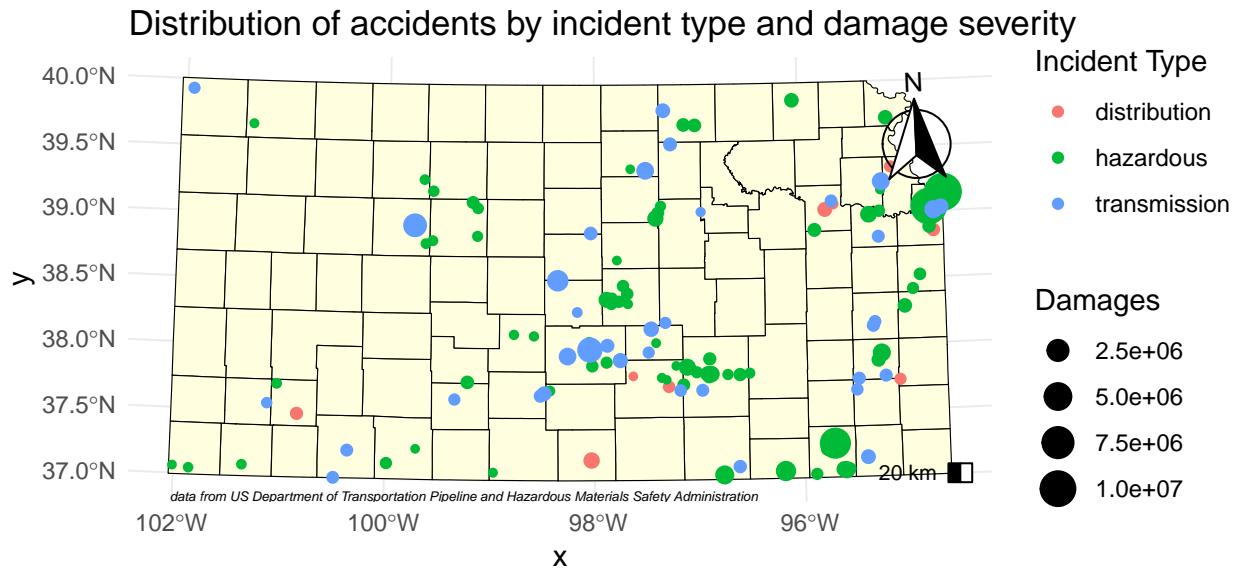
```

3.1 Spatial distribution of pipeline accidents in the state of Kansas

```

# Get the extent of the Kansas counties
xmin <- min(st_coordinates(kansas_counties_p)[, 1])
xmax <- max(st_coordinates(kansas_counties_p)[, 1])
ymin <- min(st_coordinates(kansas_counties_p)[, 2])
ymax <- max(st_coordinates(kansas_counties_p)[, 2])
# Plot
ggplot() +
  geom_sf(data = kansas_counties_p, fill = "lightyellow", color = "black") +
  geom_sf(data = kansas_accidents_p, aes(color = `Incident Type`, size = `Damages`)) +
  scale_color_discrete(name = "Incident Type") +
  annotation_scale(type = "bar", location = "br", width_hint = 0.04) +
  annotation_north_arrow(location = "tr", which_north = "true", style = north_arrow_fancy_orienteering() +
  annotate(geom = "text", x = -Inf, y = -Inf, hjust = 0, vjust = 0,
    label =
      data from US Department of Transportation Pipeline and Hazardous Materials Safety Administration
      size = 1.9, fontface = "italic", color = "black") +
  coord_sf(xlim = c(xmin, xmax), ylim = c(ymin, ymax)) +
  theme_minimal() + theme(plot.margin = margin(t = 2, r = 2, b = 2, l = 2, "pt")) +
  labs(title = "Distribution of accidents by incident type and damage severity",
    color = "Incident Type")

```



counting accidents by incident type

```

# Count of 'hazardous' accidents
hazardous_count <- sum(kansas_accidents_p$"Incident Type" == 'hazardous', na.rm = TRUE)

# Count of 'transmission' accidents
transmission_count <- sum(kansas_accidents_p$"Incident Type" == 'transmission', na.rm = TRUE)

# Count of 'distribution' accidents
distribution_count <- sum(kansas_accidents_p$"Incident Type" == 'distribution', na.rm = TRUE)

# Print the results
print(paste("Hazardous accidents: ", hazardous_count))

## [1] "Hazardous accidents: 100"
print(paste("Transmission accidents: ", transmission_count))

## [1] "Transmission accidents: 42"
print(paste("Distribution accidents: ", distribution_count))

## [1] "Distribution accidents: 13"

```

A combined study of the map above and **Figure 2** reveals that accident incidents due to pipelines carrying hazardous material(points in green) are the most common and most damaging(big points) in the state of Kansas. 100 accident incidents attributed to pipelines carrying hazardous material were calculated. They are clustered and dominant in the counties of Butler, Mc Pherson, Montgomery and Wyandotte. Accident

incidents due to transmission pipelines(points in blue) were 42 occupying second position in terms of frequency and damages caused and are dominant and clustered in the counties of Reno and Harvey. Accidents due to distribution pipelines(point in light red) were only 13 but dispersed and exhibit the least damages as evidenced by their small point sizes.

Identifying the number of pipeline accidents per county

```
# Kansas state has 105 counties. We calculate number of accidents per county
lengths(st_intersects(kansas_counties_p,kansas_accidents_p))

## [1] 2 1 0 0 0 0 0 0 4 2 0 3 0 10 2 0 1 0 0 0 0 0 0 4 1
## [26] 8 0 1 0 2 1 0 0 4 15 0 0 0 1 3 0 0 7 1 2 0 2 0 0 1
## [51] 1 0 1 1 1 0 0 1 0 0 0 0 2 1 0 2 3 0 0 0 2 2 1 1 2
## [76] 1 0 7 1 3 0 1 1 3 0 0 1 0 0 0 1 0 2 3 0 0 1 1 1 1
## [101] 7 0 0 6 0

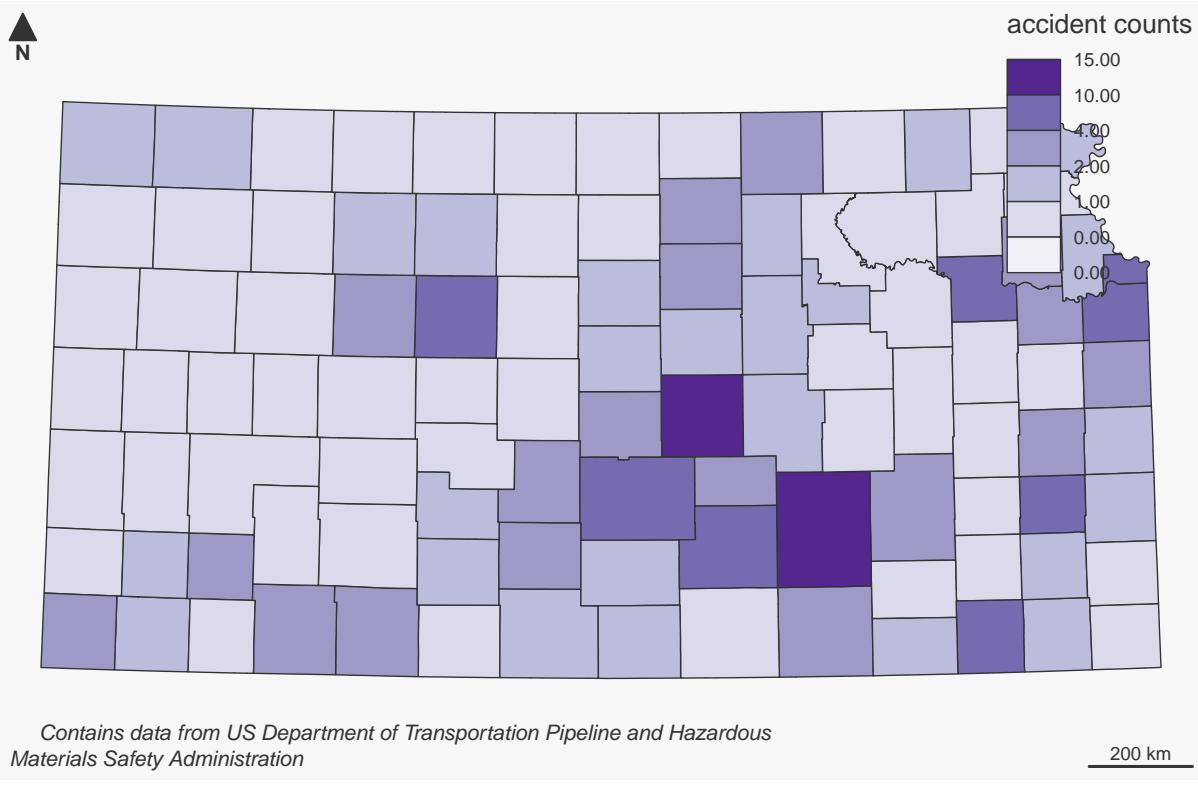
# placing the number of accidents as a new attribute called counts
kansas_counties_p$counts <- lengths(st_intersects(kansas_counties_p,kansas_accidents_p))
```

Map showing the accidents counts per county

