STP598sta: Spatiotemporal Analysis

Homework 1

Name: Jiseon Yang; NetID:

Due 11:59pm Sunday Sept 15 2024

# Question 1

1. Given latitude (*θ*) and longitude (*λ*) of two locations *P*1 and *P*2. Obtain the formula to calculate the geodesic distance *D*.

A math equations on a piece of paper

Description automatically generated

1. Use the earth radius *R* = 6371 km. Can you compute the geodesic distances:

(1) between Phoenix (33.28N, 112.03W) and Chicago (41.52N, 87.39W), and

A close-up of a math problem

Description automatically generated

(2) between New York (40.43N, 73.56W) and Los Angeles (34.03N, 118.15W)?

A close-up of a math note

Description automatically generated

# Question 2

Consider the time series, *Yt* = *X* = sin(*ωt* + *θ*) (so *X* is the amplitude, *ω* is the frequency and *θ* is the phase) where *X* is distributed with mean 0 and variance 1 independent of *θ* ∼ unif(−*π, π*)). Show that *Yt* is weakly stationary.

A page of math equations

Description automatically generated with medium confidence

# Question 3

Show that a valid variogram *γ*(·) satisfies the negative definiteness condition, i.e. for any set of locations

**s**1*,* · · · *,* **s***n* and any set of *a*1*,* · · · *, an* such that Σ *n ai* = 0, we have

*i*=1

ΣΣ ai aj γ(si - sj) ≤ 0

# A math equations on a piece of paper Description automatically generated

# Question 4

Consider the coalash data frame in the gstat package in R and available from [here.](https://www.counterpointstat.com/uploads/1/1/9/3/119383887/coal.ash.txt) This data comes from the Pittsburgh coal seam on the Robena Mine Property in Greene County, PA (Cressie, 1993, p. 32). This data frame contains 208 coal ash core samples (the variable coal in the data frame) collected on a grid given by *x* and *y* planar coordinates (*not* latitude and longitude).

1. Plot the sampled sites embedded on a map of the region. Add contour lines to the plot.

A graph of a graph showing a graph of a coal ash

Description automatically generatedA graph showing a structure of a coal ash interpolated surface

Description automatically generated

A diagram of a graph

Description automatically generated

1. Provide a descriptive summary (histograms, stems, quantiles, means, range, etc.) of the variable coal in the data frame.

A close up of numbers

Description automatically generated

A number on a white background

Description automatically generated

A close-up of numbers

Description automatically generated

A graph of coal content and a box plot of coal content

Description automatically generated

1. Plot variogms and correlograms of the response and comment on the need for spatial analysis here.

A comparison of a diagram

Description automatically generated with medium confidenceA graph with lines and dots

Description automatically generated

* The semivariance is small when the distance between points is small
* Both the classical and robust semivariograms show that nearby points are more similar,
* As the distance increases, the semivariance increases, showing that points further apart are less similar.
* There is **a strong relationship between proximity and similarity**.
* **Correlogram:** there is a **positive correlation for nearby observations** (up to around 5 units), but this correlation diminishes and turns negative as the distance increases.
* After a certain distance (around 15-20 units), the semivariance levels off. This indicates that points beyond this distance no longer exhibit spatial correlation.
* The variograms indicate **a strong need for spatial analysis** because the similarity between data points decreases with increasing distance, and this spatial dependence must be accounted for in any modeling or prediction tasks.

1. If you think that there is need for spatial analysis, arrive at your best estimates of the range, nugget, and sill.

A group of graphs with blue lines

Description automatically generated

1. **Range (**∅): the distance at which the spatial correlation becomes insignificant
   * Spatial dependence could exist for distances of up to 30 or even 40 units, meaning that samples within this range exhibit spatial correlation. Beyond this distance, the spatial autocorrelation diminishes.
2. **Sill (**σ2): the total variance when spatial correlation is no longer present (when the semivariogram levels off)
   * the sill is centered around 1.5 to 2.0, meaning the total variance in coal ash content beyond the spatial correlation is in this range.
3. **Nugget (**τ2): spatial variability at very short distances (or noise)
   * the nugget appears to vary between 0.4 and 0.8, indicating some short-range variability or measurement noise in the coal ash content data.

The data exhibits clear spatial patterns, and **the need for spatial analysis is justified by the presence of spatial dependence over a range of approximately 30 units. The nugget and sill provide additional insights into the small-scale variability and overall variance in the data.**

The SAR, CAR, and Bayesian kriging models can help in understanding the spatial patterns and in providing more accurate predictions by incorporating spatial dependencies. The spatial patterns are shown below graphs.

A map of the same type of structure

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

A screenshot of a computer program

Description automatically generated

Code :

# Install and load necessary packages

# install.packages("gstat")

# install.packages("akima")

# install.packages("maps")

library**(**gstat**)**

library**(**akima**)**

library**(**maps**)**

# Load the coalash data

data**(**coalash, package **=** "gstat"**)**

### (a)

# Directly plot the sampled sites using x and y planar coordinates and add conotour lines to the plot

plot**(**coalash**$**x, coalash**$**y, main **=** "Sampled Sites of Coal Ash", xlab **=** "X Coordinate", ylab **=** "Y Coordinate", pch **=** 20, col **=** "blue"**)**

