

Spatiotemporal models

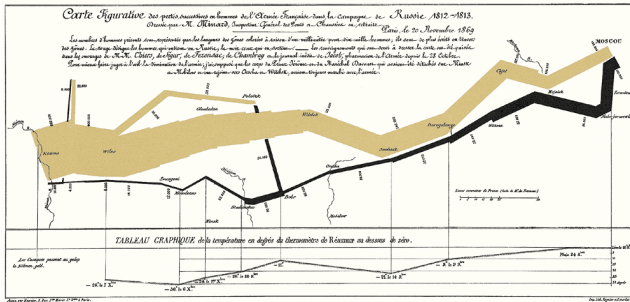
“Space is the place” - Sun Ra

Samuel Robinson, Ph.D.

Oct 27, 2023

Outline

- Spatial and temporal data
 - Some basic GIS (sf)
- How to think about space and time
 - Plotting
 - Variograms
 - “Continuous” random effects
- Some common modeling approaches
 - GLS (covariance)
 - Basis functions (GAMs)



Some common problems

- My data were sampled over time or space. I'm not really interested in time or space *per se*, so can I just ignore them and run my models?
- I am actually interested in how something changes over time or space. Can I just use day or location (lat/lon) as another term in my model?
- My supervisor told me to look for something called autocorrelation, and it sounds scary

A common approach: random effects

“Can I just use day or site as a random effect?”

- Short answer: “Yes”
- Long answer: You might be able to do better, because of the **1st Law of Geography**:

“... everything is related to everything else, but near things are more related than distant things.” Waldo Tobler

- If you have spatial or temporal information, this can help R to estimate random effects more accurately
 - Can improve prediction accuracy (smaller p-values)
 - Can give you hints about the underlying causal mechanisms

Part 1: Time and Space in \mathbb{R}

How R deals with time

- Dealing with time in R is somewhat annoying, but not complicated
- Common methods: `as.Date` (days), `as.POSIXlt` (date + time)
- Both require a date/time format: see `?strptime` for examples
- You can transform to specific formats (e.g. day of year) using `format`
- `difftime` is useful for getting differences in time points

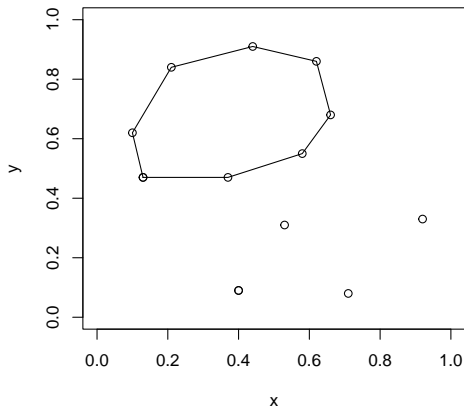
```
##      x          d1          d2
## 1  5 2010-05-06 2010-06-13
## 2 10 2021-11-14 2022-10-14

#Convert data to Date format
dateForm <- '%Y-%m-%d'
dExamp %>%
  mutate(across(c(d1,d2),
                  ~as.Date(.x,format=dateForm))) %>%
  #Get day of year
  mutate(doy=format(d1,format='%j')) %>%
  #Get difference in time between d2 and d1
  mutate(dChange=difftime(d2,d1,units='days'))

##      x          d1          d2  doy  dChange
## 1  5 2010-05-06 2010-06-13 126   38 days
## 2 10 2021-11-14 2022-10-14 318  334 days
```

