GLS for spatial data

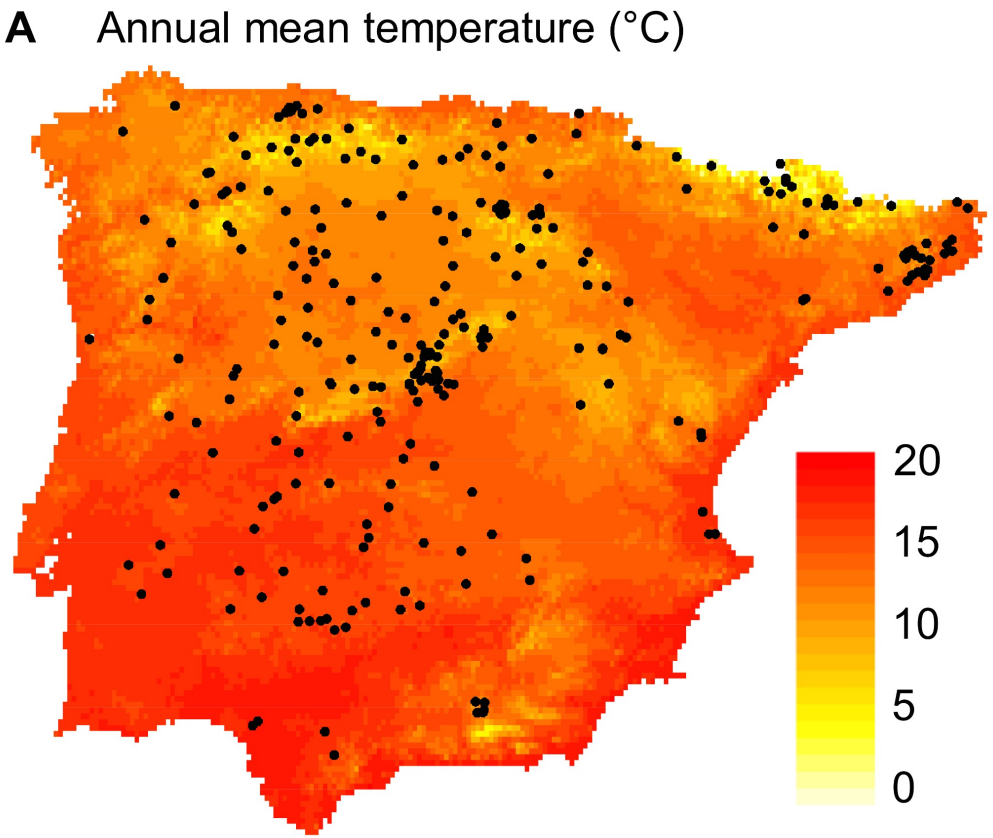
Same idea

We have some spatially extensive data, and we want to use it for a regression-type analysis

To account for potential spatial autocorrelation, we can model the residuals

Now we don’t have regular intervals (space is continuous).

- Often true for time series as well



Manzano-Piedras et al. 2014; *Arabidopsis*

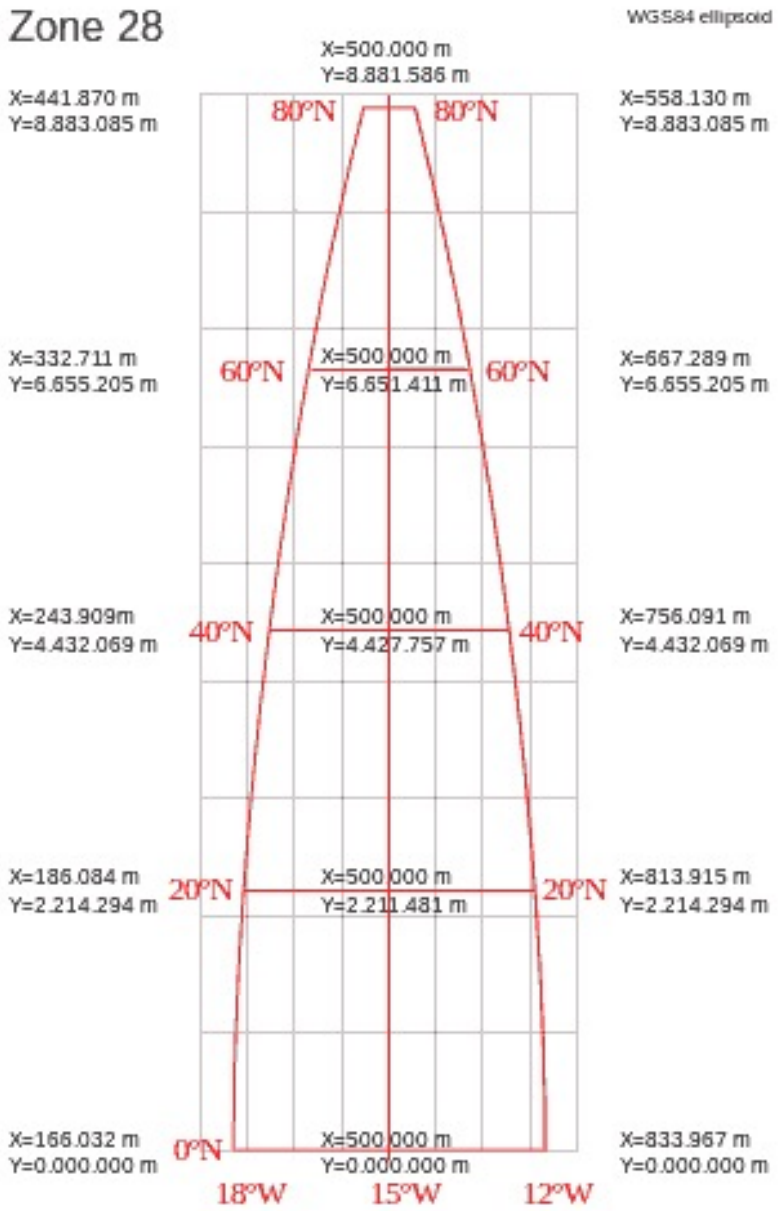
genetic variation in life history traits vs. environment

279 accessions. Seeds grown in greenhouse, then planted in common garden experiment

We'll focus on flowering time: how long from seed sown to flowers.

