Spatial Autocorrelation and Areal Data

HES 505 Fall 2024: Session 20

Carolyn Koehn

Objectives

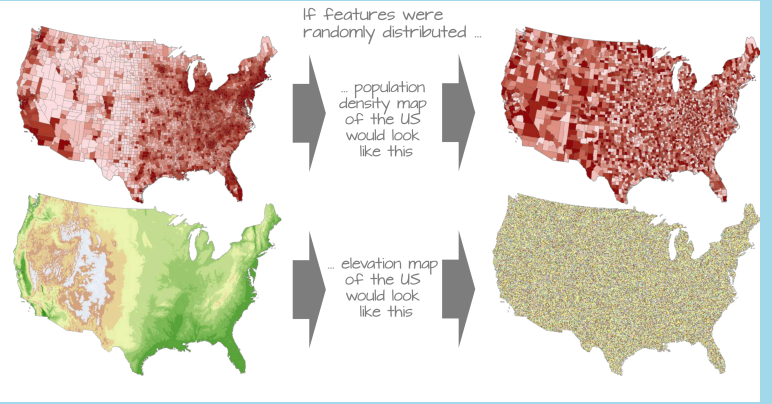
By the end of today you should be able to:

- Use the `spdep` package to identify the neighbors of a given polygon based on proximity, distance, and minimum number
- Understand the underlying mechanics of Moran's I and calculate it for various neighbors
- Distinguish between global and local measures of spatial autocorrelation
- Visualize neighbors and clusters

Revisiting Spatial Autocorrelation

Spatial Autocorrelation

- Attributes (features) are often non-randomly distributed
- Especially true with aggregated data
- Interest is in the relationship between proximity and the feature
- Difference from kriging and semivariance

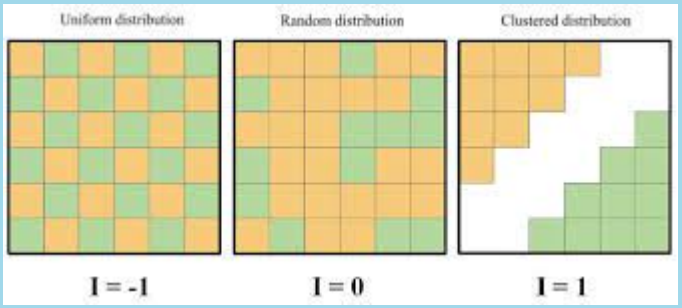


From Manuel Gimond

Moran's I

- Moran's I

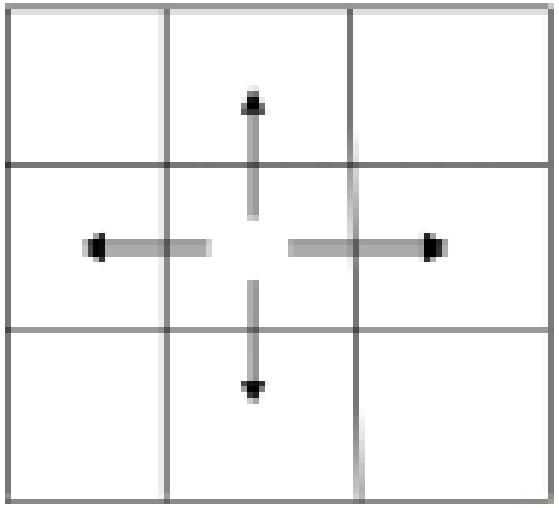
$$I(d) = \frac{\sum_{i}^{n}\sum_{j\neq i}^{n} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{S^2 \sum_{i}^{n}\sum_{j\neq i}^{n} w_{ij}}$$



