

# coefficient\_investigation

*Daivd Leslie*

*Sunday, April 12, 2015*

The first step in investigating coefficients will be to setup data storage for generated data. In order to accomplish this, an array of matrixes containing generated data will be created. To determind the size of the array, a grid size must be provided.

```
# Set grid size
n = 100

# Select desired number of matrices
numMat = 7

# Select number of columns
numCols = 6

# Names of columns
# Must match the number of coulms
colNames = c('y', 'x1', 'x2', 'x3', 'xCoord', 'yCoord')

# Create array filled with NAs
resultsArray = array(1:(numCols*n^2), dim=c(n^2, numCols, numMat), dimnames = list(1:n^2, colNames ,NULL))
```

Now that the array has been created, we can now begin to fill the array with data that we will using for the investigation. In order to accomplish this spatailly autocorrelated data will be generated using a modified version of the Spatail leave-one-out method developed by Le Rest.

```
# Import Statement(s)
library(RandomFields)

# Set spatail range for distance between points
spat_range = seq(0.001, 66, 10)

# Determine range of varience
var_range = var_range = seq(1, 10, 10)

for (i in 1:numMat) {
  # Create a model with spatial dependence
  mod_spat_dep = RMexp(var=var_range, scale=spat_range[i])

  # Create spatially autocorralated predictor variables
  x1 = RFsimulate(mod_spat_dep, x=1:n, y=1:n, grid=T)
  x2 = RFsimulate(mod_spat_dep, x=1:n, y=1:n, grid=T)
  x3 = RFsimulate(mod_spat_dep, x=1:n, y=1:n, grid=T)

  # Create spatail error term
  spat_err = RFsimulate(mod_spat_dep, x=1:n, y=1:n, grid=T)

  # Convert objects variables to vectors and store in columns
  spat_err = as.vector(spat_err)
```

```

resultsArray[, 2, i] = as.vector(x1)
resultsArray[, 3, i] = as.vector(x2)
resultsArray[, 4, i] = as.vector(x3)
resultsArray[, 1, i] = 2*resultsArray[, 2, i] + resultsArray[, 3, i] + 3*resultsArray[, 4, i] + spat_

# Coords
resultsArray[, 5, i] = rep(x=1:n, times=n)
resultsArray[, 6, i] = rep(x=1:n, each=n)
}

head(resultsArray[, ,1])

```

```

##           y           x1           x2           x3 xCoord yCoord
## 1  6.421884982  2.0957423 -0.180029288  0.3575364      1      1
## 2  0.196744407  1.4863601  1.780451871 -1.4387711      2      1
## 3 -6.012740636 -0.1780704 -1.322147576 -1.5527633      3      1
## 4  1.996978252  0.1907701  0.003238538  0.1155776      4      1
## 5  2.534213710  0.3808149  1.930306917 -0.3236428      5      1
## 6 -0.002183207  2.0702616 -2.212999111 -0.5104996      6      1

```

```
head(resultsArray[, ,2])
```

```

##           y           x1           x2           x3 xCoord yCoord
## 1 -8.1227968 -1.38445758 -1.290509 -0.7813126      1      1
## 2 -4.0700354 -1.35015940 -1.538540  0.5486604      2      1
## 3 -0.7329537 -0.76113197 -1.082091  1.1335277      3      1
## 4 -1.2146709 -0.60308286 -1.411893  0.7421933      4      1
## 5  0.5367984  0.05549135 -1.112315  0.7723346      5      1
## 6 -1.4318387 -0.01261631 -1.511336  0.4200884      6      1

```

```
head(resultsArray[, ,3])
```

```

##           y           x1           x2           x3 xCoord yCoord
## 1 -0.4171707 -0.8799954  1.1882150 -0.2207979      1      1
## 2 -0.3325561 -0.9698059  1.0457130 -0.2158512      2      1
## 3  1.8988833 -0.6280729  0.9914783  0.2755304      3      1
## 4  1.1268488 -0.5463980  1.4946485 -0.3214773      4      1
## 5  2.3604333  0.2504568  1.2368282 -0.2746849      5      1
## 6  0.2247045  0.2035984  1.2286794 -0.8657236      6      1

```

Now that we have the array filled with spatially autocorrelated data, let's see if coefficients generated by the model changes as the spatial range of the autocorrelation increases (Remembering that they should be close to the expected values of:  $x1=2$ ,  $x2=1$ , and  $x3=3$ ).