```
mf_map(kansas_counties_p, type = "choro", var = "counts",pal = brewer.pal(6,"Purples"),
nbreaks = 6, breaks = 'jenks', leg_title = "accident counts")
mf_title("Accident counts per county in Kansas state from 1986 to 2017", "center", bg = "white",fg = "black",
= FALSE)
# Adding a scale bar
mf_scale()

# Adding a north arrow
mf_arrow()
# adding credits
mf_credits(" Contains data from US Department of Transportation Pipeline and Hazardous
Materials Safety Administration ",
cex = 0.7,
pos = "bottomleft")
```

Accident counts per county in Kansas state from 1986 to 2017



A combined study of the map of accident counts per county above and **Figure 2**, shows that the neighbouring counties of Mc Pherson and Butler had the highest accident counts of between 10 to 15 accidents. Neighbouring counties of Reno and Sedgwick had accident counts of between 4 to 10 while most neighbouring counties in the far west of Kansas state such as Sharman, wallace, Greeley and Hamilton had zero accident counts. The map shows a trend where areas with high accident counts are near other areas with high counts while areas with low accident counts are near areas with low counts. This is evidence of existense of positive spatial auto-correlation.

3.2 Spatial autocorrelation analysis of accidents

```
# Create a grid over Kansas
ks_grid <- st_make_grid(kansas_counties_p, square = TRUE)

# Count the number of accidents in each grid cell
accident_counts <- st_intersects(ks_grid, kansas_accidents_p, sparse = FALSE) %>%
  rowSums() %>%
  `names<-`(seq_along(.))

# Create a data frame with the grid cells and accident counts
accident_density <- data.frame(
  id = names(accident_counts),
  accident_counts = accident_counts,
  geometry = st_geometry(ks_grid)
)

# Convert to sf object
```

```

accident_density <- st_as_sf(accident_density)

# Convert to SpatialPolygonsDataFrame for poly2nb function
accident_density_sp <- as(accident_density, "Spatial")

# Create a neighborhood object
nb <- poly2nb(accident_density_sp, queen = TRUE)
listw <- nb2listw(nb)

# Calculate Moran's I
moran.test(accident_density$accident_counts, listw)

##
## Moran I test under randomisation
##
## data: accident_density$accident_counts
## weights: listw
##
## Moran I statistic standard deviate = 2.7582, p-value = 0.002906
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##          0.127554348     -0.010101010     0.002490749

```

The output above of Moran's I test for spatial autocorrelation is interpreted below as follows:

standard deviate equal to 2.7582: This is the value of Moran's I statistic transformed into a z-score. According to Aghajan et al.(2017), this value enables us to compare the observed value of Moran's I to what we would expect if the data were randomly distributed under the null hypothesis of spatial randomness. A positive z-score indicates that neighboring areas are more similar than would be expected if the data were randomly distributed implying positive spatial autocorrelation, while a negative z-score indicates that neighboring areas are more dissimilar than would be expected by chance implying a negative spatial autocorrelation(Aghajan et al.,2017;Huanxin et al.,2022).

p-value equal to 0.002906: The p-value tells us the probability of observing a value of Moran's I as extreme or more extreme than the one we calculated, given that the null hypothesis of spatial randomness is true(Aghajan et al.,2017). A small p-value below 0.05 allows us to reject the null hypothesis and conclude that there is significant spatial autocorrelation in our accident data(Huanxin et al.,2022).

Alternative hypothesis equal to greater: This indicates the alternative hypothesis for the test. In this case, it's a one-sided test where the alternative hypothesis is that the true value of Moran's I is greater than the expectation under the null hypothesis of spatial randomness(Aghajan et al.,2017).

In conclusion, the test result suggests there is significant positive spatial autocorrelation in the accident counts across grid cells in Kansas. In other words, areas with high counts of accidents tend to be near other areas with high counts, and areas with low counts of accidents tend to be near other areas with low counts, more than would be expected if the accident counts were randomly distributed across space.

Moran scatterplot

```

# Calculate the standardized values of accident counts
standardized_counts <- scale(accident_density$accident_counts)

# Calculate the spatially lagged standardized values
lagged_counts <- lag.listw(listw, standardized_counts)

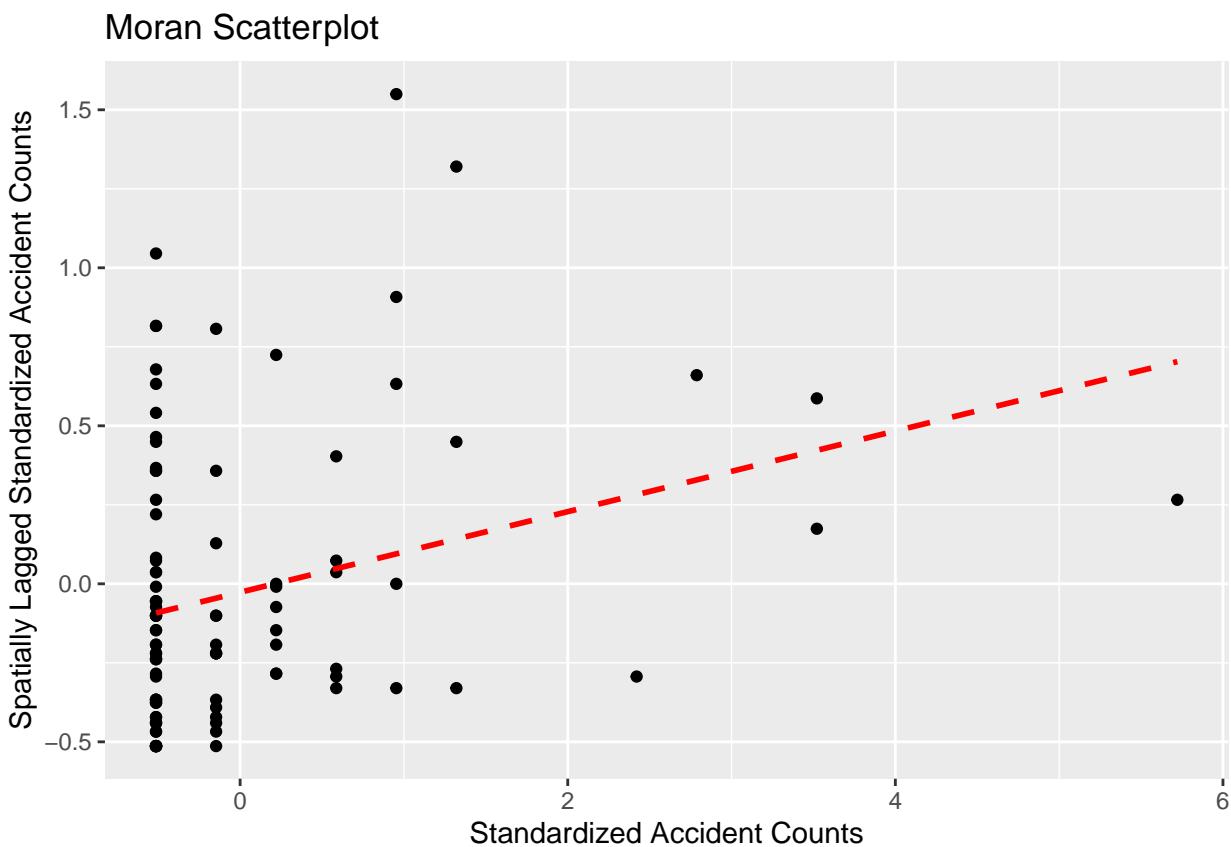
```

```

# Create a data frame
df <- data.frame(standardized_counts = standardized_counts,
                 lagged_counts = lagged_counts)

