Spatial Data Analysis in R

Spatial Dependence and Autocorrelation 2

Eco 697DR – University of Massachusetts, Amherst – Spring 2022
Michael France Nelson

Announcements

- I’m nearly caught up grading, I aim to have lab grades released by Wednesday.
- Becky is giving a workshop on interactive webmapping on Wednesday during our lab time
  - Show of hands, who would like to dedicate this week’s lab to her workshop?
- How was your break?

Examples of Spatial Autocorrelation

The Schelling Model of Segregation

Spatially-explicit agent-based model of residential patterns.

Agents are of two types.

Agents may relocate to vacant squares.

Parameters:

- Global: proportion of available housing
- Agent: preference for being near the same type.
- Neighborhood type: 4- or 8-neighbor
- In each round, agents may move to a vacant house if the neighborhood if they are 'unhappy'

RESEARCH ARTICLE | SOCIAL SCIENCES | FREE ACCESS

Understanding the social context of the Schelling segregation model

William A. V. Clark and Mark Fossett Authors Info & Affiliations

March 18, 2008 | 105 (11) | https://doi.org/10.1073/pnas.0708155105

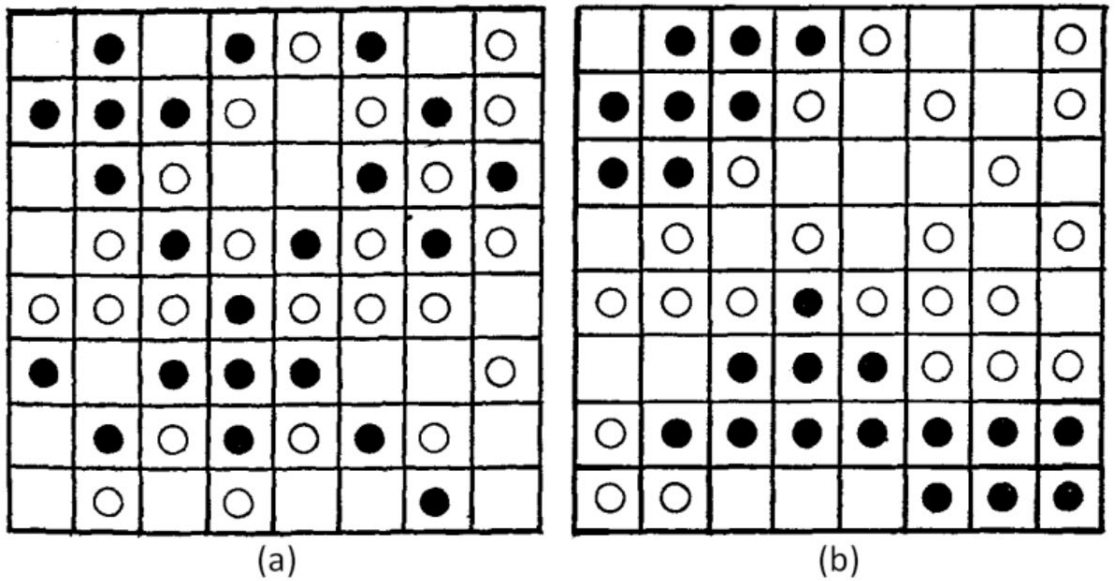


Figure 1. (a) initial condition of one of Schelling's experiments; (b) stable segregated pattern obtained in several iterations (Schelling 1974)

The Schelling Model of Segregation

- Schelling model is deliberately simple
- Stable patterns achieved in different number of steps.
- Above 75% similar wanted: stable configuration is never reached
- Check out the NetLogo model, how could you implement in R?

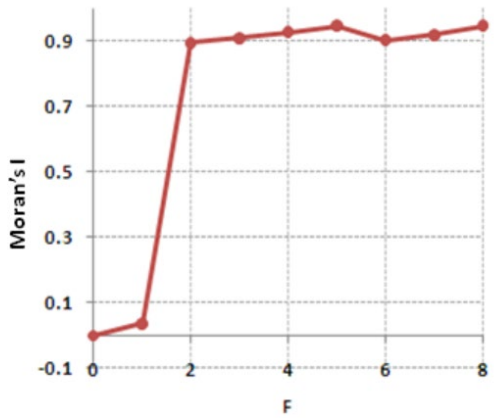
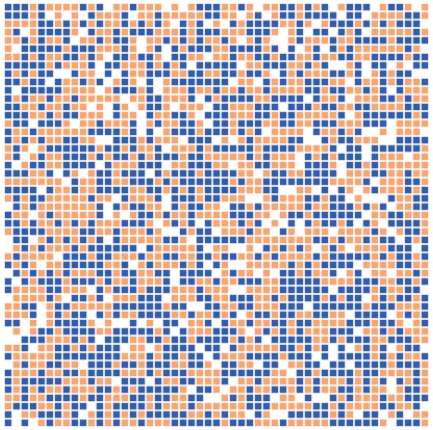
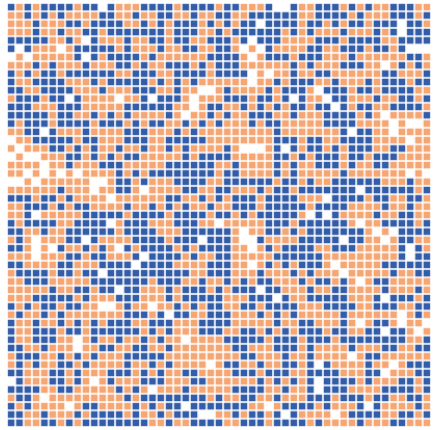


Figure 12 in Hatna, Erez, and Itzhak Benenson. "The Schelling Model of Ethnic Residential Dynamics: Beyond the Integrated - Segregated Dichotomy of Patterns." *Journal of Artificial Societies and Social Simulation* 15, no. 1 (2010): 6.

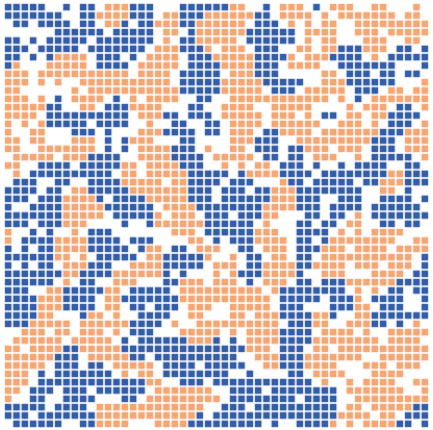
0% similar wanted



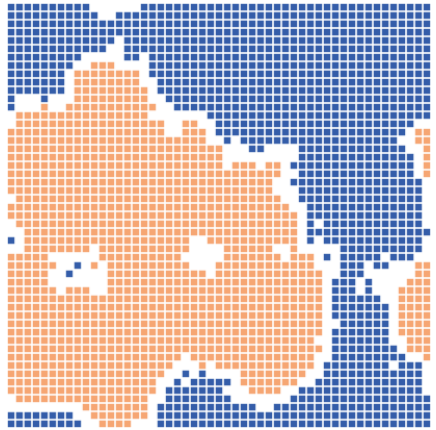
25% similar wanted



50% similar wanted



75% similar wanted



The Schelling Model of Segregation

- Schelling behavior is an example of emergent properties: system-level properties that may not be predicted by the micro-scale rules
- Variations on the original model can produce complex behavior:
  - More than 2 types of agents
  - Unequal proportions
  - Different utility functions
  - Random movements
  - Solid vs. liquid occupancy

This article has a great background on the model and describes a compelling application:

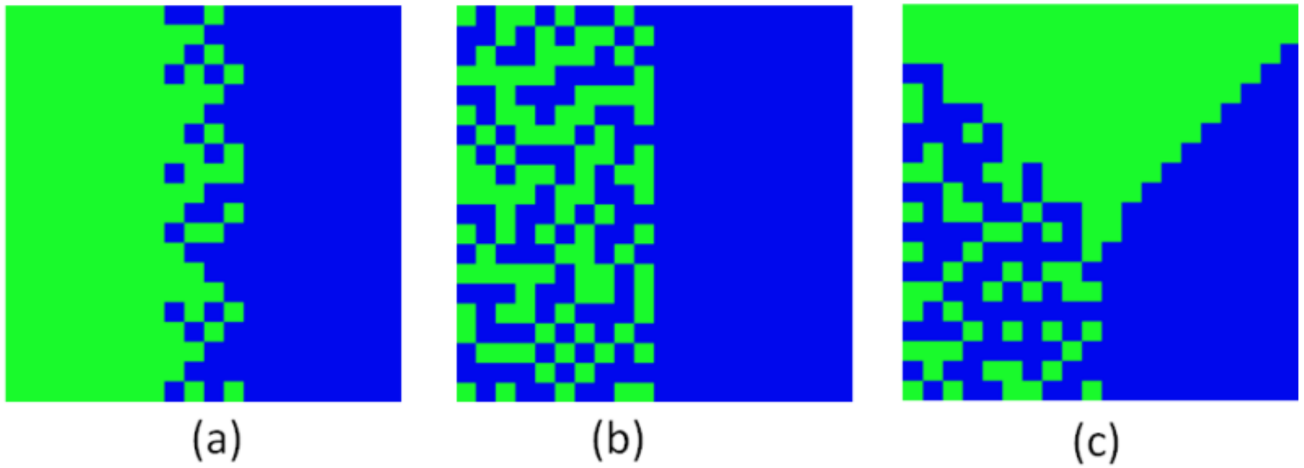
Erez Hatna[a] and Itzhak Benenson[b] (2012)

[a]Johns Hopkins University, United States; [b]Tel Aviv University, Israel

The Schelling Model of Ethnic Residential Dynamics: Beyond the Integrated - Segregated Dichotomy of Patterns

Journal of Artificial Societies and Social Simulation **15** (1) 6
<https://www.jasss.org/15/1/6.html>
DOI: 10.18564/jasss.1873

What kind of spatial autocorrelation is in these figures?



Hatna and Benenson 2016

Figure 4. Three qualitatively different patterns in Yaffo and Ramle and the corresponding Schelling-like patterns: (a) two segregated patches; (b) segregated patch adjacent to integrated patch; (c) integrated patches adjacent to two segregated patches

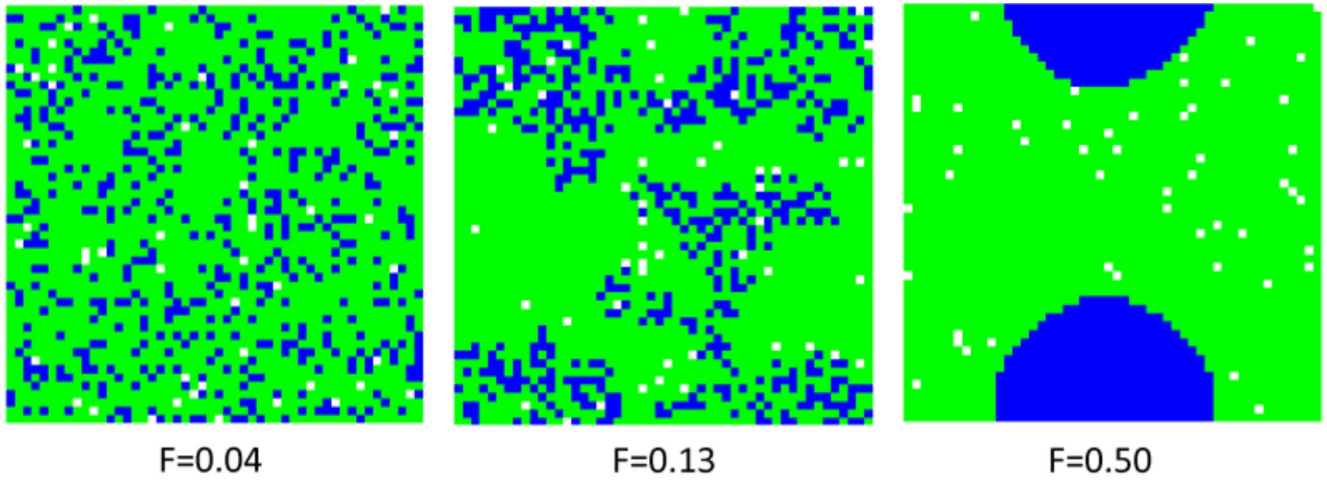


Figure 5. (a) integrated (b) mixed and (c) segregated persistent patterns produced by the Schelling model for the 0.2:0.8 Blue-to-Green size ratio

Factors contributing to negative autocorrelation

1. Disturbance
    1. Creates nearby dissimilar patches

2. Negative species interactions
    1. Competition for resources
    2. Territoriality

Research Article | Published: 23 January 2017

Disturbance increases negative spatial autocorrelation in species diversity

Shekhar R. Biswas, Rebecca L. MacDonald & Han Y. H. Chen

Landscape Ecology **32**, 823–834 (2017) | Cite this article

"However, since competition is the strongest driver of species evenness (Martin et al. 2005; Wilsey and Stirling 2007), negative spatial autocorrelation in evenness is likely related to competition."

