Spatial Dependence & Autocorrelation

1. Preface & Background

Examples of Mathematical Spatial Analytics

- **Geostatistics**: Applying statistical methods to spatial data attributes to describe patterns and make predictions.
  - Mean, median, mode, max, min, range of attributes in a spatial dataset etc.
- **Point Pattern Analysis**: How are observations distributed in space, and are there areas that contain more observations than others for a good reason?
- **Prediction Analytics**: Looking to predict unobserved observations in space by using current observations as a proxy for our best intuition.
  - Example: Species Distribution Modeling, Occupancy Modeling, HotSpot Analytics etc.
- **Landscape Metrics**: Quantify the structure, function, and change of landscapes.

Preface: Assumptions in (Spatial) Statistics

- **Spatial Dependency**: Nearby locations are more likely to share similar values.
- **Uniformity**: The statistical properties of the spatial process are spatially uniform.
- **Isotropy**: Spatial relationships are consistent in all directions.
- **Homogeneity** of Spatial Processes: The processes shaping spatial patterns are consistent throughout the area.
- **Independence of Errors**
- **Spatial Extent**: The study area is correctly defined.
- Modifiable Areal Unit Problem: We assume the scale of the analysis is appropriate.
    - Different spatial units or the scale at which data is analyzed can significantly affect conclusions

What is Spatial Autocorrelation?

- The phenomenon where similar values in a dataset are more closely located in space than would be expected if the values were randomly distributed.
- The degree to which a **variable is correlated with itself through space**.
- Example: When plotting temperature values on a map, we notice clusters of high and low temperatures in specific areas. This analysis has a good chance of containing spatial autocorrelation because the influence of geographical features such as elevation at proximity to water may be playing a role.
    - Elevation and other external variables are skewing the analysis!

Spatial Autocorrelation - Nuance

- **Spatial autocorrelation is not inherently good or bad**, it is just a characteristic of spatial data that indicates the degree to which objects close to each other in space are also similar in value.
- Whether spatial autocorrelation is considered beneficial or problematic depends on the context of the study.
- Good: Can be very useful in understanding spatial patterns in ecology, as spatial autocorrelation can indicate habitat clustering or the presence of environmental gradients.
- Bad: In statistical models that do not account for spatial autocorrelation, the presence of this autocorrelation can bias the output, and potentially lead to incorrect conclusions.

2. Spatial Autocorrelation, Moran's I, & Semivariance

Spatial Autocorrelation - why it’s a problem

- **Violation of Statistical Independence**: Many conventional statistical methods assume that observations are independent of each other. However, spatial autocorrelation implies that observations are not independent.
- **Model Misfit**: Models that do not account for spatial dependencies might lead to incorrect conclusions about the relationship between variables.
- **Spatial Confounding**: The effect of an explanatory variable is mixed up with the effect of its spatial location.
- **Scale Sensitivity**: The degree of spatial autocorrelation can vary significantly with scale as you introduce more space.
- **Complexity**: Spatial autocorrelation often requires more complex statistical models, to be viable–and these models require additional assumptions, parameters, and computational resources.
  - Spatial regression models
  - Geostatistical methods (e.g., kriging)

Measuring Spatial Autocorrelation - Moran's I

- moran() function from **spdep** package
- Measures spatial autocorrelation of spatial attribute data.
- **Compares the value of a variable at one location with the values of the same variable at neighboring locations**.
- How to interpret:
  - Values range from **-1 (perfect dispersion)** to **+1 (perfect correlation)**
  - Values close to 0 suggest a random spatial pattern
  - Positive values indicate that similar values are clustered together
  - Negative values indicate that dissimilar values are adjacent to each other
    - i.e., high values are near low values and vice versa

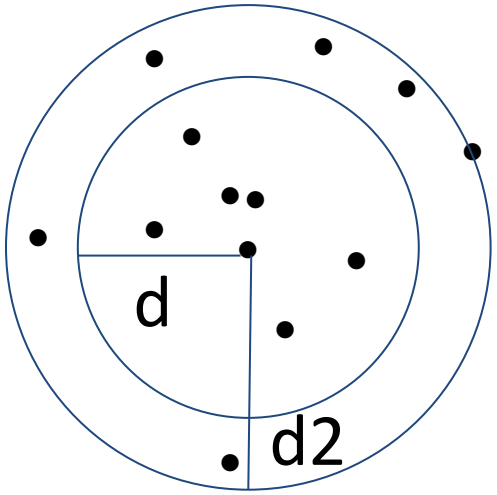
Moran's I

Moran's I requires a neighborhood definition: it is a **global statistic**.

- Answers the question of whether autocorrelation is present.
- Doesn't directly tell us anything about distance

Moran's I at different distances

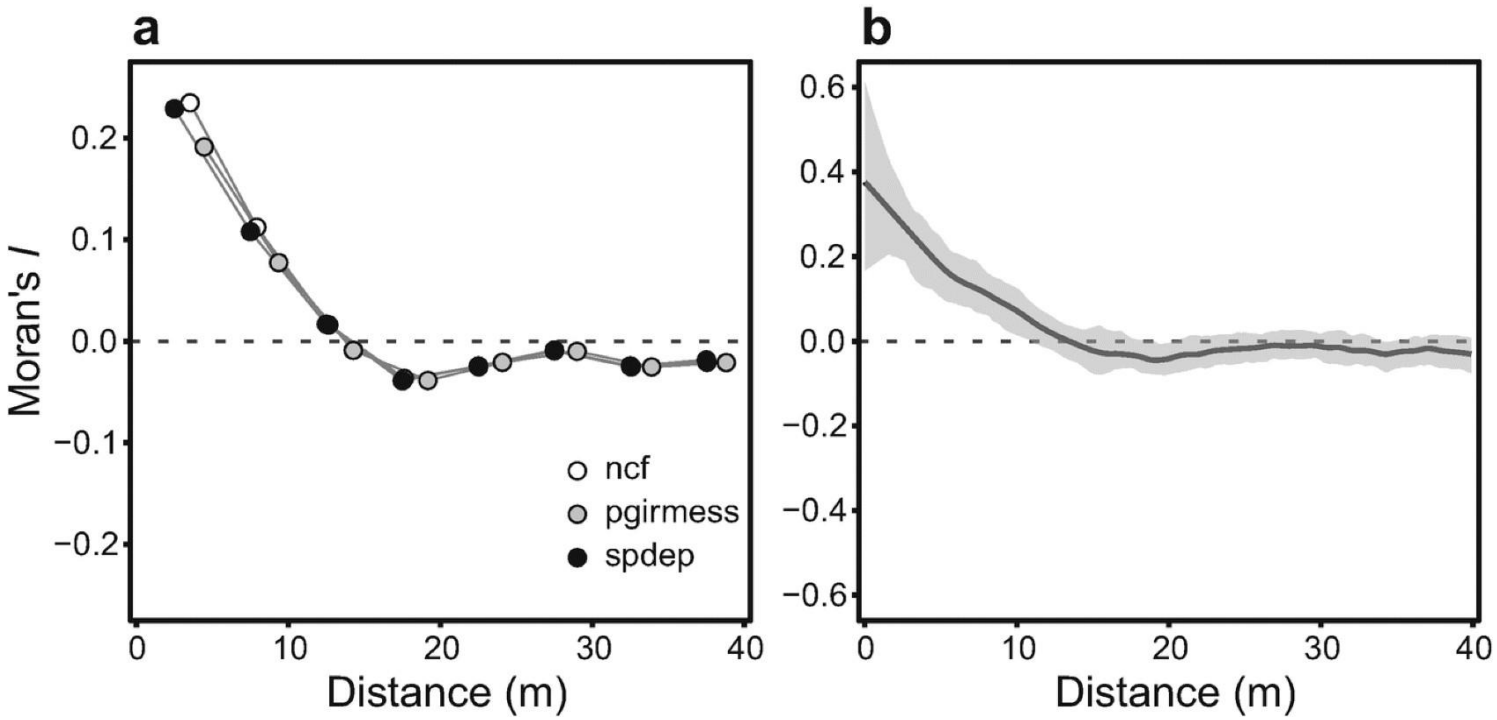
- We can quantify Moran's I for different neighborhoods.
- Define distance range, d, for 'ring' style neighborhoods.
  - Reminiscent of the pair correlation function
- Plot of I vs. distance is a **correlogram**.



Correlograms

We can calculate I for different distances, or distance classes, between points by using different weight matrices.

If we make a plot $I\,(d)$ vs d we get a correlogram.



Quantifying Autocorrelation: Semivariance and variograms

Semivariance Intuition

Semivariance is a measure of variability at a fixed distance, or distance class/bin.

- Interested in the variability of values
- Mathematically: sum of differences in values of points within x distance.

With positive autocorrelation/dependence, we expect:

- Autocorrelation (as measured by I) to be high at short distances.
- Semivariance to be low at short distances.
- It is low because the values will all be similar due to the autocorrelation.