A graph of a graph of a graph

Description automatically generated with medium confidence

# Interpolate the data to create a continuous surface

coalash\_interp **<-** interp**(**coalash**$**x, coalash**$**y, coalash**$**coal, extrap **=** **TRUE)**

contour**(**coalash\_interp, add **=** **TRUE)**

A diagram of a coal ash

Description automatically generated

# Add image plot and contour lines

image**(**coalash\_interp, xlim **=** range**(**coalash**$**x**)**, ylim **=** range**(**coalash**$**y**)**, xlab **=** "X Coordinate", ylab **=** "Y Coordinate", main **=** "Coal Ash Interpolated Surface"**)**

contour**(**coalash\_interp, add **=** **TRUE)**

A yellow and orange oval with black dots

Description automatically generated with medium confidence

# Generate a 3D surface plot using persp()

persp**(**coalash\_interp**$**x, coalash\_interp**$**y, coalash\_interp**$**z,

theta **=** **-**30, phi **=** 30, col **=** "lightblue",

xlab **=** "X Coordinate", ylab **=** "Y Coordinate", zlab **=** "Coal Ash Content",

main **=** "3D Surface Plot of Coal Ash Content"**)**

A graph of coal ash content

Description automatically generated

### (b)

par**(**mfrow **=** c**(**1, 2**))**

# Descriptive statistics for the coal variable

summary**(**coalash**$**coal**)**



# Plot a histogram

hist**(**coalash**$**coal, main **=** "Histogram of Coal Ash Content", xlab **=** "Coal Ash Content", col **=** "lightblue", border **=** "black"**)**

A graph of a graph

Description automatically generated with medium confidence

# Stem-and-leaf plot

stem**(**coalash**$**coal**)**

A number on a white background

Description automatically generated

# Quantiles of coal ash content

quantile**(**coalash**$**coal**)**

A number with a number of numbers

Description automatically generated with medium confidence

# Boxplot

boxplot**(**coalash**$**coal, main **=** "Boxplot of Coal Ash Content", ylab **=** "Coal Ash Content", col **=** "lightgreen"**)**

A graph with a green and black line

Description automatically generated

### (c) Geostatistical analysis of coalash data

#install.packages("geoR") # Install geoR if not already installed

library**(**geoR**)**

obj **=** cbind**(**coalash**$**x, coalash**$**y, coalash**$**coal**)**

# Convert the matrix into a geo-referenced object using geoR's as.geodata function

coalash.geo **=** as.geodata**(**obj, coords.col **=** 1**:**2, data.col **=** 3**)**

# Calculate the classical semivariogram using the variog function

coalash.var **=** variog**(**coalash.geo, estimator.type **=** 'classical'**)**

# Calculate the robust semivariogram

coalash.var.robust **=** variog**(**coalash.geo, estimator.type **=** 'modulus'**)**

# Set up the plotting layout for side-by-side semivariograms

par**(**mfrow **=** c**(**1, 2**))**

# Plot the classical semivariogram

plot**(**coalash.var, main **=** 'Classical Semivariogram'**)**

# Plot the robust semivariogram

plot**(**coalash.var.robust, main **=** 'Robust Semivariogram'**)**

A diagram of a semivarigram

Description automatically generated with medium confidence

# Fit an exponential model to the robust semivariogram

coalash.var.fit **=** variofit**(**coalash.var.robust,

ini.cov.pars **=** c**(**2.0, 0.5**)**,

cov.model **=** 'exponential',

fix.nugget **=** **FALSE**,

nugget **=** 1.0**)**

print**(**coalash.var.fit**)**

A white background with black text

Description automatically generated

# Exponential covariance model was fitted to the variogram using weighted least squares (WLS)

# parameter estimates:

# tausq: 0.6835, spatial variability at very small distance, probability due to measurement noise or microscale vriability

# sigmasq: 1.2135, partial sill, The larger the partial sill, the more variance is explained by spatial correlation

# phi: 7.1813, range parameter, which controls how quickly the covariance between two points decays as a function of distance.

# In an exponential model, the covariance decays rapidly, but spatial dependence is still present beyond this range.

# How far apart points can be before they are no longer spatially correlated.

# Practical Range with cor=0.05 for asymptotic range: 21.51322

# :the distance at which the spatial correlation becomes negligible (when the correlation drops to ~5%).

# :samples located within 21.51 units of each other are likely to exhibit some degree of spatial autocorrelation, while those farther apart are not strongly correlated.

# variofit: minimised weighted sum of squares = 236.0256

# : how well the model fits the observed variogram. The smaller the value, the better the fit.

# : a reasonable fit, yet still could be improved by adjusting the model or covariance structure.

# Perform likelihood-based fitting for the exponential model using likfit

coalash.lik.fit **=** likfit**(**coalash.geo,

ini.cov.pars **=** c**(**2.0, 0.5**)**,

cov.model **=** 'exponential',

trend **=** 'cte',

fix.nugget **=** **FALSE**,

nugget **=** 1.0,

nospatial **=** **TRUE**,

lik.method **=** 'ML'**)**

print**(**coalash.lik.fit**)**

# likfit: estimated model parameters:

# beta tausq sigmasq phi

# "9.6769" "1.0350" "0.6896" "7.0211"

# Practical Range with cor=0.05 for asymptotic range: 21.03324

#

# likfit: maximised log-likelihood = -321

### Correlograms

# Create correlogram using the coalash data

# install.packages("ncf")

library**(**ncf**)**

# Define coordinates and data

x **<-** coalash**$**x

y **<-** coalash**$**y

z **<-** coalash**$**coal

# Generate correlogram

coalash.corr **<-** correlog**(**x, y, z, increment **=** 1, resamp **=** 0**)**

# Plot the correlogram

plot**(**coalash.corr**)**

A graph with lines and dots

Description automatically generated

#################################################### Kriging

### (d)

# Bayesian kriging provides posterior distributions for the spatial parameters,

# rather than just point estimates (as in frequentist methods).

# The advantage is that it incorporates uncertainty in these parameters

# by giving you a range of possible values (posterior distribution) instead of a single point estimate.

library**(**geoR**)**

# Create the geo-referenced object using coalash data

obj **=** cbind**(**coalash**$**x, coalash**$**y, coalash**$**coal**)**

coalash.geo **=** as.geodata**(**obj, coords.col **=** 1**:**2, data.col **=** 3**)**