# Plot
ggplot(df, aes(standardized_counts, lagged_counts)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +
  labs(x = "Standardized Accident Counts",
       y = "Spatially Lagged Standardized Accident Counts",
       title = "Moran Scatterplot")

```



According to Huanxin et al.(2022), In a Moran scatter plot, each point represents a spatial unit and in this case, a grid cell, with its standardized accident count on the x-axis and the mean standardized accident count of its neighbors on the y-axis. The red dashed line is the linear regression line.

Positive spatial autocorrelation where similar values are close together is characterized by a positive slope, as we have in this case, given the positive Moran's I statistic, while negative spatial autocorrelation where dissimilar values are close together would be characterized by a negative slope. The slope of the regression line is also equal to the Moran's I statistic(Huanxin et al.,2022).

plotting local Moran's I for accident density

```

# Calculate local Moran's I
local_moran <- localmoran(accident_density$accident_counts, listw)

```

```

# Add the local Moran's I values to the accident_density data frame
accident_density$local_moran <- local_moran[,1]

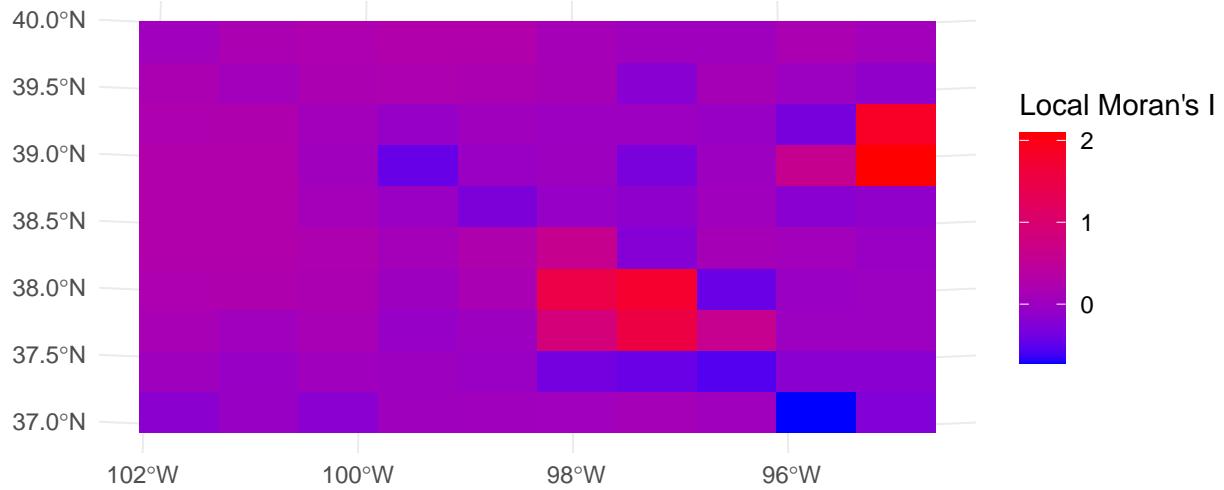
# View the first few rows of the updated data frame
head(accident_density)

## Simple feature collection with 6 features and 3 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: 204917.7 ymin: 363400.2 xmax: 1505733 ymax: 475049.7
## Projected CRS: NAD83 / Kansas LCC (ftUS)
##   id accident_counts           geometry local_moran
## 1 1             3 POLYGON ((204917.7 363400.2... -0.15942029
## 2 2             0 POLYGON ((421720.3 363400.2... -0.03804348
## 3 3             3 POLYGON ((638522.9 363400.2... -0.17391304
## 4 4             1 POLYGON ((855325.5 363400.2... 0.03260870
## 5 5             1 POLYGON ((1072128 363400.2,... 0.05434783
## 6 6             1 POLYGON ((1288931 363400.2,... 0.06521739

# Plot
ggplot() +
  geom_sf(data = accident_density, aes(fill = local_moran), color = NA) +
  scale_fill_gradient(low = "blue", high = "red") +
  theme_minimal() +
  labs(fill = "Local Moran's I",
       title = "Spatial distribution of Local Moran's I values for accident counts")

```

Spatial distribution of Local Moran's I values for accident counts



From the plot above, areas of high accident density shown in bright red colour also have high positive local Moran's I values between +1 and +2. They are areas of high-high value clustering. From **Figure 2**, a cluster exists in the North east of Kansas of counties with high accident density comprised of Jafferson, Johnson, Leavenworth, Douglas and Shawnee. Another cluster with high accident density in the south central of Kansas exists with high-high values comprised of counties like Sedwick, Butler, Mc Pherson and Reno. Areas with blue have negative Moran's values and are outliers or areas of dispersion. Areas in pink have low positive Moran's values near zero showing low-low value clustering. These are clusters of counties with low accident density and accident counts. Such counties include those in the far west such as Sharman, Wallace, Greeley and Hamilton. Generally, positive auto correlation is prevalent in this plot.

3.3 Hot spot analysis

```

# Create a spatial weights matrix
listw <- knn2nb(knearneigh(st_coordinates(kansas_accidents_p), k = 8))
listw <- nb2listw(listw)

# Calculate the Getis-Ord Gi* statistic based on 'Damages'
g_star <- localG(kansas_accidents_p$Damages, listw)

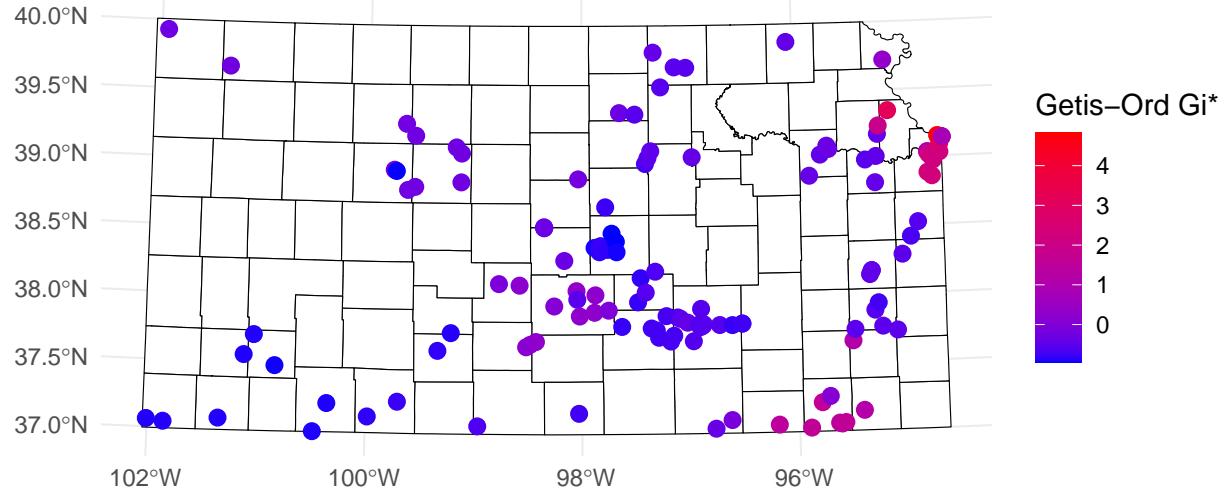
# Add the Getis-Ord Gi* statistic to the accident data frame
kansas_accidents_p$g_star <- g_star

# Get the extent of the Kansas counties
xmin <- min(st_coordinates(kansas_counties_p)[, 1])
xmax <- max(st_coordinates(kansas_counties_p)[, 1])
ymin <- min(st_coordinates(kansas_counties_p)[, 2])
ymax <- max(st_coordinates(kansas_counties_p)[, 2])

# Plot
ggplot() +
  geom_sf(data = kansas_counties_p, fill = "white", color = "black") +
  layer_spatial(kansas_accidents_p, aes(color = g_star), size = 2.5) + # increased point size to 3
  scale_color_gradient(low = "blue", high = "red") +
  coord_sf(xlim = c(xmin, xmax), ylim = c(ymin, ymax)) +
  theme_minimal() +
  labs(color = "Getis-Ord Gi*",
       title = "Hot Spot Analysis of Damages severity in Kansas")

```

Hot Spot Analysis of Damages severity in Kansas



The map points in red have high positive z-scores showing a hot spot and cluster of accidents that caused high damages. Comparing with **Figure 2**, The first such hot spot is in Wyandotte and Johnson counties and another is in Montgomery and Chautauqua counties. The blue map points have low negative z-scores showing accidents clusters of low damages and such cold spots include the counties of Mc Pherson, Harvey and Butler.

3.4 Kernel Density map of pipeline accident incidents in Kansas state