A key life history trait: how does this trait evolve as temperature, precipitation vary?

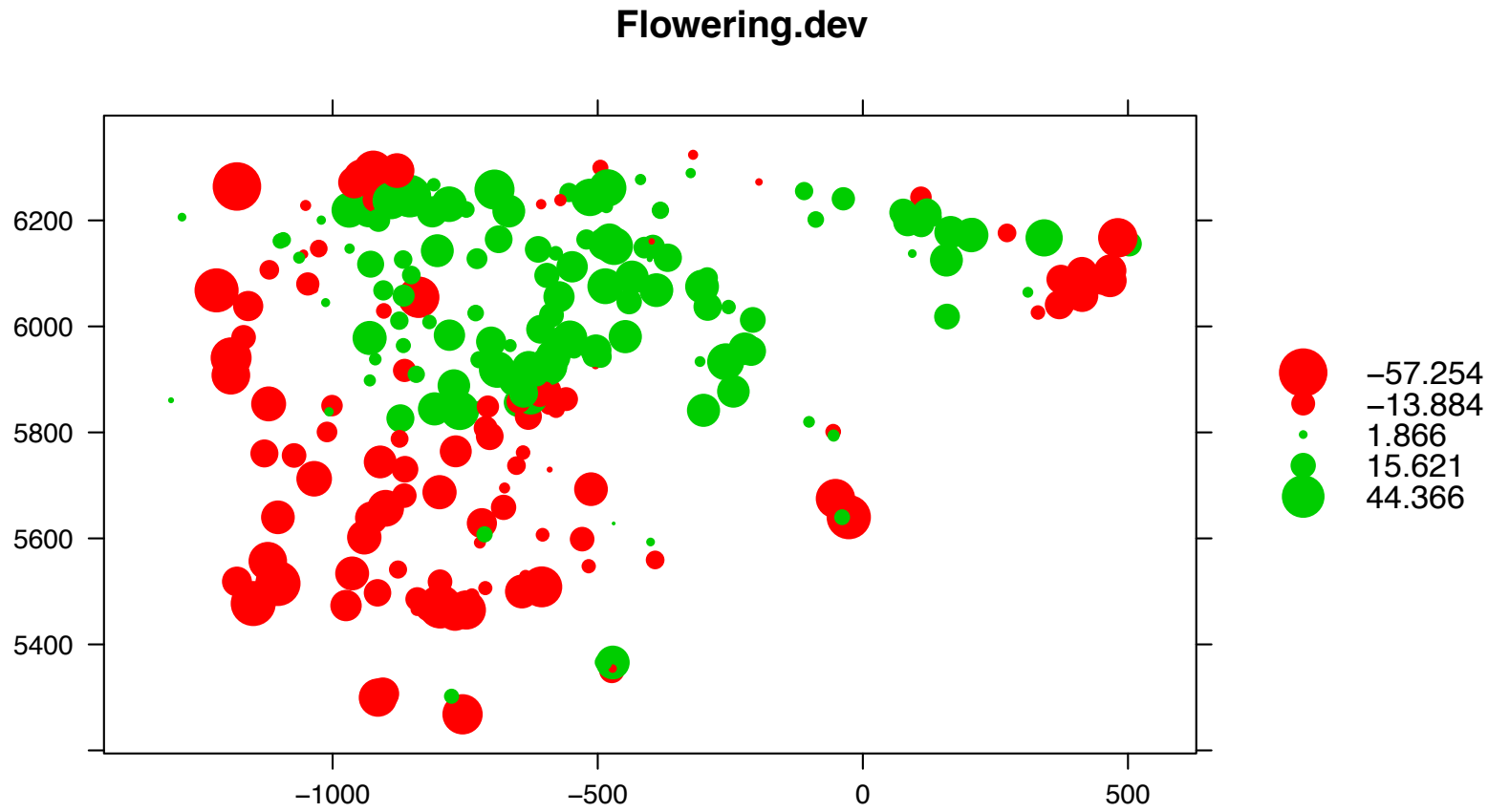
First step: convert Lat + Lon to Universal Transverse Mercator coordinates



First step: convert Lat + Lon to Universal Transverse Mercator coordinates

```
library(rgdal)
xy = cbind(arab$Latitude, arab$Longitude)
utms = project(xy, "+proj=utm +zone=30 ellps=WGS84")
arab$northing = utms[,1]/1000
arab$easting = utms[,2]/1000
```

```r
library(sp)
arab.sp = arab
coordinates(arab.sp) = c('easting', 'northing')
arab.sp$Flowering.dev = arab.sp$Flowering.time - mean(arab.sp$Flowering.time)
bubble(arab.sp, zcol = 'Flowering.dev', scales = list(draw = T))
```




sciLands
scientific landscapes
www.scilands.de
0
100
200km

How to visualize spatial autocorrelation: variogram

$$\hat{\gamma}(h) = \frac{1}{2N(h)} \sum_{(i,j) \in N(h)} (z_i - z_j)^2$$

What is the variance in flowering time among sites that are *h* kilometers apart?