# Perform Bayesian kriging

coalash.bayes **=** krige.bayes**(**coalash.geo, locations **=** 'no', borders **=** **NULL**,

model **=** model.control**(**trend.d **=** 'cte', cov.model **=** 'exponential'**)**,

prior **=** prior.control**(**beta.prior **=** 'flat', sigmasq.prior **=** 'reciprocal',

tausq.rel.prior **=** 'uniform', tausq.rel.discrete **=** seq**(**0, 1, .01**)))**

**krige.bayes: no prediction locations provided.**

**Only samples of the posterior for the parameters will be returned.**

**krige.bayes: computing the discrete posterior of phi/tausq.rel**

**krige.bayes: argument `phi.discrete` not provided, using default values**

**krige.bayes: computing the posterior probabilities.**

**Number of parameter sets: 5050**

**1, 101, 201, 301, 401, 501, 601, 701, 801, 901, 1001, 1101, 1201, 1301, 1401, 1501, 1601, 1701, 1801, 1901, 2001, 2101, 2201, 2301, 2401, 2501, 2601, 2701, 2801, 2901, 3001, 3101, 3201, 3301, 3401, 3501, 3601, 3701, 3801, 3901, 4001, 4101, 4201, 4301, 4401, 4501, 4601, 4701, 4801, 4901, 5001,**

**krige.bayes: sampling from posterior distribution**

**krige.bayes: sample from the (joint) posterior of phi and tausq.rel**

**[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]**

**phi 41.56568 48.33218 41.56568 43.49897 44.46561 26.09938 28.99931 40.59903 43.49897 42.53232 45.43225 46.3989 48.33218 25.13274**

**[,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28]**

**phi 28.99931 40.59903 44.46561 47.36554 41.56568 18.36623 34.79917 47.36554 48.33218 18.36623 26.09938 31.89924 36.73246 38.66575**

**[,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38] [,39] [,40] [,41] [,42]**

**phi 42.53232 44.46561 28.03267 29.96595 44.46561 46.3989 47.36554 19.33287 29.96595 31.89924 35.76582 38.66575 41.56568 43.49897**

**[,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50] [,51] [,52] [,53] [,54] [,55] [,56]**

**phi 44.46561 48.33218 17.39959 28.03267 28.99931 32.86589 33.83253 34.79917 36.73246 37.6991 40.59903 42.53232 47.36554 29.96595**

**[,57] [,58] [,59] [,60] [,61] [,62] [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70]**

**phi 31.89924 35.76582 36.73246 37.6991 41.56568 45.43225 47.36554 31.89924 35.76582 39.63239 43.49897 44.46561 18.36623 20.29952**

**[,71] [,72] [,73] [,74] [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84]**

**phi 25.13274 26.09938 27.06602 28.99931 43.49897 44.46561 45.43225 27.06602 28.03267 30.9326 34.79917 36.73246 38.66575 39.63239**

**[,85] [,86] [,87] [,88] [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96] [,97] [,98]**

**phi 40.59903 45.43225 46.3989 47.36554 48.33218 16.43294 27.06602 34.79917 35.76582 36.73246 39.63239 17.39959 18.36623 23.19945**

**[,99] [,100] [,101] [,102] [,103] [,104] [,105] [,106] [,107] [,108] [,109] [,110] [,111] [,112]**

**phi 25.13274 28.99931 29.96595 31.89924 32.86589 33.83253 44.46561 45.43225 21.26616 25.13274 27.06602 28.99931 31.89924 32.86589**

**[,113] [,114] [,115] [,116] [,117] [,118] [,119] [,120] [,121] [,122] [,123] [,124] [,125] [,126]**

**phi 38.66575 39.63239 42.53232 45.43225 20.29952 25.13274 28.03267 30.9326 31.89924 33.83253 35.76582 38.66575 39.63239 41.56568**

**[,127] [,128] [,129] [,130] [,131] [,132] [,133] [,134] [,135] [,136] [,137] [,138] [,139] [,140]**

**phi 42.53232 47.36554 15.4663 16.43294 23.19945 26.09938 27.06602 28.99931 29.96595 32.86589 34.79917 36.73246 37.6991 40.59903**

**[,141] [,142] [,143] [,144] [,145] [,146] [,147] [,148] [,149] [,150] [,151] [,152] [,153] [,154]**

**phi 27.06602 28.03267 29.96595 31.89924 32.86589 34.79917 35.76582 40.59903 41.56568 45.43225 46.3989 9.666437 18.36623 21.26616**

**[,155] [,156] [,157] [,158] [,159] [,160] [,161] [,162] [,163] [,164] [,165] [,166] [,167] [,168]**

**phi 22.2328 24.16609 25.13274 28.03267 32.86589 43.49897 48.33218 16.43294 21.26616 27.06602 36.73246 39.63239 42.53232 45.43225**

**[,169] [,170] [,171] [,172] [,173] [,174] [,175] [,176] [,177] [,178] [,179] [,180] [,181] [,182]**

**phi 16.43294 20.29952 24.16609 25.13274 28.99931 33.83253 35.76582 37.6991 38.66575 41.56568 42.53232 43.49897 46.3989 48.33218**

**[,183] [,184] [,185] [,186] [,187] [,188] [,189] [,190] [,191] [,192] [,193] [,194] [,195] [,196]**

**phi 13.53301 16.43294 17.39959 18.36623 21.26616 27.06602 30.9326 34.79917 38.66575 39.63239 40.59903 47.36554 48.33218 14.49966**

**[,197] [,198] [,199] [,200] [,201] [,202] [,203] [,204] [,205] [,206] [,207] [,208] [,209] [,210]**

**phi 16.43294 18.36623 25.13274 26.09938 28.03267 29.96595 30.9326 32.86589 35.76582 37.6991 38.66575 40.59903 41.56568 42.53232**