Inferring process from pattern?

Checkerboard of clearcuts – southwest of Eugene, OR

- Historical clearcuts have created a patchwork pattern in the Coast Range of western Oregon.
- How would the variograms and correlograms of DBH look?

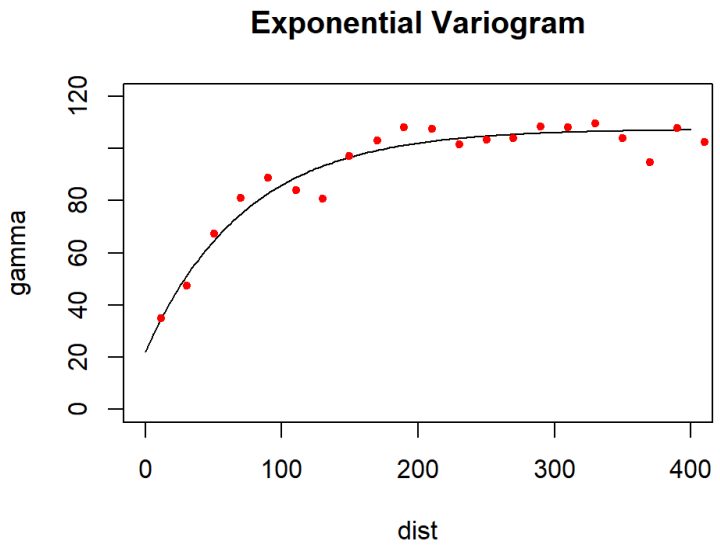
Variograms and Kriging

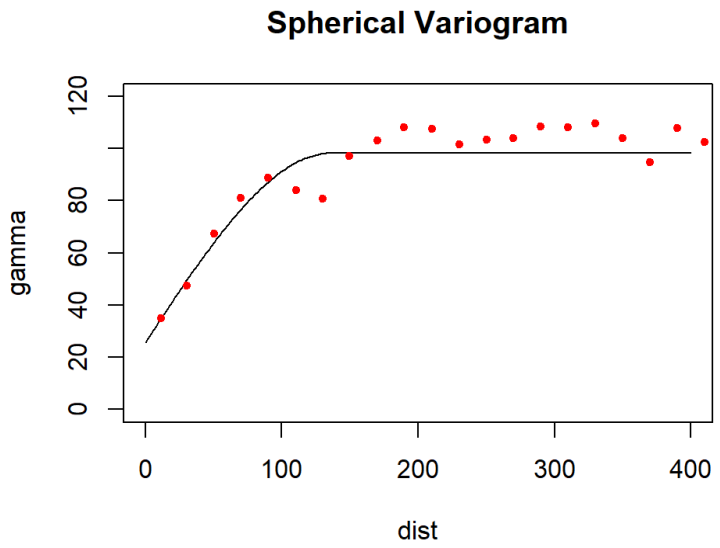
Theoretical Variograms

Mathematical functions with formulae derived from probabiltiy theory.

Many flavors: exponential, Gaussian, spherical, etc…

We use software to find a ‘best’ fit variogram.





Interpolation: Kriging

Kriging is a type of probabilistic smoothing.

Reminiscent of General Additive Models (GAMs), like Locally Weighted Regression (LOESS/LOWESS).

“Environmental surveys are almost always based on samples, but in general the measurements represent a continuum in space from which the sample has been drawn.” – Oliver and Webster 2014

The Kriging Family

A typical spatial statistical assumption: Stationarity

- What is stationarity?

Family of Kriging techniques

- Ordinary Kriging - assumes stationarity
- Universal Kriging - assumes linear or polynomial trend, i.e. non-stationarity
- Indicator Kriging – discrete response
- Co-Kriging – two variables

Kriging Walkthrough

California Ozone Pollution

Procedure for Kriging in gstat

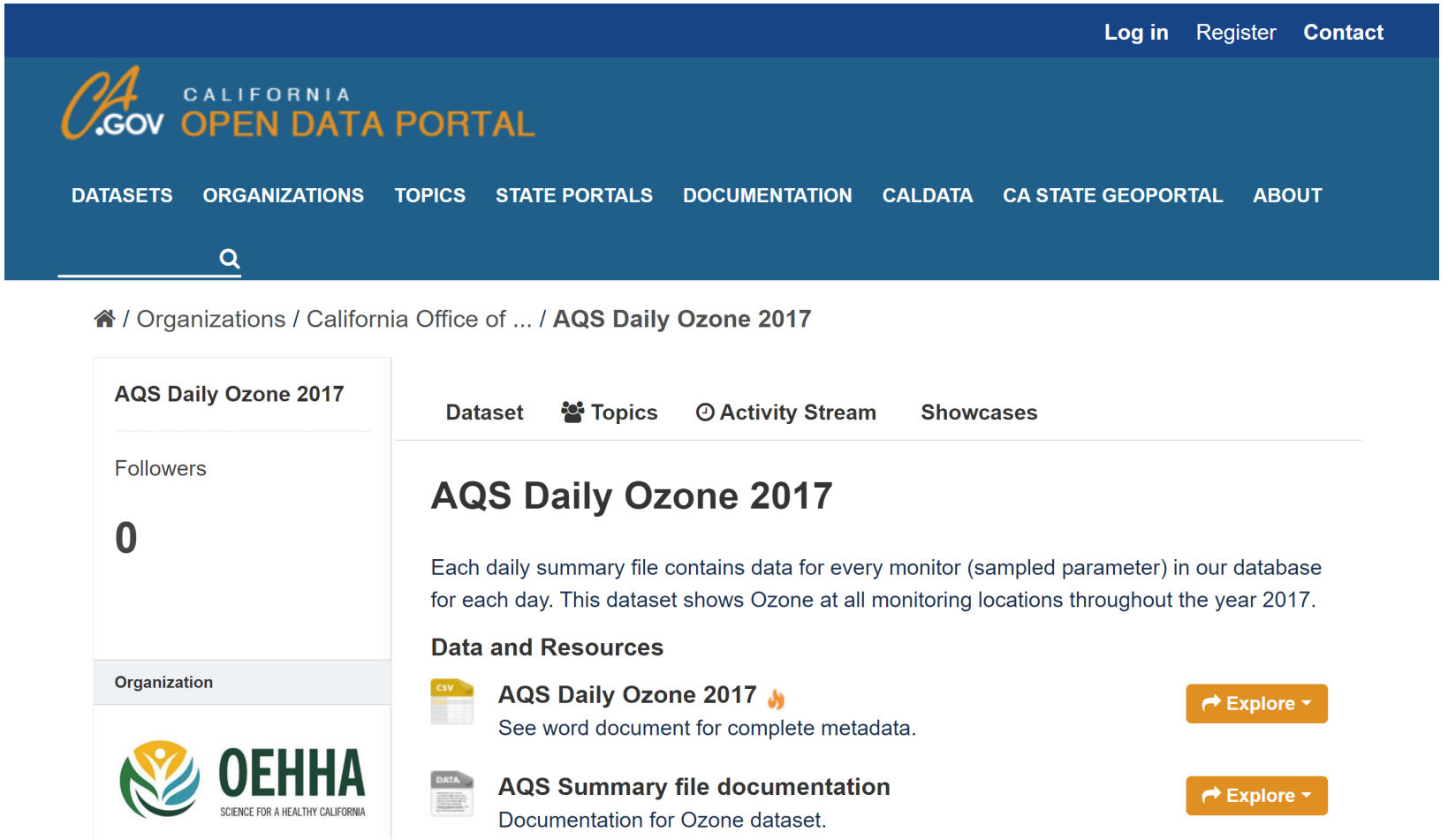
1. Acquire/Load data (London home prices in GWmodel package)
2. Visualize your data
3. Create a gstat object
4. Fit variograms, variogram model selection
5. Create a template grid
6. Krige!

We'll walk through the process with some California ozone pollution data.