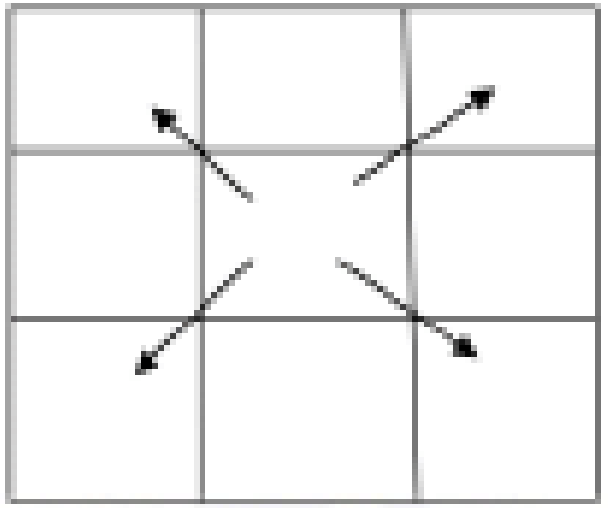
Finding Neighbors - Contiguity

- How do we define $I(d)$ for areal data?
- What about w_{ij}?
- We can use `spdep` for that!!

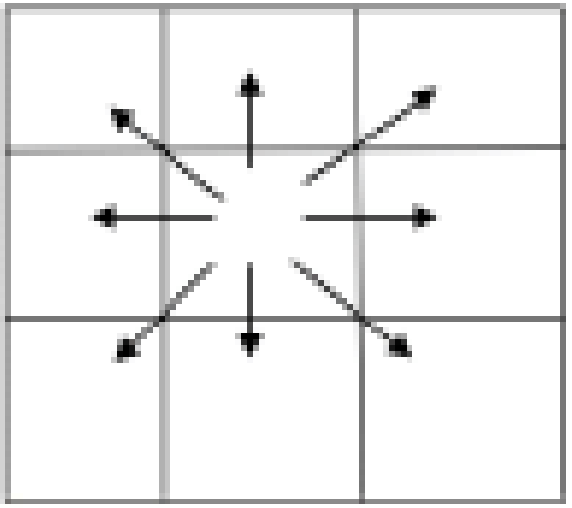
A: Rook's Contiguity



B: Bishop's contiguity

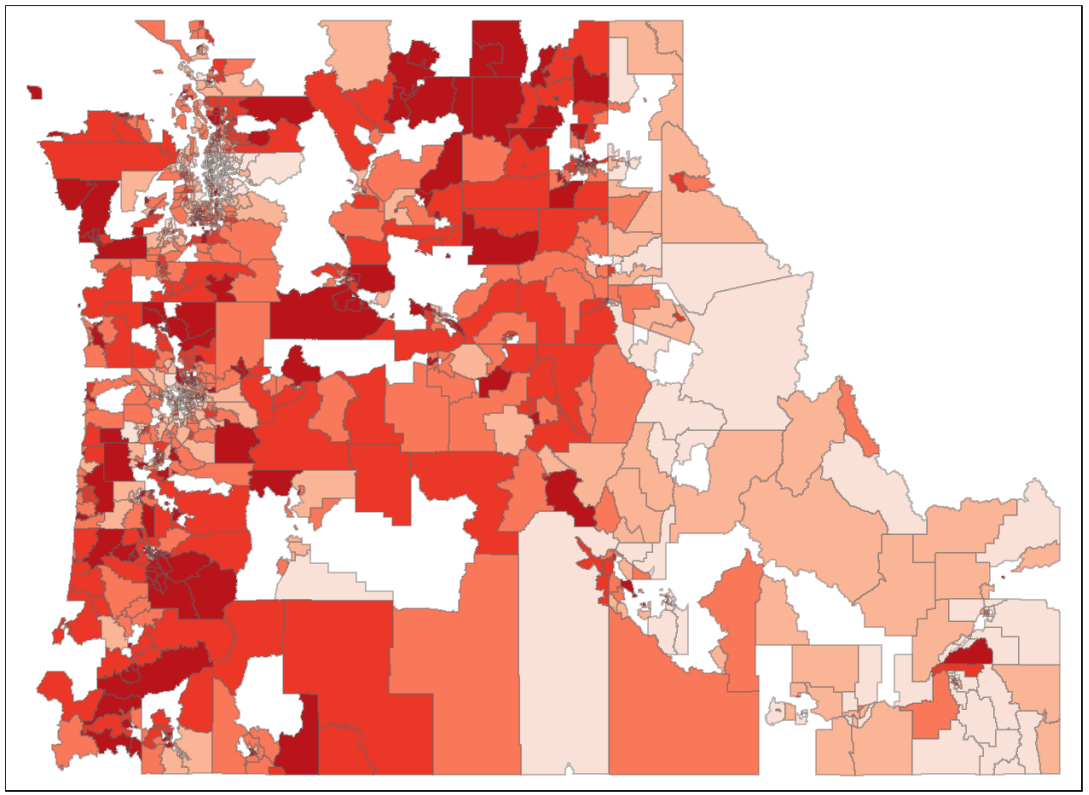


C: Queen's contiguity



Using spdep

```
1  cdc <- read_sf("data/opt/data/2023/vectorexample/cdc_nw.shp") %>%