**[,211] [,212] [,213] [,214] [,215] [,216] [,217] [,218] [,219] [,220] [,221] [,222] [,223] [,224]**

**phi 14.49966 17.39959 26.09938 29.96595 30.9326 32.86589 33.83253 34.79917 38.66575 39.63239 43.49897 44.46561 45.43225 18.36623**

**[,225] [,226] [,227] [,228] [,229] [,230] [,231] [,232] [,233] [,234] [,235] [,236] [,237] [,238]**

**phi 28.03267 30.9326 34.79917 35.76582 37.6991 41.56568 43.49897 45.43225 9.666437 21.26616 29.96595 30.9326 31.89924 32.86589**

**[,239] [,240] [,241] [,242] [,243] [,244] [,245] [,246] [,247] [,248] [,249] [,250] [,251] [,252]**

**phi 34.79917 37.6991 40.59903 46.3989 47.36554 48.33218 18.36623 19.33287 21.26616 24.16609 28.03267 28.99931 31.89924 34.79917**

**[,253] [,254] [,255] [,256] [,257] [,258] [,259] [,260] [,261] [,262] [,263] [,264] [,265] [,266]**

**phi 35.76582 38.66575 44.46561 9.666437 10.63308 17.39959 18.36623 22.2328 23.19945 28.03267 34.79917 36.73246 37.6991 42.53232**

**[,267] [,268] [,269] [,270] [,271] [,272] [,273] [,274] [,275] [,276] [,277] [,278] [,279] [,280]**

**phi 47.36554 48.33218 15.4663 22.2328 27.06602 28.99931 29.96595 32.86589 34.79917 35.76582 36.73246 38.66575 39.63239 40.59903**

**[,281] [,282] [,283] [,284] [,285] [,286] [,287] [,288] [,289] [,290] [,291] [,292] [,293] [,294]**

**phi 41.56568 43.49897 45.43225 9.666437 12.56637 15.4663 16.43294 19.33287 22.2328 26.09938 27.06602 30.9326 31.89924 33.83253**

**[,295] [,296] [,297] [,298] [,299] [,300] [,301] [,302] [,303] [,304] [,305] [,306] [,307] [,308]**

**phi 43.49897 44.46561 12.56637 15.4663 20.29952 25.13274 28.03267 28.99931 30.9326 35.76582 36.73246 38.66575 41.56568 18.36623**

**[,309] [,310] [,311] [,312] [,313] [,314] [,315] [,316] [,317] [,318] [,319] [,320] [,321] [,322]**

**phi 23.19945 24.16609 33.83253 35.76582 38.66575 39.63239 40.59903 43.49897 9.666437 14.49966 21.26616 24.16609 31.89924 33.83253**

**[,323] [,324] [,325] [,326] [,327] [,328] [,329] [,330] [,331] [,332] [,333] [,334] [,335] [,336]**

**phi 34.79917 37.6991 40.59903 41.56568 46.3989 47.36554 48.33218 8.699793 13.53301 14.49966 19.33287 21.26616 22.2328 25.13274**

**[,337] [,338] [,339] [,340] [,341] [,342] [,343] [,344] [,345] [,346] [,347] [,348] [,349] [,350]**

**phi 26.09938 28.99931 35.76582 37.6991 38.66575 14.49966 19.33287 21.26616 25.13274 26.09938 28.03267 34.79917 36.73246 37.6991**

**[,351] [,352] [,353] [,354] [,355] [,356] [,357] [,358] [,359] [,360] [,361] [,362] [,363] [,364]**

**phi 38.66575 10.63308 17.39959 18.36623 21.26616 23.19945 24.16609 29.96595 31.89924 33.83253 34.79917 35.76582 44.46561 8.699793**

**[,365] [,366] [,367] [,368] [,369] [,370] [,371] [,372] [,373] [,374] [,375] [,376] [,377] [,378]**

**phi 16.43294 22.2328 25.13274 26.09938 28.03267 32.86589 40.59903 42.53232 45.43225 14.49966 17.39959 24.16609 25.13274 27.06602**

**[,379] [,380] [,381] [,382] [,383] [,384] [,385] [,386] [,387] [,388] [,389] [,390] [,391] [,392]**

**phi 37.6991 38.66575 40.59903 42.53232 43.49897 45.43225 48.33218 7.733149 12.56637 22.2328 27.06602 29.96595 31.89924 32.86589**

**[,393] [,394] [,395] [,396] [,397] [,398] [,399] [,400] [,401] [,402] [,403] [,404] [,405] [,406]**

**phi 35.76582 37.6991 38.66575 6.766506 10.63308 13.53301 15.4663 19.33287 20.29952 22.2328 26.09938 27.06602 28.03267 29.96595**

**[,407] [,408] [,409] [,410] [,411] [,412] [,413] [,414] [,415] [,416] [,417] [,418] [,419] [,420]**

**phi 33.83253 34.79917 36.73246 40.59903 45.43225 46.3989 7.733149 10.63308 21.26616 24.16609 25.13274 29.96595 37.6991 38.66575**

**[,421] [,422] [,423] [,424] [,425] [,426] [,427] [,428] [,429] [,430] [,431] [,432] [,433] [,434]**

**phi 39.63239 40.59903 44.46561 47.36554 13.53301 17.39959 20.29952 37.6991 10.63308 15.4663 23.19945 26.09938 32.86589 44.46561**

**[,435] [,436] [,437] [,438] [,439] [,440] [,441] [,442] [,443] [,444] [,445] [,446] [,447] [,448]**

**phi 6.766506 12.56637 19.33287 20.29952 27.06602 28.03267 34.79917 35.76582 36.73246 38.66575 39.63239 46.3989 8.699793 9.666437**

**[,449] [,450] [,451] [,452] [,453] [,454] [,455] [,456] [,457] [,458] [,459] [,460] [,461] [,462]**

**phi 11.59972 12.56637 13.53301 14.49966 16.43294 17.39959 24.16609 29.96595 30.9326 31.89924 32.86589 42.53232 46.3989 48.33218**