Data Source

I aggregated ozone readings by sensor site to get an annual average

- CA Open Data Portal: https://data.ca.gov/dataset/aqs-daily-ozone-2017

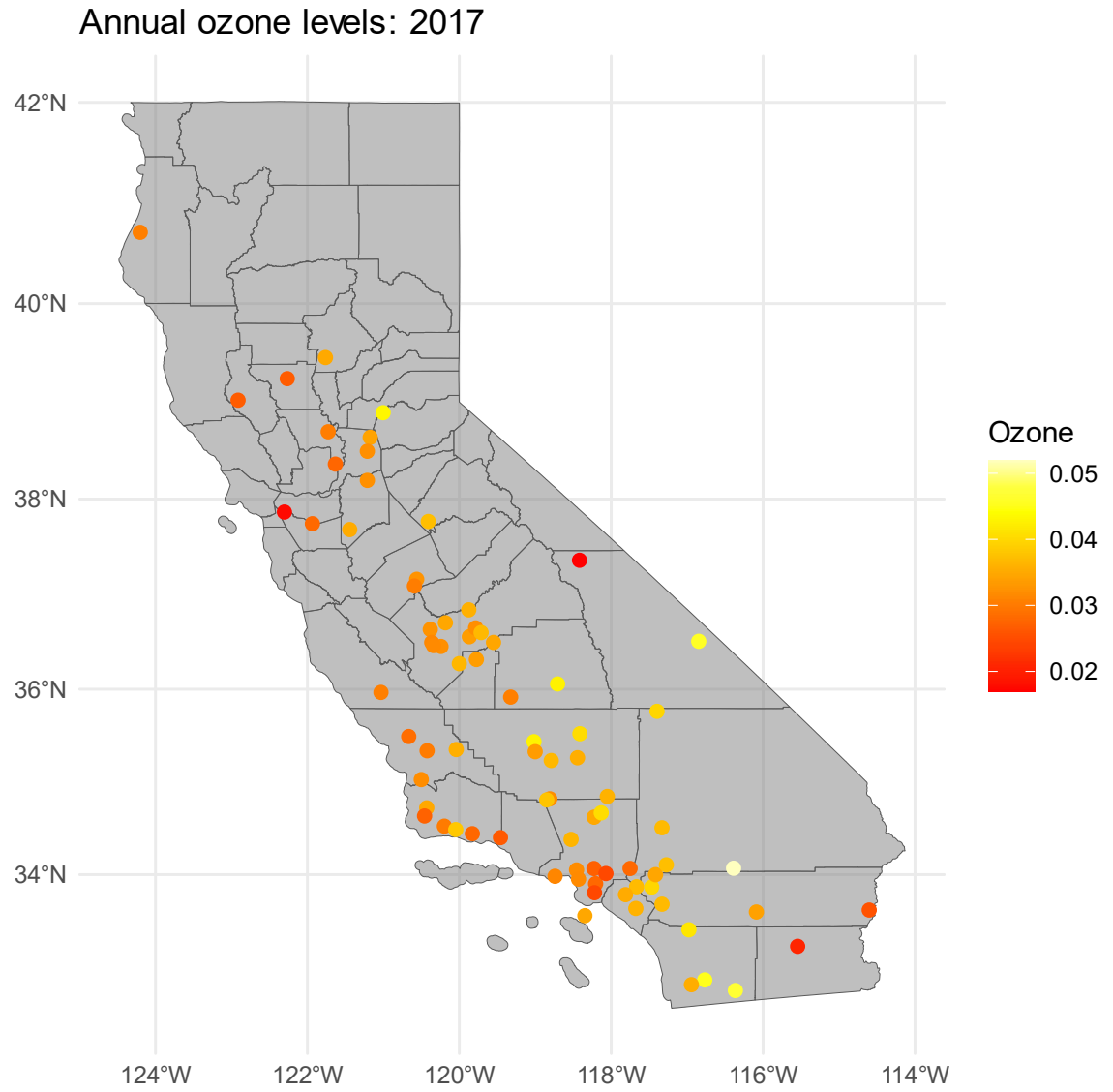


Visualize the Data!

- I used sf objects and ggplot to make a map:

```
ca_sf = st_as_sf(ca_cnty)
oz_sf = st_as_sf(ca_ozone)

ggplot() +
  geom_sf(data = ca_sf, lwd = 0.5, fill = gray(0.5, 0.5)) +
  geom_sf(data = oz_sf, mapping = aes(colour = ozone)) +
  scale_color_gradientn(colours = heat.colors(10), name = "Ozone") +
  theme_minimal() +
  ggtitle("Annual ozone levels: 2017")
```

Ozone Sensor Locations



Visualize the Data with ggplot!

ca_cnty and ca_ozone are SPDFs

st_as_sf() coerces to sf objects.

```
ca_sf = st_as_sf(ca_cnty)
oz_sf = st_as_sf(ca_ozone)

ggplot() +
  geom_sf(data = ca_sf, lwd = 0.5, fill = gray(0.5, 0.5)) +
  geom_sf(data = oz_sf, mapping = aes(colour = ozone)) +
  scale_color_gradientn(colours = heat.colors(10), name = "Ozone") +
  theme_minimal() +
  ggtitle("Annual ozone levels: 2017")
```

Visualize the Data!

geom_sf() works with sf objects:

- use the color aesthetic to map values in a column to a color on the map.
  - This works for point and polygon features, it's great for creating choropleth maps!

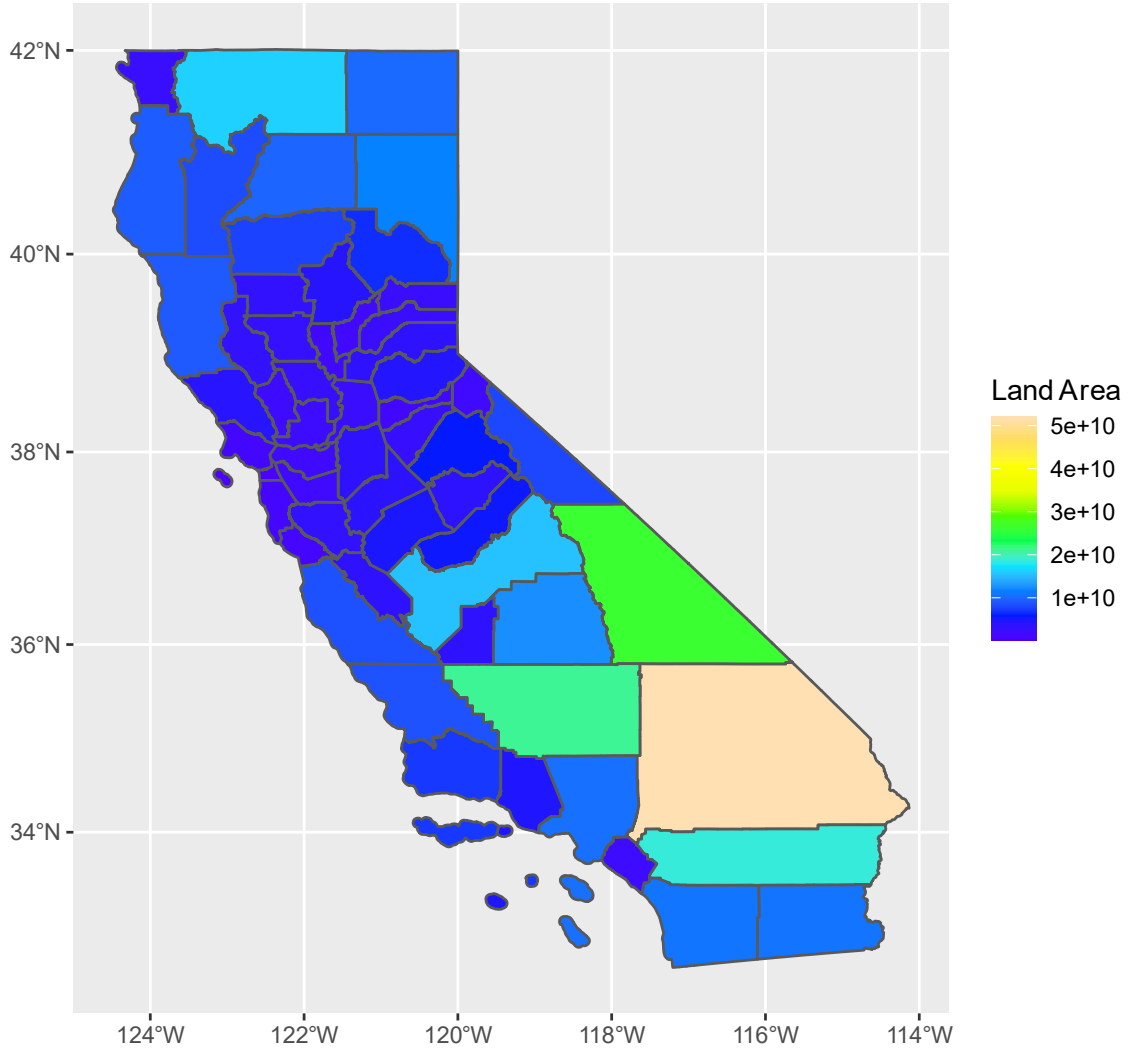
```
ca_sf = st_as_sf(ca_cnty)
oz_sf = st_as_sf(ca_ozone)

ggplot() +
  geom_sf(data = ca_sf, lwd = 0.5, fill = gray(0.5, 0.5)) +
  geom_sf(data = oz_sf, mapping = aes(colour = ozone)) +
  scale_color_gradientn(colours = heat.colors(10), name = "Ozone") +
  theme_minimal() +
  ggtitle("Annual ozone levels: 2017")
```

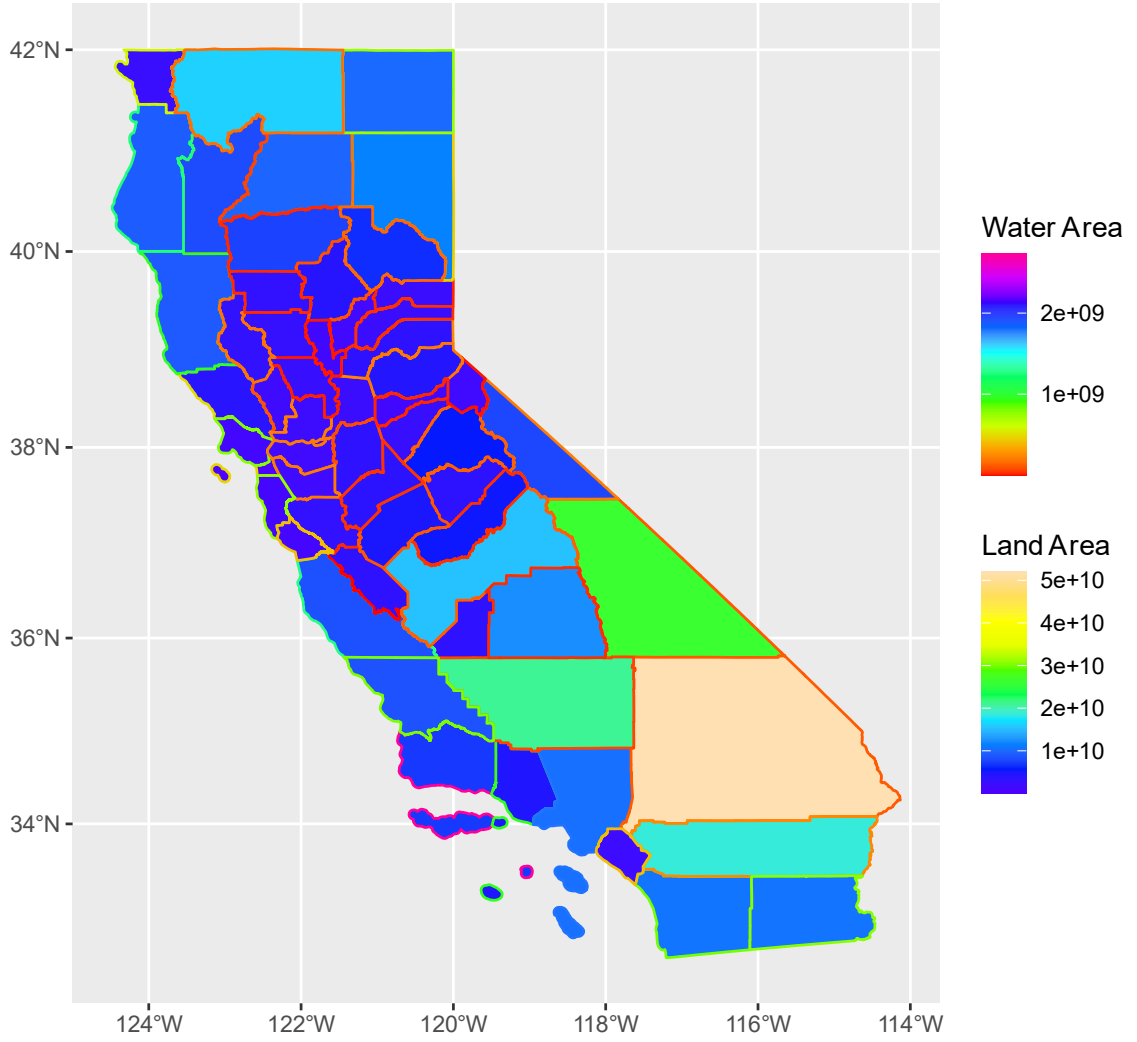
Detour: choropleth with sf and ggplot

I used the fill aesthetic with the land area (coerced to numeric) column to color the county polygons.

```
ggplot() +
  geom_sf(
    data = ca_sf,
    mapping = aes(fill = as.numeric(ALAND))) +
  scale_fill_gradientn(
    colors = topo.colors(10),
    name = "Land Area")
```