If nearby sites have similar flowering time, then the variance will be smaller when *h* is smaller

How to visualize spatial autocorrelation: variogram

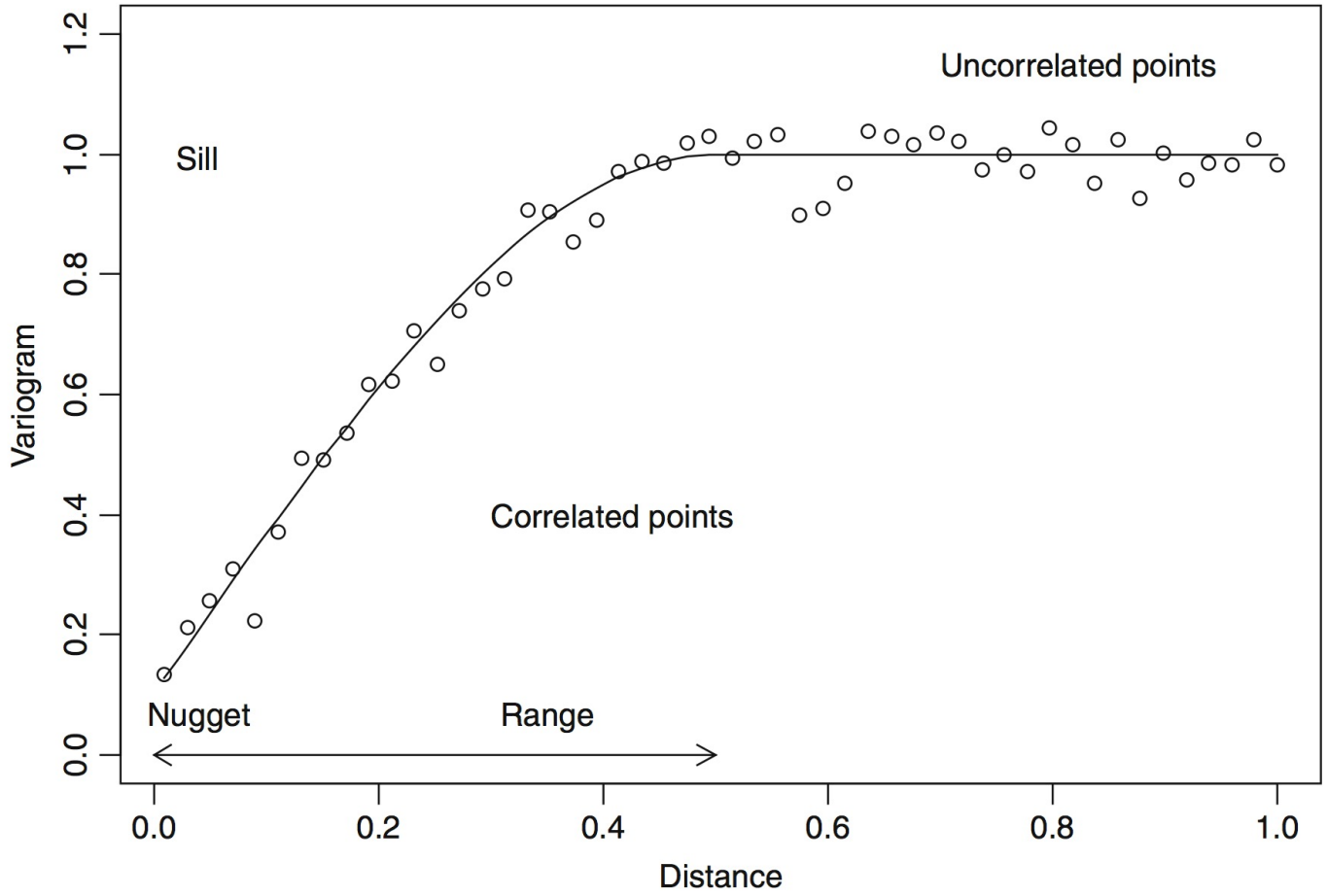
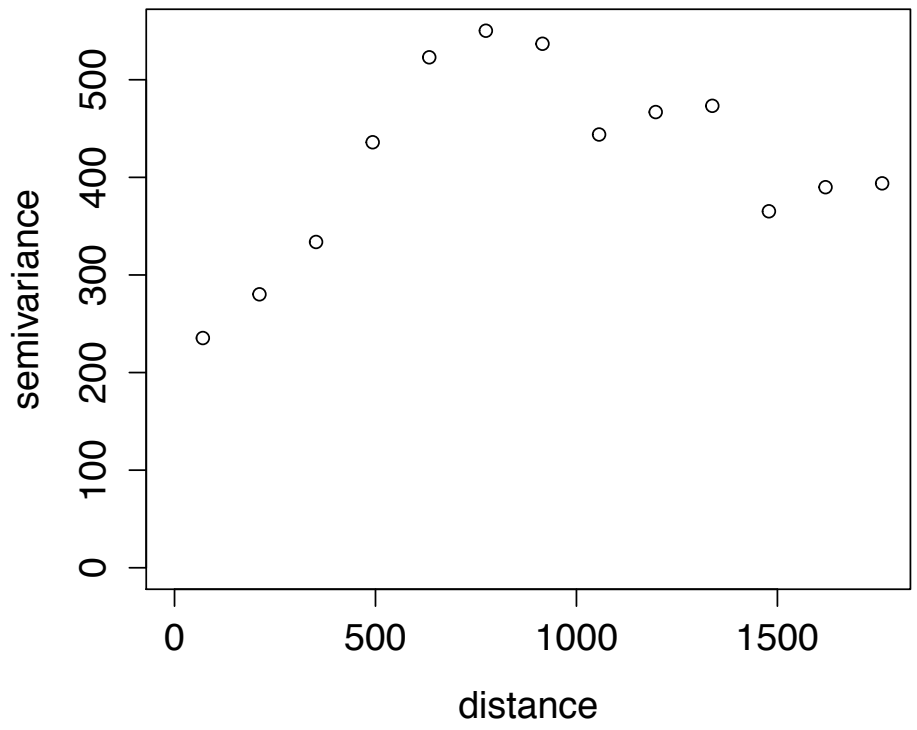