```

# Convert accidents data to a ppp object which is used by spatstat package
accidents_ppp <- as.ppp(st_coordinates(kansas_accidents_p), st_bbox(kansas_counties_p))

# Calculating the Kernel density with the density.ppp function
accidents_kde <- density.ppp(accidents_ppp)

# Converting the kde to a raster object
accidents_kde_raster <- raster(accidents_kde)

# Defining the CRS of the raster object manually
crs(accidents_kde_raster) <- crs(kansas_counties_p)

# Converting the raster object to a SpatialPixelsDataFrame
accidents_kde_sp <- as(accidents_kde_raster, "SpatialPixelsDataFrame")

# Converting the SpatialPixelsDataFrame to an sf object
accidents_kde_sf <- st_as_sf(accidents_kde_sp)

# Plotting the KDE with base R

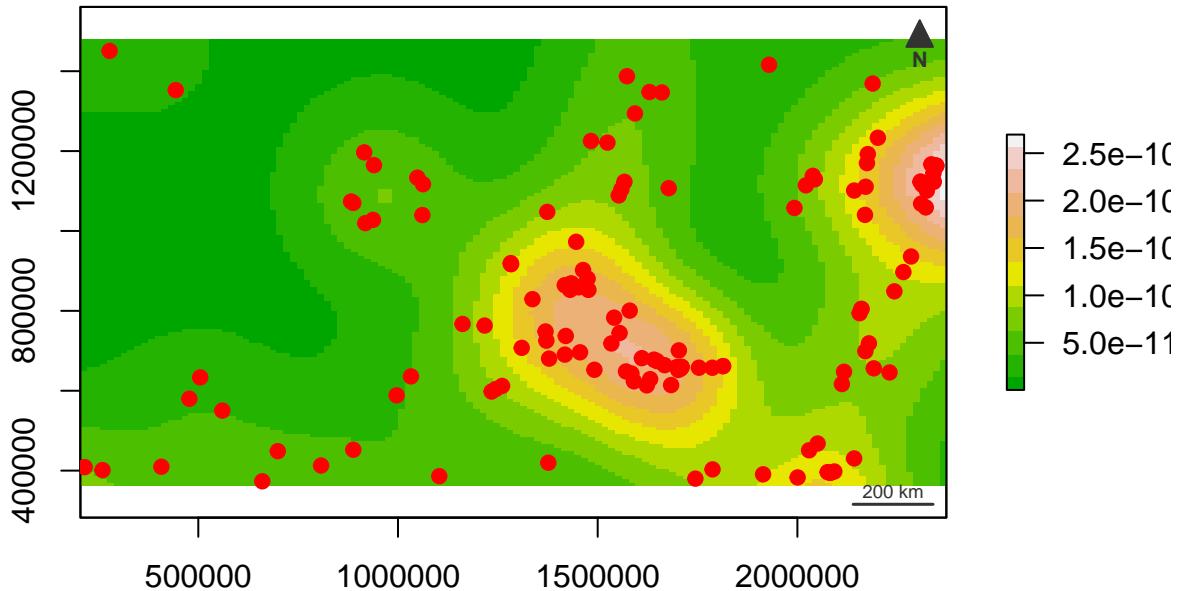
```

```

plot(accidents_kde_raster, main = "Kernel Density Estimation of pipeline accidents in Kansas state",
  col = terrain.colors(12), legend = TRUE)
points(st_coordinates(kansas_accidents_p), pch = 19, col = "red")
mf_arrow(pos = "topright")
mf_scale(pos = "top")

```

Kernel Density Estimation of pipeline accidents in Kansas state



The Kernel Density Estimation map of accidents across counties in Kansas reveals a number of significant hotspots and coldspots. Two peak hotspot areas (light red colour) of relatively high accident density are noticed in the North Eastern and Southcentral areas of Kansas. The North eastern hot spot consists of Wyandotte, Johnson, Douglas and Shawnee counties while the Southern central hotspot consists of Mc Pherson, Harvey, Butler and Sedwick counties. These areas require further investigation to determine contributing factors to the high rate of accidents. Conversely, coldspots or valleys(lighter green colour) were identified in Cheyenne, Morton, Stevens and Seward counties signifying areas of lower accident density. It would be beneficial to study the factors in these areas that contribute to a lower rate of accidents, which might be applied as preventative measures in accident-prone areas.

Reading in and pre-processing environmental data for Kansas

```

# Reading in shapefile for surface water, rivers and streams
kansas_streams <- st_read("Surface_Water_Register_-_Streams.shp")

## Reading layer `Surface_Water_Register_-_Streams` from data source
##   `C:\Users\ivan innocent\Desktop\GeospatialDataAnalytics\practical_1\Surface_Water_Register_-_Streams.shp` using driver `ESRI Shapefile'
## Simple feature collection with 2041 features and 18 fields
## Geometry type: MULTILINESTRING

```

```

## Dimension:      XY
## Bounding box:  xmin: -102.0518 ymin: 36.99798 xmax: -94.58854 ymax: 40.00315
## Geodetic CRS:  WGS 84

# data is in WGS84 crs. We transform it to Kansas LCC/NAD83 projected crs
# EPSG code is 6923
kansas_streams_p <- kansas_streams %>%
  st_transform(6923)
kansas_streams_p %>%
  st_crs()

## Coordinate Reference System:
##   User input: EPSG:6923
##   wkt:
##   PROJCRS["NAD83 / Kansas LCC (ftUS)",
##             BASEGEOCRS["NAD83",
##                         DATUM["North American Datum 1983",
##                                ELLIPSOID["GRS 1980",6378137,298.257222101,
##                                          LENGTHUNIT["metre",1]]],
##                         PRIMEM["Greenwich",0,
##                                ANGLEUNIT["degree",0.0174532925199433]],
##                         ID["EPSG",4269]],
##             CONVERSION["Kansas DOT Lambert (US Survey feet)",
##                         METHOD["Lambert Conic Conformal (2SP)",
##                                ID["EPSG",9802]],
##                         PARAMETER["Latitude of false origin",36,
##                                   ANGLEUNIT["degree",0.0174532925199433],
##                                   ID["EPSG",8821]],
##                         PARAMETER["Longitude of false origin",-98.25,
##                                   ANGLEUNIT["degree",0.0174532925199433],
##                                   ID["EPSG",8822]],
##                         PARAMETER["Latitude of 1st standard parallel",39.5,
##                                   ANGLEUNIT["degree",0.0174532925199433],
##                                   ID["EPSG",8823]],
##                         PARAMETER["Latitude of 2nd standard parallel",37.5,
##                                   ANGLEUNIT["degree",0.0174532925199433],
##                                   ID["EPSG",8824]],
##                         PARAMETER["Easting at false origin",1312333.3333,
##                                   LENGTHUNIT["US survey foot",0.304800609601219],
##                                   ID["EPSG",8826]],
##                         PARAMETER["Northing at false origin",0,
##                                   LENGTHUNIT["US survey foot",0.304800609601219],
##                                   ID["EPSG",8827]]],
##             CS[Cartesian,2],
##               AXIS["easting (X)",east,
##                     ORDER[1],
##                     LENGTHUNIT["US survey foot",0.304800609601219]],
##               AXIS["northing (Y)",north,
##                     ORDER[2],
##                     LENGTHUNIT["US survey foot",0.304800609601219]],
##             USAGE[
##               SCOPE["Topographic mapping (small scale)."],
##               AREA["United States (USA) - Kansas."],
##               BBOX[36.99,-102.06,40.01,-94.58]],
##               ID["EPSG",6923]]

```

Reading in and pre-processing Kansas state census tracts shape file data

```
# reading in shape file of census tracts in Kansas state
kansas_tracts <- st_read("tl_2022_20_tract.shp")

## Reading layer `tl_2022_20_tract` from data source
##   `C:\Users\ivan innocent\Desktop\GeospatialDataAnalytics\practical_1\tl_2022_20_tract.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 829 features and 12 fields
## Geometry type: POLYGON
## Dimension:      XY
## Bounding box:  xmin: -102.0518 ymin: 36.99302 xmax: -94.58839 ymax: 40.00317
## Geodetic CRS:  NAD83

