Areal Data Overview

Proximity Matrix

Similar to the distance matrix with point-reference data, a proximity matrix W is used to model areal data.

Given measurements $Y_i, \ldots, Y_n$ associated with areal units $1, \ldots, n$, the elements of W, w_{ij} connect units i and j

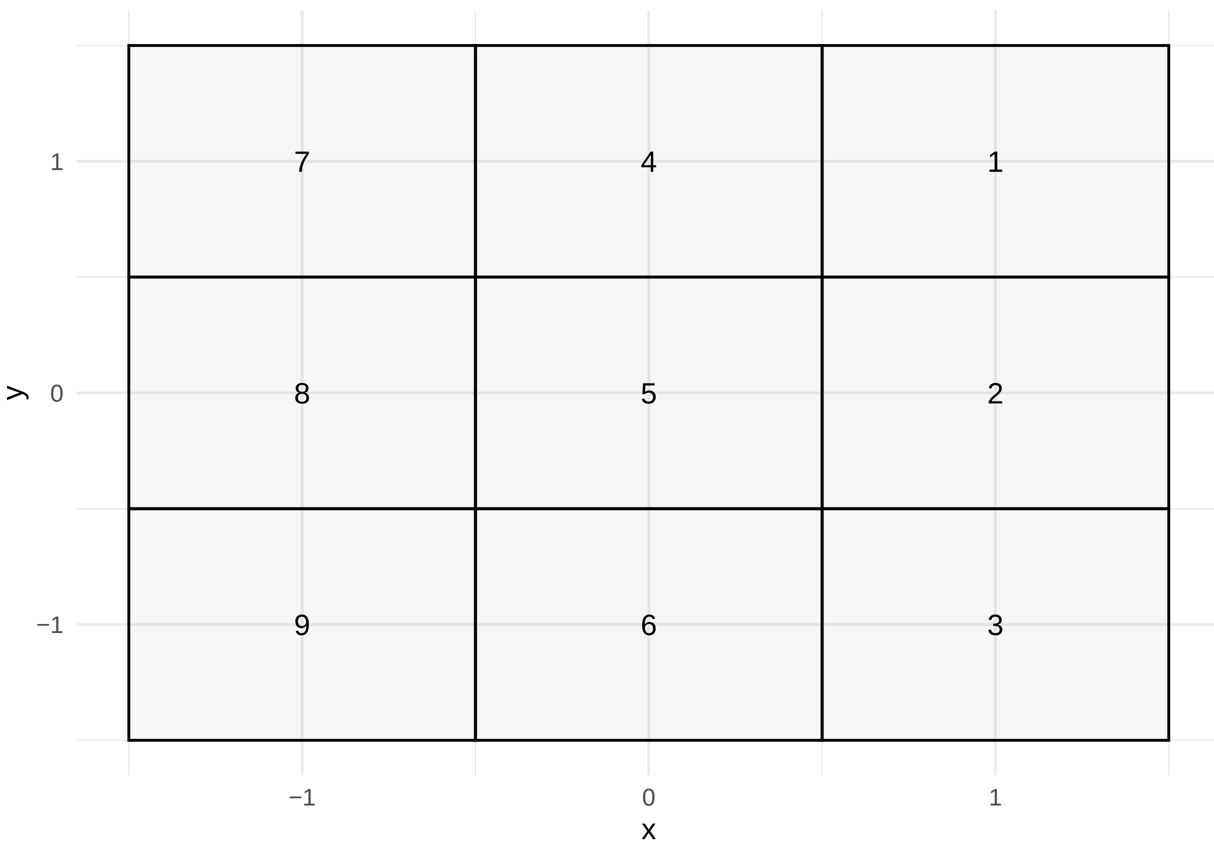
Common values for w_{ij} are

$$w_{ij} = \begin{cases} 1 & \text{if i and j are adjacent} \\ 0 & \text{otherwise (or if i=j)} \end{cases}$$

Grid Example

Create an adjacency matrix with diagonal neigbors

Create an adjacency matrix without diagonal neigbors



Spatial Association

There are two common statistics used for assessing spatial association: Moran's I and Geary's C.