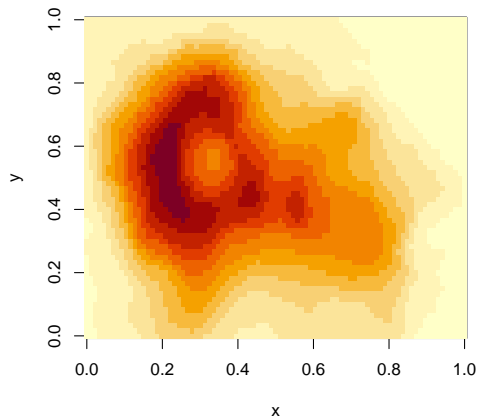
Two main types of spatial data

Vector data: points, lines, and polygons



R packages: `sf`, `sp`, `gstat`, `spdep`

Raster data: cells



R packages: `stars`, `terra`

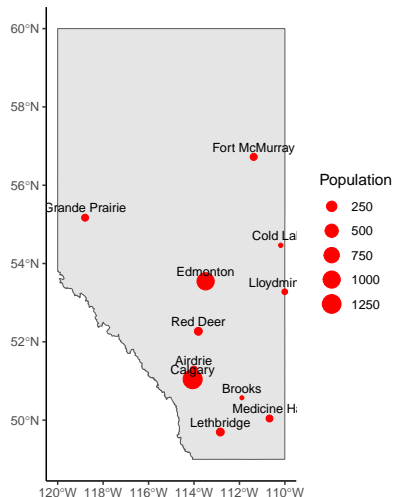
R as a GIS

- A **Geographic Information System** (GIS) is a system for organizing, analyzing, and displaying spatial information
- Common platforms and tools: ArcGIS, QGIS, PostGIS, Python
- A number of R packages are specifically written for dealing with GIS data, usually specific to raster or vector formats
- Ecologists mostly deal with vector data (site locations, boundary polygons) but raster data is sometimes used (NDVI, land cover classes)
- I'll show you a couple practical tips for using the `sf` package (see [here](#) also), but there are [many other packages](#) out there

If you're dealing with large amounts of spatial data *I would encourage you to take a formal GIS course*, as there is a LOT to learn!

Common tasks: making maps

- Vector data are often encoded as *shapefiles* (set of several files)
- Point data can also be read in as *csv* files, which need to be turned into an *sf* object
- *sf* objects can be displayed in *ggplot* using *geom_sf*. Common aesthetics (colour, size) can be mapped onto the plot
 - Objects are layered on the map in order of coding
- Be careful: shapefiles can be very large, which can easily crash R!

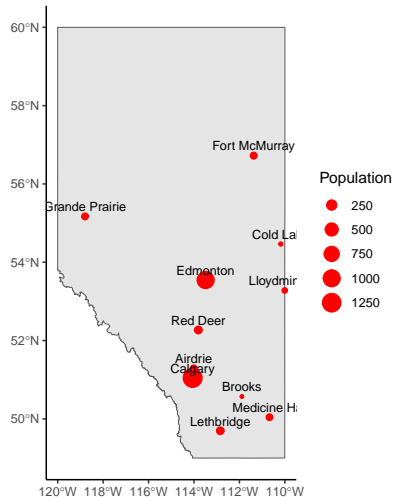


Common tasks: making maps (cont.)

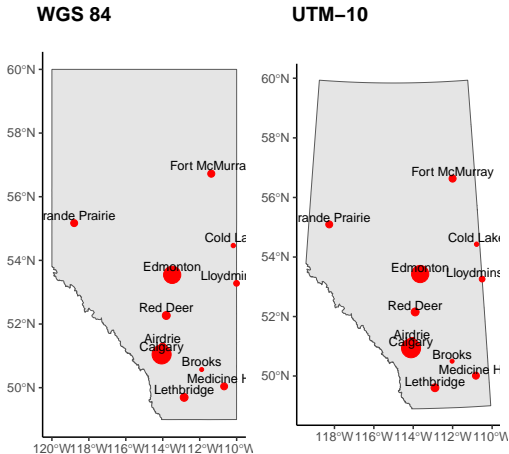
```
#Reads AB boundary shapefile
abBound <- read_sf('./shapefiles/AB_only.shp') %>%
  st_transform(4326)

#Reads city csv
csvPath <- './shapefiles/abCities.csv'
abCities <- read.csv(csvPath) %>%
  #Converts to sf
  st_as_sf(coords = c('lon','lat'),crs=4326)
#NOTE: crs 4326 is common lat/lon format

#Make map
(p1 <- ggplot()+
  #Add boundary
  geom_sf(data=abBound)+
  #Add cities
  geom_sf(data=abCities,aes(size=pop),col='red')+
  #Add labels
  geom_sf_text(data=abCities,aes(label=name),
    size=3,nudge_y=0.25)+
  labs(x=NULL,y=NULL,size="Population"))
```



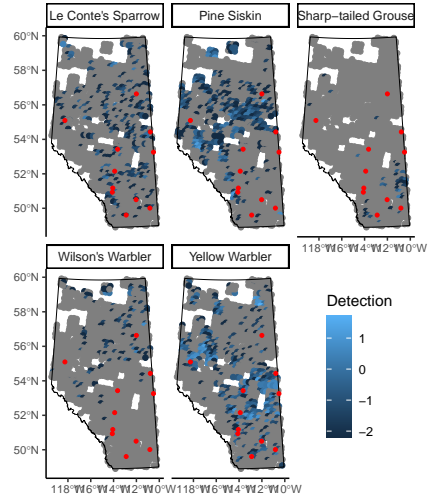
Common tasks: reprojection



- The world is not flat: all maps have to “bend” the data somehow. This is called the map **projection**
- Some map projections preserve *area*, others preserve *distance*. Degrees are not all the same distance apart!
- Usually we’re interested in absolute distance between locations, so *Mercator* (UTM) is a good choice, but be careful which UTM zone you choose!
- sf uses crs codes: **4326** is for lat/lon (WGS 84), 3401 is an Alberta-specific UTM projection
- **Many** others are available