Some terrible maps (for illustration only)

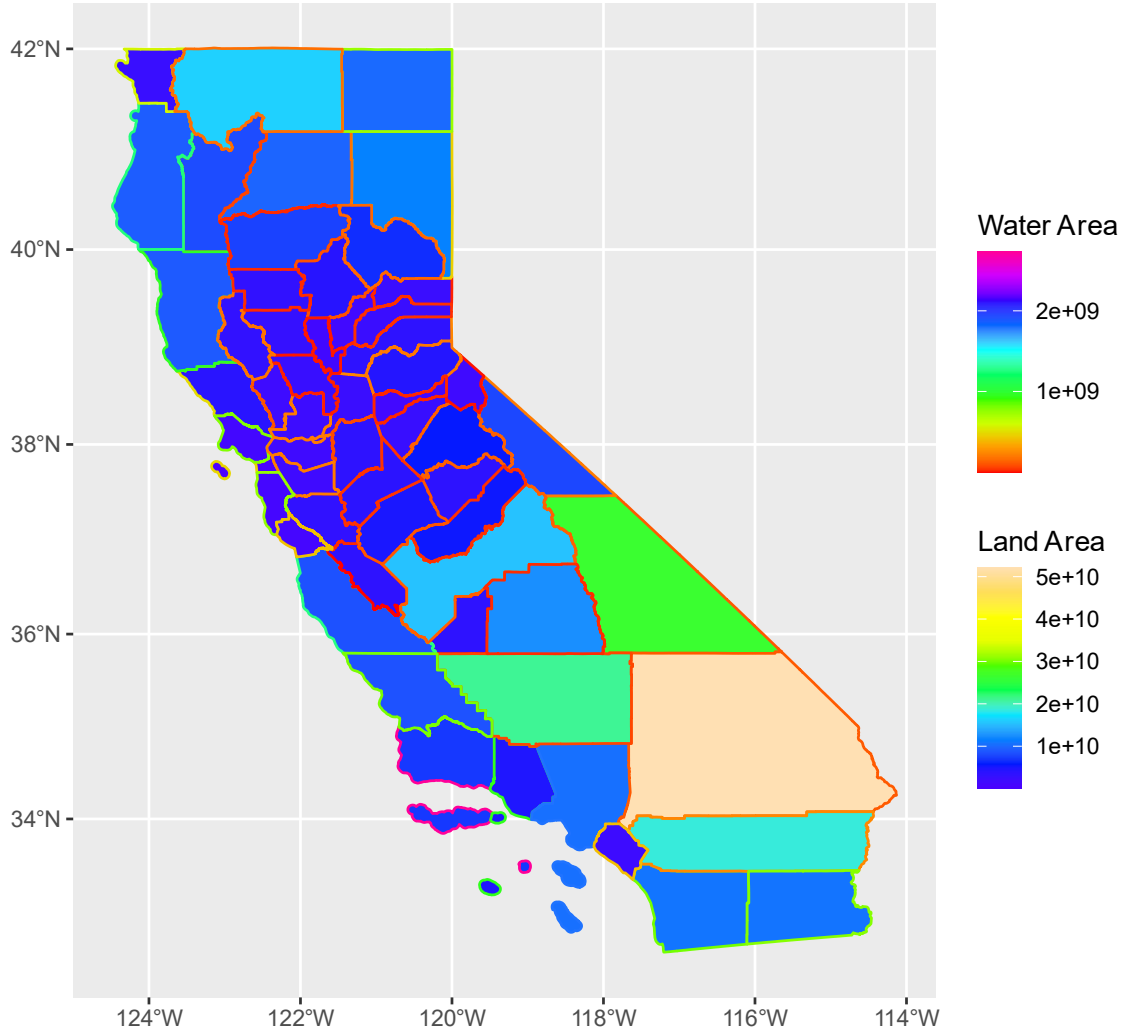




Detour: choropleth with sf and ggplot

The color aesthetic refers to the border of polygons

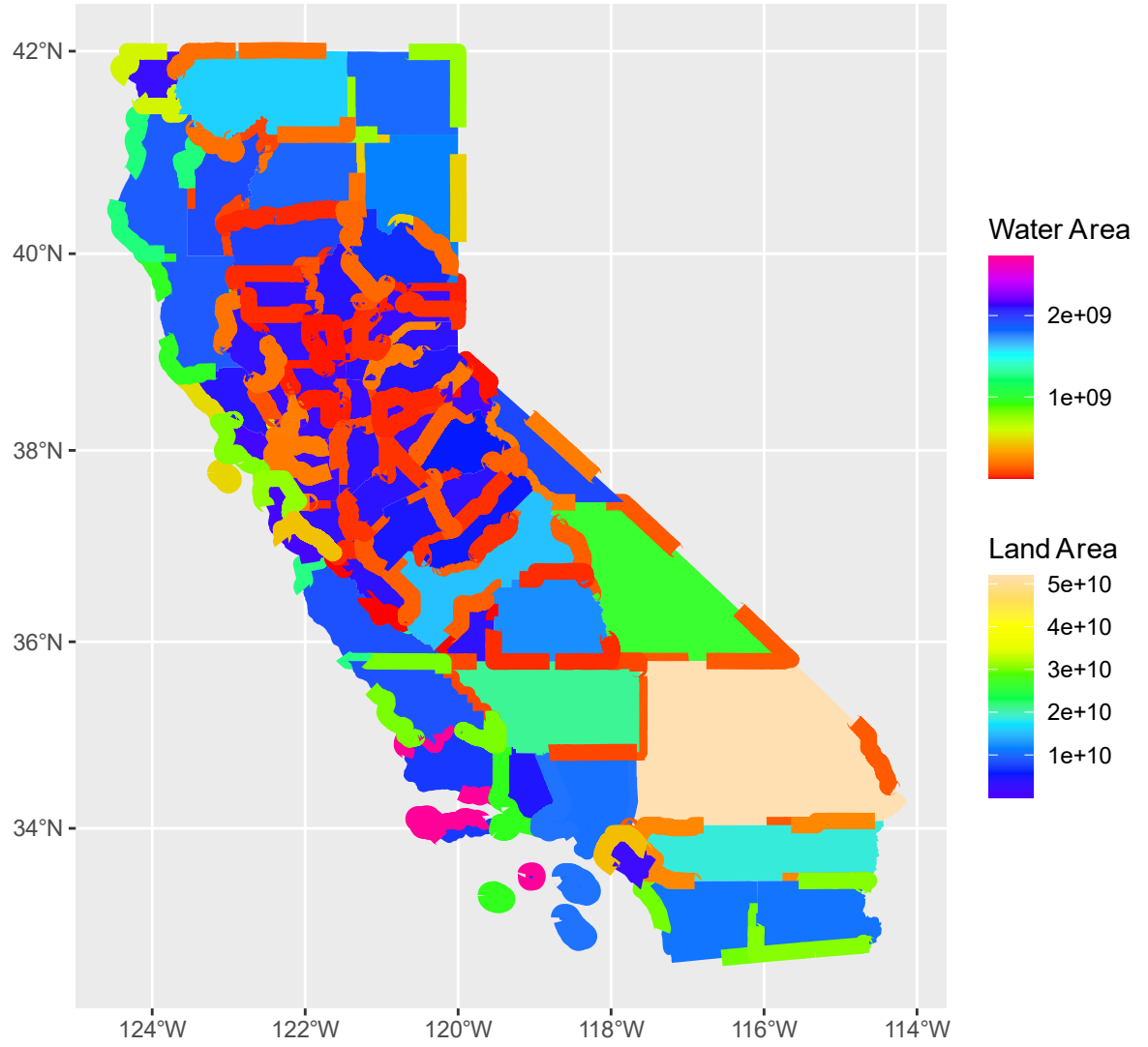
```
ggplot() +
  geom_sf(
    data = ca_sf,
    mapping = aes(
      fill = as.numeric(ALAND),
      colour = as.numeric(AWATER)))
  scale_fill_gradientn(
    colors = topo.colors(10),
    name = "Land Area") +
  scale_colour_gradientn(
    colors = rainbow(10),
    name = "Water Area")
```



Other Map Options

- I can use these arguments to adjust the boundary line type and width attributes:
- lty
- lwd

```
ggplot() +
  geom_sf(
    data = ca_sf,
    mapping = aes(
      fill = as.numeric(ALAND),
      colour = as.numeric(AWATER)),
    lwd = 3, lty = 2) +
```



Back to Kriging

Next we need to create a gstat object:

```
oz_gs = gstat(
   formula = ozone ~ 1,
   locations = ca_ozone)
```

gstat() expects:

- A formula.
  - We use ozone~1 since we're interpolating ozone without covariates.
  - You could specify a simple, or more complex, model with covariates.
- A set of locations at which the variables specified in the formula were observed.
  - This is usually a SpatialPointsDataFrame

Variograms!

Fitting a variogram with gstat requires a few steps:

1. Create the ‘empirical variogram’
2. Propose a model type and specify initial parameter values to optimize
3. Use the empirical and proposed model to build a fitted variogram model

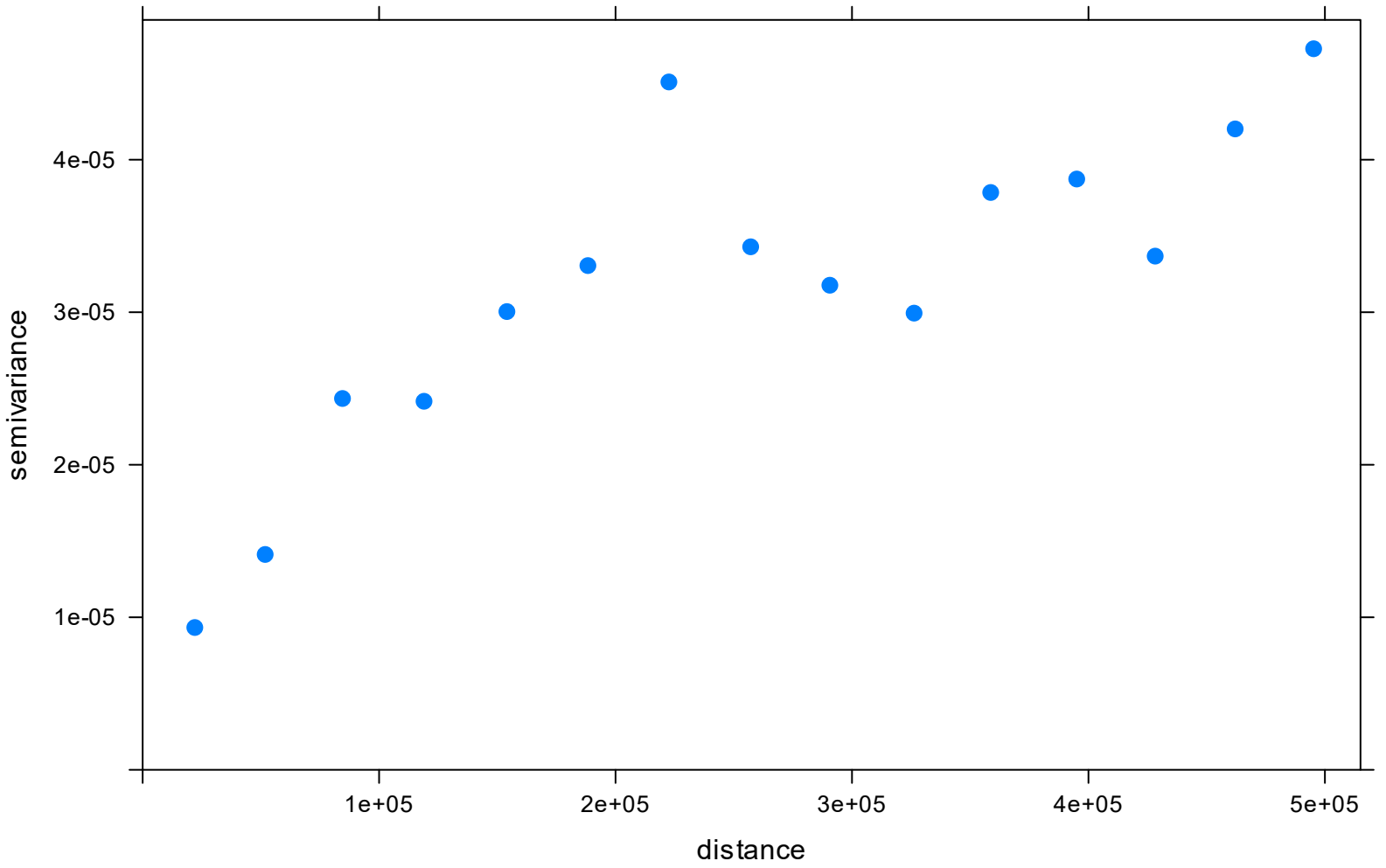
Note: There are other packages that do kriging and variograms in R, your book describes how to do it with geoR.