Moran's I

$$I = \frac{n \sum_i \sum_j w_{ij}(Y_i - \bar{Y})(Y_j - \bar{Y})}{(\sum_{i \neq j \ w_{ij}}) \sum_i (Y_i - \bar{Y})^2}$$

Moran's I is analogous to correlation, where values close to 1 exhibit spatial clustering and values near -1 show spatial regularity (checkerboard effect).

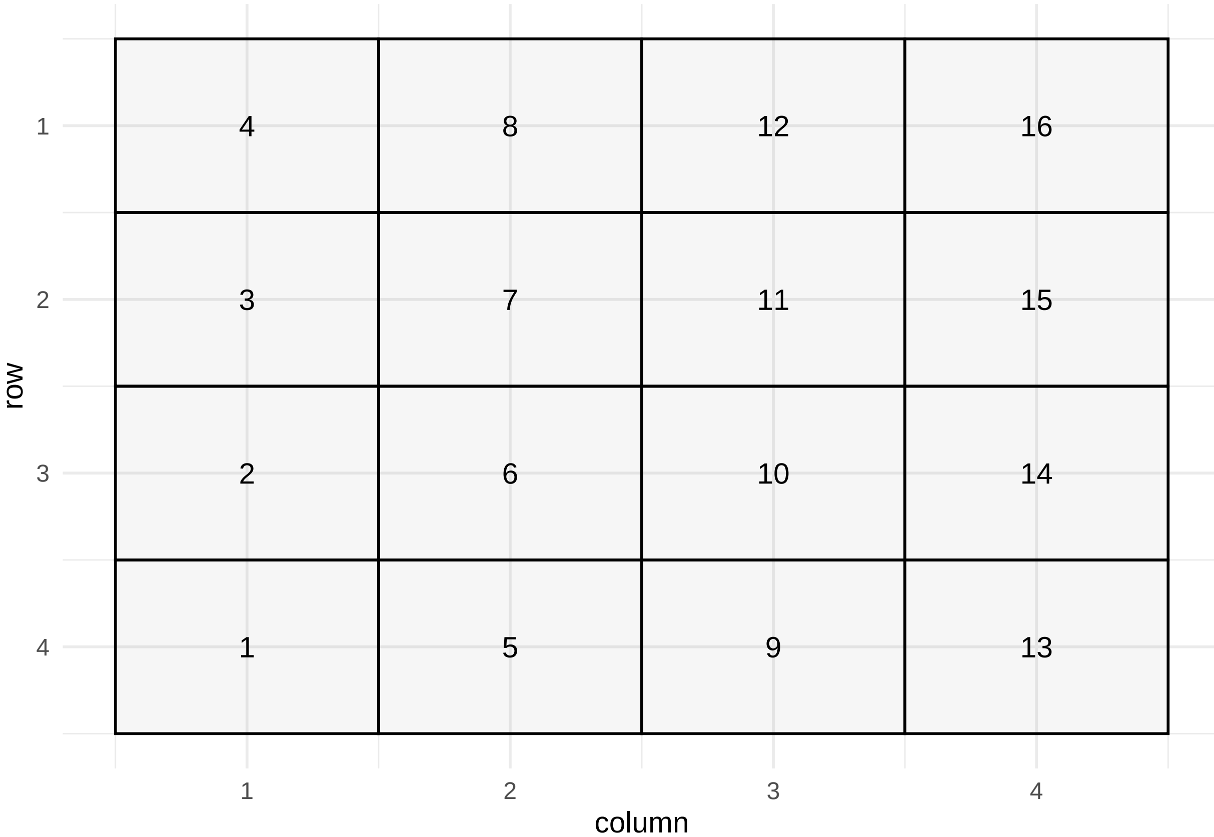
Geary's C

$$C = \frac{(n-1) \sum_i \sum_j w_{ij}(Y_i - Y_j)^2}{2(\sum_{i \neq j \ w_{ij}}) \sum_i (Y_i - \bar{Y})^2}$$

Geary's C is more similar to a variogram (has a connection to Durbin-Watson in 1-D). The statistics ranges from 0 to 2; values close to 2 exhibit regularity and values close to 1 show clustering.