**[,463] [,464] [,465] [,466] [,467] [,468] [,469] [,470] [,471] [,472] [,473] [,474] [,475] [,476]**

**phi 9.666437 16.43294 19.33287 22.2328 24.16609 26.09938 27.06602 28.99931 34.79917 37.6991 38.66575 40.59903 8.699793 10.63308**

**[,477] [,478] [,479] [,480] [,481] [,482] [,483] [,484] [,485] [,486] [,487] [,488] [,489] [,490]**

**phi 14.49966 17.39959 19.33287 25.13274 31.89924 36.73246 37.6991 39.63239 42.53232 44.46561 47.36554 12.56637 14.49966 20.29952**

**[,491] [,492] [,493] [,494] [,495] [,496] [,497] [,498] [,499] [,500] [,501] [,502] [,503] [,504]**

**phi 21.26616 23.19945 25.13274 30.9326 32.86589 36.73246 37.6991 42.53232 44.46561 9.666437 14.49966 16.43294 17.39959 21.26616**

**[,505] [,506] [,507] [,508] [,509] [,510] [,511] [,512] [,513] [,514] [,515] [,516] [,517] [,518]**

**phi 24.16609 25.13274 26.09938 30.9326 31.89924 35.76582 36.73246 40.59903 9.666437 12.56637 15.4663 16.43294 18.36623 20.29952**

**[,519] [,520] [,521] [,522] [,523] [,524] [,525] [,526] [,527] [,528] [,529] [,530] [,531] [,532]**

**phi 22.2328 23.19945 24.16609 26.09938 28.99931 29.96595 32.86589 43.49897 44.46561 11.59972 12.56637 17.39959 18.36623 39.63239**

**[,533] [,534] [,535] [,536] [,537] [,538] [,539] [,540] [,541] [,542] [,543] [,544] [,545] [,546]**

**phi 42.53232 9.666437 12.56637 18.36623 21.26616 24.16609 27.06602 28.99931 29.96595 34.79917 35.76582 36.73246 14.49966 18.36623**

**[,547] [,548] [,549] [,550] [,551] [,552] [,553] [,554] [,555] [,556] [,557] [,558] [,559] [,560]**

**phi 26.09938 28.03267 40.59903 45.43225 11.59972 19.33287 21.26616 22.2328 25.13274 26.09938 30.9326 40.59903 6.766506 8.699793**

**[,561] [,562] [,563] [,564] [,565] [,566] [,567] [,568] [,569] [,570] [,571] [,572] [,573] [,574]**

**phi 11.59972 16.43294 17.39959 18.36623 19.33287 20.29952 22.2328 23.19945 25.13274 29.96595 30.9326 37.6991 8.699793 10.63308**

**[,575] [,576] [,577] [,578] [,579] [,580] [,581] [,582] [,583] [,584] [,585] [,586] [,587] [,588]**

**phi 13.53301 18.36623 20.29952 23.19945 28.03267 28.99931 29.96595 30.9326 31.89924 32.86589 33.83253 35.76582 43.49897 10.63308**

**[,589] [,590] [,591] [,592] [,593] [,594] [,595] [,596] [,597] [,598] [,599] [,600] [,601] [,602]**

**phi 14.49966 19.33287 25.13274 27.06602 29.96595 42.53232 4.833218 9.666437 12.56637 15.4663 17.39959 20.29952 21.26616 22.2328**

**[,603] [,604] [,605] [,606] [,607] [,608] [,609] [,610] [,611] [,612] [,613] [,614] [,615] [,616] [,617]**

**phi 28.99931 32.86589 37.6991 46.3989 8.699793 12.56637 14.49966 17.39959 21.26616 22.2328 23.19945 24.16609 37.6991 41.56568 43.49897**

**[,618] [,619] [,620] [,621] [,622] [,623] [,624] [,625] [,626] [,627] [,628] [,629] [,630] [,631]**

**phi 10.63308 12.56637 16.43294 17.39959 22.2328 26.09938 28.99931 37.6991 8.699793 9.666437 11.59972 12.56637 13.53301 14.49966**

**[,632] [,633] [,634] [,635] [,636] [,637] [,638] [,639] [,640] [,641] [,642] [,643] [,644] [,645]**

**phi 15.4663 16.43294 20.29952 21.26616 23.19945 26.09938 30.9326 31.89924 34.79917 39.63239 41.56568 11.59972 12.56637 13.53301**

**[,646] [,647] [,648] [,649] [,650] [,651] [,652] [,653] [,654] [,655] [,656] [,657] [,658] [,659]**

**phi 17.39959 22.2328 26.09938 27.06602 31.89924 4.833218 5.799862 8.699793 11.59972 15.4663 17.39959 18.36623 23.19945 25.13274**

**[,660] [,661] [,662] [,663] [,664] [,665] [,666] [,667] [,668] [,669] [,670] [,671] [,672] [,673]**

**phi 28.03267 33.83253 35.76582 40.59903 6.766506 11.59972 12.56637 13.53301 16.43294 24.16609 25.13274 26.09938 33.83253 35.76582**

**[,674] [,675] [,676] [,677] [,678] [,679] [,680] [,681] [,682] [,683] [,684] [,685] [,686] [,687]**

**phi 37.6991 44.46561 4.833218 10.63308 15.4663 17.39959 24.16609 39.63239 40.59903 41.56568 17.39959 18.36623 21.26616 27.06602**

**[,688] [,689] [,690] [,691] [,692] [,693] [,694] [,695] [,696] [,697] [,698] [,699] [,700] [,701]**

**phi 35.76582 43.49897 10.63308 12.56637 17.39959 20.29952 21.26616 24.16609 27.06602 28.03267 28.99931 30.9326 31.89924 32.86589**

