Griffin Shelor

13-12-2023

Spatial Analysis

Spatial Analysis Final Exam

1. Answer:

Plot envelopes to visualize clustering

Mad.test on envelopes to statistically examine clustering

Envelope objects

Envelope function from spatstat package using Gest and Kest functions

ppp object

as.ppp()

sf points object

st\_as\_sf(), assign CRS of NAD 83 UTM Zone 11

Dataframe with x and y data

read.csv()

CSV with x and y data

Use bw.diggle() from spatstat package to estimate bandwidth for Kernel Density Estimation (KDE)

No

Intensity of impacts is constant throughout bounding box of campsite and activity points

Yes

Check CSR

Use density function to create kernel surface

Plot kernel to visualize point density

1. Answer:

terra::rast() to read in rasters as spatrasters

ppp object from question 1

Make sure rasters are projected in same CRS as points

Unmarked ppp

Unmark ppp

Use ppm(ppp ~ covariates) as well as (ppp ~ 1) to test null since ppm is very useful for modeling pattern of data driven by covariates

Transform data if needed

Summarize models to examine statistical significance of covariates used in each model

Evaluate models

Plot fitted trend of best model

Compare AIC of models

1. Answer: IDW analysis because it is useful for interpolating values like the question asks for and also is more commonly used than kriging in atmospheric and hydrological applications

Find minimum distance threshold using min\_distthreshold

st\_as\_sf(), coords = (x, y), assign appropriate CRS

CSV of station points with values

Create prediction grid

library(spdep)

library(rgeoda)

sf object

Create distance-based nearest neighbors

IDW model

Moran.test for spatial autocorrelation in pollutant data

Identify IDP which best fits points

Plot IDW output as spatraster

vect(), rasterize()

1. Answer:

terra::rast() to read in raster layers as spatrasters

Or

Shapefile/gpkg of plant points with values

CSV of plant points with values

st\_as\_sf(), assign appropriate crs

read.csv()

read\_sf(), assign appropriate crs

Assign same CRS as sf object

Data frame

Extract raster values to points

sf object

Check spatial autocorrelation of plant height, two-sided

spatrasters

Compare variograms with different covariates by creating linear model comparing fitted variogram values to observed values

terra::as.points()

sf points with raster values

st\_as\_sf()

Transform data if necessary

Fit variogram with covariates

Compare AIC of linear models

Rasters as sf object

krige() predicting plant height based on covariates in best model

rasterize(vect()) and plot kriged output

1. Areal analysis, given that the data is reported aggregated at a polygon level

csv of dropout data by school zones, if necessary

Shapefile/gpkg of income data

Shapefile/gpkg of school zones

Create raster with same extent and CRS as larger geometry

read\_sf(), assign same crs as school zones

read\_sf(), assign crs

Rasterize income data

If necessary, join dropout data to school zone sf object

Aggregate income data to school zone level

OR

sf object of school zones and dropout data

Extract income raster values to school zones as new field

Subset geometries so only the polygons which align/overlap remain

Remaining school zones with dropout data and aggregated income

st\_interpolate() to grab intersecting portions of remaining polygons and average income data by amount of area, assign to school zone sf object

Assign school zone ID back to school zone sf object

knn2nb() to use k nearest neighbors to assign nearest neighbors to each school zone

Remaining school zones with dropout data and aggregated income

Moran.test() for spatial autocorrelation

Create lists of neighbor weights using nb2listw()

spautolm(dropout ~ income) to run SAR model with different weights lists

School zones with dropout, income, and fitted trend data

Add fitted values from best model to school zones

Plot trend

AIC on different SAR models

1. The authors found that spatial autocorrelation existed up to 1600 km with regards to species richness, then decreased up to 3700 km where it had negative spatial autocorrelation, and then no significant correlation for distances greater than 3700 km. Mean daily temperature (MINT), annual mean temperature (ANNT), and annual potential evapotranspiration (PET) all showed continuously decreasing trends in spatial autocorrelation, with Moran’s I scores at short distances but negative spatial autocorrelation as distance increased. Only annual actual evapotranspiration (AET) showed a different trend, with high spatial autocorrelation at short and large distances but slightly negative spatial autocorrelation at medium distances. According to this paper, GLS can potentially improve our understanding of these systems by taking spatial autocorrelation into account in the residuals, which OLS does not do. Since the residuals are assumed to have an exponential relationship in a GLS model, scale effects are factored into the error calculations for the model. This is meaningful since it causes GLS to weight variables which are more impactful at smaller scales rather than variables which may be considered globally spatially autocorrelated. GLS therefore prioritizes variables with more local impacts. While GLS does deprioritize variables with high global spatial autocorrelation, it does not mean that spatial autocorrelation should be ignored. Instead, it means that GLS is better suited to help us understand systems with relationships between predictor and response variable that are correlated at long distances with no obvious local cause. GLS can better reveal local spatial patterns in these sorts of situations and, as in the paper, provide more detailed insight into what processes drive spatial variation in species richness.