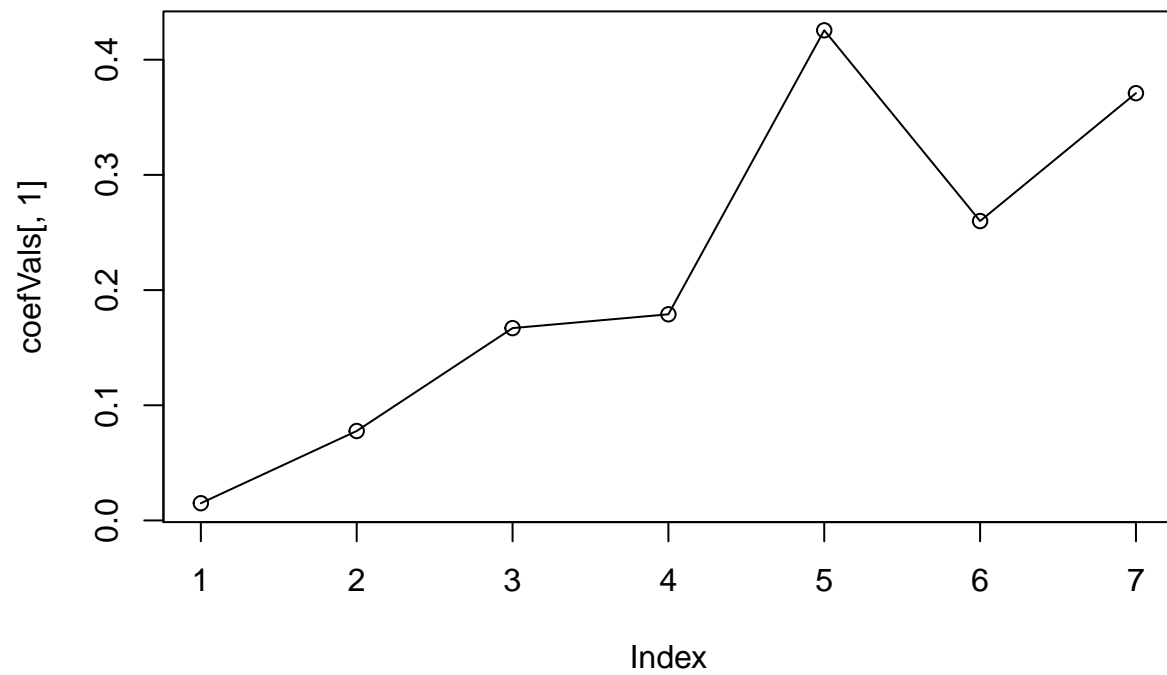
```

coefVals = matrix(NA, nrow=numMat, ncol=3, dimnames=list(1:numMat, c('x1_dif', 'x2_dif', 'x3_dif')))
for(i in 1:numMat) {
  model = glm(y ~ x1 + x2 + x3, data = as.data.frame(resultsArray[, ,i]))
  coefVals[i,] = coef(model)[2:4]
}
coefVals[,1] = abs(coefVals[,1] - 2)

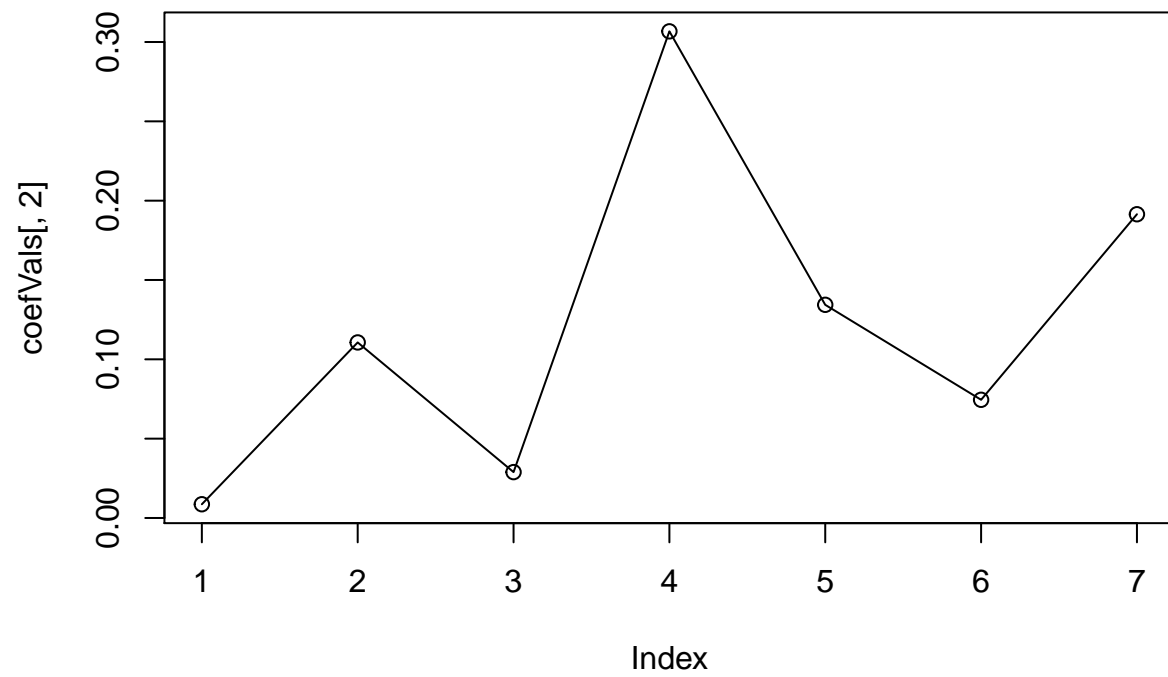
```

```
coefVals[,2] = abs(coefVals[,2] - 1)
coefVals[,3] = abs(coefVals[,3] - 3)
coefVals = cbind(coefVals, spat_range)
```

```
plot(coefVals[,1])
lines(coefVals[,1])
```



```
plot(coefVals[,2])
lines(coefVals[,2])
```



```
plot(coefVals[,3])  
lines(coefVals[,3])
```