Spatial Association Exercise

Consider the following scenarios and use the following 4-by-4 grid



and proximity matrix

```
W <- matrix(0, 16, 16)
for (i in 1:16){
  W[i,] <- as.numeric((d4$rpos[i] == d4$rpos & (abs(d4$cpos[i] - d4$cpos) == 1)) |
                        (d4$cpos[i] == d4$cpos & (abs(d4$rpos[i] - d4$rpos) == 1)))
}
head(W)
```

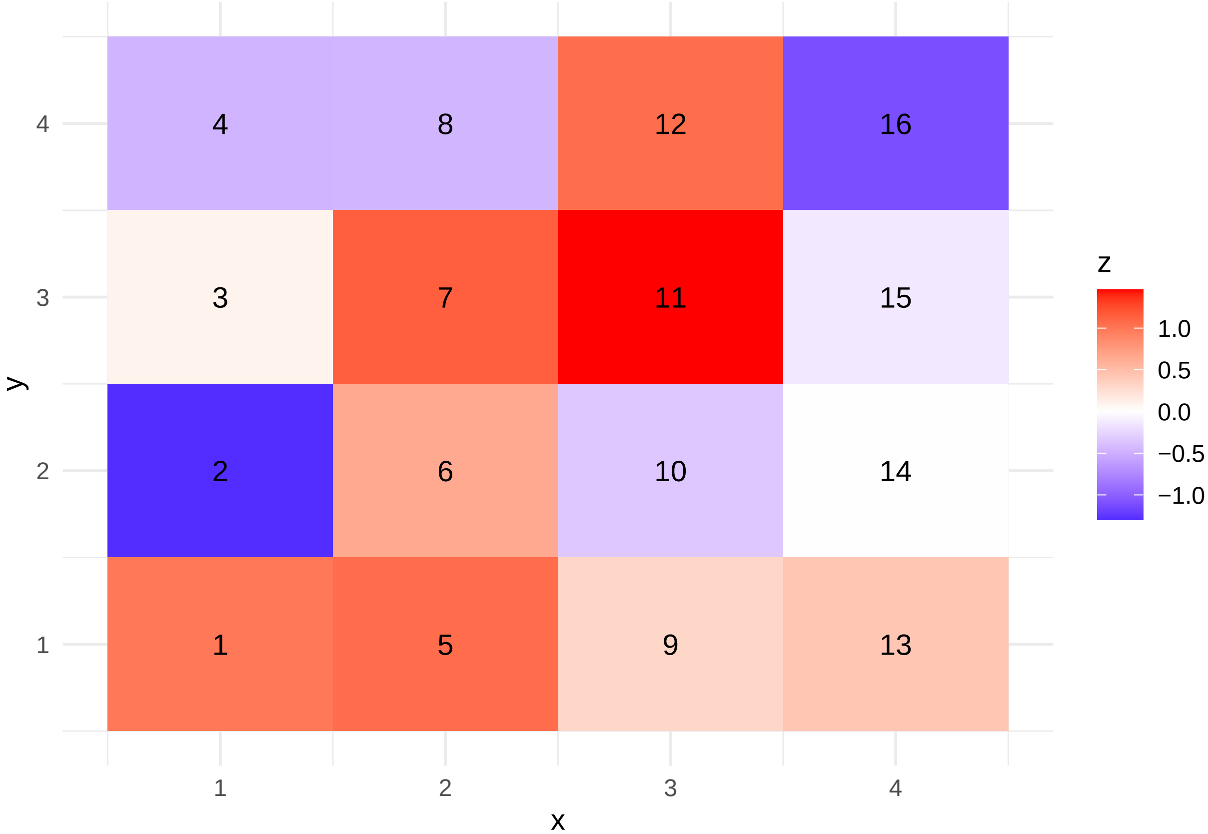
```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]     0    1    0    0    1    0    0    0    0     0     0     0     0
## [2,]     1    0    1    0    0    1    0    0    0     0     0     0     0
## [3,]     0    1    0    1    0    0    1    0    0     0     0     0     0
## [4,]     0    0    1    0    0    0    0    1    0     0     0     0     0
## [5,]     1    0    0    0    0    1    0    0    1     0     0     0     0
## [6,]     0    1    0    0    1    0    1    0    0     1     0     0     0
##       [,14] [,15] [,16]
## [1,]      0     0     0
## [2,]      0     0     0
## [3,]      0     0     0
## [4,]      0     0     0
## [5,]      0     0     0
## [6,]      0     0     0
```