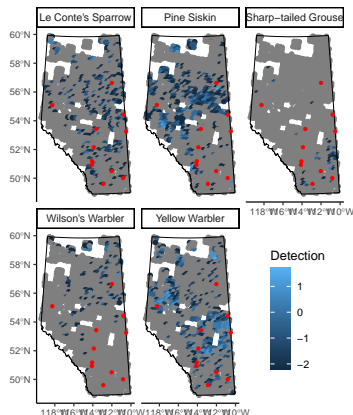
First challenge

- Make this map of bird counts from the ABMI dataset, with Alberta cities and provincial boundaries overlaid on top
- `medDetects` is the median detection rate at each site over several years (I used $\log(\text{medDetects})$ for this map)



First challenge results

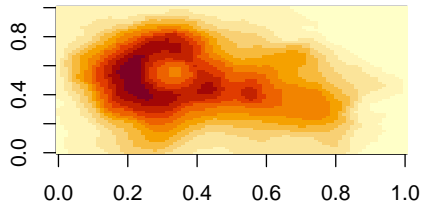
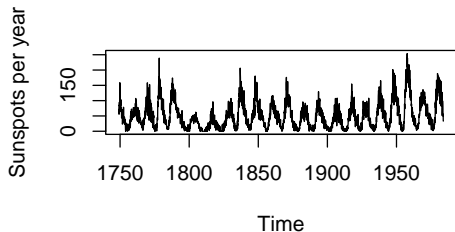
```
abBound <- read_sf('./shapefiles/AB_only.shp') %>%  
  st_transform(3401)  
birdDat <- read.csv('./shapefiles/birdDat.csv') %>%  
  st_as_sf(coords = c('lon', 'lat'), crs=4326) %>%  
  st_transform(st_crs(abBound))  
abCities <- read.csv('./shapefiles/abCities.csv') %>%  
  st_as_sf(coords = c('lon', 'lat'), crs=4326) %>%  
  st_transform(st_crs(abBound))  
  
ggplot()+  
  geom_sf(data=birdDat, aes(col=log(medDetects)))+  
  geom_sf(data=abBound, fill=NA, col='black')+  
  geom_sf(data=abCities, col='red', size=1)+  
  facet_wrap(~Common.Name)+  
  labs(col='Detection')+  
  theme(axis.text = element_text(size=8),  
        legend.position=c(0.85,0.25))
```



Part 2: Spatiotemporal modeling

Temporal or Spatial Data

- Correlation is often present in temporal data or spatial data; causes may be unknown or “uninteresting”
- Usually we are interested in accounting for these patterns, in order to better estimate the “interesting” patterns on top of them
- Last week we talked about *cross*-correlation (i.e. correlation between columns of data); this week we’re talking about *auto*-correlation (i.e. correlation between individual data points in a single column)



Covariance

- Normal distributions¹ don't just have a single σ , but a matrix of values
- If our data y are *independent*, then it looks like this:

$$y \sim \text{Normal}(\hat{y}, \Sigma)$$

$$\hat{y} = [\mu_1, \mu_2, \mu_3]$$

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

- Zeros mean “ μ_1 , μ_2 , & μ_3 aren't related to each other”
- Diagonal elements = *variance*, off-diagonal = *covariance*

¹Multivariate Normal

Covariance and Correlation

In real life, things may not be independent from each other. For example:

- $\sigma = 2$ (variance = $\sigma^2 = 4$)
- μ_1 and μ_2 are strongly correlated ($r=0.7$), but μ_3 is not related to anything ($r=0$).
Shown here as a *correlation matrix* (R):

$$R = \begin{bmatrix} 1 & 0.7 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- When multiplied by the variance, this becomes the *covariance matrix* (Σ)