2    select(stateabbr, countyname, countyfips, casthma_cr)
```

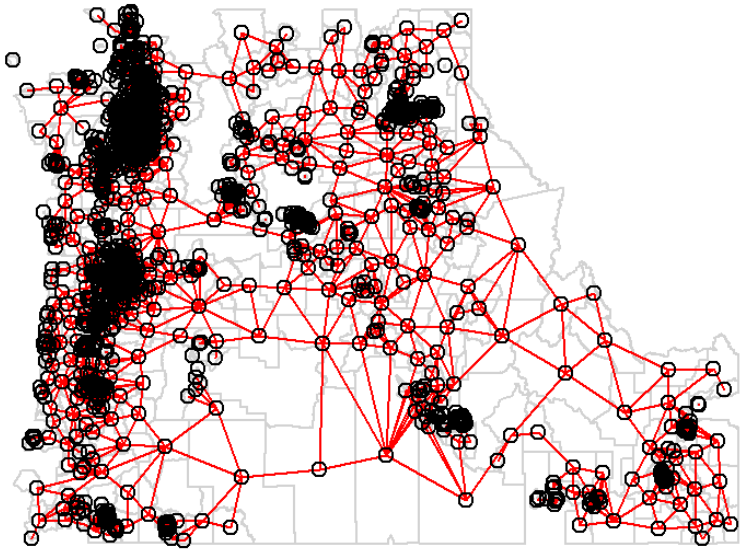


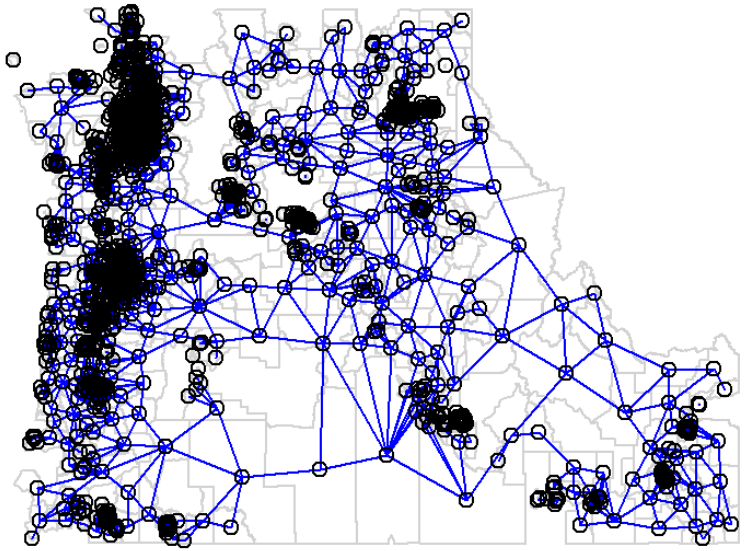
Finding Neighbors

- Queen, rook, (and bishop) cases impose neighbors by contiguity
- Weights calculated as a $1/num.\,of neighbors$

```
1 nb.qn <- poly2nb(cdc, queen=TRUE)
2 nb.rk <- poly2nb(cdc, queen=FALSE)
```