**[,702] [,703] [,704] [,705] [,706] [,707] [,708] [,709] [,710] [,711] [,712] [,713] [,714] [,715]**

**phi 1.933287 8.699793 10.63308 11.59972 12.56637 13.53301 17.39959 18.36623 22.2328 23.19945 30.9326 31.89924 34.79917 35.76582**

**[,716] [,717] [,718] [,719] [,720] [,721] [,722] [,723] [,724] [,725] [,726] [,727] [,728] [,729]**

**phi 37.6991 6.766506 9.666437 17.39959 22.2328 24.16609 32.86589 8.699793 11.59972 14.49966 16.43294 17.39959 24.16609 27.06602**

**[,730] [,731] [,732] [,733] [,734] [,735] [,736] [,737] [,738] [,739] [,740] [,741] [,742] [,743]**

**phi 4.833218 8.699793 9.666437 10.63308 11.59972 12.56637 13.53301 15.4663 16.43294 17.39959 34.79917 35.76582 6.766506 10.63308**

**[,744] [,745] [,746] [,747] [,748] [,749] [,750] [,751] [,752] [,753] [,754] [,755] [,756] [,757]**

**phi 12.56637 13.53301 15.4663 16.43294 17.39959 34.79917 38.66575 8.699793 10.63308 11.59972 12.56637 16.43294 17.39959 21.26616**

**[,758] [,759] [,760] [,761] [,762] [,763] [,764] [,765] [,766] [,767] [,768] [,769] [,770] [,771]**

**phi 23.19945 28.03267 28.99931 30.9326 31.89924 34.79917 46.3989 7.733149 10.63308 14.49966 16.43294 17.39959 22.2328 31.89924**

**[,772] [,773] [,774] [,775] [,776] [,777] [,778] [,779] [,780] [,781] [,782] [,783] [,784] [,785]**

**phi 38.66575 7.733149 9.666437 15.4663 16.43294 17.39959 18.36623 19.33287 20.29952 25.13274 28.03267 37.6991 39.63239 3.866575**

**[,786] [,787] [,788] [,789] [,790] [,791] [,792] [,793] [,794] [,795] [,796] [,797] [,798] [,799]**

**phi 5.799862 12.56637 16.43294 18.36623 21.26616 25.13274 27.06602 37.6991 41.56568 7.733149 10.63308 20.29952 22.2328 24.16609**

**[,800] [,801] [,802] [,803] [,804] [,805] [,806] [,807] [,808] [,809] [,810] [,811] [,812] [,813]**

**phi 29.96595 31.89924 35.76582 1.933287 11.59972 14.49966 15.4663 22.2328 27.06602 35.76582 38.66575 41.56568 3.866575 12.56637**

**[,814] [,815] [,816] [,817] [,818] [,819] [,820] [,821] [,822] [,823] [,824] [,825] [,826] [,827]**

**phi 13.53301 16.43294 18.36623 19.33287 28.03267 31.89924 39.63239 4.833218 5.799862 6.766506 9.666437 13.53301 16.43294 17.39959**

**[,828] [,829] [,830] [,831] [,832] [,833] [,834] [,835] [,836] [,837] [,838] [,839] [,840] [,841]**

**phi 24.16609 28.03267 32.86589 7.733149 15.4663 20.29952 27.06602 29.96595 6.766506 9.666437 10.63308 18.36623 24.16609 25.13274**

**[,842] [,843] [,844] [,845] [,846] [,847] [,848] [,849] [,850] [,851] [,852] [,853] [,854]**

**phi 26.09938 28.99931 29.96595 46.3989 8.699793 16.43294 17.39959 19.33287 26.09938 29.96595 34.79917 36.73246 39.63239**

**[ reached getOption("max.print") -- omitted 2 rows ]**

# Larger phi suggest a broader spatial correlation,

# meaning that samples farther apart are still correlated.

# Smaller phi imply that spatial correlation decays quickly,

# meaning that samples need to be close to each other to exhibit correlation.

# Extract posterior samples

out **=** coalash.bayes**$**posterior**$**sample

# Plot posterior distributions of parameters

par**(**mfrow **=** c**(**2, 2**))**

hist**(**out**$**beta, breaks **=** 20, freq **=** **FALSE**, main **=** **NULL)**

lines**(**density**(**out**$**beta**)**, col **=** 'blue', lwd **=** 2**)**

title**(**'Mean'**)**

hist**(**out**$**sigmasq, breaks **=** 20, freq **=** **FALSE**, main **=** **NULL)**

lines**(**density**(**out**$**sigmasq**)**, col **=** 'blue', lwd **=** 2**)**

title**(**'Sill'**)**

hist**(**out**$**phi, breaks **=** 20, freq **=** **FALSE**, main **=** **NULL)**

lines**(**density**(**out**$**phi**)**, col **=** 'blue', lwd **=** 2**)**

title**(**'Range'**)**

hist**(**out**$**tausq, breaks **=** 20, freq **=** **FALSE**, main **=** **NULL)**

lines**(**density**(**out**$**tausq**)**, col **=** 'blue', lwd **=** 2**)**

title**(**'Nugget'**)**

A diagram of a normal distribution

Description automatically generated with medium confidence

############################################################################################################################################