Create the empirical variogram

Creating and plotting the empirical variogram from the gstat object is easy!

```
vgm_emp = variogram(oz_gs)
plot(vgm_emp)
```

This one actually looks pretty good!

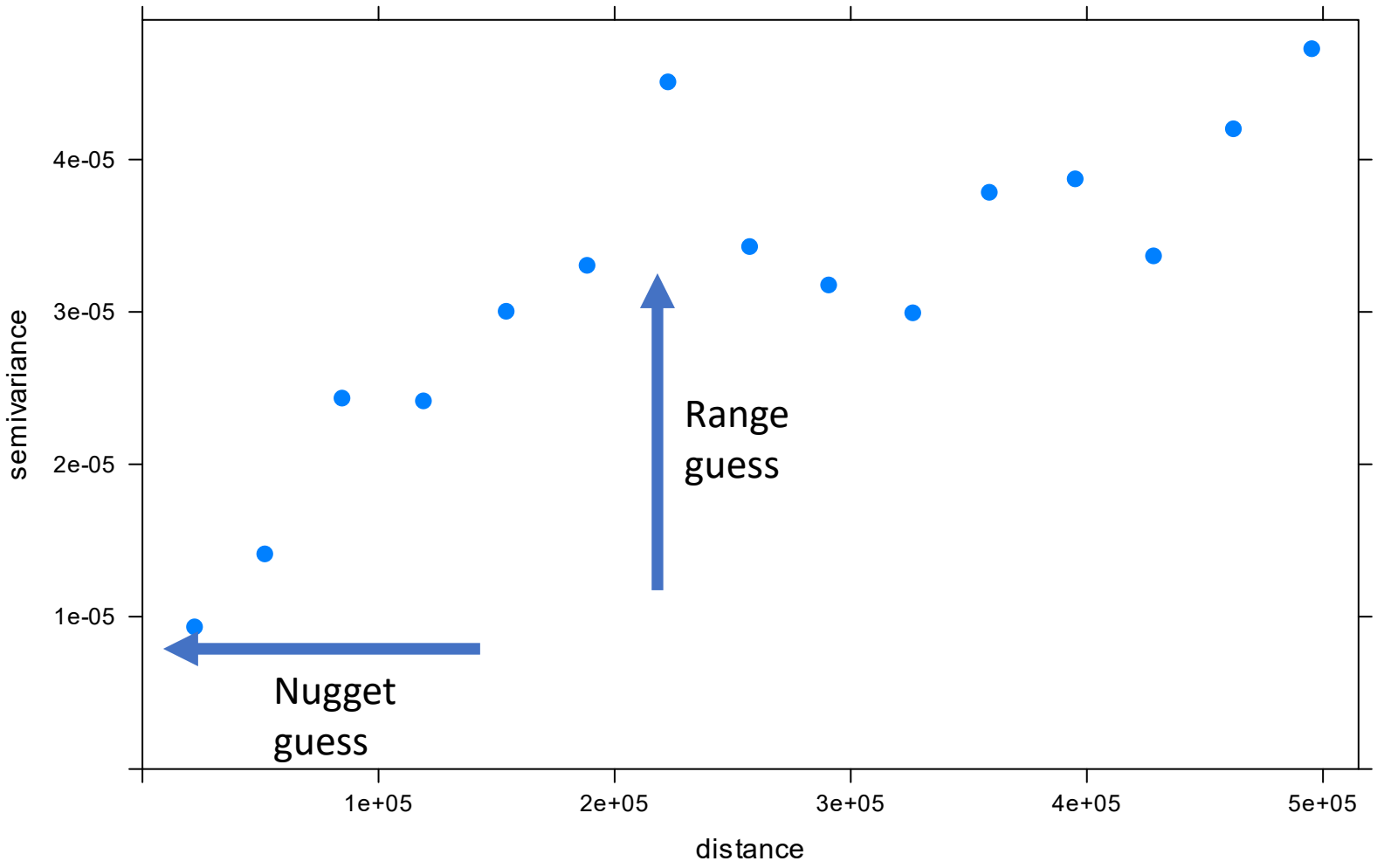


Propose a variogram model

Next step is to propose a type of model to optimize.

You can visually estimate some starting parameters:

- It looks like the nugget is around 1e-5
- My guess for the range is around 2e5 meters



Propose a variogram model

Next step is to propose a type of model to optimize.

You can visually estimate some starting parameters:

- It looks like the nugget is around 1e-5
- My guess for the range is around 2e5 meters

Use the vgm() function to create your proposed model. I'll create exponential and spherical model templates:

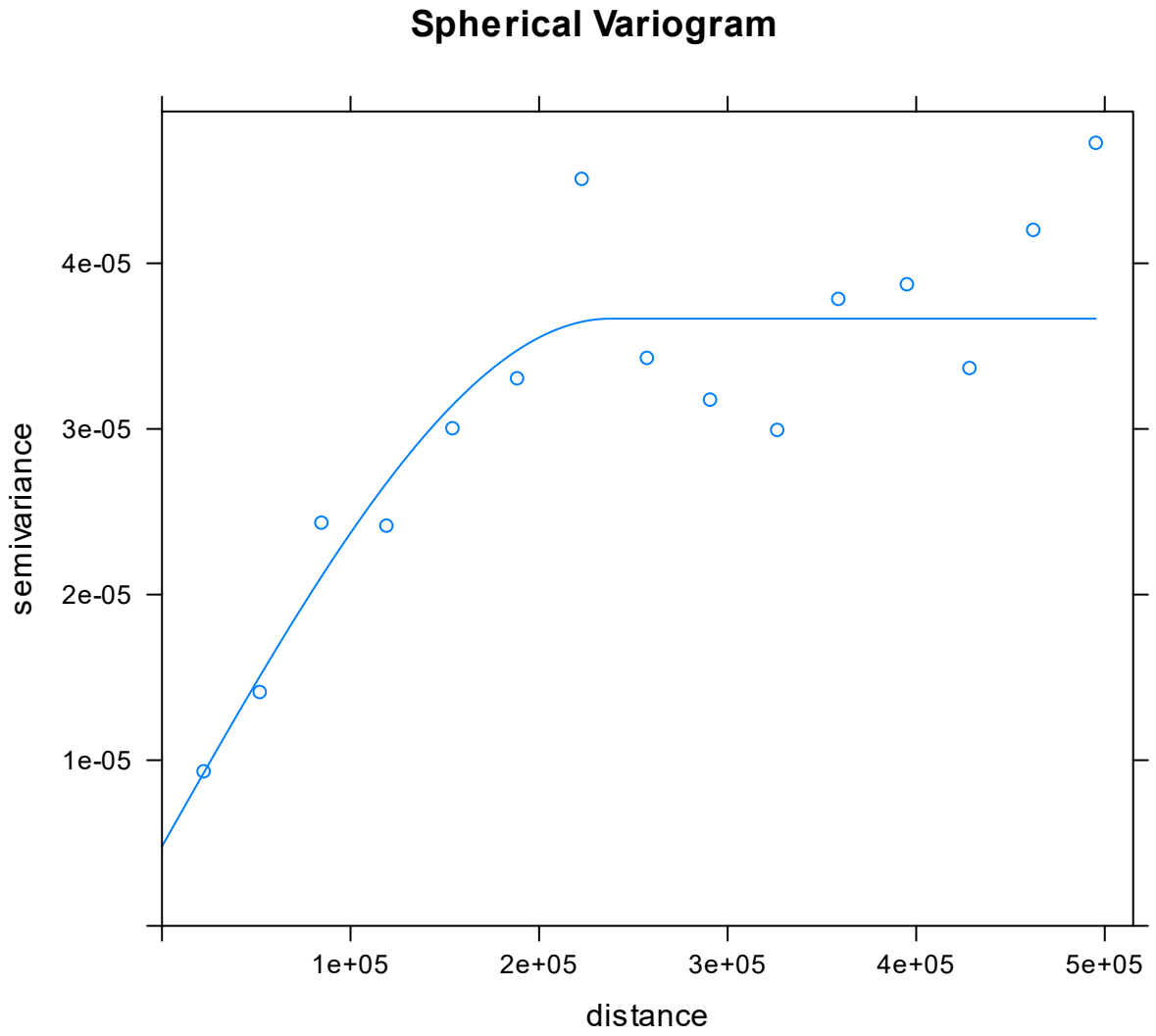
```
vgm_mod_exp = vgm(
  model = "Exp",
  nugget = 1e-5,
  range = 2e5)
vgm_mod_sph = vgm(
  model = "Sph",
  nugget = 1e-5,
  range = 2e5)
```