Finding Neighbors





Getting Weights and Distance

```
1  # get weights
2  lw.qn <- nb2listw(nb.qn, style="W", zero.p
3  lw.qn$weights[1:5]
```

```
[[1]]
[1] 0.5 0.5

[[2]]
[1] 0.25 0.25 0.25 0.25

[[3]]
[1] 0.2 0.2 0.2 0.2 0.2

[[4]]
[1] 0.3333333 0.3333333 0.3333333

[[5]]
[1] 1
```

```
1  # get average neighboring asthma values
2  asthma.lag <- lag.listw(lw.qn, cdc$casthma
```

```
asthma.lag
[1,] "Camas"       "9.9"
"10.3"
[2,] "Kootenai"    "10.4"
"9.575"
[3,] "Kootenai"    "10"
"9.88"
[4,] "Kootenai"    "9.5"
"10.2666666666667"
[5,] "Twin Falls" "10.2"
"9.5"
[6,] "Twin Falls" "10.4"
"9.9"
```

Fit a model

- Moran's I coefficient is the slope of the regression of the *lagged* asthma percentage vs. the asthma percentage in the tract
- More generally it is the slope of the lagged average to the measurement

```
1  M <- lm(asthma.lag ~ cdc$casthma_cr)
```

```
cdc$casthma_cr
     0.6357449
```