Fig. 7.2 Variogram with fitted line. The sill is the asymptotic value and the range is the distance where this value occurs. Pairs of points that have a distance larger than the range are uncorrelated. The nugget effect occurs if $\hat{\gamma}(\mathbf{h})$ is far from 0 for small $\boldsymbol{h}$

How to visualize spatial autocorrelation: variogram

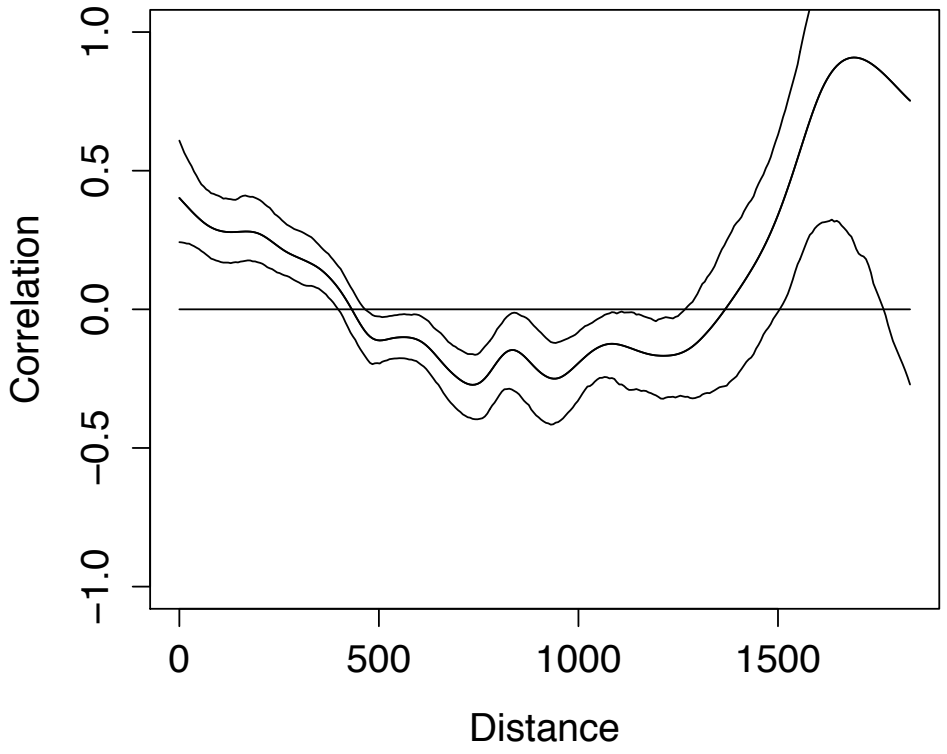
```
library(geoR)

v1 <- variog(coords = arab[,c('easting', 'northing')], data = arab$Flow
ering.time)

plot(v1)
```



How to visualize spatial autocorrelation: spline correlogram

Similar to acf(), but over continuous distance

```
ncf.scor <- spline.correlog(arab$easting, arab$northing, arab$Flowering.time,
resamp=500, quiet = TRUE)
plot(ncf.scor)
```



How to visualize spatial autocorrelation: spline correlogram

These plots are useful for exploring the data

But we shouldn't expect them to show the 'classic' geostatistical patterns

E.g. the classic variogram assumes a random stationary process

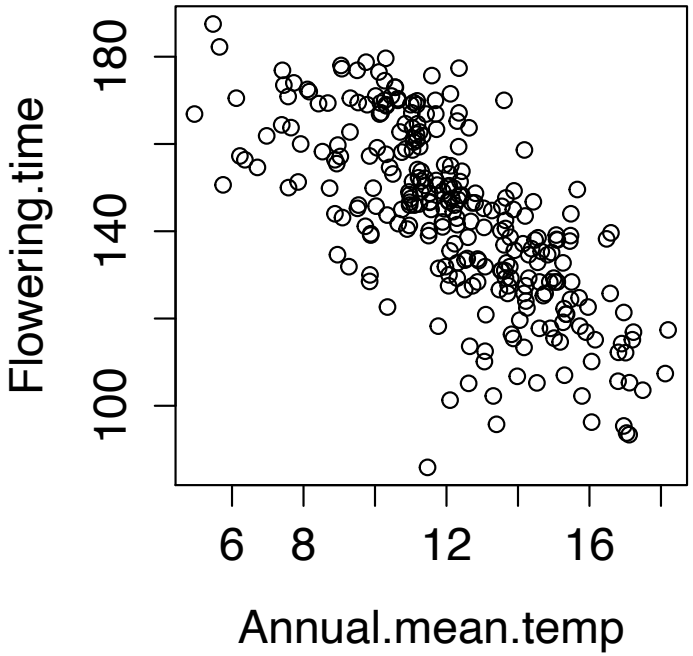
- The only thing creating patterns is autocorrelation