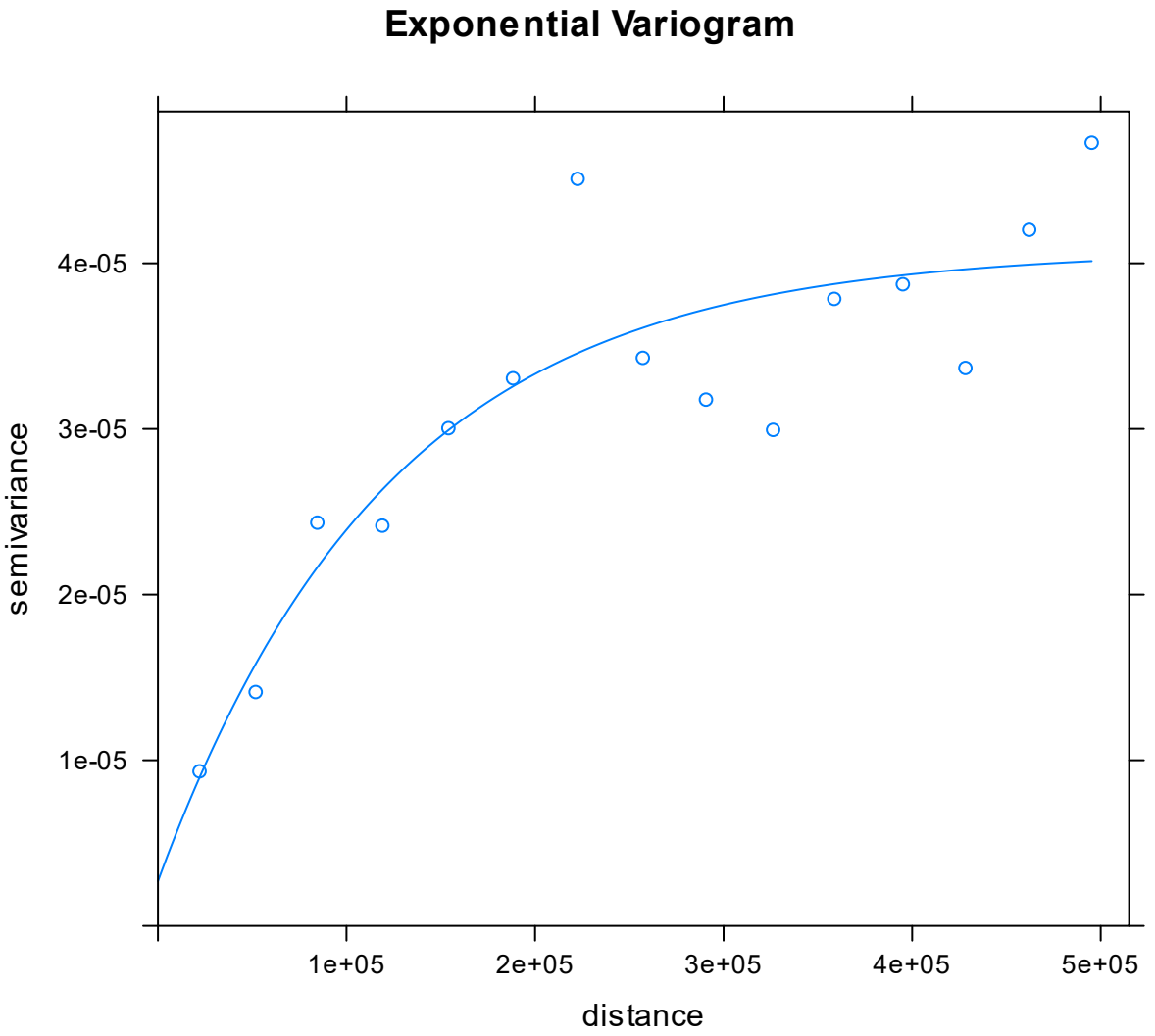
Fit your variogram model to the data

- The fit.variogram() function will use your empirical variogram and proposed variogram model to optimize the parameters.
- Plotting is straightforward, you just need to pass your empirical and fitted variograms to plot()

```
vgm_fit_sph = fit.variogram(
  vgm_emp, vgm_mod_sph)

vgm_fit_exp = fit.variogram(
  vgm_emp, vgm_mod_exp)

plot(vgm_emp, vgm_fit_sph)
plot(vgm_emp, vgm_fit_exp)
```

Which one is the better fit?





Variograms

- You can examine the fitted variogram objects to see the optimized parameter values.
- Which of the two variograms seems like a better fit?
- There are formal methods for testing goodness of fit, you can read more in F + F.
- Now that we've got our variograms, let's build our template and Krige!

Template for interpolation

To perform an interpolation, we need a set of points for which to interpolate values.

- The procedure is somewhat awkward with gstat, but a process that works is:

1. Create a template raster, set the CRS
    1. This is simple because you can just use the raster function and supply either the desired number of rows/columns or the resolution.
    2. You should manually set the CRS of the template raster before proceeding
2. Make a spatial points object from the template raster
3. Optional: crop the spatial points to a bounding polygon

Template for interpolation

1. Create a template raster, set the CRS
2. Make a spatial points object from the template raster
3. Optional: crop the spatial points to a bounding polygon

```
temp_rast = raster(
  ca_cnty, nrow = 200, ncol = 180)
crs(temp_rast) = crs(ca_cnty)
temp_grid_sp = as(temp_rast, "SpatialPoints")
temp_grid_sp = crop(temp_grid_sp, ca_cnty)
```

This step helps avoid mismatched CRS errors with kriging in the gstat package. Comments in the CRS are not transferred to the template raster, but manually setting them equal will avoid difficult to diagnose issues later on.

Ready, Set, Krige!

- The syntax for the krige() function is straightforward.
- I'll build an interpolation for each of our fitted variogram models

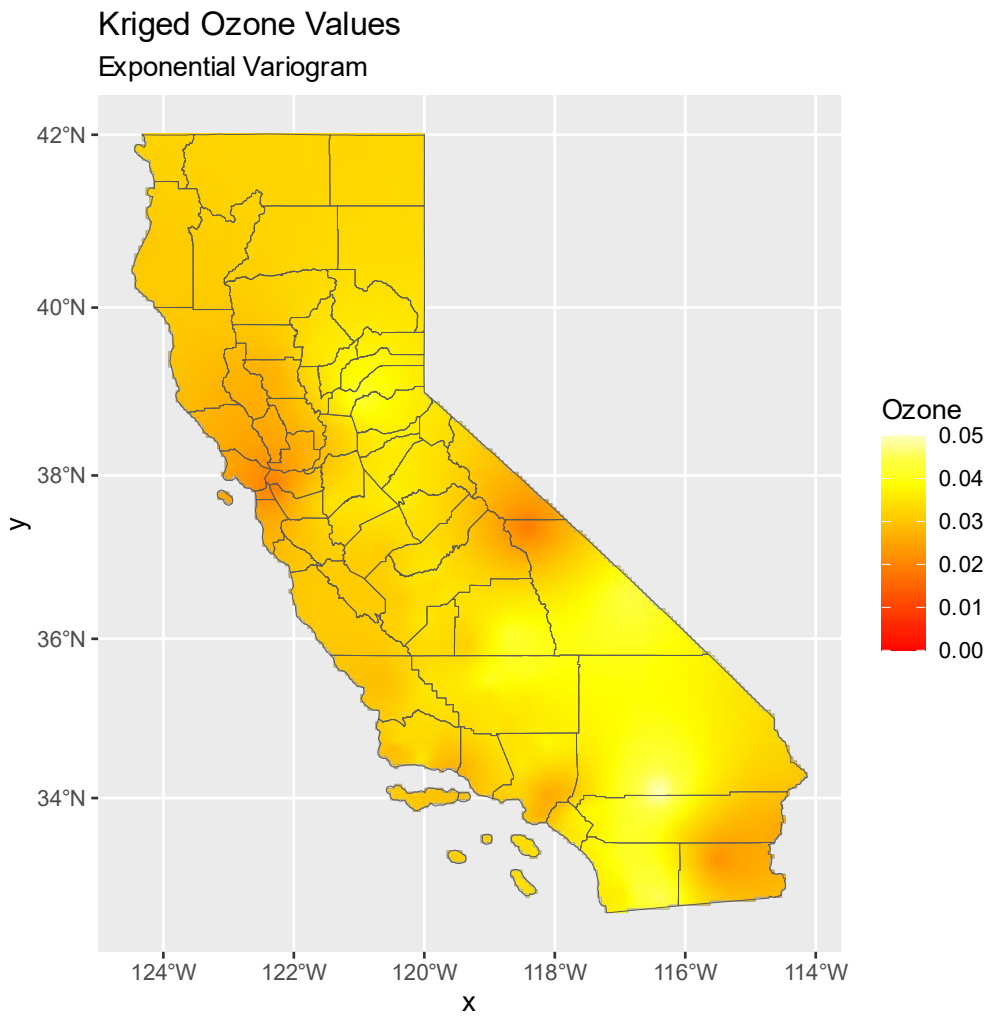
```
oz_krig_exp =  gstat::krige(
  ozone ~ 1,
  locations = ca_ozone,
  newdata = temp_grid_sp,
  model = vgm_fit_exp)

oz_krig_sph =  gstat::krige(
  ozone ~ 1,
  locations = ca_ozone,
  newdata = temp_grid_sp,
  model = vgm_fit_sph)
```

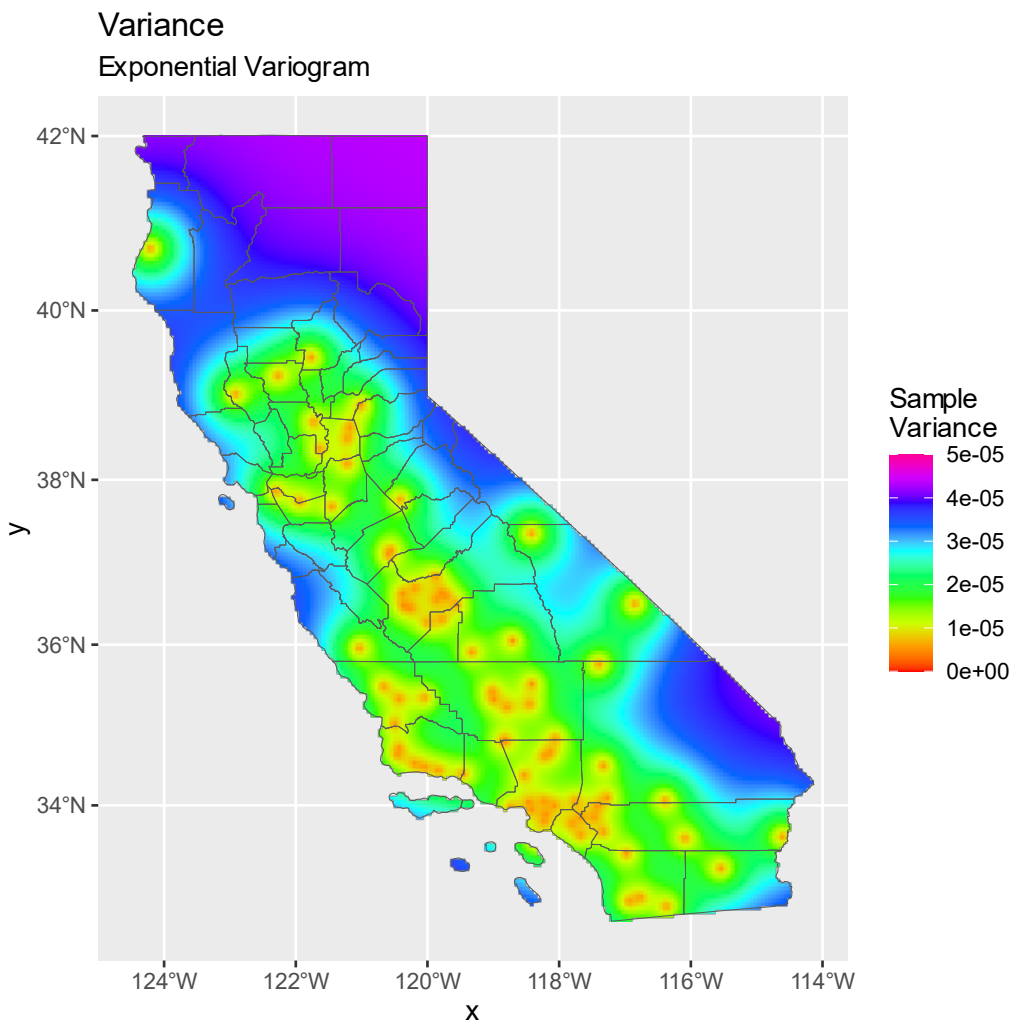
Plot your interpolations: predicted values and var

```
ggplot() +
  geom_raster(
    data = as.data.frame(oz_krig_exp),
    mapping = aes(
      x = x, y = y, fill = var1.pred)) +
  ggtitle(
    "Kriged Ozone Values",
    "Exponential Variogram") +
  scale_fill_gradientn(
    colours = heat.colors(10),
    name = "Ozone",
    limits = c(0, 0.05))
```