# transforming Kansas_counties shape file from WGS84 to Kansas LCC/NAD83
# EPSG code is 6923
kansas_tracts_p <- kansas_tracts %>%
  st_transform(6923)
kansas_tracts_p %>%
  st_crs()

## Coordinate Reference System:
##   User input: EPSG:6923
##   wkt:
## PROJCRS["NAD83 / Kansas LCC (ftUS)",
##   BASEGEOGCRS["NAD83",
##     DATUM["North American Datum 1983",
##       ELLIPSOID["GRS 1980",6378137,298.257222101,
##         LENGTHUNIT["metre",1]]],
##     PRIMEM["Greenwich",0,
##       ANGLEUNIT["degree",0.0174532925199433]],
##     ID["EPSG",4269]],
##   CONVERSION["Kansas DOT Lambert (US Survey feet)",
##     METHOD["Lambert Conic Conformal (2SP)",
##       ID["EPSG",9802]],
##     PARAMETER["Latitude of false origin",36,
##       ANGLEUNIT["degree",0.0174532925199433],
##       ID["EPSG",8821]],
##     PARAMETER["Longitude of false origin",-98.25,
##       ANGLEUNIT["degree",0.0174532925199433],
##       ID["EPSG",8822]],
##     PARAMETER["Latitude of 1st standard parallel",39.5,
##       ANGLEUNIT["degree",0.0174532925199433],
##       ID["EPSG",8823]],
##     PARAMETER["Latitude of 2nd standard parallel",37.5,
##       ANGLEUNIT["degree",0.0174532925199433],
##       ID["EPSG",8824]],
##     PARAMETER["Easting at false origin",1312333.3333,
##       LENGTHUNIT["US survey foot",0.304800609601219],
##       ID["EPSG",8826]],
##     PARAMETER["Northing at false origin",0,
##       LENGTHUNIT["US survey foot",0.304800609601219],
##       ID["EPSG",8827]],
##   CS[Cartesian,2],
##   AXIS["easting (X)",east,
```

```

##           ORDER[1],
##           LENGTHUNIT["US survey foot",0.304800609601219]],
##           AXIS["northing (Y)",north,
##           ORDER[2],
##           LENGTHUNIT["US survey foot",0.304800609601219]],
##           USAGE[
##               SCOPE["Topographic mapping (small scale)."],
##               AREA["United States (USA) - Kansas."],
##               BBOX[36.99,-102.06,40.01,-94.58]],
##               ID["EPSG",6923]]

# setting up 5km buffer around pipeline accident sites in Kansas tracts since 5km is approximately 1640
# create the buffer
kansas_accidents_buf1 <- kansas_accidents_p %>%
  st_buffer(16405)

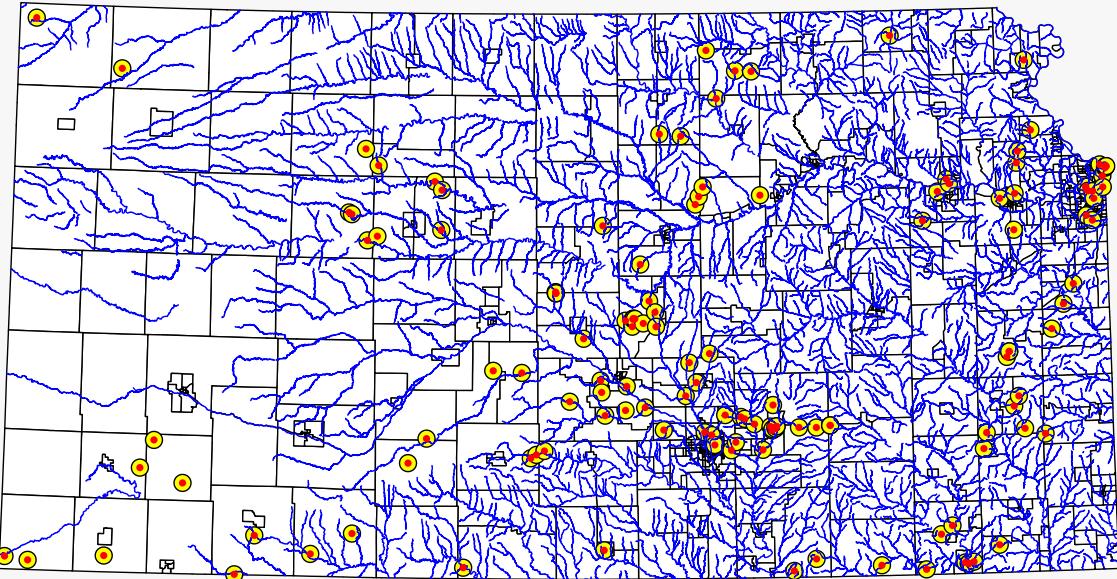
# plotting the 5km undissolved buffers around accident sites
kansas_tracts_p %>%
  mf_map(type = "base",
         col = "white",
         border = "black",
         lty = 1)
kansas_accidents_buf1 %>%
  mf_map(type = "base",
         col = "yellow",
         border = "black",
         lty = 1,
         add = TRUE)
kansas_streams_p %>%
  mf_map(type = "base",
         col = "blue",
         pch = 16,
         cex = 1,
         add = TRUE)

kansas_accidents_p %>%
  mf_map(type = "base",
         col = "red",
         pch = 16,
         cex = 0.5,
         add = TRUE)
# add a title and orientation
mf_title("Accidents sites and rivers with undissolved 5km buffers", 'center',
         bg = "white",
         fg = "black",
         cex = 1.5,
         tab = FALSE)
mf_arrow()
mf_scale(pos = "bottomleft")
mf_credits("surface water data is from kansas geoportal, https://hub.kansascgis.org/",
           cex = 0.7,
           pos = "bottomright")

```

Accidents sites and rivers with undissolved 5km buffers

N



200 km

surface water data is from kansas geoportal, <https://hub.kansasgis.org/>

```
# merging all individual 5km accident buffers in kansas tracts
kansas_buf_merge <- st_union(kansas_accidents_buf1)

# plotting the dissolved 5km buffers around accident sites
kansas_tracts_p %>%
  mf_map(type = "base",
         col = "white",
         border = "black",
         lty = 1)

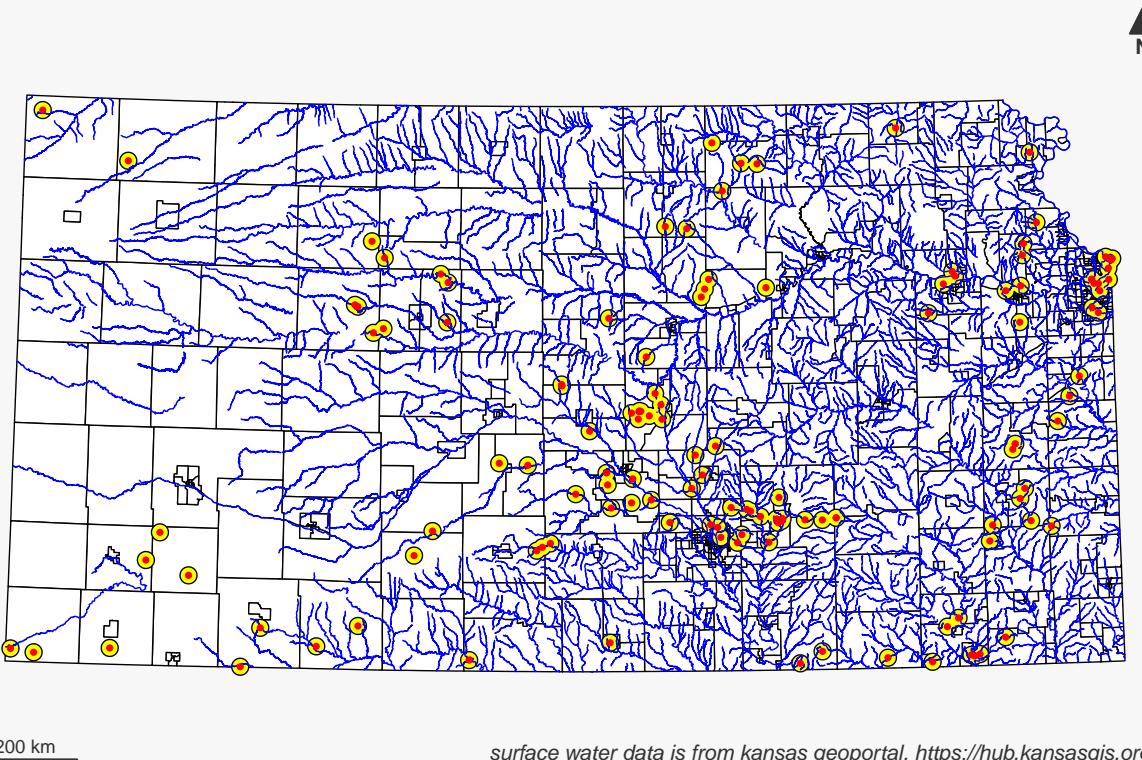
kansas_buf_merge %>%
  mf_map(type = "base",
         col = "yellow",
         border = "black",
         lty = 1,
         add = TRUE)
kansas_streams_p %>%
  mf_map(type = "base",
         col = "blue",
         pch = 16,
         cex = 1,
         add = TRUE)
kansas_accidents_p %>%
  mf_map(type = "base",
         col = "red",
         pch = 16,
```