Variograms

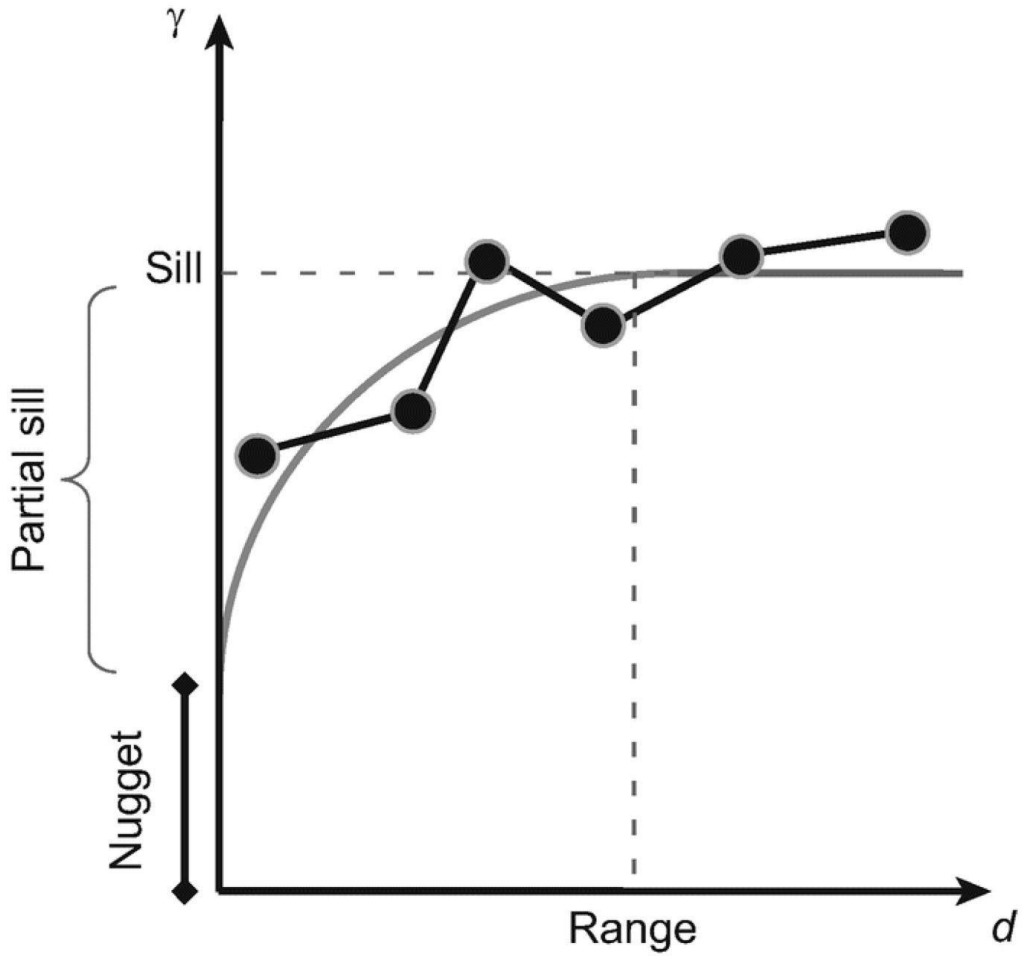
Variograms are plots of γ versus distance.

Describes spatial dependence by dist.

Variograms are associated with a fun set of terms.

- sill
- nugget
- range

- F+F figure 5.2



Variograms

Nugget: amount of variation at short distances.

- Amount of variation with local influence
- Background noise, measurement error, unmeasured variables

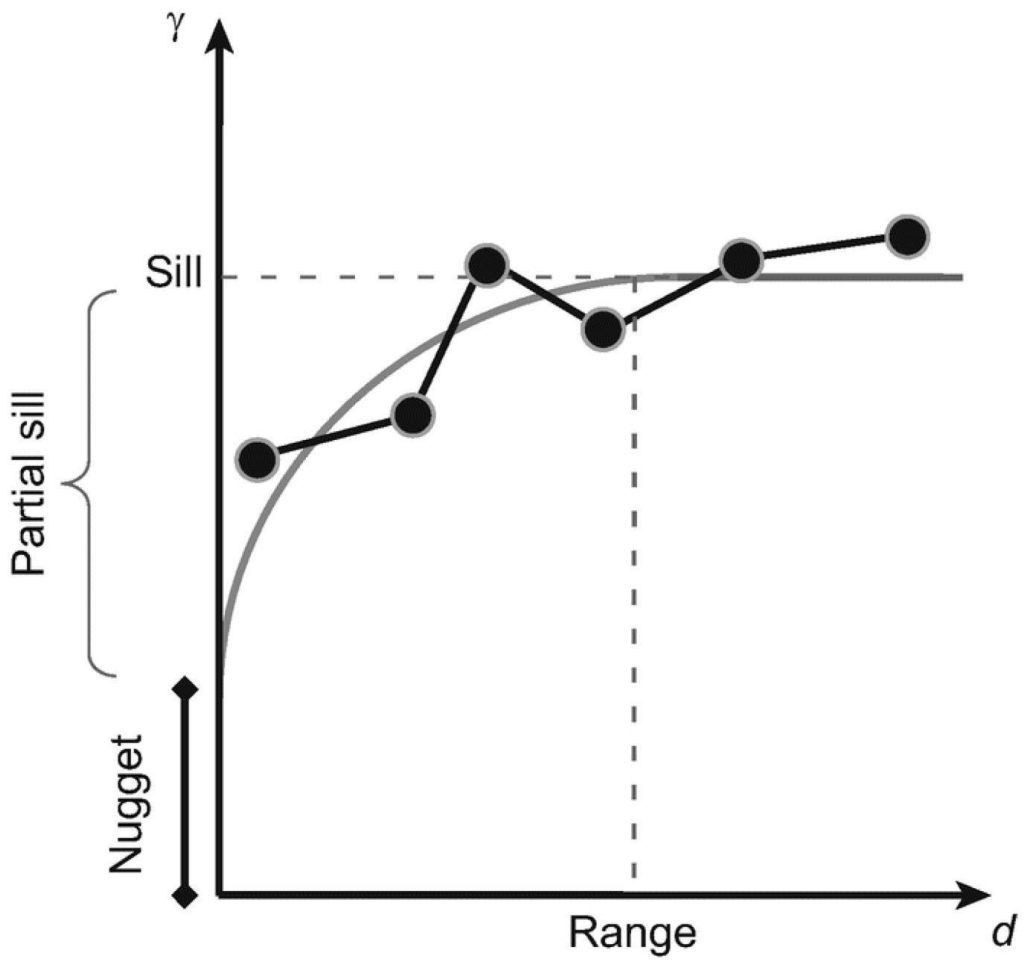
Range: distance above which there is no spatial dependence

- Points separated by this distance are no longer autocorrelated

Sill: Amount of variation at long distances.

- Amount of variation without local influence.

- F+F figure 5.2

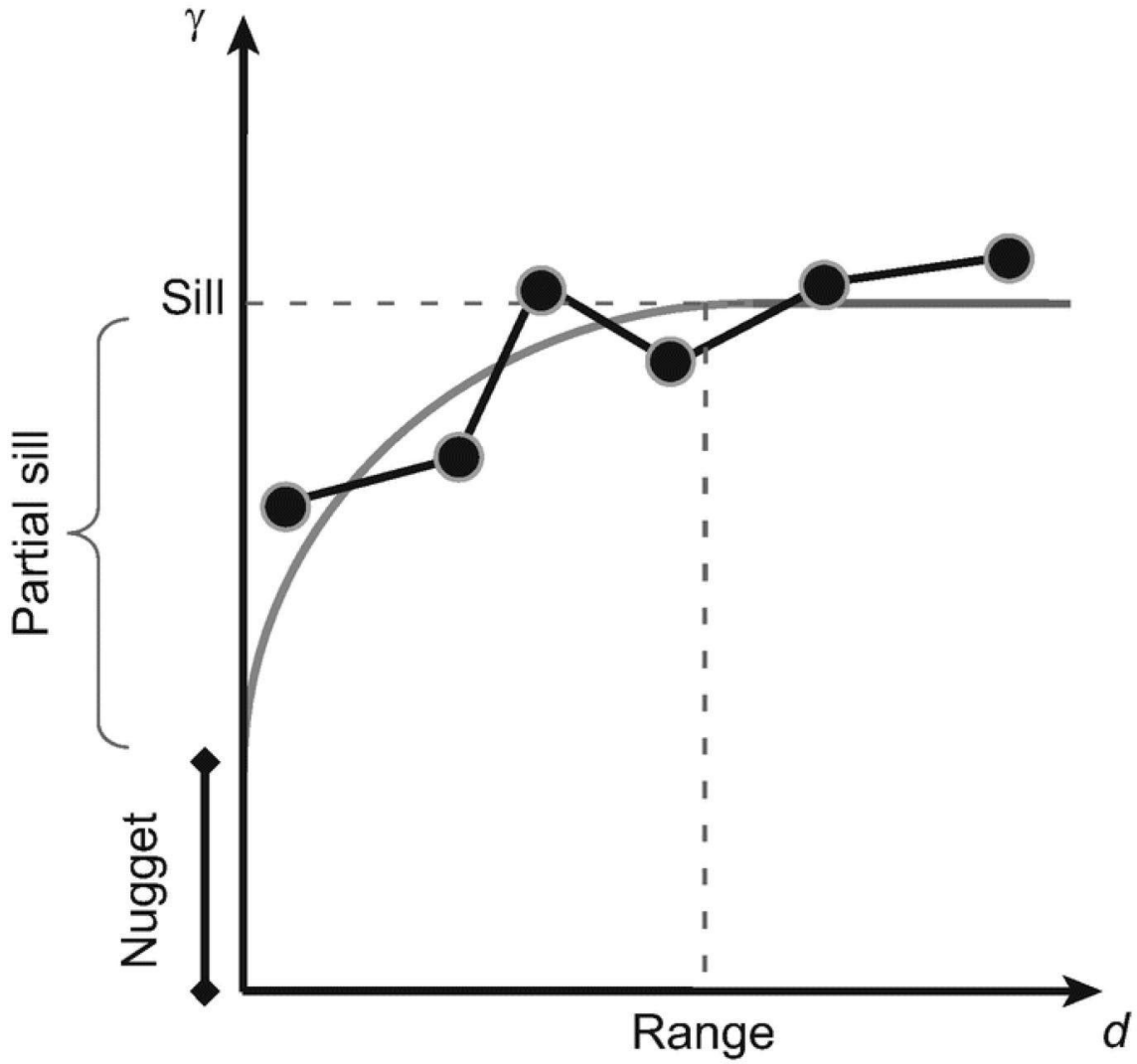


Spatial Statistics and Geostatistics

- Moran's I: spatial form of Pearson's correlation
- A correlogram is a plot of $I\,(d)$
- Semivariance (γ): spatial form of variance
- A variogram is a plot of $\gamma\ (d)$
- Interpolation methods:
  - nearest-neighbor interpolation
  - inverse distance weighted interpolation
  - **Kriging**