for each scenario plot the grid, calculate I and G, along with permutation-based p-values.

1. Simulate data where the responses are i.i.d. N(0,1).

```
d4$z <- rnorm(16)

ggplot() +
  geom_tile(data=d4, mapping=aes(x = x, y = y, fill = z)) +
  geom_text(data=d4, aes(x=xmin+(xmax-xmin)/2, y=ymin+(ymax-ymin)/2, label=id), size=4) +
  theme_minimal() + scale_fill_gradient2(midpoint=0, low="blue", mid="white",
                      high="red", space ="Lab" )
```




```
moran.test(d4$z, mat2listw(W), alternative = 'two.sided')
```

```
##
##  Moran I test under randomisation
##
## data:  d4$z
## weights: mat2listw(W)
##
## Moran I statistic standard deviate = 0.14855, p-value = 0.8819
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##       -0.03921374       -0.06666667        0.03415414
```

```
#moran.plot(d4$z, mat2listw(W))
geary.test(d4$z, mat2listw(W), alternative = 'two.sided')
```

```
##
##  Geary C test under randomisation
```

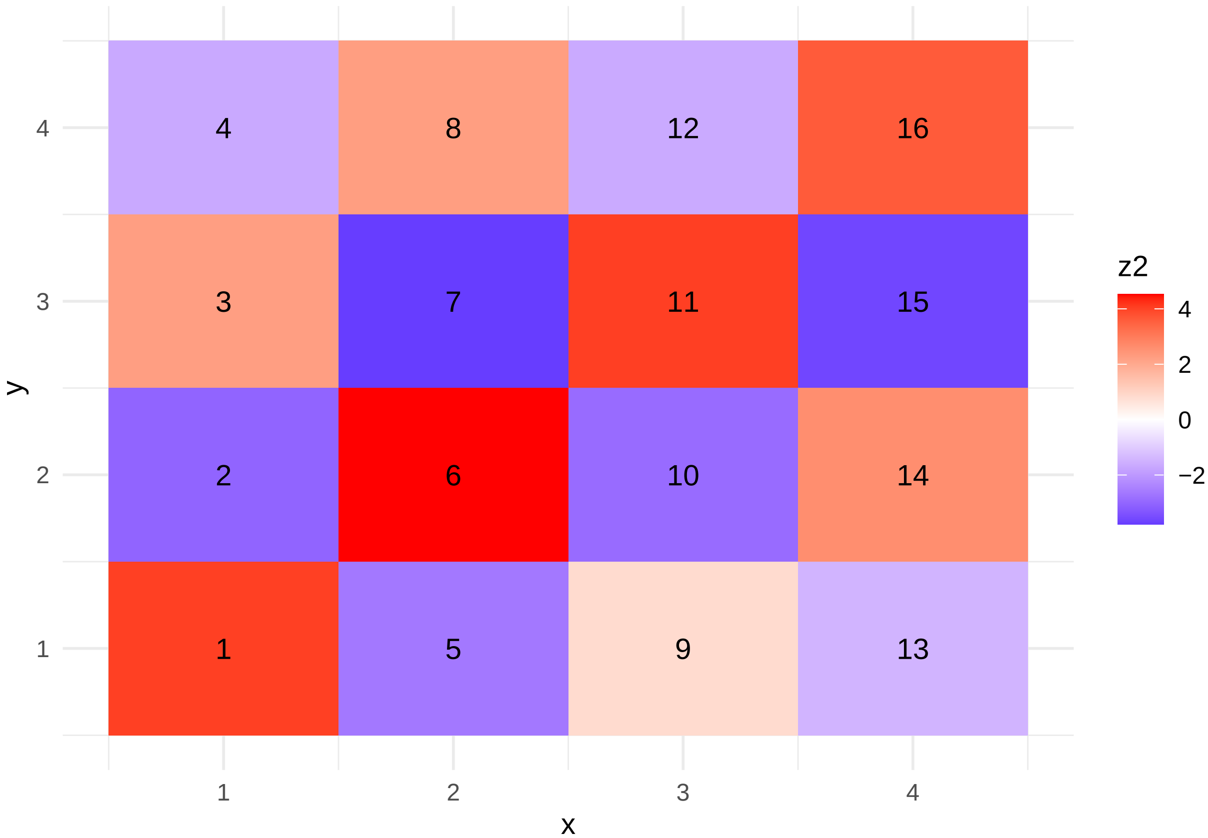
```
##
## data:  d4$z
## weights: mat2listw(W)
##
## Geary C statistic standard deviate = 0.12803, p-value = 0.8981
## alternative hypothesis: two.sided
## sample estimates:
## Geary C statistic       Expectation          Variance
##        0.97620132        1.00000000        0.03455024
```