```

  cex = 0.5,
  add = TRUE)
# add a title and orientation
mf_title("Accidents sites and rivers with dissolved 5km buffers", 'center',
  bg = "white",
  fg = "black",
  cex = 1.5,
  tab = FALSE)
mf_arrow(pos = "topright")
mf_scale(pos = "bottomleft")
mf_credits("surface water data is from kansas geoportal, https://hub.kansascgis.org/",
  cex = 0.7,
  pos = "bottomright")

```

Accidents sites and rivers with dissolved 5km buffers



```

# identify the summation of all river with buffer intersection instances
st_intersects(kansas_streams_p, kansas_buf_merge) %>%
  lengths()%>%
  sum()

## [1] 228

```

create a map of rivers which fall within the 5 km accidents' buffer

```

# intersect the merged buffer area with another polygon feature
accident_streams <- st_intersection(kansas_streams_p, kansas_buf_merge)

```

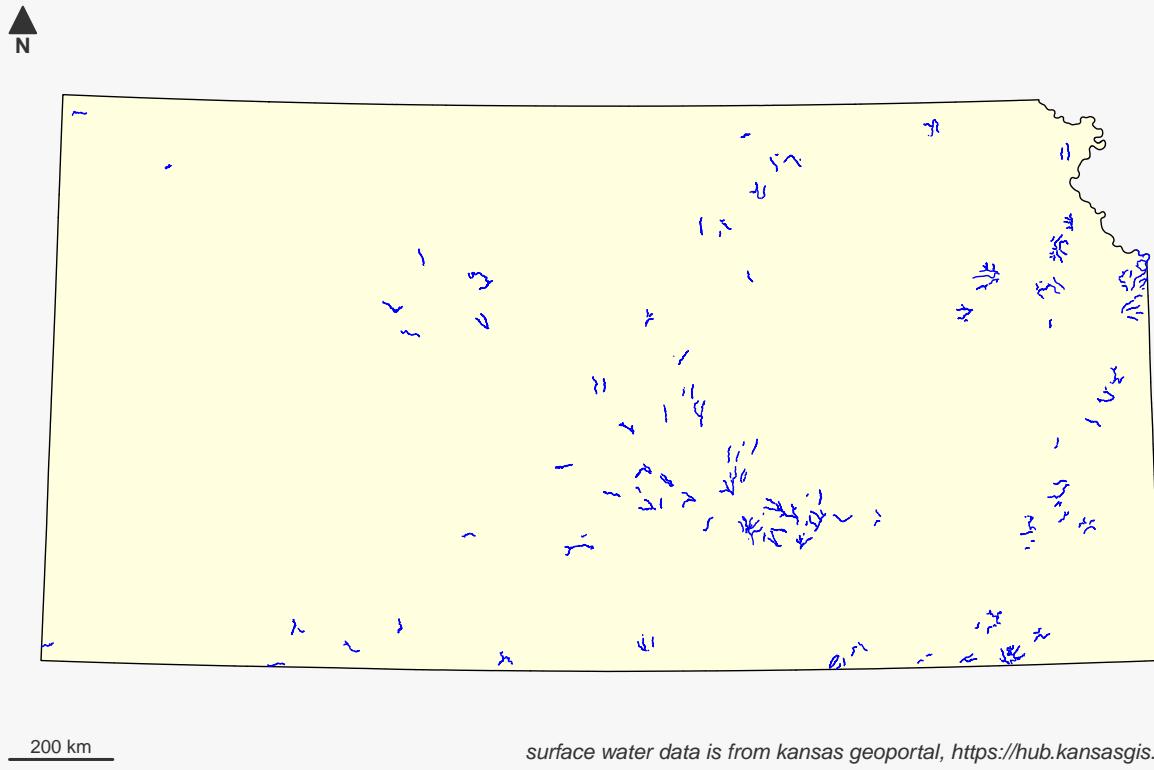
```

# Merge all Kansas county boundaries
kansas_boundary <- st_union(kansas_counties_p)

# plot the results
kansas_boundary %>%
  mf_map(type = "base",
         col = "lightyellow",
         border = "black",
         lty = 1)
accident_streams %>%
  mf_map(type = "base",
         col = "blue",
         border = "black",
         lty = 1,
         add = TRUE)
# add a title and orientation
mf_title("Stream and river sections within the 5km buffer", 'center',
         bg = "white",
         fg = "black",
         cex = 1.5,
         tab = FALSE)
mf_arrow()
mf_scale(pos = "bottomleft")
mf_credits("surface water data is from kansas geoportal, https://hub.kansascgis.org/",
           cex = 0.7,
           pos = "bottomright")

```

Stream and river sections within the 5km buffer



counting the total number of streams and rivers in Kansas state

```
# Count the number of distinct river names that are in the 5km accident buffer
total_distinct_river_count <- n_distinct(kansas_streams_p$RNAME)

# Print the result
print(total_distinct_river_count)

## [1] 924
```

counting the number of rivers and streams that traverse the 5km buffer

```
# Count the number of distinct river names that are in the 5km accident buffer
in_buffer_distinct_river_count <- n_distinct(accident_streams$RNAME)

# Print the result
print(in_buffer_distinct_river_count)

## [1] 160
```

percentage of rivers and streams susceptible to accident damage and pollution

```
160/924*100
```

```
## [1] 17.31602
```

The buffer analysis of oil pipeline accidents and their proximity to rivers and streams in Kansas created a 5 kilometer buffer around each accident location. Calculations show that out of a total of 924 rivers in Kansas, 160 fell within these buffer zones making 228 intersection instances with the buffer zone. This represents a percentage of 17.3% of all rivers in Kansas. The results show that spills from these accidents pose a potential risk to nearby riverine ecosystems by poisoning the water and aquatic life such as fish. In addition, since rivers and streams interconnect via underground networks, the pollution from the 160 rivers in the buffer can still be passed into the remaining rivers that are not in the buffer.

Downloading the 2016 total population data for Kansas census tracts

```
# accessing tidycensus API to download the 2016 total population data
census_api_key("528195527f3b1814bdd4d4052c06e5430b8596e5", install = TRUE, overwrite = TRUE)

## [1] "528195527f3b1814bdd4d4052c06e5430b8596e5"

# creating a data frame of 2016 total population per tract in Kansas state
# KS is the state code for Kansas and B01003_001 is the total population variable
kansas_pop <- get_acs(geography = "tract",
                       variables = "B01003_001",
                       state = "KS",
                       year = 2016,
                       survey = "acs5",
                       output = "wide")

# creating an left join between total population per tract and census tracts
kansas_tracts_pop <- left_join(kansas_tracts_p, kansas_pop, by = "GEOID")

# checking to confirm that CRS is in NAD83/Kansas LCC projected crs
st_crs(kansas_tracts_pop)

## Coordinate Reference System:
##   User input: EPSG:6923
##   wkt:
##   PROJCRS["NAD83 / Kansas LCC (ftUS)",
##             BASEGEOGCRS["NAD83",
##                         DATUM["North American Datum 1983",
##                                ELLIPSOID["GRS 1980",6378137,298.257222101,
##                                         LENGTHUNIT["metre",1]]],
##                         PRIMEM["Greenwich",0,
##                                ANGLEUNIT["degree",0.0174532925199433]],
##                         ID["EPSG",4269]],
##             CONVERSION["Kansas DOT Lambert (US Survey feet)",
##                        METHOD["Lambert Conic Conformal (2SP)",
##                               ID["EPSG",9802]],
##                        PARAMETER["Latitude of false origin",36,
##                                  ANGLEUNIT["degree",0.0174532925199433],
##                                  ID["EPSG",8821]],
##                        PARAMETER["Longitude of false origin",-98.25,
##                                  ANGLEUNIT["degree",0.0174532925199433],
##                                  ID["EPSG",8822]],
##                        PARAMETER["Latitude of 1st standard parallel",39.5,
##                                  ANGLEUNIT["degree",0.0174532925199433],
##                                  ID["EPSG",8823]],
##                        PARAMETER["Latitude of 2nd standard parallel",37.5,
##                                  ANGLEUNIT["degree",0.0174532925199433],
```