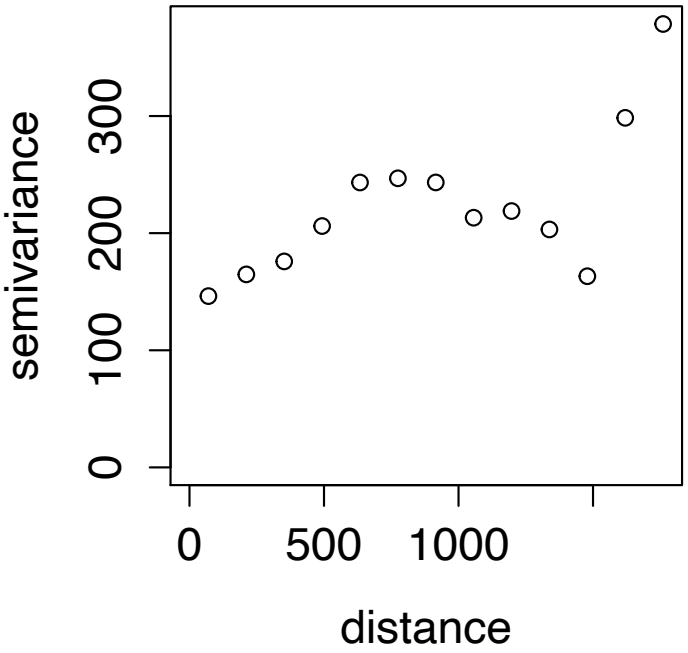
But our data has clear geographic structure, not stationary

For purposes of analysis, what really matters is the residuals

```
mod.nocorr = lm(Flowering.time ~ Annual.mean.temp, data = arab)
v1 <- variog(coords = arab[,c('easting', 'northing')], data =
resid(mod.nocorr))
```

This has a weaker correlation structure: the predictor may be sufficient





How to model spatial correlation

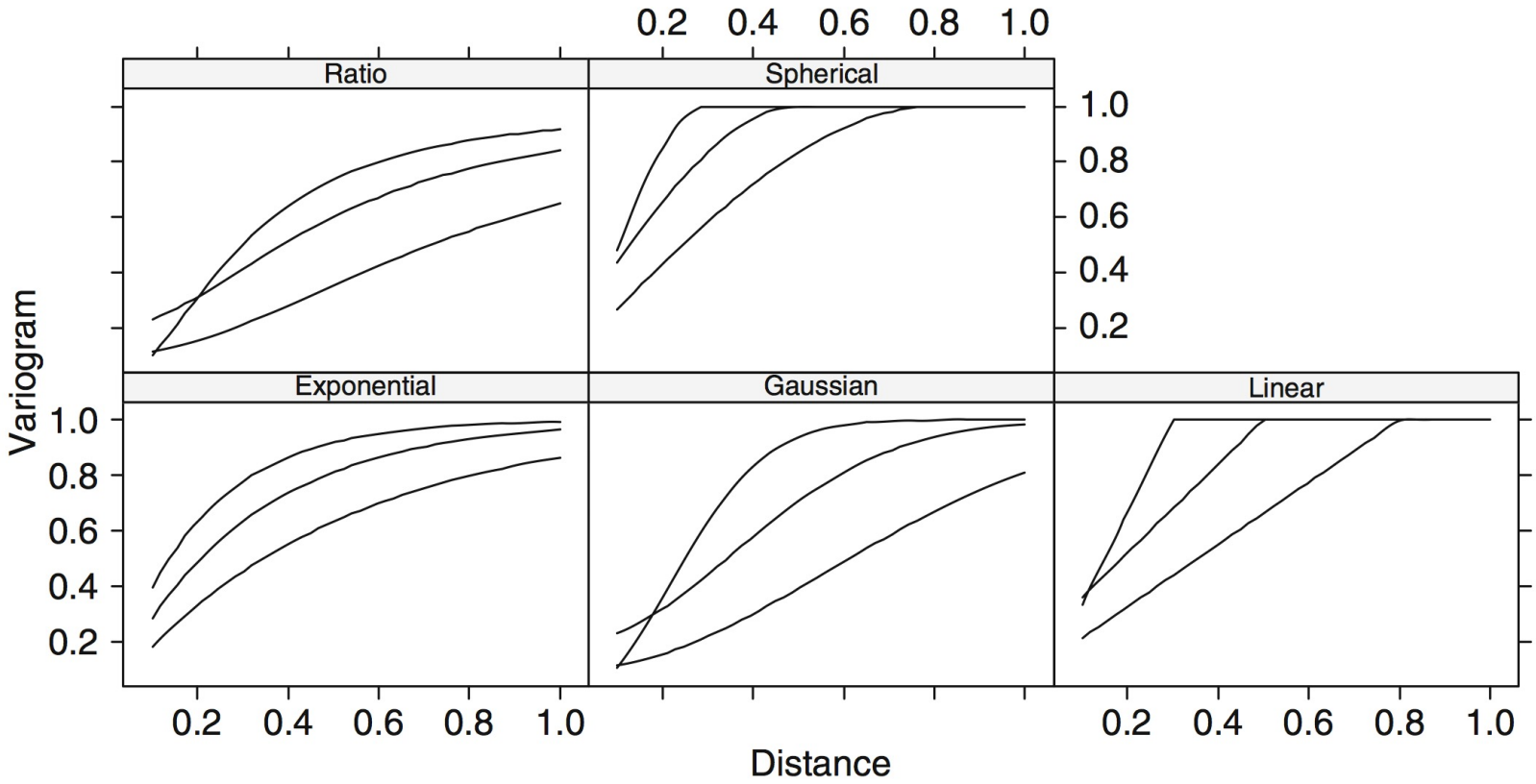


Fig. 7.4 Different variogram patterns. The three lines in the same panel were obtained using different values for the range and nugget

Exponential

$$\gamma(s,p) = 1 - e^{-\frac{s}{p}}$$

$$\gamma(s,p) = c_0 + (1 - c_0)(1 - e^{-\frac{s}{p}})$$

- Similar to AR(1); can use for irregularly spaced time series data

How to model spatial correlation

```
mod.gls.nopred = gls(Flowering.time ~1, data = arab, correlation = corExp(form
= ~easting + northing, nugget = TRUE, value = c(1000, 0.8)))
```

