

How do I fit a variogram model to my spatial data in R using regression commands?

Authored by
stats writer

June 30, 2024

RECOMMENDED CITATION

stats writer (2024). *How do I fit a variogram model to my spatial data in R using regression commands?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=161741>

The process of fitting a variogram model to spatial data in R using regression commands involves using statistical regression techniques to analyze the spatial relationships within the data. Variogram models are used to describe the spatial variability and autocorrelation of the data, and fitting them allows for a better understanding of the underlying patterns and trends. The process typically involves importing the data into R, performing exploratory data analysis, selecting an appropriate variogram model, and using regression commands to fit the model to the data. This approach allows for the identification of any significant spatial relationships and can aid in predicting values at unobserved locations. By utilizing regression commands, this process provides a formal and efficient method for analyzing spatial data in R.

How do I fit a variogram model to my spatial data in R using regression commands? | R FAQ

We often examine data with the aim of making predictions. Spatial data analysis is no exception. Given measurements of a variable at a set of points in a region, we might like to extrapolate to points in the region where the variable was not measured or, possibly, to points outside the region that we believe will behave similarly. We can base these predictions on our measured values alone by kriging or we can incorporate covariates and make predictions using a regression model. In both scenarios, we will need to first

fit a variogram model to our data.

You can fit a variogram model graphically using the variog command to calculate and then plot the points and assess the points with possible models in mind; or you can fit several variogram models using lme and compare the model fits. This page walks through the second approach. For an example of the other approach, see R FAQ: How do I generate a variogram for spatial data in R?.

There are several shapes that a variogram might follow and, in fitting a variogram model, we aim to mathematically describe the shape. Some commonly used variogram models are the spherical, exponential and Gaussian models. In all three of these models, the variogram increases with distance at small distances and then levels off. This general shape is suggestive of a spatial correlation that is positive and strong at small distances and

becomes less so as distances increase until reaching a certain distance d . Pairs of points separated by a distance greater than d appear uncorrelated.

The lme linear mixed-effects regression command in the nlme R package supports these three as covariance structures. We will be using the thick dataset provided in the SAS documentation for proc variogram, which includes the measured thickness of coal seams at different coordinates (we have converted this to a .csv file for easy use in R). The code below installs and loads the nlme package and reads in the data we will use.

```
install.packages("nlme")  
library(nlme)  
spdata <-  
read.table("https://stats.idre.ucla.edu/stat/r/faq/thick.csv",  
header = T, sep = ",")
```

We will be looking at the fit statistics of models with our three different covariance structures and comparing the likelihoods of these models. For a baseline likelihood, we can run a model without specifying a covariance structure and obtain the likelihood of a "null" model that we hope to improve with information about the spatial structure of the data. The lme command requires a grouping variable. Since we do not have a grouping variable in our data, we can create a dummy variable that has the same value for all 75 observations.

```
dummy <- rep(1, 75)
```

```
spdata <- cbind(spdata, dummy)
```

```
null.model <- lme(fixed = thick ~ 1, data = spdata,  
random = ~ 1 | dummy)
```

```
summary(null.model)
```

Linear mixed-effects model fit by REML

Data: spdata

AIC BIC logLik

347.7511 354.6633 -170.8756

Random effects:

Formula: ~1 | dummy

(Intercept) Residual

StdDev: 0.7284064 2.365569

Fixed effects: thick ~ 1

Value Std.Error DF t-value p-value

(Intercept) 40.13867 0.7779383 74 51.59621 0

Standardized Within-Group Residuals:

Min Q1 Med Q3 Max

**-2.55273316 -0.54475968 -0.01634561 0.68116097
1.92821818**

Number of Observations: 75

Number of Groups: 1

Next, we can run the same model with spatial correlation structures. We can specify such a structure with the correlation option of lme. The structures are generated using corExp for exponential,

corGaus for Gaussian, or **corSpher** for spherical. These are the three we will show on this page, though there are two more spatial options for **lme**: **corLin** for linear and **corRatio** for rational quadratics.