### Moran’s I and Geary’s C

# Load necessary packages

install.packages**(**"spdep"**)**

library**(**spdep**)**

# Define coordinates for coalash

coords **=** cbind**(**coalash**$**x, coalash**$**y**)**

# Define neighbors using nearest neighbors (k = 5)

coalash.nb **=** knn2nb**(**knearneigh**(**coords, k **=** 5**))**

# Create a list of spatial weights

coalash.listw **=** nb2listw**(**coalash.nb, style **=** "W"**)**

# Perform Moran's I test

moran.out **=** moran.test**(**coalash**$**coal, listw **=** coalash.listw**)**

moran.I **=** moran.out**$**estimate**[**1**]**

moran.I.se **=** sqrt**(**moran.out**$**estimate**[**3**])**

moran\_tab **=** data.frame**(**I **=** moran.I, se **=** moran.I.se**)**

print**(**moran\_tab**)**

A number with a black dot

Description automatically generated with medium confidence

# Perform Geary's C test

geary.out **=** geary.test**(**coalash**$**coal, listw **=** coalash.listw**)**

geary.C **=** geary.out**$**estimate**[**1**]**

geary.C.se **=** sqrt**(**geary.out**$**estimate**[**3**])**

geary\_tab **=** data.frame**(**C **=** geary.C, se **=** geary.C.se**)**

print**(**geary\_tab**)**



### SAR and CAR Model Fitting

library**(**spatialreg**)**

# Create nearest neighbor structure for coalash data (k = 5)

coords **=** cbind**(**coalash**$**x, coalash**$**y**)**

coalash.nb **=** knn2nb**(**knearneigh**(**coords, k **=** 5**))**

# Create spatial weights list

coalash.listw **=** nb2listw**(**coalash.nb, style **=** "W"**)**

# Fit a SAR model

coalash.sar.out **=** lagsarlm**(**coalash **~** 1, data **=** coalash, listw **=** coalash.listw**)**

summary**(**coalash.sar.out**)**

> summary(coalash.sar.out)

Call:lagsarlm(formula = coalash ~ 1, data = coalash, listw = coalash.listw)

Residuals:

Min 1Q Median 3Q Max

-2.21396 -0.65860 -0.12544 0.57185 7.25876

Type: lag

Coefficients: (asymptotic standard errors)

Estimate Std. Error z value Pr(>|z|)

(Intercept) 4.62230 0.79708 5.799 6.67e-09

Rho: 0.52501, LR test value: 37.247, p-value: 1.041e-09

Asymptotic standard error: 0.081508

z-value: 6.4412, p-value: 1.1852e-10

Wald statistic: 41.489, p-value: 1.1852e-10

Log likelihood: -326.7814 for lag model

ML residual variance (sigma squared): 1.2867, (sigma: 1.1343)

Number of observations: 208

Number of parameters estimated: 3

AIC: 659.56, (AIC for lm: 694.81)

LM test for residual autocorrelation

test value: 5.687, p-value: 0.017091

# Fit a CAR model

coalash.car.out **=** spautolm**(**coalash **~** 1, data **=** coalash, family **=** "CAR", listw **=** coalash.listw**)**

summary**(**coalash.car.out**)**

> summary(coalash.car.out)

Call: spautolm(formula = coalash ~ 1, data = coalash, listw = coalash.listw, family = "CAR")

Residuals:

Min 1Q Median 3Q Max

-2.43589 -0.71114 -0.17281 0.65562 6.84459

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 9.37357 0.24372 38.461 < 2.2e-16

Lambda: 0.90471 LR test value: 43.934 p-value: 3.3967e-11

Numerical Hessian standard error of lambda: NaN

Log likelihood: -323.4377

ML residual variance (sigma squared): 1.1772, (sigma: 1.085)

Number of observations: 208

Number of parameters estimated: 3

AIC: 652.88

# Add fitted values to the coalash dataset

coalash**$**fitted.sar **=** fitted**(**coalash.sar.out**)**

coalash**$**fitted.car **=** fitted**(**coalash.car.out**)**

###################################################################################

# Load required libraries

library**(**maps**)**

library**(**classInt**)**

library**(**RColorBrewer**)**

library**(**sp**)**

# Create a SpatialPointsDataFrame for coalash data

coords **<-** cbind**(**coalash**$**x, coalash**$**y**)**

coalash\_sp **<-** SpatialPointsDataFrame**(**coords, data **=** data.frame**(**coal **=** coalash**$**coal, fitted.sar **=** coalash**$**fitted.sar**))**

# Define breaks and color palette

brks **<-** c**(**min**(**coalash**$**coal**)**, quantile**(**coalash**$**coal, c**(**0.25, 0.5, 0.75**))**, max**(**coalash**$**coal**))**

color.palette **<-** rev**(**brewer.pal**(**4, "RdBu"**))**

# Create class intervals and color codes for raw and fitted data

class.raw **<-** classIntervals**(**var **=** coalash\_sp**$**coal, n **=** 4, style **=** "fixed", fixedBreaks **=** brks, dataPrecision **=** 4**)**

color.code.raw **<-** findColours**(**class.raw, color.palette**)**

class.fitted **<-** classIntervals**(**var **=** coalash\_sp**$**fitted.sar, n **=** 4, style **=** "fixed", fixedBreaks **=** brks, dataPrecision **=** 4**)**

color.code.fitted **<-** findColours**(**class.fitted, color.palette**)**

# Create legend text

leg.txt **<-** c**(**paste**(**"<", round**(**brks**[**2**]**, 2**))**,

paste**(**round**(**brks**[**2**]**, 2**)**, "-", round**(**brks**[**3**]**, 2**))**,

paste**(**round**(**brks**[**3**]**, 2**)**, "-", round**(**brks**[**4**]**, 2**))**,

paste**(**">", round**(**brks**[**4**]**, 2**)))**

# Set up the plotting area

par**(**mfrow **=** c**(**2,1**)**, oma **=** c**(**0,0,4,0**)** **+** 0.1, mar **=** c**(**0,0,1,0**)** **+** 0.1**)**

# Plot raw coal ash content

plot**(**coalash\_sp, col **=** color.code.raw, pch **=** 20, cex **=** 2**)**

title**(**"a) Raw Coal Ash Content"**)**

legend**(**"bottomleft", legend **=** leg.txt, cex **=** 1.25, bty **=** "n", horiz **=** **FALSE**, fill **=** color.palette**)**