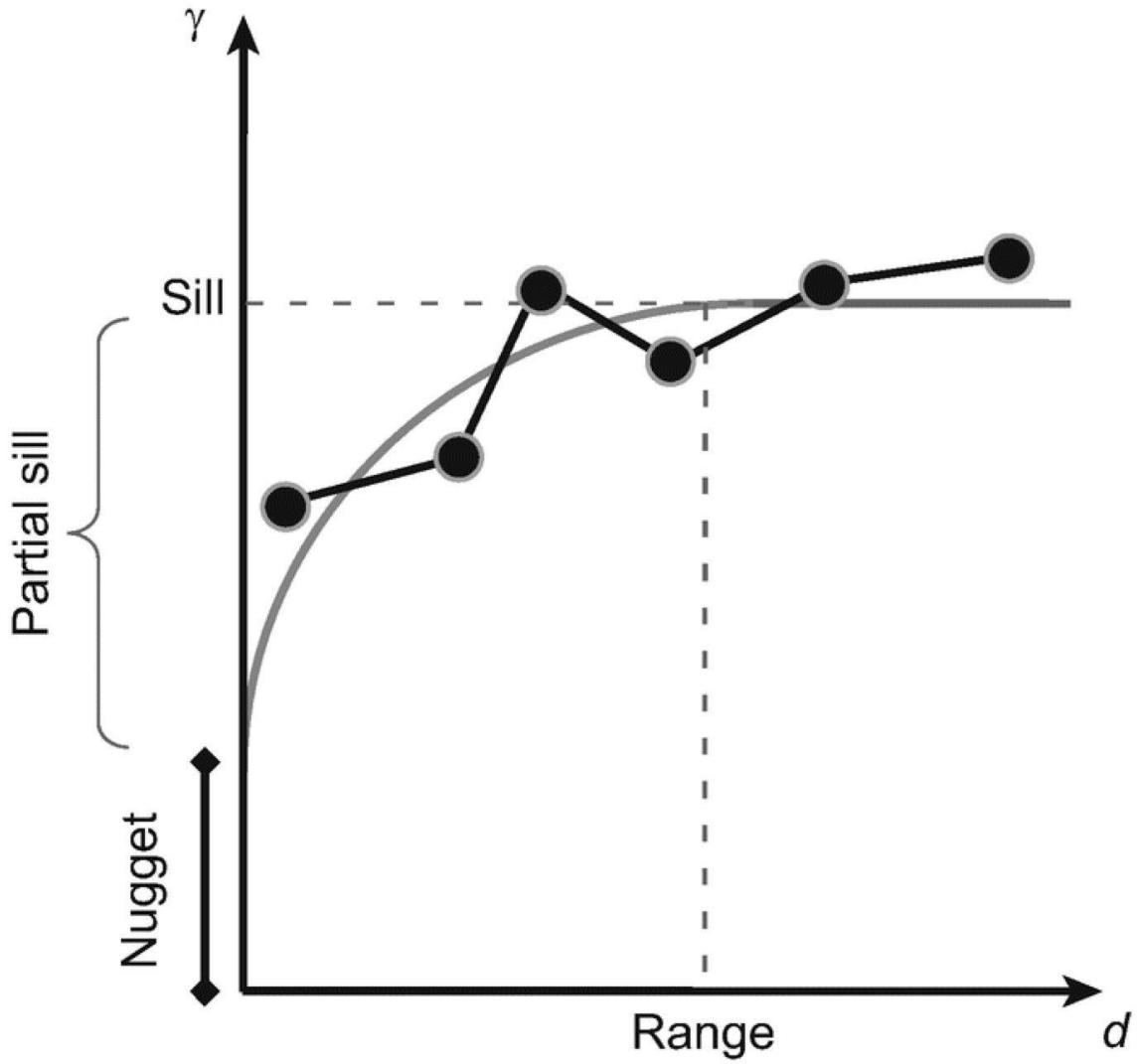
Variogram Components: The Nugget

How much variability do we expect at very nearby points?



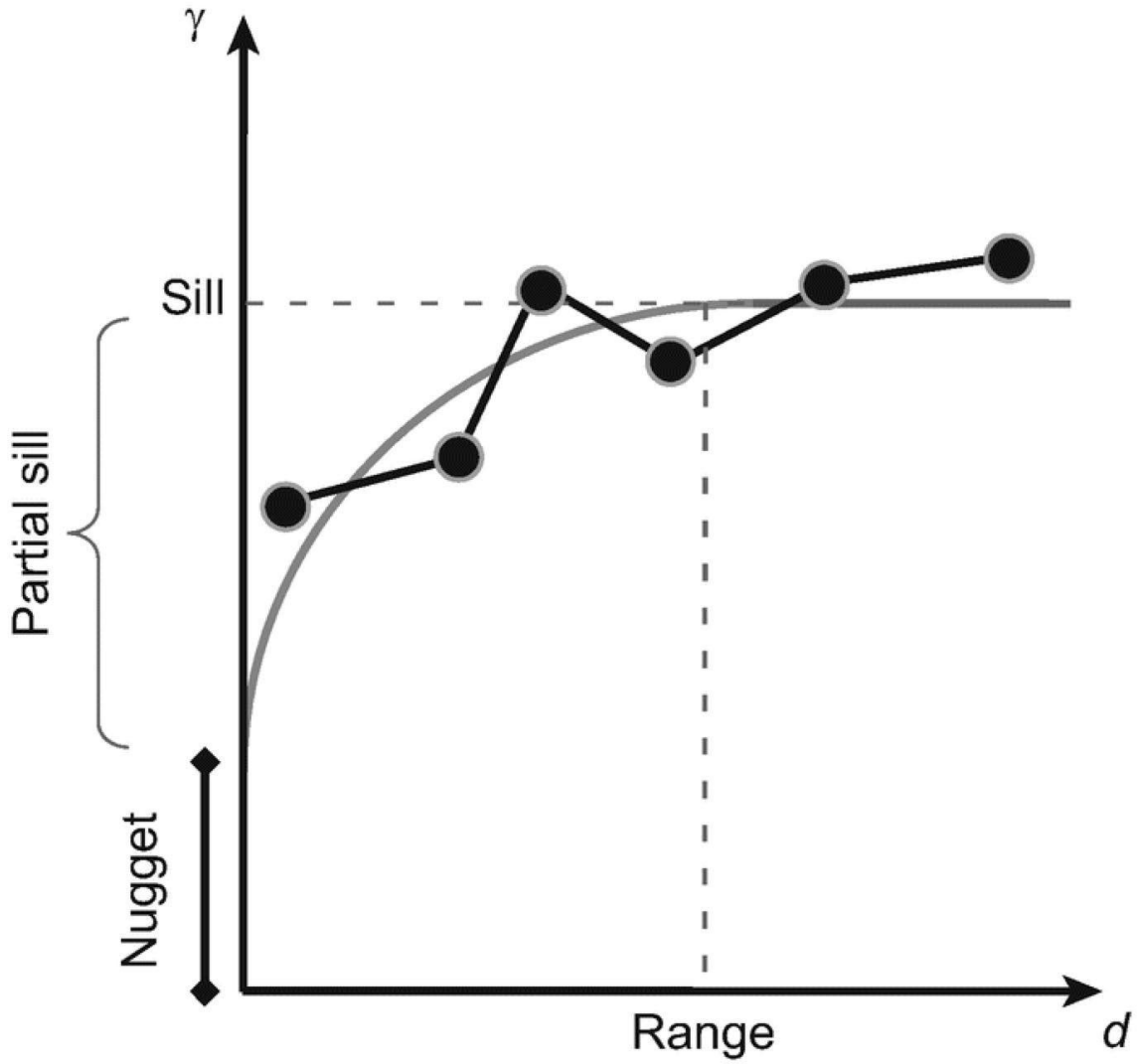
Variogram Components: The Range

What happens as we increase the distance between pairs of points?



Variogram Components: The Sill

What happens between pairs of points separated by large distances?



Anatomy of a Variogram

The variogram components help us answer different questions:

- nugget: How much variability is not explained by spatial proximity?
- range: How far do points have to be separated for spatial dependence to break down?
- sill: What is the variability among points that are distant enough to be spatially independent?

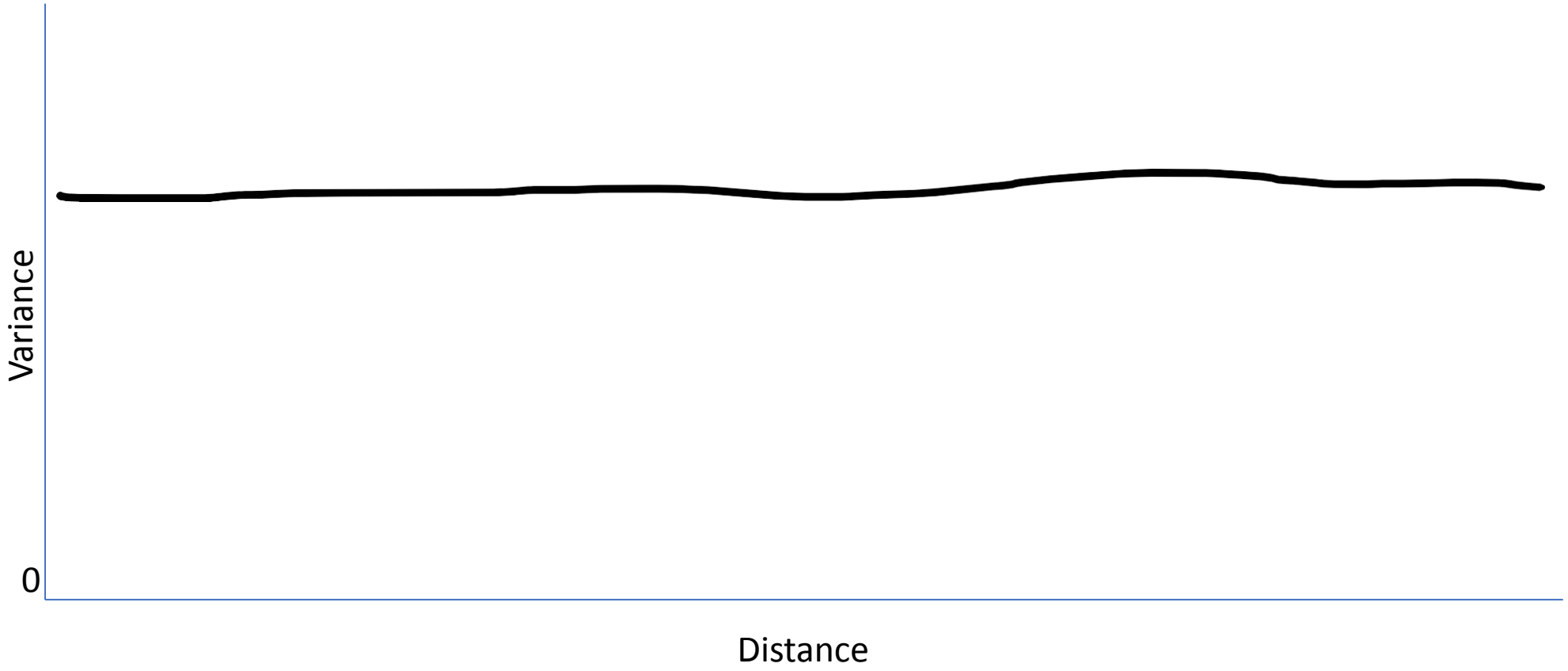
Sketching Variograms and Correlograms

Let's draw some variograms and correlograms:

- We have observed a value of Z at one location.
  - Scenario 1: Knowing the outcome of a stochastic process, z_i, tells us nothing about any other realizations (nearby or far)
  - Scenario 2: Nearby points are similar, separated points are different.
  - Scenario 3: No spatial dependence.
  - Scenario 4: Nearby points are identical, far points are not correlated

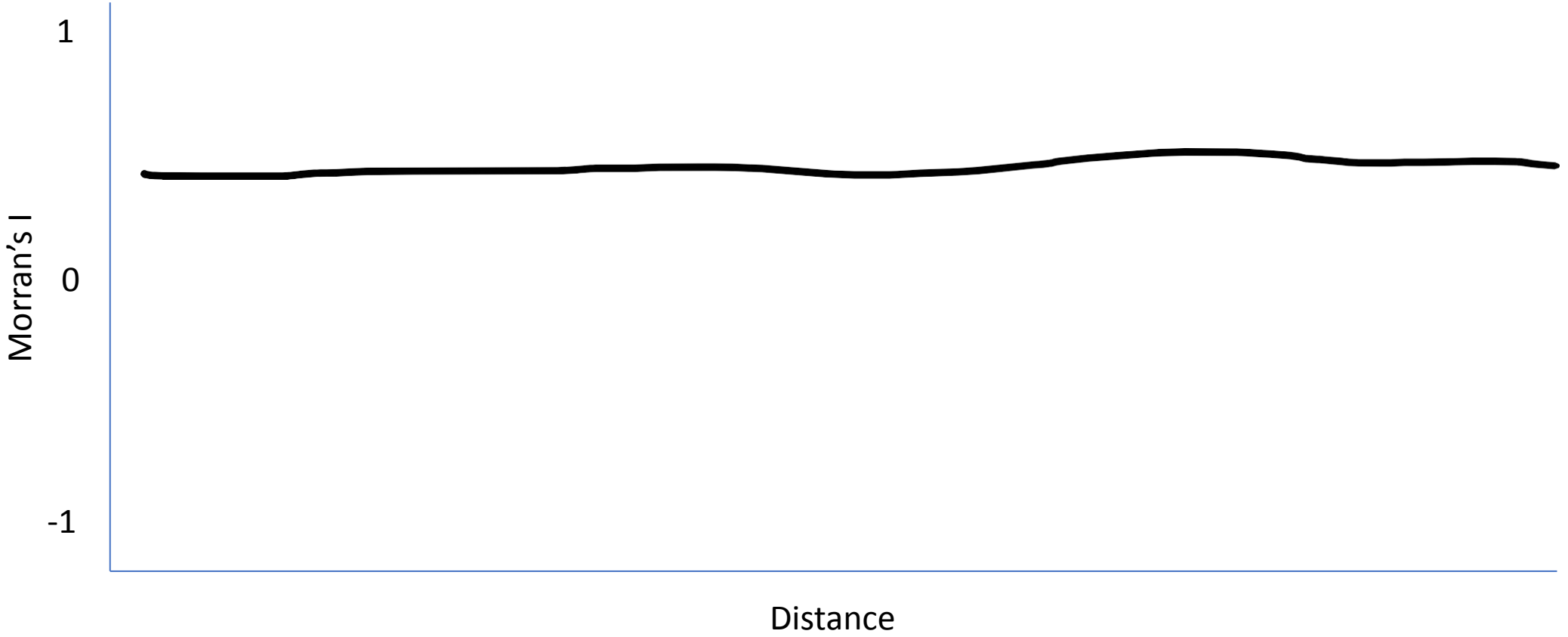
Variogram 1

- Scenario 1: Knowing the outcome of a stochastic process, z_i, tells us nothing about any other realizations (nearby or far)



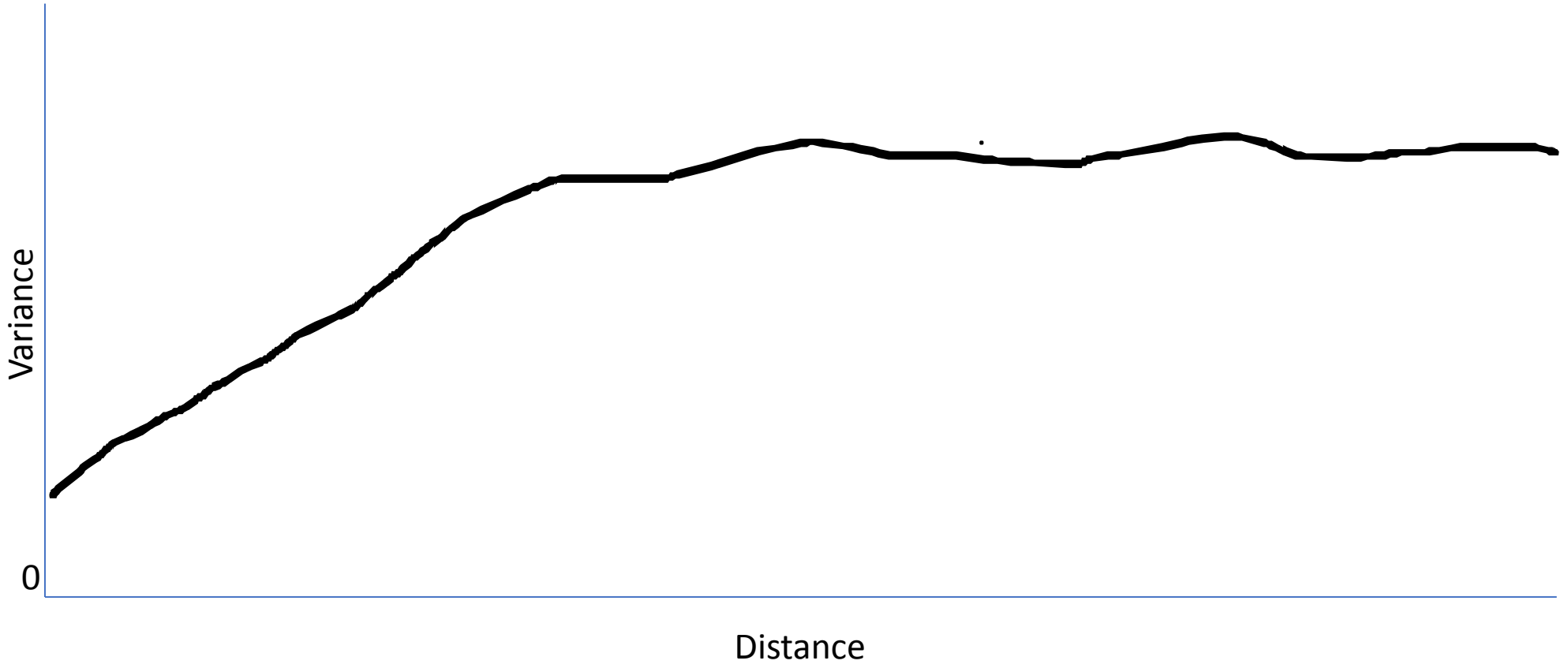
Correlogram 1

- Scenario 1: Knowing the outcome of a stochastic (random) process, z_i, tells us nothing about any other realizations (nearby or far)