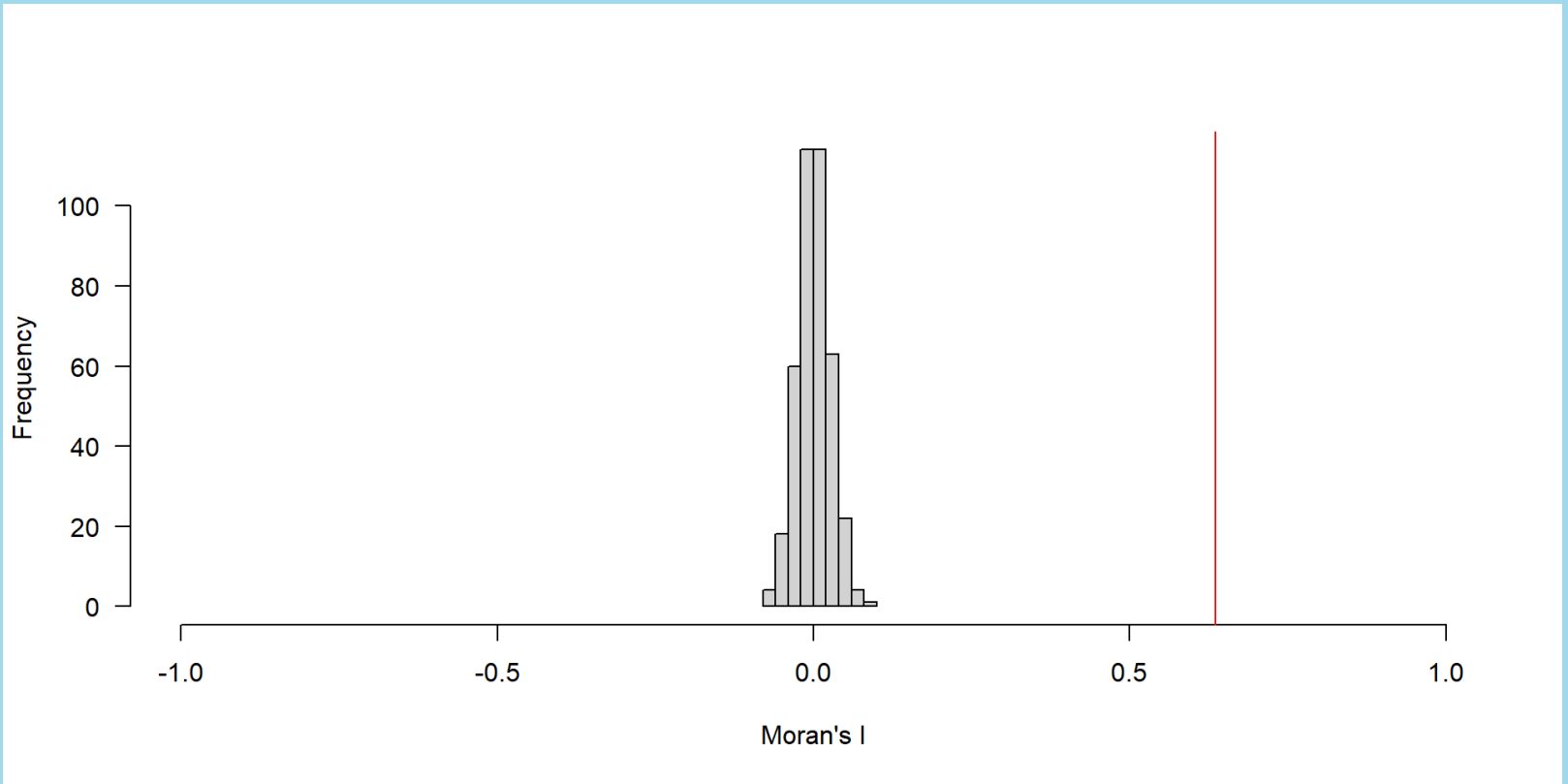
Comparing observed to expected

- We can generate the expected distribution of Moran's I coefficients under a Null hypothesis of no spatial autocorrelation
- Using permutation and a loop to generate simulations of Moran's I

```
n <- 400L   # Define the number of simulations
I.r <- vector(length=n)  # Create an empty vector

for (i in 1:n){
  # Randomly shuffle income values
  x <- sample(cdc$casthma_cr, replace=FALSE)
  # Compute new set of lagged values
  x.lag <- lag.listw(lw.qn, x)
  # Compute the regression slope and store its value
  M.r    <- lm(x.lag ~ x)
```

Comparing observed to expected

```r
# manual p-value
# hist is null hypothesis of no spatial autocorrelation
# red line is our value
hist(I.r, main=NULL, xlab="Moran's I", las=1, xlim = c(-1, 1))
abline(v=coef(M)[2], col="red")
```



Compare to Moran's I test

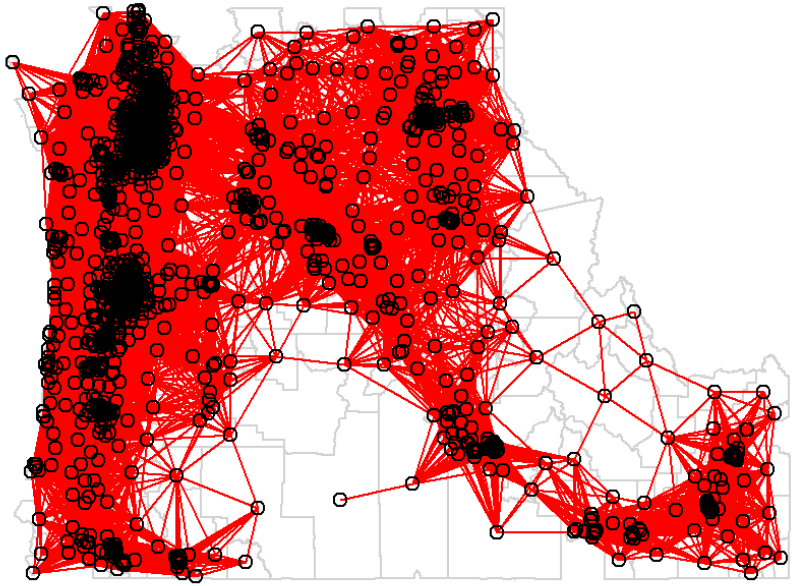
```
1 moran.test(cdc$casthma_cr, lw.qn)
```

```
    Moran I test under randomisation

data:  cdc$casthma_cr
weights: lw.qn
n reduced by no-neighbour observations

Moran I statistic standard deviate = 40.826, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.6381428057     -0.0005037783      0.0002447034
```