If we wish to see the correlation matrix we will be specifying, we can first use the **corExp** command to indicate how many and which variables are to be used in calculating distances. We will be using our coordinate variables east and north. With the **Initialize** command, we provide our distance structure with our dataset. Then, based on that data and structure, a correlation matrix based on the distances is calculated. For a dataset with n observations, this will yield an $n \times n$ matrix (we have printed just a portion below).

```
cs1Exp <- corExp(1, form = ~ east + north)
```

```
cs1Exp <- Initialize(cs1Exp, spdata)  
corMatrix(cs1Exp)
```

```
1.000000e+00 8.899995e-11 1.116596e-07 3.560630e-04  
8.899995e-11 1.000000e+00 3.247567e-04 9.157127e-14  
1.116596e-07 3.247567e-04 1.000000e+00 2.066026e-10  
3.560630e-04 9.157127e-14 2.066026e-10 1.000000e+00  
1.087634e-04 1.063903e-07 3.067433e-04 5.905301e-07  
2.334579e-11 3.296153e-21 7.576799e-18 3.591706e-08  
3.038054e-12 4.342979e-22 1.011711e-18 4.741565e-09  
5.823563e-07 1.664325e-16 4.223333e-13 1.526776e-03  
8.442416e-10 2.043360e-19 5.295179e-16 1.950855e-06  
6.292598e-27 1.033177e-36 2.692782e-33 1.074585e-23
```

When we update our null model to include an Exponential spatial correlation structure, this will be the matrix used. To make this update, we can use the update command and use the correlation option in lme to input the form of the correlation. As we had done outside of the model, we will do this with corExp.

```
exp.sp <- update(null.model, correlation = corExp(1,  
form = ~ east + north), method = "ML")
```

```
summary(exp.sp)
```

Linear mixed-effects model fit by maximum likelihood

Data: spdata

AIC BIC logLik

167.1209 176.3909 -79.56047

Random effects:

Formula: ~1 | dummy

(Intercept) Residual

StdDev: 0.0003758452 2.959959

Correlation Structure: Exponential spatial correlation

Formula: ~east + north | dummy

Parameter estimate(s):

range

205.4266

Fixed effects: thick ~ 1

Value Std.Error DF t-value p-value

(Intercept) 42.39964 2.496176 74 16.98584 0

Standardized Within-Group Residuals:

Min Q1 Med Q3 Max

-2.8039702 -1.1992182 -0.7769150 -0.2194749 0.7771606

Number of Observations: 75

Number of Groups: 1

We can look at some of the model fit statistics to see if our model with spatial correlation fits better than one without. We do not have any predictor variables in the model, so we are essentially just looking at whether or not our outcome is spatially autocorrelated. The fit statistics suggest that there is. We can look at two other spatial correlation structures to see which appears to fit our data best.

```
gau.sp <- update(null.model, correlation = corGaus(1,  
form = ~ east + north), method = "ML")  
summary(gau.sp)
```

Linear mixed-effects model fit by maximum likelihood

Data: spdata

AIC BIC logLik

89.55566 98.82561 -40.77783

Random effects:

Formula: ~1 | dummy

(Intercept) Residual

StdDev: 7.954591e-05 2.088646

Correlation Structure: Gaussian spatial correlation

Formula: ~east + north | dummy

Parameter estimate(s):

range

20.43577

Fixed effects: thick ~ 1

Value Std.Error DF t-value p-value

(Intercept) 40.34101 0.5807745 74 69.46071 0

Standardized Within-Group Residuals:

Min Q1 Med Q3 Max

-2.9880634 -0.7138627 -0.1153888 0.6745968 2.0869951

Number of Observations: 75

Number of Groups: 1

From the fit statistics of these models, we can see that the Gaussian

covariance structure best fits our data. If we were to further model coal seam thickness in this dataset and wished to indicate a spatial correlation in the outcome, we would choose Gaussian. This is consistent with the findings of the graphical fitting of a variogram model seen in SAS FAQ: How do I fit a variogram model to my spatial data in SAS using Proc Variogram?.

See the

R online documentation for lme for further details on modeling options.

See also

References