- Calculates euclidean distance using coordinates
- Use the nugget
- Give it starting values
- Nugget: 0 for no nugget; 1 for no autocorrelation

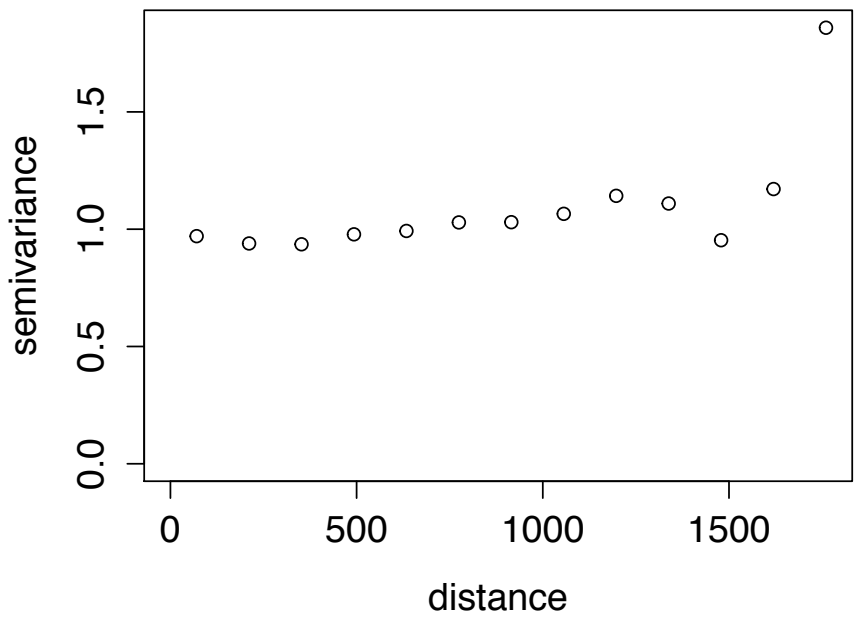
How to model spatial correlation

```
mod.gls.nopred = gls(Flowering.time ~1, data = arab, correlation = corExp(form
= ~easting + northing, nugget = TRUE, value = c(1000, 0.8)))
```

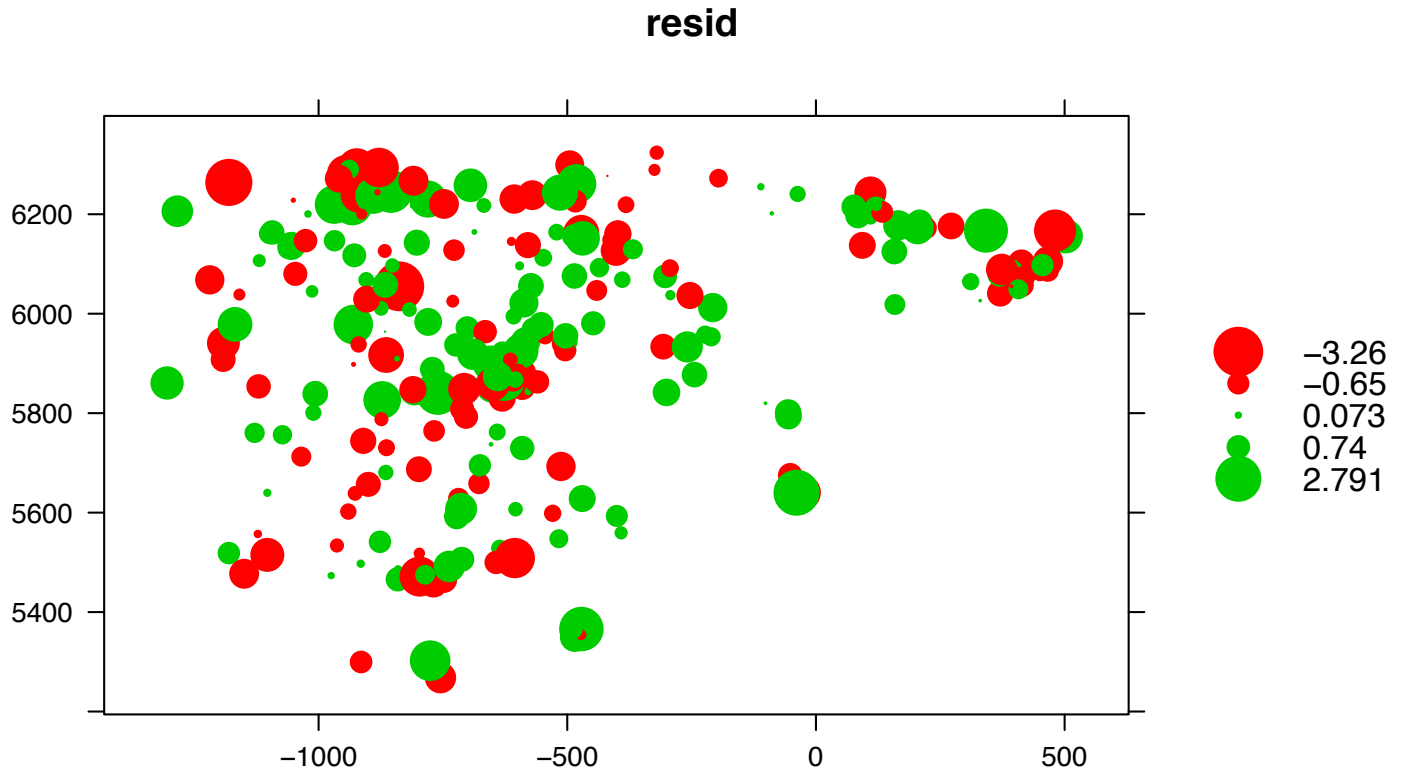
```
summary(mod.gls.nopred)
```

```
## Generalized least squares fit by REML
##   Model: Flowering.time ~ 1
##   Data: arab
##
## Correlation Structure: Exponential spatial correlation
##  Formula: ~easting + northing
##  Parameter estimate(s):
##    range   nugget
## 323.0969   0.2615
##
## Coefficients:
##             Value Std.Error t-value p-value
## (Intercept) 132.1     8.102    16.3       0
```

```
v1 <- variog(coords = arab[,c('easting', 'northing')], data = resid(mod.gls.no
pred, type = "normalized"))

plot(v1)
```