Finding Neighbors - Distance

```
cdc.pt <- cdc %>% st_point_on_surface(.)
# get nearest neighbor
geog.nearnb <- knn2nb(knearneigh(cdc.pt, k = 1), row.names = cdc.pt$GEOID,
#estimate distance to first nearest neighbor
nb.nearest <- dnearneigh(cdc.pt,
                         # minimum distance to search
                         d1 = 0,
                         # maximum distance to search
                         d2 = max( unlist(nbdists(geog.nearnb, cdc.pt))))
```

Getting Weights



```
1  lw.nearest <- nb2listw(nb.nearest, style="W")
2  asthma.lag <- lag.listw(lw.nearest, cdc$casthma_cr)
```

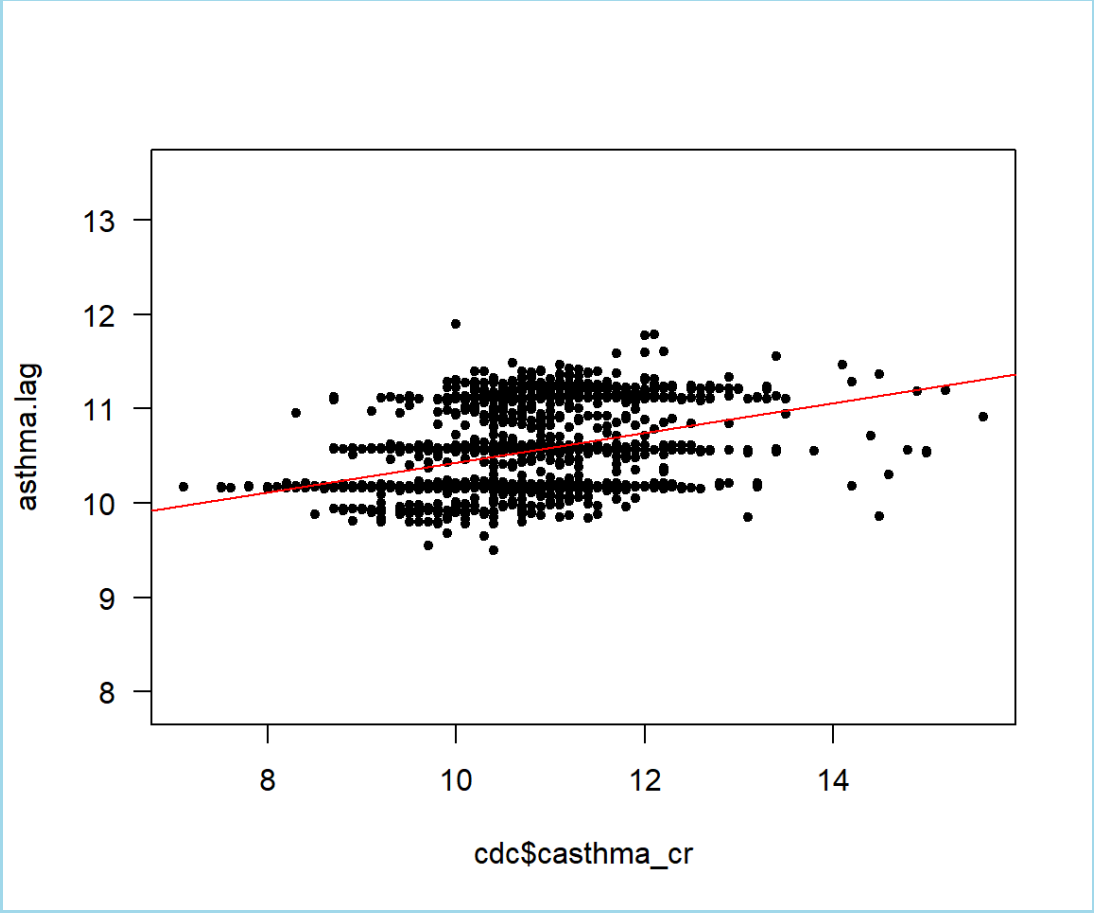
Fit a model

- Moran's I coefficient is the slope of the regression of the *lagged* asthma percentage vs. the asthma percentage in the tract
- More generally it is the slope of the lagged average to the measurement

Fit a model

```
1  M <- lm(asthma.lag ~ cdc$casthma_cr)
```

```
cdc$casthma_cr
     0.1577524
```

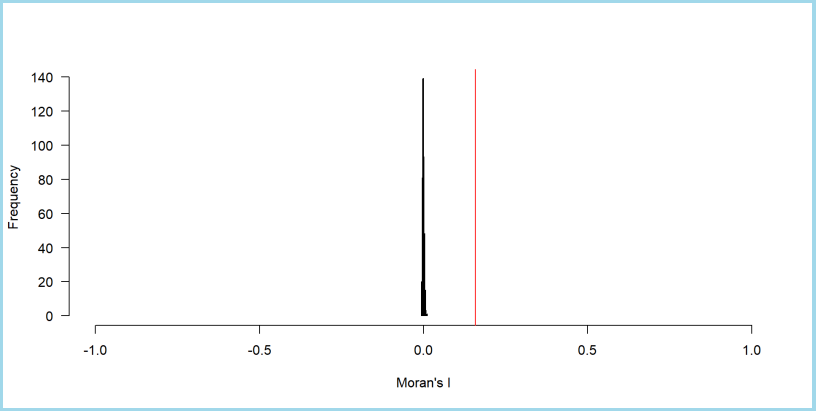


Comparing observed to expected

- We can generate the expected distribution of Moran's I coefficients under a Null hypothesis of no spatial autocorrelation
- Using permutation and a loop to generate simulations of Moran's I

Comparing observed to expected

```
n <- 400L   # Define the number of simulations