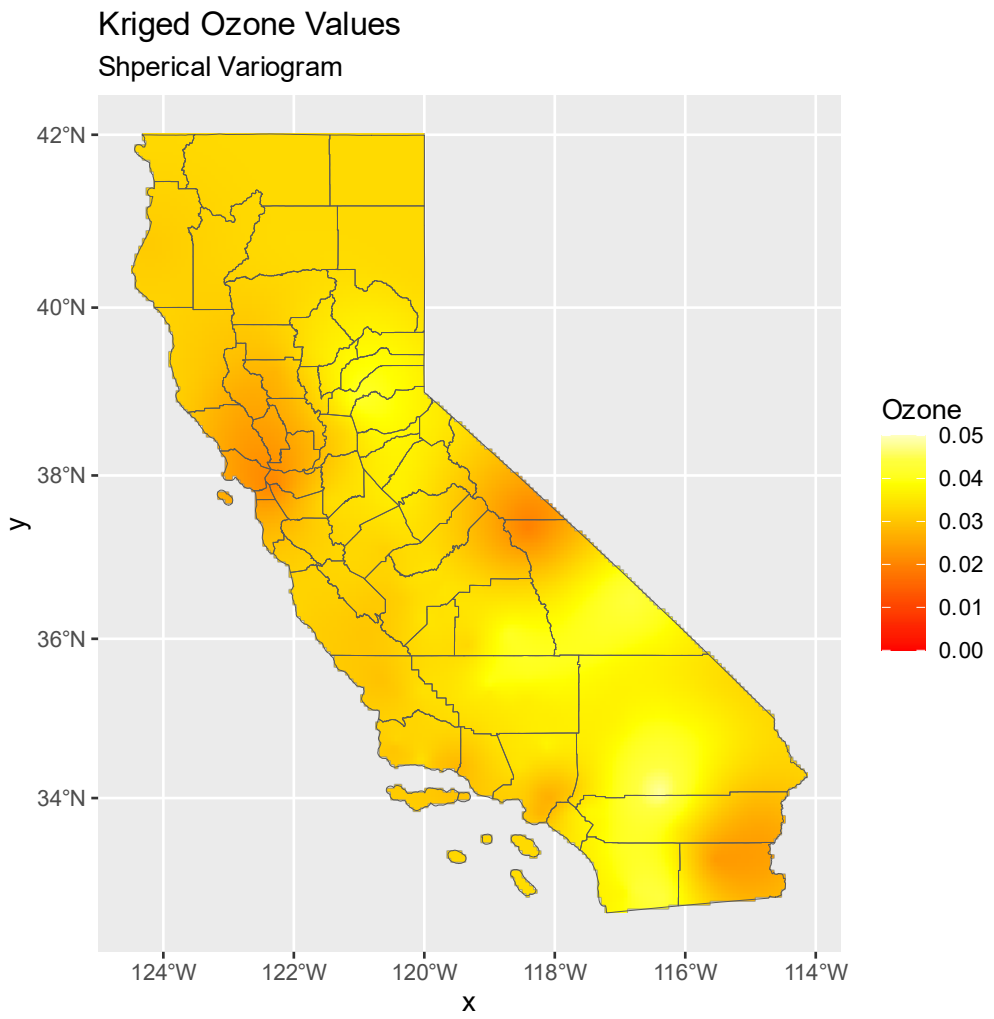
```
ggplot() +
  geom_raster(
    data = as.data.frame(oz_krig_exp),
    mapping = aes(
      x = x, y = y, fill = var1.var)) +
  ggtitle(
    "Variance",
    "Exponential Variogram") +
  scale_fill_gradientn(
    colours = rainbow(10),
    name = "Sample\nVariance",
    limits = c(0, 5e-5))
```