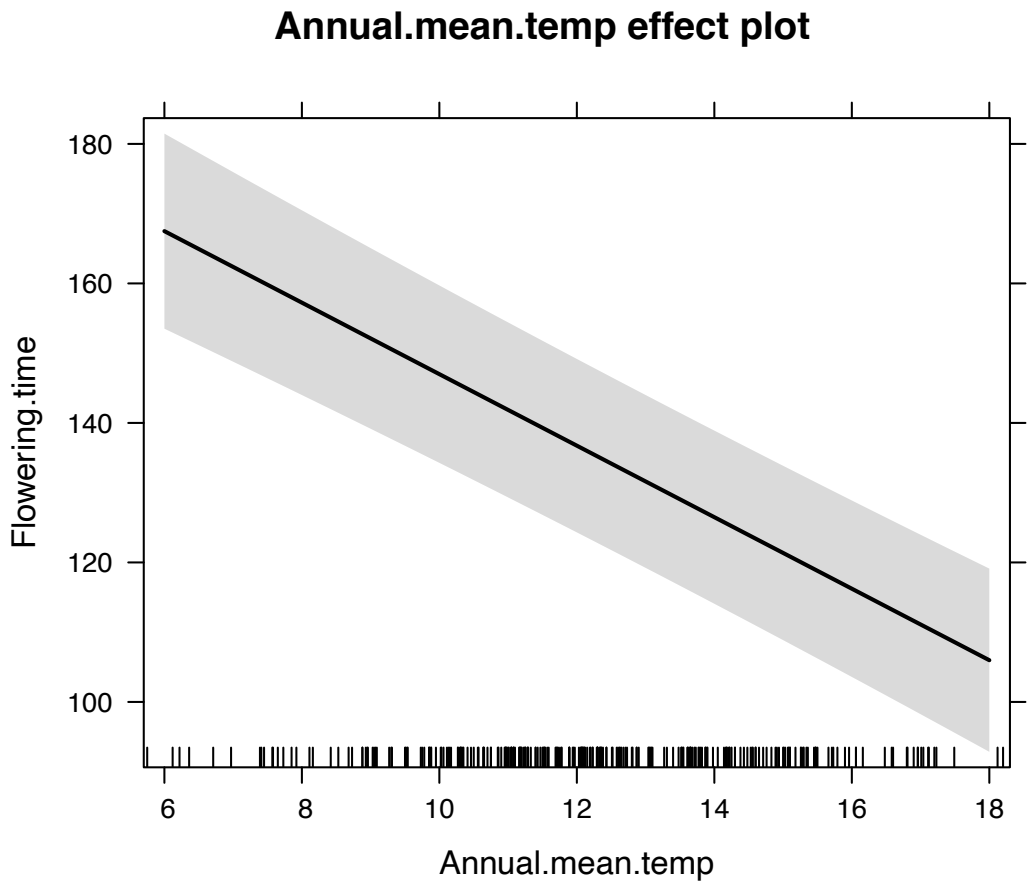
An exponential correlation function does a good job of accounting for the spatial structure of this data


resid
6200
6000
5800
5600
5400
−1000
−500
0
500
−3.26
−0.65
0.073
0.74
2.791

Combining a linear model with the correlation function

```
mod.gls1 = gls(Flowering.time ~ Annual.mean.temp, data = arab, correlation = c
orExp(form = ~easting + northing, nugget = TRUE, value = c(1000, 0.8)))
summary(mod.gls1)

## Generalized least squares fit by REML
##   Model: Flowering.time ~ Annual.mean.temp
##   Data: arab
##
## Correlation Structure: Exponential spatial correlation
##  Formula: ~easting + northing
##  Parameter estimate(s):
##    range   nugget
## 510.6430   0.4853
##
## Coefficients:
##                   Value Std.Error t-value p-value
## (Intercept)      198.25     8.733   22.70       0
## Annual.mean.temp  -5.13     0.460  -11.15       0
##
```