```

##           ID["EPSG",8824]],
##           PARAMETER["Easting at false origin",1312333.3333,
##           LENGTHUNIT["US survey foot",0.304800609601219],
##           ID["EPSG",8826]],
##           PARAMETER["Northing at false origin",0,
##           LENGTHUNIT["US survey foot",0.304800609601219],
##           ID["EPSG",8827]],
##           CS[Cartesian,2],
##           AXIS["easting (X)",east,
##           ORDER[1],
##           LENGTHUNIT["US survey foot",0.304800609601219]],
##           AXIS["northing (Y)",north,
##           ORDER[2],
##           LENGTHUNIT["US survey foot",0.304800609601219]],
##           USAGE[
##           SCOPE["Topographic mapping (small scale)."],
##           AREA["United States (USA) - Kansas."],
##           BBOX[36.99,-102.06,40.01,-94.58]],
##           ID["EPSG",6923]]

# filtering out tracts with missing values(NA) for total population
kansas_tracts_pop <- kansas_tracts_pop %>% filter(!is.na(B01003_001E))

```

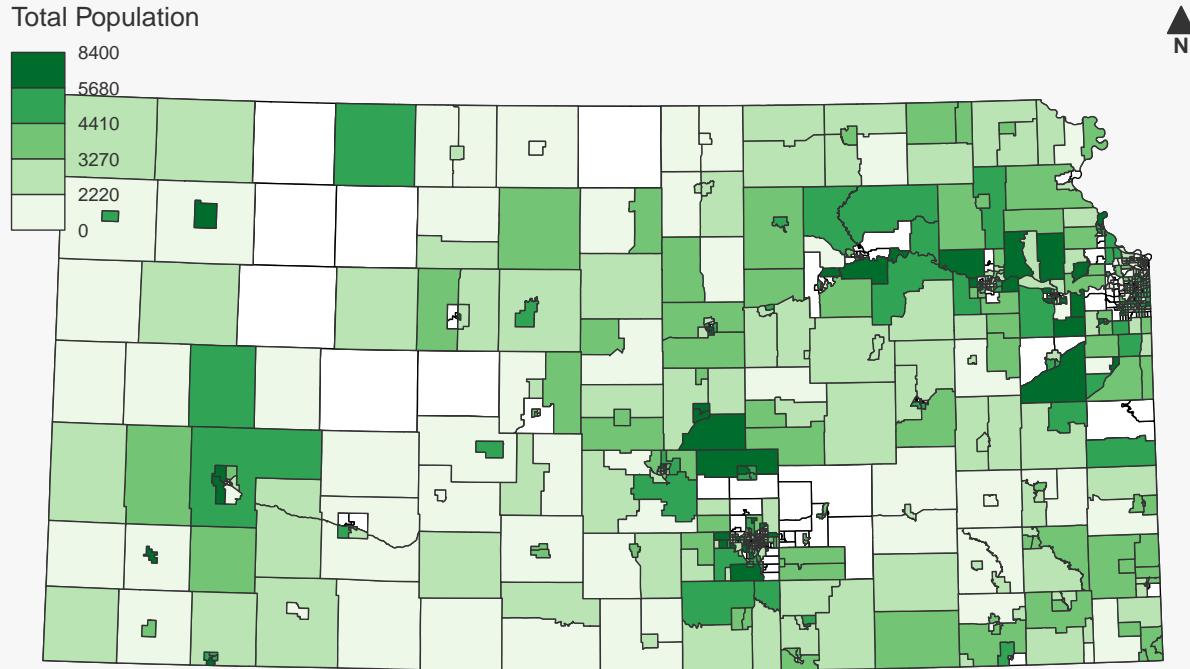
2016 Total population per census tract map for Kansas

```

kansas_tracts_p %>%
  mf_map(type = "base",
         col = "white",
         border = "black",
         lty = 1)
kansas_tracts_pop %>%
  mf_map(type = "choro",
         var = "B01003_001E",
         pal = brewer.pal(5,"Greens"),
         nbreaks = 5,
         breaks = 'jenks',
         leg_title = "Total Population",
         leg_val_rnd = -1,add=TRUE)
# add a title and orientation
mf_title("2016 Total Population in Kansas census tracts ", 'center',
         bg = "white",
         fg = "black",
         cex = 1.5,
         tab = FALSE)
mf_arrow(pos = "topright")
mf_scale(pos = "bottomleft")
mf_credits("population census data is from tidy census library in R",
           cex = 0.7,
           pos = "bottomright",)

```

2016 Total Population in Kansas census tracts



population census data is from tidy census library in R

```
## creating 5km accident buffers in populated census tracts
# merging or dissolving all individual accident buffers in kansas tracts
kansas_buf_merge <- st_union(kansas_accidents_buf1)

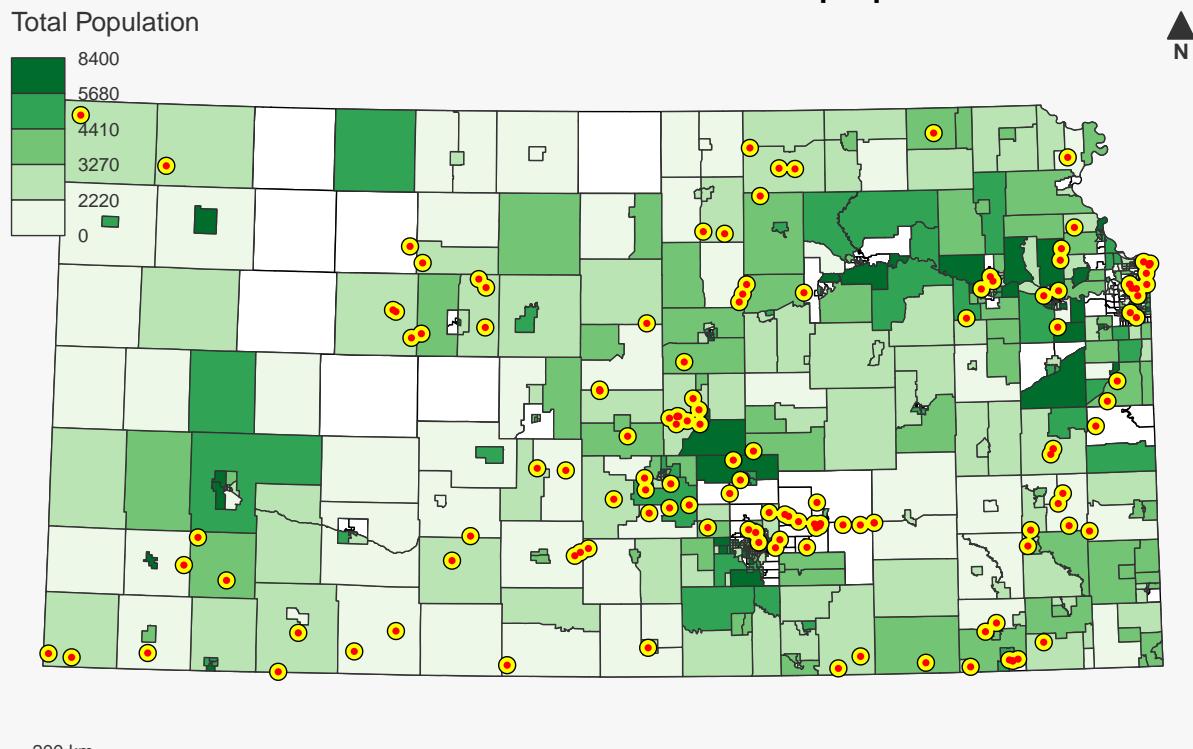
# plotting the 5km dissolved buffers around accident sites
kansas_tracts_p %>%
  mf_map(type = "base",
         col = "white",
         border = "black",
         lty = 1)
kansas_tracts_pop %>%
  mf_map(type = "choro",
         var = "B01003_001E",
         pal = brewer.pal(5,"Greens"),
         nbreaks = 5,
         breaks = 'jenks',
         leg_title = "Total Population",
         leg_val_rnd = -1,add = TRUE)
kansas_buf_merge %>%
  mf_map(type = "base",
         col = "yellow",
         border = "black",
         lty = 1,
         add = TRUE)
```