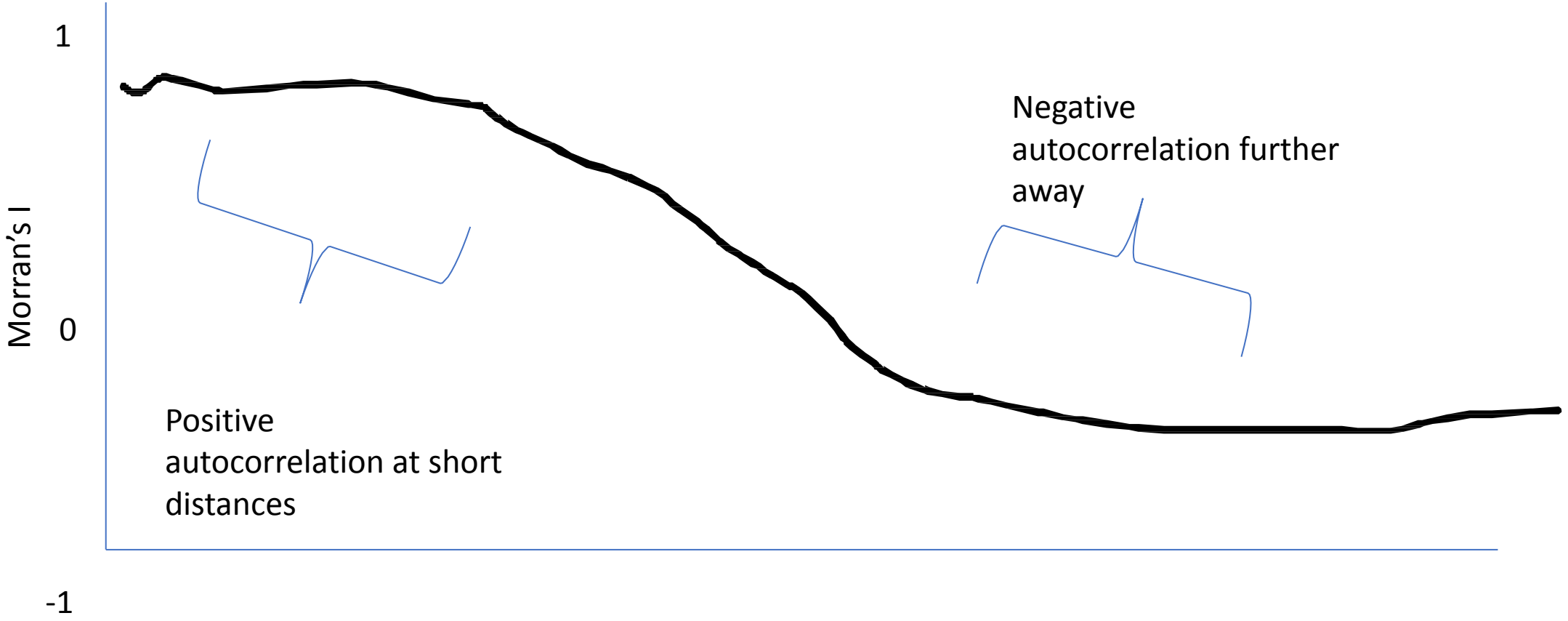
Variograms 2

- Scenario 2: Nearby points are similar, separated points are different.



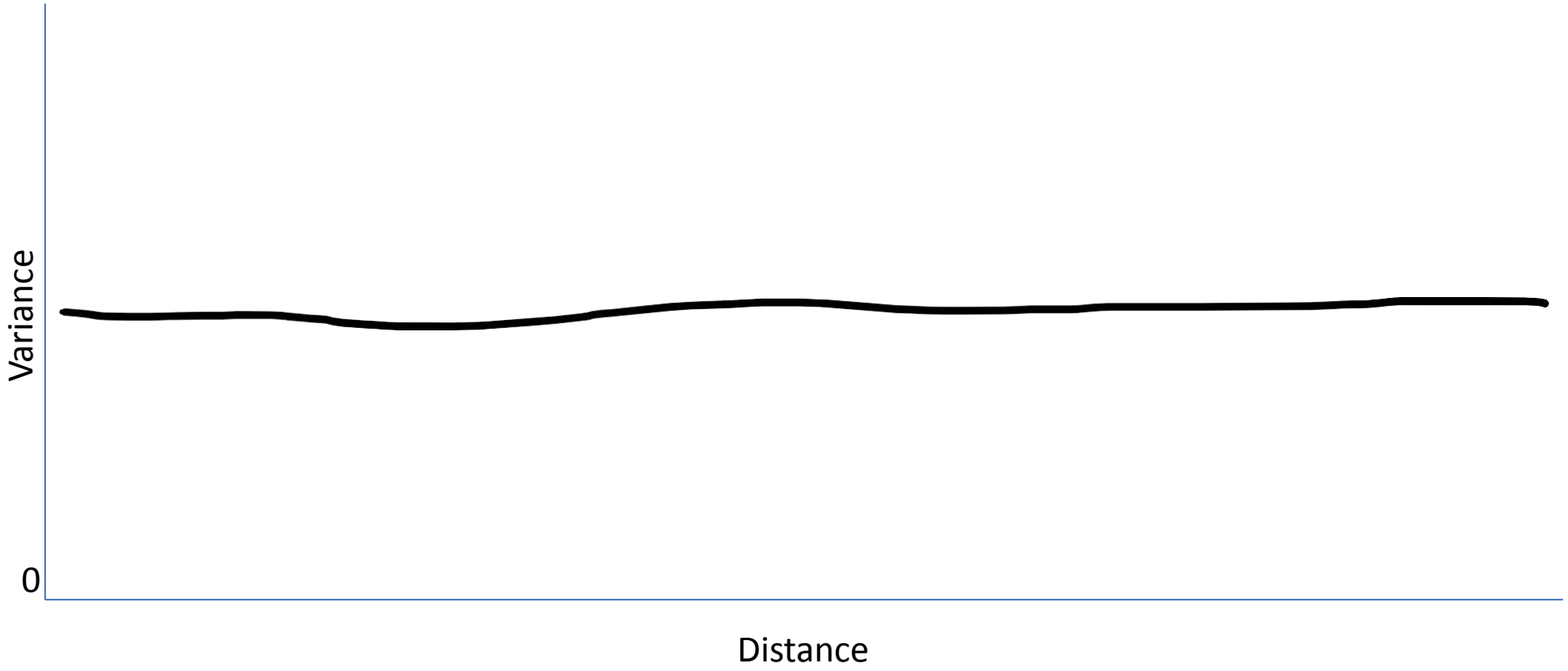
Correlogram 2

- Scenario 2: Nearby points are similar, separated points are different: kind of like overdispersion



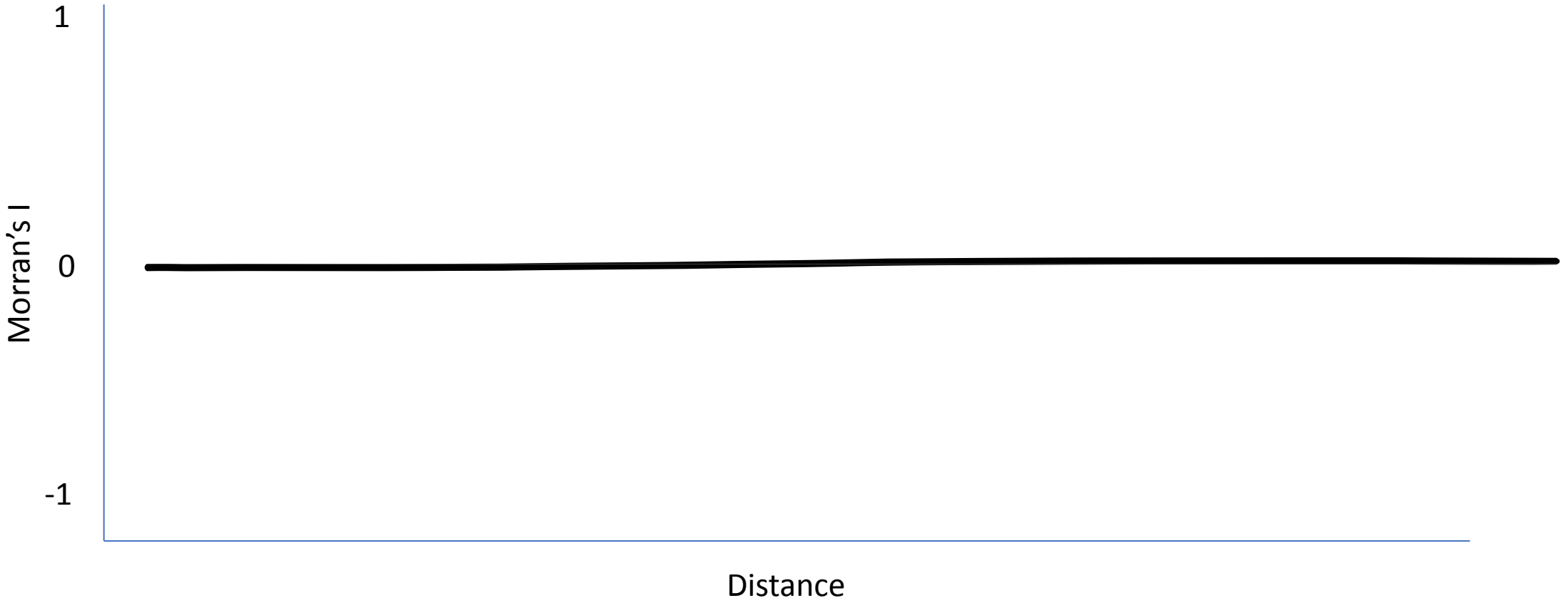
Variogram 3

- Scenario 3: No spatial dependence.



Correlogram 3

- Scenario 3: No spatial dependence.



Sketching Variograms

- Scenario 4: Nearby points are identical, far points are not correlated

Correlogram 4

- Scenario 4: Nearby points are identical, far points are not correlated

Moran's I and Semivariance

We can think of the differences with the help of the following informal definitions:

$I\ (\mathrm{d})$: "correlation as a function of distance"

$\gamma\ (d)$: "variance as a function of distance"

Variograms

- variogram() or vgm()
- A variogram is a function describing the degree of spatial dependence of a spatial random field or stochastic process!

3. Why is this useful?

Types of Variograms

- **Empirical Models**: The base model, it is calculated directly from the observed data without assuming any specific model
- **Spherical Models**: The most commonly used specific model, with a somewhat linear behavior at small separation distances near the origin, but flattening out at larger distances and reaching a sill limit.
- **Exponential Models**: Reach the sill asymptotically, with the practical range defined as that distance at which the variogram value is 95% of the sill.
  - Like the spherical model, the exponential model is linear at small distances near the origin, yet rises more steeply and flattens out more gradually.
  - Erratic data sets can sometimes be fit better with exponential models.

Kriging

- Kriging is a statistical technique used to predict the value of a variable at unmeasured locations based on observed values at known locations.
- The core idea behind kriging is that spatially closer points are more likely to have similar values than points that are farther apart.
  - You can see why this is related to spatial autocorrelation!
- Kriging predicts values and provides an estimate of the uncertainty or variance of the prediction.
  - This helps in understanding how confident we can be about the predictions

Connecting Variograms & Kriging

- Kriging is a spatial interpolation of data points based on variograms.
  - Variograms describe the amount of variation at each distances continuously!
- Variograms give Kriging functions the details on how to interpolate on the map with mathematically sound weights for the continuous distance from each point.
- The **variogram acts as the weighting data for the krige**, so it tells you how much weight should be attributed to points close and far away.
- Most of the time spent on a kriging analysis is in choosing the best variogram!