2. Simulate data and calculate I and G for a 4-by-4 grid with a chess board approach, where "black squares" $\sim N(-2, 1)$ and "white squares" $\sim N(2, 1)$.

```
d4$z2 <- 0

for (i in 1:16){
  if ((d4$rpos[i] + d4$cpos[i])%% 2 == 1) {
    d4$z2[i] <- rnorm(1, mean = 3)
  } else {
    d4$z2[i] <- rnorm(1, mean = -3)
  }
}

ggplot() +
  geom_tile(data=d4, mapping=aes(x = x, y = y, fill = z2)) +
  geom_text(data=d4, aes(x=xmin+(xmax-xmin)/2, y=ymin+(ymax-ymin)/2, label=id), size=4) +
  theme_minimal() + scale_fill_gradient2(midpoint=0, low="blue", mid="white",
                    high="red", space ="Lab" )
```




```
moran.test(d4$z2, mat2listw(W), alternative = 'two.sided')
```

```
##
##  Moran I test under randomisation
##
## data:  d4$z2
## weights: mat2listw(W)
##
## Moran I statistic standard deviate = -4.7733, p-value = 1.812e-06
```

```
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##       -0.97385534       -0.06666667        0.03612088
```

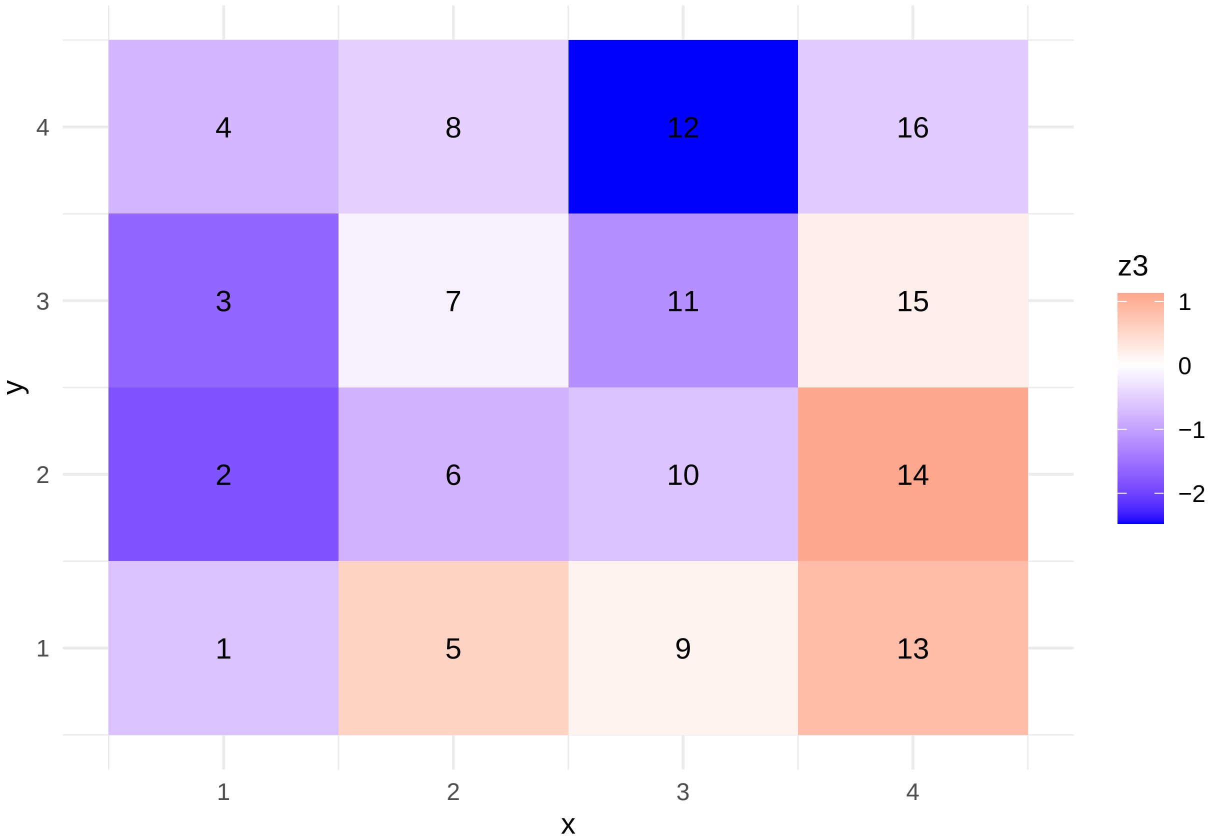
```
geary.test(d4$z2, mat2listw(W), alternative = 'two.sided')
```

```
##
##  Geary C test under randomisation
##
## data:  d4$z2
## weights: mat2listw(W)
##
## Geary C statistic standard deviate = -4.9744, p-value = 6.545e-07
## alternative hypothesis: two.sided
## sample estimates:
## Geary C statistic       Expectation          Variance
##        1.90780887        1.00000000        0.03330465
```