# Plot fitted coal ash content from SAR model

plot**(**coalash\_sp, col **=** color.code.fitted, pch **=** 20, cex **=** 2**)**

title**(**"b) Fitted Coal Ash Content from SAR model"**)**

legend**(**"bottomleft", legend **=** leg.txt, cex **=** 1.25, bty **=** "n", horiz **=** **FALSE**, fill **=** color.palette**)**

# Add overall title

mtext**(**"Coal Ash Content: Raw vs. Fitted from SAR Model", outer **=** **TRUE**, cex **=** 1.5**)**

A graph of a graph showing a graph of a graph

Description automatically generated with medium confidence

############################################################################################################

# Install and load necessary packages

install.packages**(**c**(**"sf", "spdep", "spatialreg", "gstat", "ggplot2", "gridExtra"**))**

library**(**sf**)**

library**(**spdep**)**

library**(**spatialreg**)**

library**(**gstat**)**

library**(**ggplot2**)**

library**(**gridExtra**)**

# Load the coalash data

data**(**coalash**)**

# Check the structure of the coalash data

str**(**coalash**)**

> str(coalash)

'data.frame': 208 obs. of 3 variables:

$ x : int 1 1 1 2 2 2 2 2 2 2 ...

$ y : int 14 15 16 8 10 11 12 14 15 16 ...

$ coalash: num 10.21 9.92 11.17 10.01 11.15 ...

# Create an sf object for coalash data

coalash\_sf **<-** st\_as\_sf**(**coalash, coords **=** c**(**"x", "y"**))**

# Create a distance-based neighborhood object

coalash.coords **<-** st\_coordinates**(**coalash\_sf**)**

coalash.knn **<-** knearneigh**(**coalash.coords**)**

coalash.knn2nb **<-** knn2nb**(**coalash.knn**)**

coalash.dist.list **<-** nbdists**(**coalash.knn2nb, coalash.coords**)**

coalash.dist.vec **<-** unlist**(**coalash.dist.list**)**

coalash.dist.max **<-** max**(**coalash.dist.vec**)**

coalash.dnn.nb **<-** dnearneigh**(**coalash.coords, 0, coalash.dist.max**)**

# Form a distance-based proximity matrix (listw object)

coalash.dnn.listw **<-** nb2listw**(**coalash.dnn.nb, style **=** "B", zero.policy **=** **TRUE)**

# Build SAR model

coalash.dnn.sar.out **<-** spautolm**(**coalash **~** 1, data **=** coalash\_sf, family **=** "SAR", listw **=** coalash.dnn.listw, zero.policy **=** **TRUE)**

coalash.dnn.sar.fitted **<-** fitted**(**coalash.dnn.sar.out**)**

coalash\_sf**$**fitted.dnn.sar **<-** coalash.dnn.sar.fitted

summary**(**coalash.dnn.sar.out**)**

> summary(coalash.dnn.sar.out)

Call: spautolm(formula = coalash ~ 1, data = coalash\_sf, listw = coalash.dnn.listw, family = "SAR", zero.policy = TRUE)

Residuals:

Min 1Q Median 3Q Max

-2.40051 -0.70372 -0.13624 0.59907 7.32201

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 9.77278 0.13637 71.664 < 2.2e-16

Lambda: 0.1182 LR test value: 29.702 p-value: 5.0374e-08

Numerical Hessian standard error of lambda: 0.019937

Log likelihood: -330.5535

ML residual variance (sigma squared): 1.3335, (sigma: 1.1548)

Number of observations: 208

Number of parameters estimated: 3

AIC: 667.11

# Build CAR model

coalash.dnn.car.out **<-** spautolm**(**coalash **~** 1, data **=** coalash\_sf, family **=** "CAR", listw **=** coalash.dnn.listw, zero.policy **=** **TRUE)**

coalash.dnn.car.fitted **<-** fitted**(**coalash.dnn.car.out**)**

coalash\_sf**$**fitted.dnn.car **<-** coalash.dnn.car.fitted

summary**(**coalash.dnn.car.out**)**

> summary(coalash.dnn.car.out)

Call: spautolm(formula = coalash ~ 1, data = coalash\_sf, listw = coalash.dnn.listw, family = "CAR", zero.policy = TRUE)

Residuals:

Min 1Q Median 3Q Max

-2.58382 -0.74905 -0.11778 0.60785 6.87723

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 9.75043 0.16484 59.152 < 2.2e-16

Lambda: 0.22084 LR test value: 34.841 p-value: 3.5779e-09

Numerical Hessian standard error of lambda: 0.021992

Log likelihood: -327.9842

ML residual variance (sigma squared): 1.2233, (sigma: 1.106)

Number of observations: 208

Number of parameters estimated: 3

AIC: 661.97

# Visualize the results

# Raw data plot

raw\_plot **<-** ggplot**(**coalash\_sf**)** **+**

geom\_sf**(**aes**(**color **=** coalash**)**, size **=** 3**)** **+**

scale\_color\_viridis\_c**()** **+**

theme\_minimal**()** **+**

ggtitle**(**"Raw Coal Ash Content"**)**

# SAR fitted values plot

sar\_plot **<-** ggplot**(**coalash\_sf**)** **+**

geom\_sf**(**aes**(**color **=** fitted.dnn.sar**)**, size **=** 3**)** **+**

scale\_color\_viridis\_c**()** **+**

theme\_minimal**()** **+**

ggtitle**(**"SAR Fitted Coal Ash Content"**)**

# CAR fitted values plot

car\_plot **<-** ggplot**(**coalash\_sf**)** **+**

geom\_sf**(**aes**(**color **=** fitted.dnn.car**)**, size **=** 3**)** **+**

scale\_color\_viridis\_c**()** **+**

theme\_minimal**()** **+**

ggtitle**(**"CAR Fitted Coal Ash Content"**)**

# Arrange plots

grid.arrange**(**raw\_plot, sar\_plot, car\_plot, ncol **=** 3**)**

A graph of a map of the state of sar

Description automatically generated with medium confidence