```

kansas_accidents_p %>%
  mf_map(type = "base",
         col = "red",
         pch = 16,
         cex = 0.5,
         add = TRUE)
# add a title and orientation
mf_title("dissolved accident buffers with 2016 population totals", 'center',
         bg = "white",
         fg = "black",
         cex = 1.5,
         tab = FALSE)
mf_arrow(pos = "topright")
mf_scale(pos = "bottomleft")
mf_credits("population census data is from tidy census library in R",
           cex = 0.7,
           pos = "bottomright")

```

dissolved accident buffers with 2016 population totals



```

# identify the summation of all instances of intersection between buffer and tracts
st_intersects(kansas_tracts_pop, kansas_buf_merge) %>%
  lengths()%>%
  sum()

```

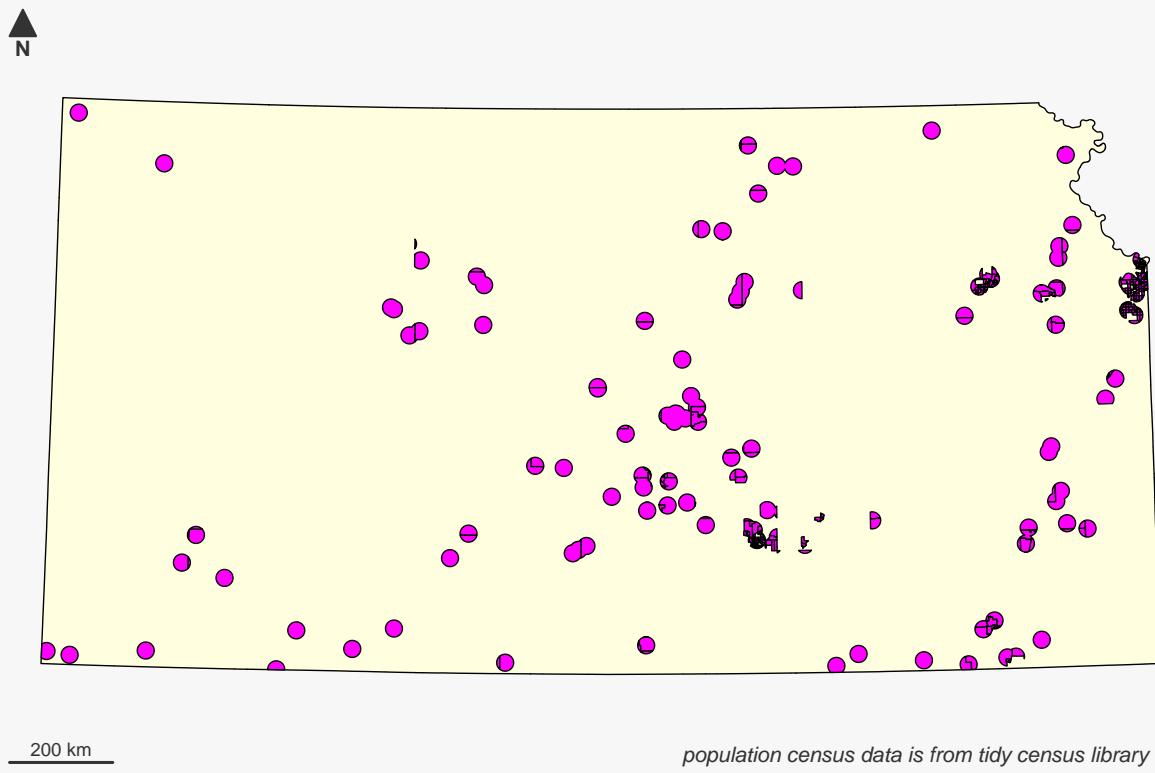
```
## [1] 329
```

create a map of census tracts which fall within the 5 km accidents' buffer

```
# intersect the merged buffer area with another polygon feature
accident_tracts <- st_intersection(kansas_tracts_pop, kansas_buf_merge)

# plot the results
kansas_boundary %>%
  mf_map(type = "base",
         col = "lightyellow",
         border = "black",
         lty = 1)
accident_tracts %>%
  mf_map(type = "base",
         col = "magenta",
         border = "black",
         lty = 1, add=TRUE)
# add a title and orientation
mf_title("Census tracts within the buffer area", 'center',
         bg = "white",
         fg = "black",
         cex = 1.5,
         tab = FALSE)
mf_arrow()
mf_scale(pos = "bottomleft")
mf_credits("population census data is from tidy census library in R",
           cex = 0.7,
           pos = "bottomright")
```

Census tracts within the buffer area



calculating the 2016 total population in Kansas

```
# Sum the population in the intersection layer
total_population_in_kansas <- sum(kansas_tracts_pop$B01003_001E, na.rm = TRUE)

# Print the result
print(total_population_in_kansas)
```

```
## [1] 2354857
```

estimating the total population in the intersection

```
# Sum the population in the intersection layer
total_population_in_intersection <- sum(accident_tracts$B01003_001E, na.rm = TRUE)

# Print the result
print(total_population_in_intersection)
```

```
## [1] 1121230
```

percentage of the population in the buffer

```
1121230/2354857*100
```

```
## [1] 47.61351
```

The buffer analysis of the population affected by the accidents reveals that of the 2,354,857 residents of Kansas, 1,121,230 residents are at potential risk to accidents since they are in the 5km buffer. This figure is 47.6% of the 2016 total population of Kansas. This figure is not realistic since some of the accidents analysed occurred before or after 2016. Generally, despite its ease of implementation in proximity analysis and impact assessment studies, buffer analysis has limitations. It assumes a homogeneous impact within the buffer zone,

ignores influence from terrain or other geographic factors, and its results will vary depending on the chosen buffer distance. In addition, the influence of directional factors such as wind or water flow is ignored.

4. Conclusion

This study utilized spatial autocorrelation, hot spot analysis, point density analysis and buffer analysis with overlays to investigate the potential impact of human and environmental exposure to pipeline accident contamination in the state of Kansas.

Accidents due to hazardous material carrying pipelines had the highest frequency of 100 and caused the greatest damage. These were followed by the 42 due to transmission and 13 accidents due to distribution. Distribution accidents were few and did the least damage. Mc Pherson and Butler counties had the highest accident counts of between 10 to 15. In Kansas, areas with high accident counts tended to be near those with high accident counts and viceversa. High accident density clusters were identified in Johnson, Leavenworth, Douglas, Mc Pherson and Remo counties. Johnson and Wyandotte counties were hot spots of accidents of highest damage. 17.3% of rivers and 47% of the total population in Kansas were at risk of damage to accidents due to being in a 5km buffer from accident sites. Buffer analysis introduced limitations in the study, thus future research should establish the unique causes of these accidents in Kansas. In addition, more efficient methods such as k-means clustering can be tried for spatial analysis to improve results' accuracy.

5. References

- Aghajani,M.A., Dezfoulian,R.S., Arjroody,A.R. and Rezaei,M. (2017) 'Applying GIS to identify the spatial and temporal patterns of road accidents using spatial statistics', *Transportation Procedia*, 25, pp.2126–2138. doi:10.1016/j.trpro.2017.05.409
- Hongfang,L., Xi,D. and Qin, G. (2023) 'Environmental risk of oil pipeline accidents', *Science of The Total Environment*, 874, 162386. doi: 10.1016/j.scitotenv.2023.162386
- Huanxin,W., Liu,Z., Wang,X. and Wang,J. (2022) 'GIS-based analysis on the spatial patterns of Global Maritime accidents', *Ocean Engineering*, 245, 110569. doi: 10.1016/j.oceaneng.2022.110569
- Prasannakumar,V.,Vijith,H.,Charutha,R. and Geetha,N. (2011) 'Spatio-Temporal Clustering of Road Accidents: GIS Based Analysis and Assessment', *Procedia Social and Behavioral Sciences*, 21, pp.317–325. doi:10.1016/j.sbspro.2011.07.020