Combining a linear model with the correlation function



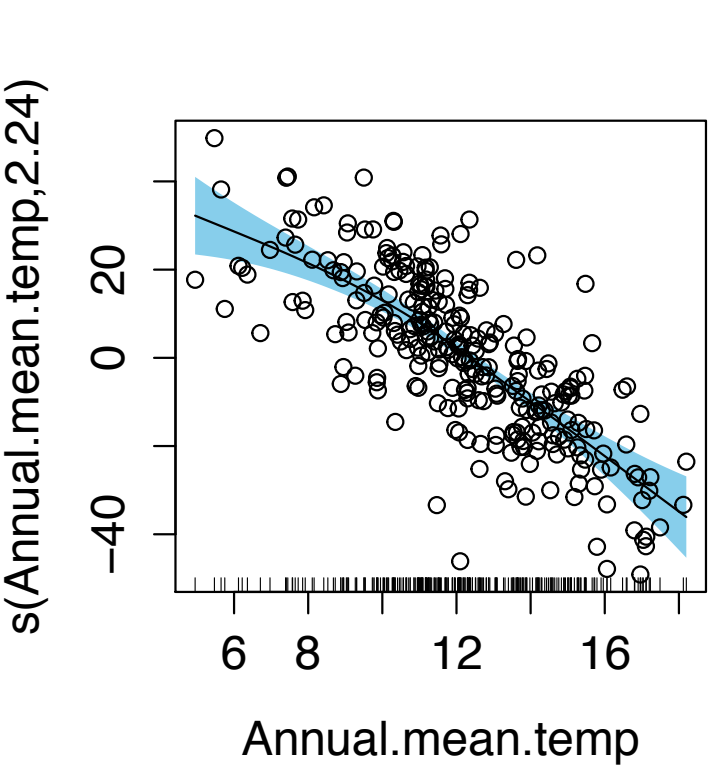
Faster development to avoid hot dry summers?

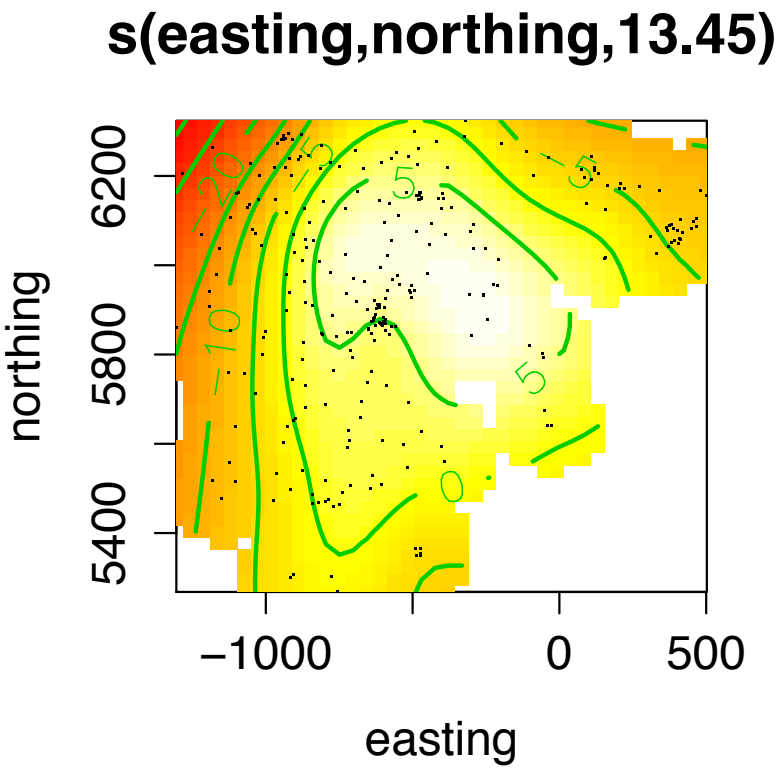
GLM vs GAM

Instead of modeling the residuals, we might account for spatial pattern with a smoother

- Will work better for bigger trends than small-scale clustering

```
mod.gam = gam(Flowering.time ~ s(easting, northing) + Annual.mean.temp, data =
 arab)
plot(mod.gam, scheme = 2, lwd = 2, labcex = 1)
```



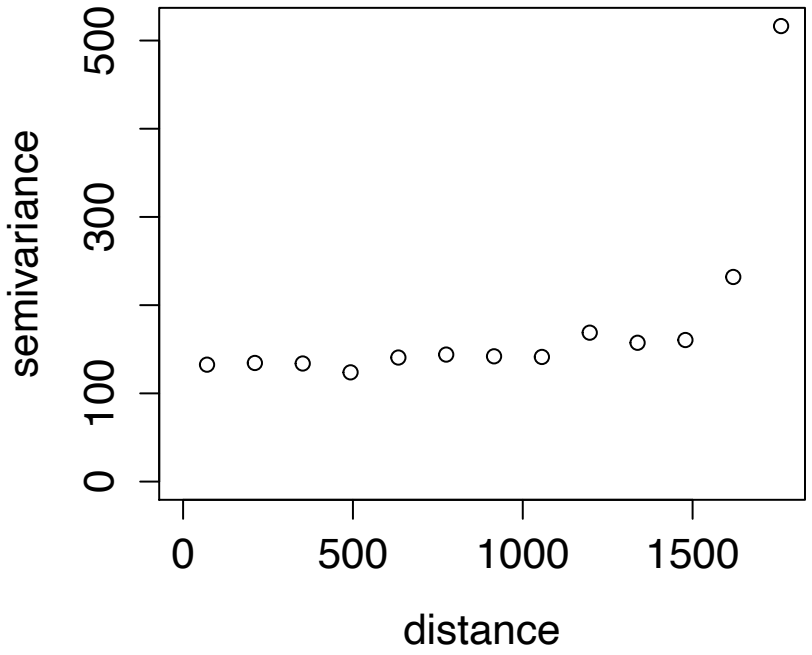


GLM vs GAM

Instead of modeling the residuals, we might account for spatial pattern with a smoother

- Will work better for bigger trends than small-scale clustering

```
v1 <- variog(coords = arab[,c('easting', 'northing')], data = resid(mod.gam))

## variog: computing omnidirectional variogram

plot(v1)
```



Ways to use spatial or time series data for regression

1) With the right predictors, autocorrelation will disappear

2) Model the residuals (GLS). For non-normal data, Generalized Estimating Equations

3) Model smooth patterns in space or time (GAM)

4) Chunk into spatial groups (sites, regions) or temporal groups (years),
   use random effects

And you may want to mix and match