I.r <- vector(length=n)  # Create an empty vector

for (i in 1:n){
  # Randomly shuffle income values
  x <- sample(cdc$casthma_cr, replace=FALSE)
  # Compute new set of lagged values
  x.lag <- lag.listw(lw.nearest, x)
  # Compute the regression slope and store its value
  M.r    <- lm(x.lag ~ x)
  I.r[i] <- coef(M.r)[2]
}
```



Significance testing

- Pseudo p-value (based on permutations)
- Analytically (sensitive to deviations from assumptions)
- Using Monte Carlo

```
#Pseudo p-value
N.greater <- sum(coef(M)[2] > I.r)
# add modifiers to stay in -1 to 1 range
(p <- min(N.greater + 1, n + 1 - N.greater) / (n + 1))

# Analytically
# Based on a normal distribution, not the distribution of your data
moran.test(cdc$casthma_cr,lw.nearest, zero.policy = TRUE)

# Monte Carlo
moran.mc(cdc$casthma_cr, lw.nearest, zero.policy = TRUE, nsim=400)
```

Significance testing

```
[1] 0.002493766


    Moran I test under randomisation

data:  cdc$casthma_cr
weights: lw.nearest

Moran I statistic standard deviate = 64.107, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     1.577524e-01     -4.990020e-04      6.093649e-06


    Monte-Carlo simulation of Moran I

data:  cdc$casthma_cr
weights: lw.nearest
number of simulations + 1: 401

statistic = 0.15775, observed rank = 401, p-value = 0.002494
alternative hypothesis: greater
```