4. Variogram Demo

Step 1: Create a gstat object

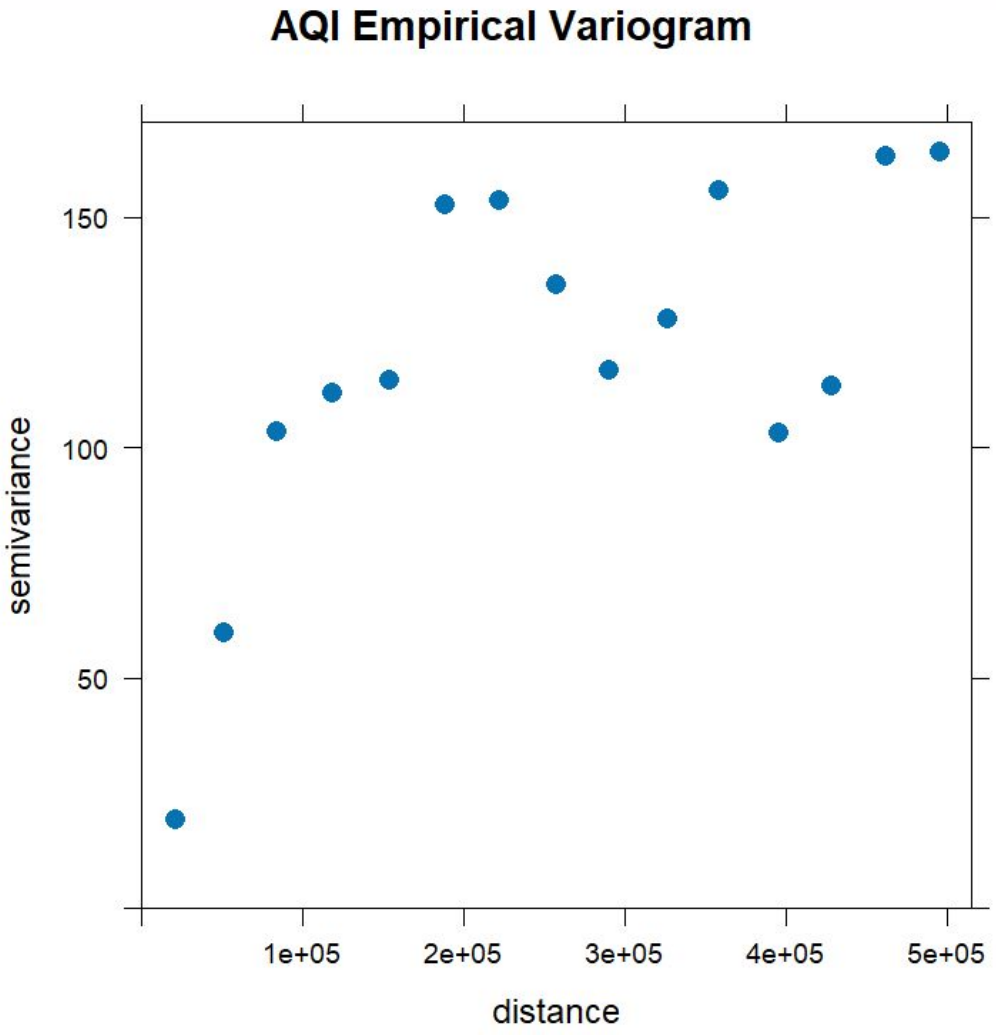
```
> gs_obj= gstat(
  formula = RESPONSE_VAR ~ 1,
  locations = SF_PTS)
```

- Why we use “RESPONSE_VAR ~ 1”: The ~1 signifies a model with an intercept only and no slope. This means that **the model does not include any explanatory variables affecting the response variable**.

Step 2: Generate Variogram w/ gstat Object

> vgm_emp = **variogram**(gs_obj)

- vgm = **V**ario**G**ram **M**odel
- emp = Empirical
- variogram() creates emp models
- This is your base model
- Don’t variograms have a line fit?



Step 3: Generate Model Fits for the Variogram

```
> vgm_mod_exp = vgm(
  model = "Exp",
  nugget = X,
  range = Y)
> vgm_mod_sph = vgm(
  model = "Sph",
  nugget = X,
  range = Y)
```

Note: You will have to inspect your empirical variogram to determine the nugget and range by visual estimation!

- vgm() creates your specific model variograms

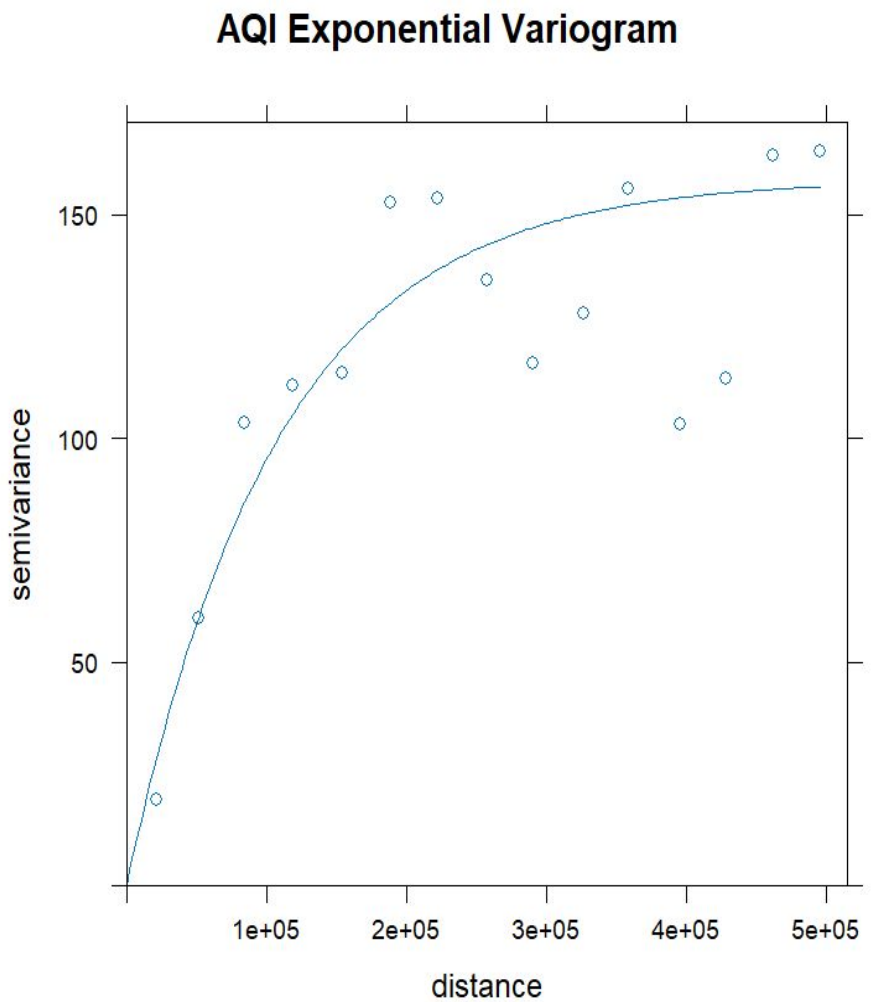
Step 4: Fit Exponential & Spherical Variograms

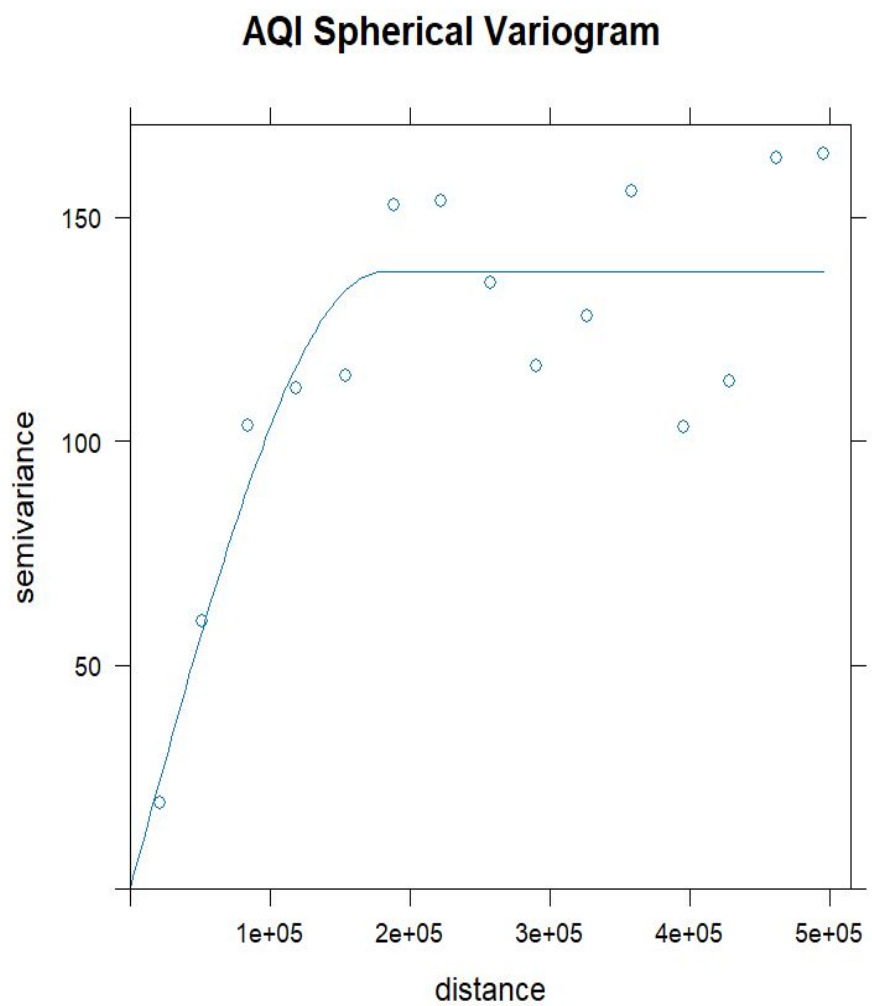
> vgm_fit_exp = **fit.variogram**(vgm_emp, vgm_mod_exp)

> vgm_fit_sph = **fit.variogram**(vgm_emp, vgm_mod_sph)

- Notice how the empirical variogram is the basis to each of these fits!
- The empirical model is the model based off of your actual data.
- The exponential & spherical models are based off samples of your data.