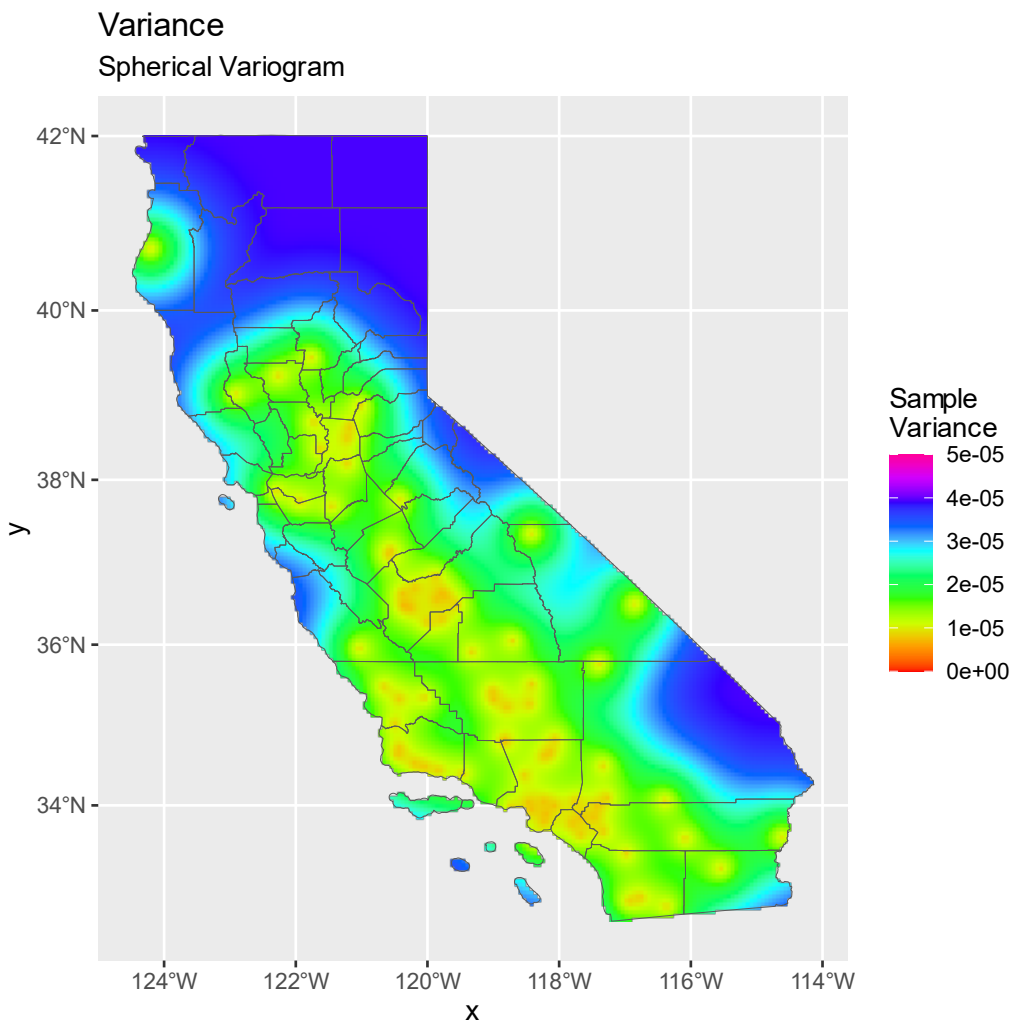
Exponential Model





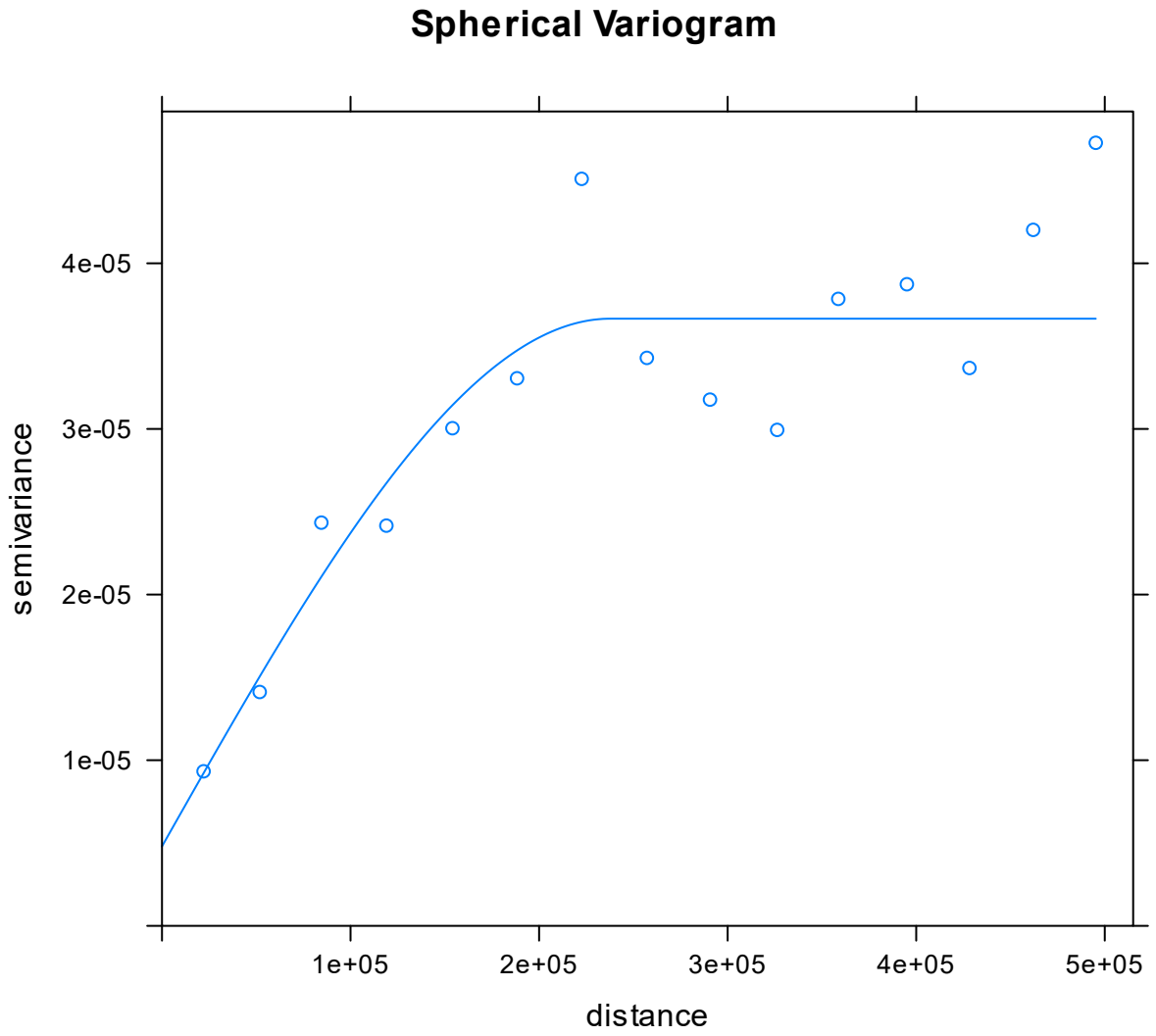
Spherical Model

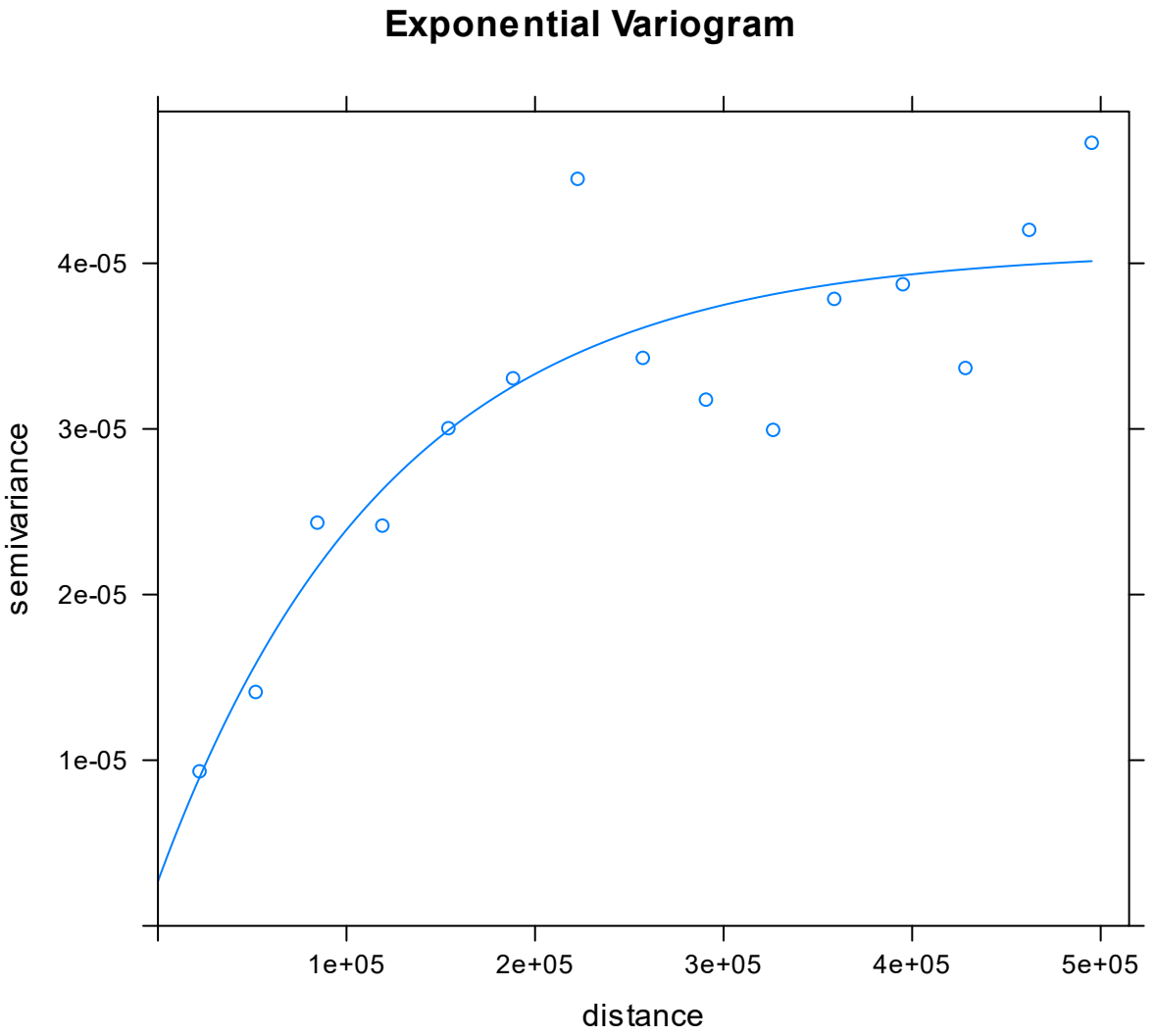




What differences did you notice?

Which one is the better fit?





Which one is the better fit?

Notice how the uncertainty (sample variance) levels off faster for the spherical model.

In practice, you should use a formal procedure to compare several variogram models to find the best fitting model.

Note that you can pass multiple proposed variogram models to fit.variogram() and it will return the 'best' fitting model. However, this process is not as explicit as the one using likfit() from package geoR in your F+F book.

- The syntax:

```
fit.variogram(
   vgm_emp,
   vgm(
      c("Mat", "Exp", "Mat"),
      nugget = 1e-5,
      range = 2e5))
```

- Returns the spherical model as the best-fitting. Do you agree based on the plots?

Spatial Dependence

Preview of Chapter 6

Preview of chapter 6: dealing with spatial dependence

- What is the primary statistical problem posed by spatial dependence?
- How could we deal with it? Some possibilities:

Preview of chapter 6: dealing with spatial dependence

- What is the primary statistical problem posed by spatial dependence?
  - Non-independent observations
- How could we deal with it? Some possibilities:
  - Ignore
  - Interpret dependence as topic of interest
  - Spatially-aware regression

Recall Stationarity

Recall the concept of stationary, a.k.a. homogeneity:

- The position of each point does not depend on the positions of other points
  - points, or values at points, are independent
- Points (values) are equally likely to occur anywhere in space
  - the intensity is constant
- Possible null models
  - CSR
  - parent/offspring patterns
  - trend surfaces

Spatial processes and data are often not stationary.

What can we do?

Avoiding Spatial Dependence: Sampling Design

Simple options?

- Ignore spatial dependence
  - observed dependence/correlation is low
- Avoid spatial dependence
  - Design sampling scheme

Sampling design for avoiding spatial dependence:

- You can use a correlograms or variograms to guess a critical distance, above which spatial dependence does not occur.
- Simply space your sampling locations greater than this critical distance.
- Any challenges or problems with this approach?

Methods for dealing with spatial dependence

- We often want to do inference and/or prediction:
  - spatially-aware regression
- Some considerations for spatial dependence in regression-like models include:
  - Consider exogenous and endogenous factors
  - Consider dependence in responses and predictors
  - Consider dependence in residuals
  - Consider model structure

Endogenous and exogenous factors

What are some potential exogenous contributors to spatial dependence?

What are some potential endogenous spatial contributors to spatial dependence?

- Consider two general classes of techniques:
  - coordinate-based
  - distance-based

Coordinate and distance paradigms

- What are the fundamental conceptual differences?
- How does each paradigm consider space?

Coordinate-based models

We can use the explicit spatial (x, y) coordinates in models via:

- Polynomial model terms of the spatial coordinates
- Fourier or wavelet methods
- Eigenvector mapping techniques

These may be effective for large spatial scale exogenous factors.

- examples?

When might projections matter?

Fourier Analysis

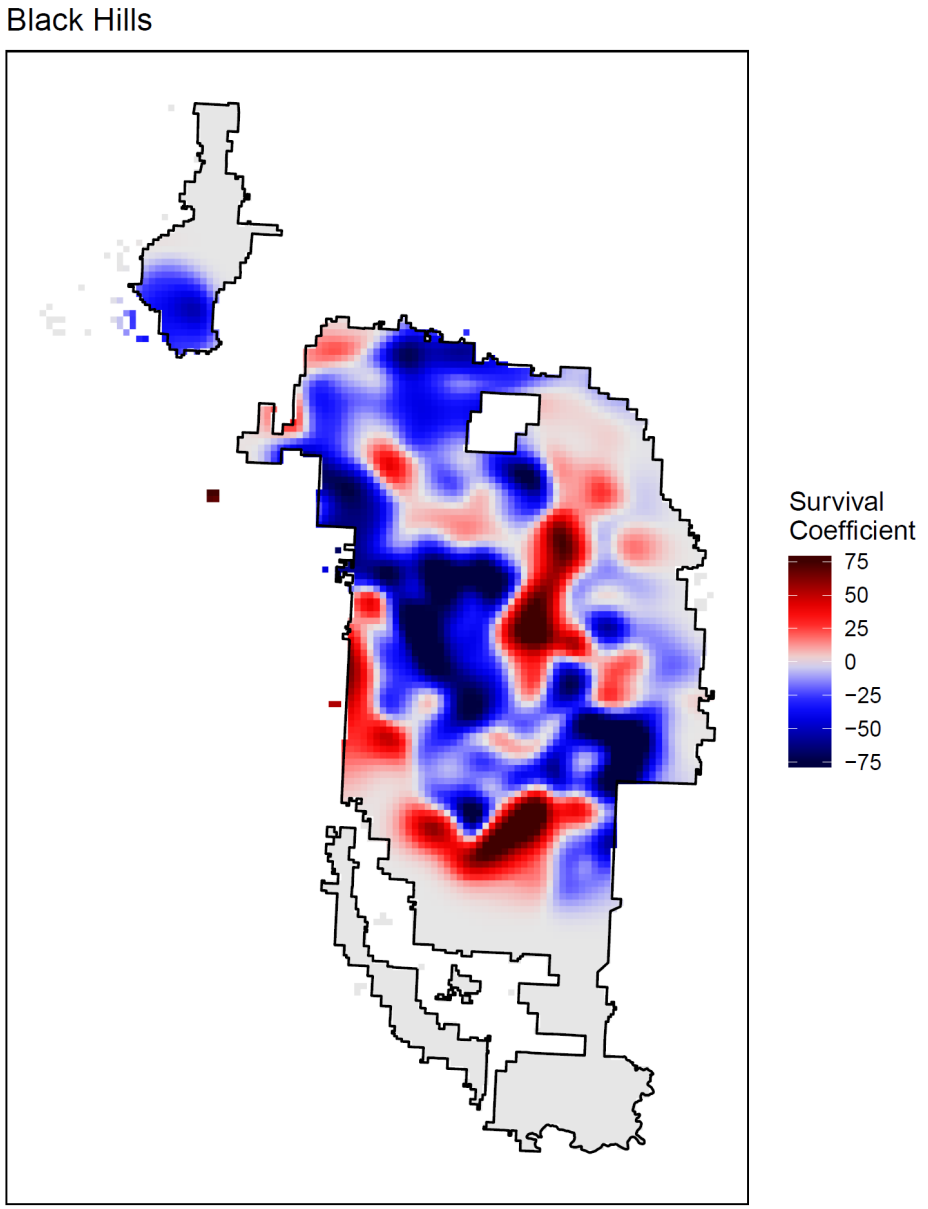
- We won't spend much time on this in class, but it's a topic you should be aware of. Here are two excellent videos that I recommend for getting a visual/intuitive understanding (without delving too far into the math)
  - Both are from the 3Blue1Brown channel (I recommend it highly)
- But what is a Fourier series? From heat flow to drawing with circles | DE4
- But what is the Fourier Transform? A visual introduction.

Additive models

General Additive Models - GAMs

Local regression

- Distance weighting
- Splines and knots
- Geographically weighted regression (GWR):
  - [tree mortality] ~ [beetle survival] + [tree cover]



Distance-based models

- We can define distance in many ways, including
  - Euclidean
  - Neighborhood
- Neighborhoods
  - first order neighbors
  - larger neighborhoods
  - distance-decay function
- Implementation via distance and weight matrices

Spatial dependence in factors

Spatial dependence can occur in the

- Predictors
- Responses
- Model residuals

How do we consider each?

Spatially-aware regression

We can often deal with spatial dependence in regression-like models by considering:

- Random and fixed effects
- Autocovariates
- Autoregression
- Dependence in predictors and errors
- Error correlation structures

- We'll focus on spatially-aware regression in the next slide decks and chapter 6 of F + F.

Some Resources

http://gsp.humboldt.edu/OLM/R/04_01_Variograms.html

https://casoilresource.lawr.ucdavis.edu/

https://mgimond.github.io/Spatial/interpolation-in-r.html

https://rspatial.org/raster/analysis/4-interpolation.html

https://webcam.srs.fs.fed.us/impacts/ozone/spatial/kriging.shtml

Bivand, R.S., Pebesma, E.J., Gomez-Rubio, V., and Pebesma, E.J. (2008). Applied spatial data analysis with R (Springer).