3. Simulate multivariate normal response on a 4-by-4 grid where $y \sim N(0, (I - \rho W)^{-1})$, where $\rho = .3$ is a correlation parameter and W is a proximity matrix.

```
d4$z3 <- mnormt::rmnorm(1, mean = 0, varcov = solve(diag(16) -  .3 * W))

ggplot() +
  geom_tile(data=d4, mapping=aes(x = x, y = y, fill = z3)) +
  geom_text(data=d4, aes(x=xmin+(xmax-xmin)/2, y=ymin+(ymax-ymin)/2, label=id), size=4) +
  theme_minimal() + scale_fill_gradient2(midpoint=0, low="blue", mid="white",
                      high="red", space ="Lab" )
```




```
moran.test(d4$z3, mat2listw(W), alternative = 'two.sided')
```

```
##
##  Moran I test under randomisation
##
## data:  d4$z3
## weights: mat2listw(W)
##
## Moran I statistic standard deviate = 2.1226, p-value = 0.03378
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##        0.31886880       -0.06666667        0.03298976
```

```
geary.test(d4$z3, mat2listw(W), alternative = 'two.sided')
```

```
##
##  Geary C test under randomisation
```

```
##
## data:  d4$z3
## weights: mat2listw(W)
##
## Geary C statistic standard deviate = 2.0748, p-value = 0.03801
## alternative hypothesis: two.sided
## sample estimates:
## Geary C statistic       Expectation          Variance
##        0.61025337        1.00000000        0.03528768
```