Step 5: Plot!


AQI Exponential Variogram
semivariance
150
100
50
1e+05
2e+05
3e+05
4e+05
5e+05
distance


AQI Spherical Variogram
semivariance
150
100
50
1e+05
2e+05
3e+05
4e+05
5e+05
distance

5. Krige Demo

Step 1: Create a template raster to store our outputs

1: Create a template raster grid for our outputs

> temp_rast = **rast**(STUDY_POLY, nrow = 200, ncol = 180) # choose row&col values per dataset

2: Project temp_rast to match ca_cnty CRS

> temp_rast = **project**(temp_rast, st_crs(STUDY_POLY)$wkt)

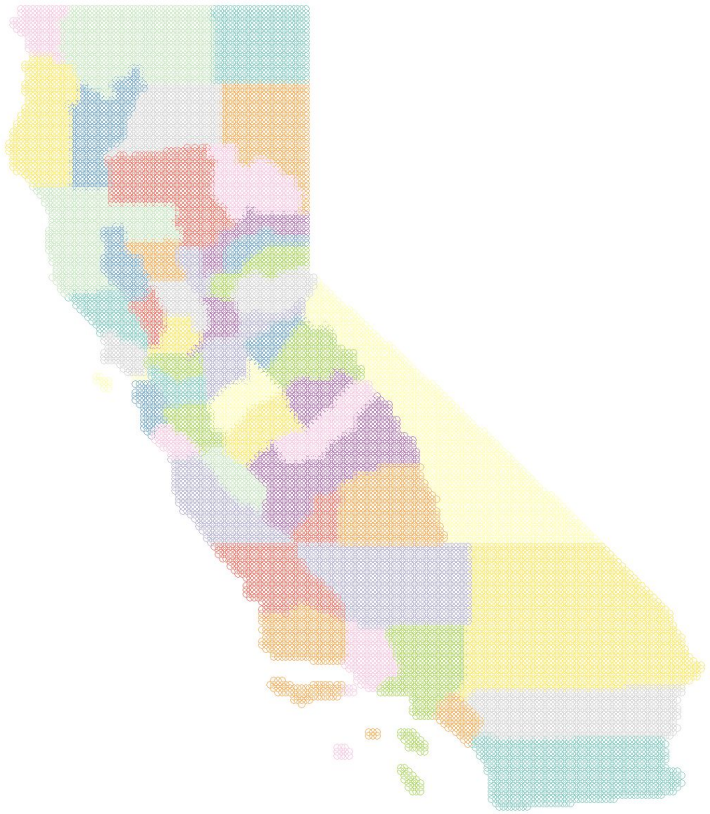
3: Convert the raster to points, and then SF

> krige_pts <- **as.points**(temp_rast)

> krige_pts <- **st_as_sf**(krige_pts)

4: Perform an intersection to have krige_pts align with ca_cnty

> krige_pts <- **st_intersection**(krige_pts, STUDY_POLY)



Step 2: Use our Variograms to Krige!

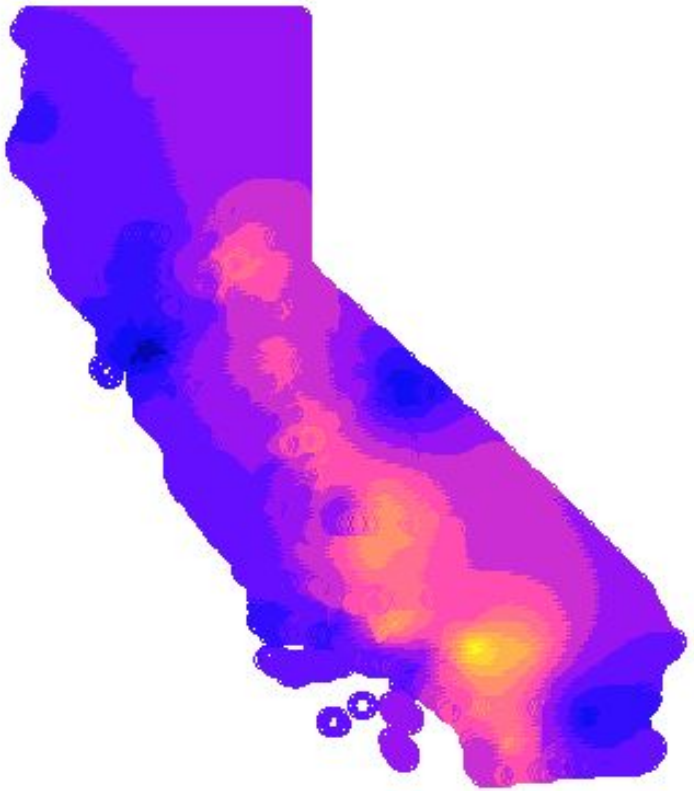
```
> VAR_krig_exp = gstat::krige(
  RESPONSE_VAR ~ 1,
  locations = SF_PTS,
  newdata = rast_pts, # Storage from Step 1
  model = vgm_fit_exp)
```

```
VAR_krig_sph = gstat::krige(
  RESPONSE_VAR ~ 1,
  locations = SF_PTS,
  newdata = rast_pts, # Storage from Step 1
  model = vgm_fit_sph)
```

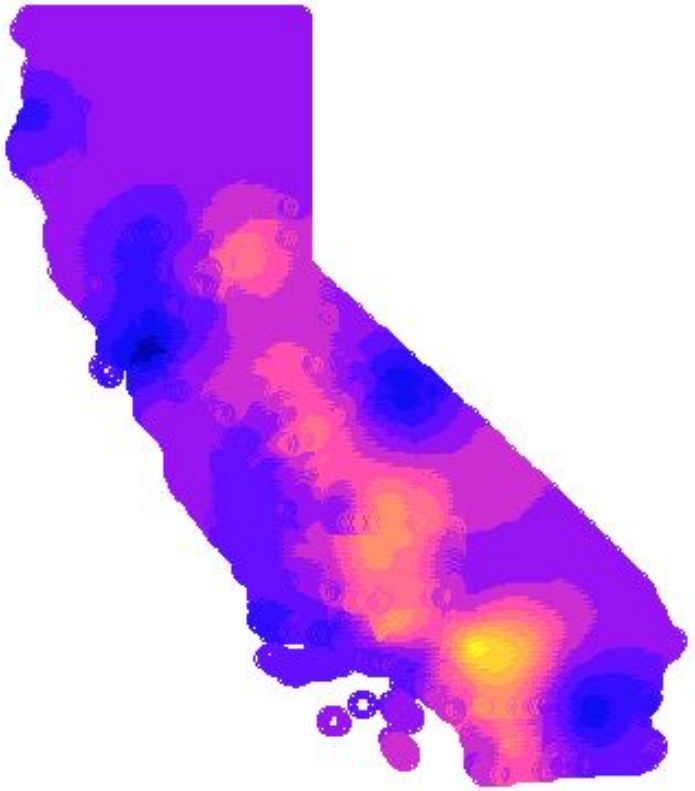
- Same response variable only model like before (RESPONSE_VAR ~ 1)
- Utilize original points for krige
- Store new predicted points values into our points template rast_pts
- **We created an exp and sph fit, and each will give us a different krige**!

Step 3: Take a look at your new maps!

Exponential



Spherical



6. Moran’s I Demo

How to Setup Moran's I Inputs for a Correlogram

1. Create a Distance Matrix
   - st_distance()
2. Determine Min/Max Distances
3. Create Distance Classes
   - This sets which distances to calculate Moran’s I on for each iteration
4. Calculate Euclidean distance between neighbors
   - dnearneigh(data, mindist, maxdist)
5. Create Weights Object
   - nb2listw() = "neighbors list weights"
6. Determine Model Error w/ a Linear Model

Step 1-3: Determine Moran's Distances of Interest

1: Create a Distance Matrix

> distmat = **st_distance**(SF_PTS)

2: Determine Maximum & Minimum Distances between California Ozone points

> maxdist = **max**(distmat)

> mindist = **min**(distmat)

3: Make a sequence of X distance classes between the min and max distances:

> n_dist_class = 10

> dist_classes = **seq**(mindist, maxdist, **length.out** = n_dist_class)

Step 4: Creating Distance Classes

1: Calculate Euclidean distance between neighbors

- dnearneigh() : Identifies neighbors of region points by Euclidean distance. It does this by a circular (radius) distance from each point in kilometers.
- Note: d1 and d2 need to be numeric, not units like dist_classes is (meters for example)

```
nbh_dist = dnearneigh(
  x = SF_PTS,
  d1 = as.numeric(dist_classes[1]), # Minimum Distance
  d2 = as.numeric(dist_classes[2]), # Maximum Distance
  longlat=F)
```

Step 5: Create a Distance Weighted Object

- nb2listw() : Supplements a neighbors list with spatial weights

```
> wts = nb2listw(
  neighbours = nbh_dist, # This is our nearest neighbor distances from Step 4
  style='W',
  zero.policy=T)
```

Step 6: Determine Variable Error

```
# 1. Create a linear model of our variables of interest

> var_lm = lm(VAR1 ~ VAR2, data = SF_PTS)

# 2. Extract residual error to SF_PTS from the model fit!

> SF_PTS$resids = residuals(var_lm)
```

Step 7: Run Moran's I

```
> mor_i = moran.test(
  SF_PTS$VAR,
  listw = wts,
  randomisation=F,
  zero.policy=T)
```

```
        Moran I test under normality

data:  ca_ozone$AQI
weights: wts_ca
n reduced by no-neighbour observations

Moran I statistic standard deviate = 5.9643, p-value = 1.229e-09
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.465038880      -0.013698630       0.006442845
```

7. Correlogram Demo

Workflow Outline

- Essentially we want to loop through different distances away from points, and calculate Moran's I for each distance.
- We can't do this continuously, so we choose a few distances between the minimum and maximum distance values.
- To make sure our output looks sound, we can calculate a **confidence interval** to quantify how 'confident' we are that we have estimated true population values based on our sample data.

DO NOT WORRY I'VE DONE THIS FOR YOU

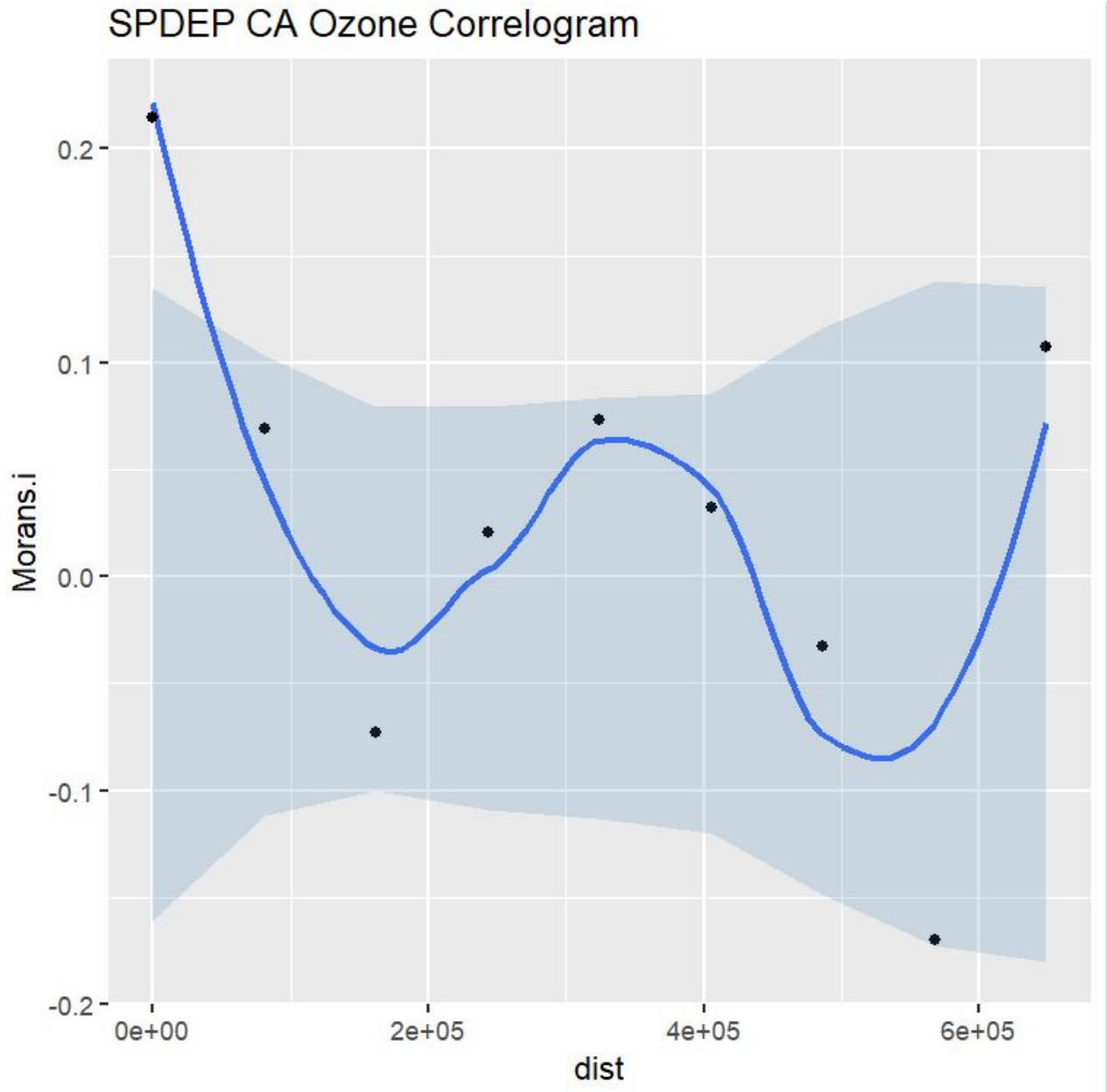
See Correlogram_Function.R

Take a look at our Looped Moran's I Output

dist	Morans.i	Null.lcl	Null.ucl	Pvalue
0.00	0.21461274	-0.1517972	0.14331189	0.002
81058.84	0.06880281	-0.1168357	0.10054886	0.070
162117.67	-0.07289021	-0.1027791	0.08550686	0.900
243176.51	0.02070692	-0.1017314	0.07756179	0.222
324235.35	0.07293153	-0.1127387	0.08691643	0.042
405294.18	0.03231131	-0.1200164	0.08387007	0.186
486353.02	-0.03238294	-0.1526775	0.12124260	0.640
567411.86	-0.16949472	-0.1695351	0.12051593	0.974
648470.69	0.10760327	-0.1690195	0.14282567	0.054

Interpretation

- Shaded blue area is the confidence interval that we calculated before.
- If our fitted line & points for ozone are within this blue area, meaning that the values are likely not spatially autocorrelated.



Quantifying Error w/ Linear Modeling - TLDR

- A linear model is a predicted fit of values based on some data.
- Linear models **predict an outcome based on a linear combination of input features**.
- Fits a "**y = mx + b**" for a dataset
- The model learns the best values m and b to fit your data
- A residual error is the difference between the expected value (prediction), and the actual values observed.
  - **Residual = Observed - Expected**
- We are interested in this residual error to account for it in our krige prediction.

How to Linear Model in R

1: Create your linear model with lm()

> aqi_lm = **lm**(**ozone ~ AQI**, data = ca_ozone)

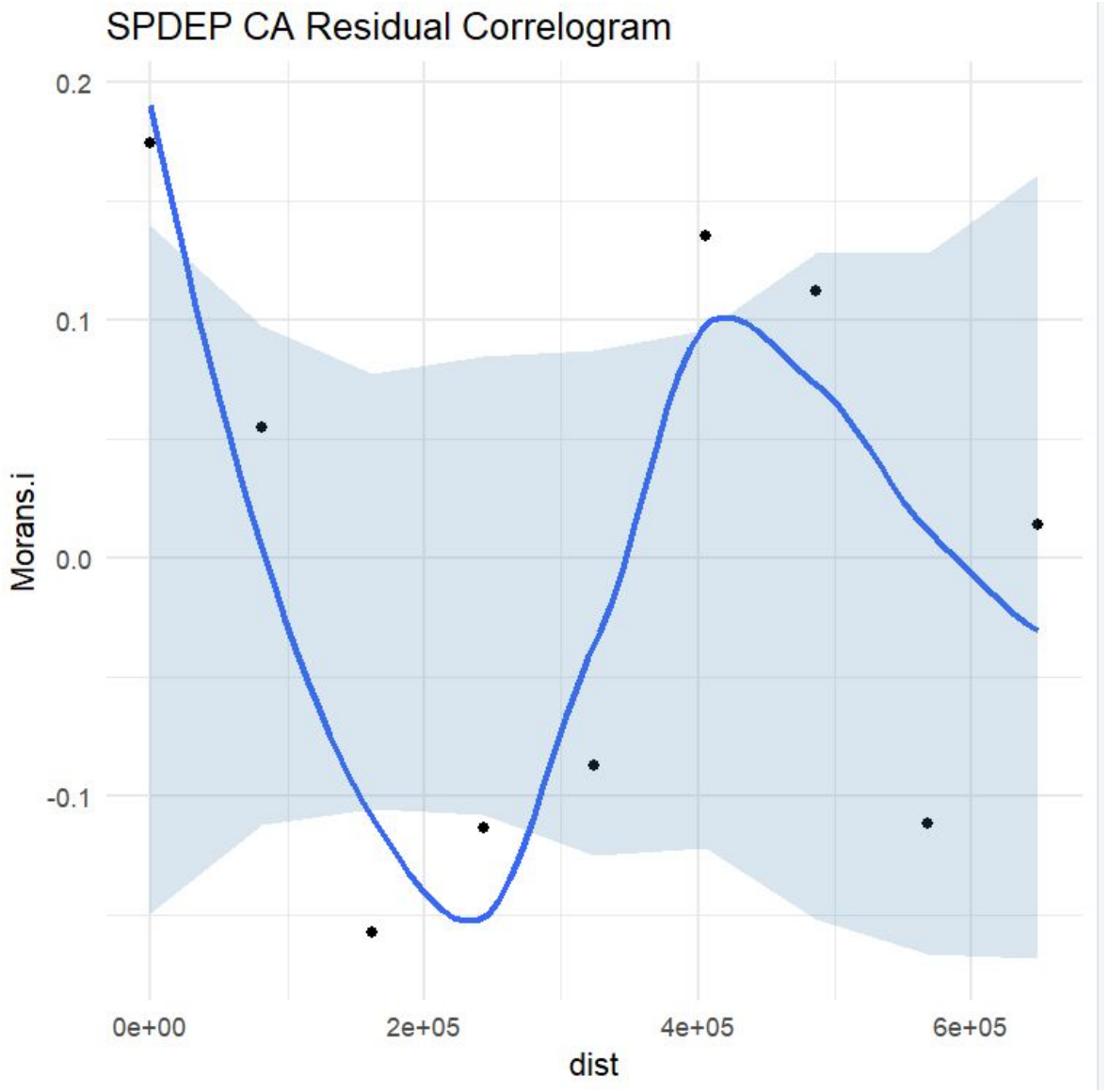
- This model determines the relationship between ozone & AQI

2: Extract residuals to ca_ozone from the model fit!

> ca_ozone**$resids** = **residuals**(aqi_lm)

Interpretation

- Most of the points are within the envelope, meaning that the residuals are likely not spatially autocorrelated!
- The envelope describes the the null hypothesis of Moran's I, which is that there is no spatial autocorrelation–so by being within the envelope the data confirms that for the most part the data isn't spatially autocorrelated!



Thank you for bearing with me!