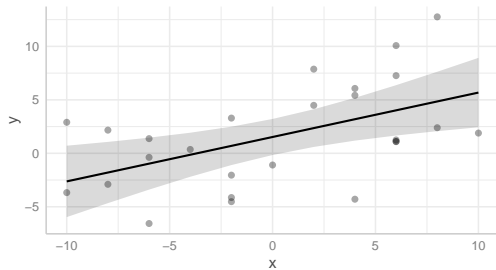
$$\Sigma = \begin{bmatrix} \sigma^2 \times 1 & \sigma^2 \times 0.7 & \sigma^2 \times 0 \\ \sigma^2 \times 0.7 & \sigma^2 \times 1 & \sigma^2 \times 0 \\ \sigma^2 \times 0 & \sigma^2 \times 0 & \sigma^2 \times 1 \end{bmatrix} = \begin{bmatrix} 4 & 2.8 & 0 \\ 2.8 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Let's start with an example

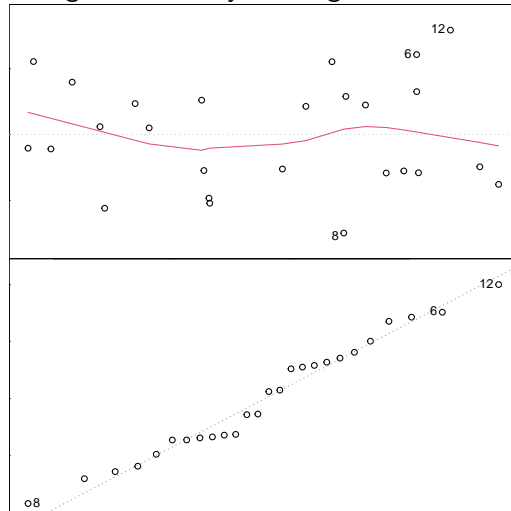
- Say we're fitting a simple linear regression on a dataset collected across space

##		y	x	site	lat	lon
## 1	-5.553336	-5.966361	a	3.884460	-0.7769030	
## 2	-2.845579	-2.399296	b	-3.619967	1.3429571	
## 3	7.936928	5.658655	c	-4.640508	-0.8192497	
## 4	-1.939703	-4.689827	d	-1.449829	-1.7533565	
## 5	1.945473	-2.277718	e	-1.210539	2.6578723	

Model: $\text{lm}(y \sim x)$

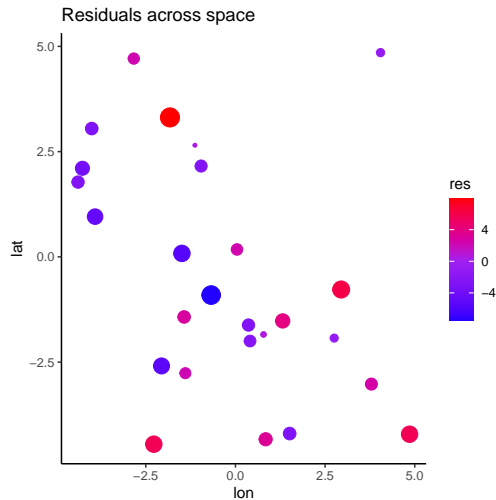
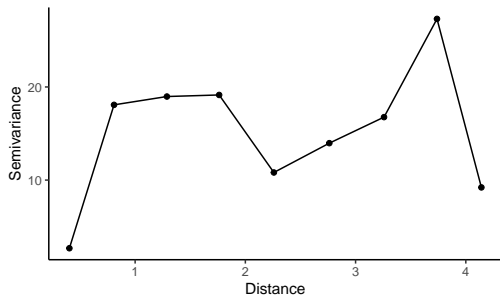


Things look mostly OK, right?



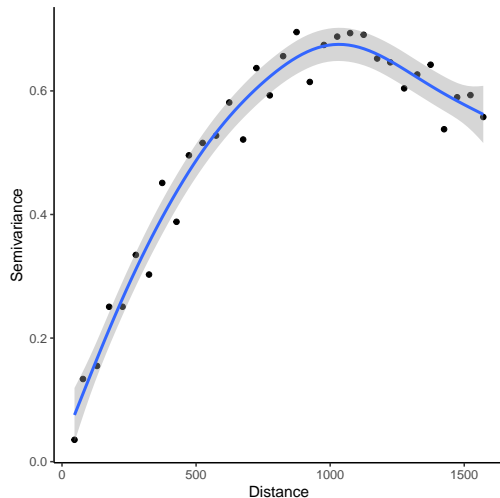
Spatial residual plot

- Residuals are spatially *non-independent!*
- *Variograms* are a common tool to examine how variance changes with distance
- Uncorrelated spatial data will have a *flat* variogram (no change in semivariance with distance)



How do variograms work?

- Variograms compare the squared difference (variance) between values spaced at different distances
- If close values are similar, variance increases with increasing distance before leveling off
- Analysis of the specific shape of the curve is called *variography*, and is important for spatial modeling



Gaussian Process Modelling

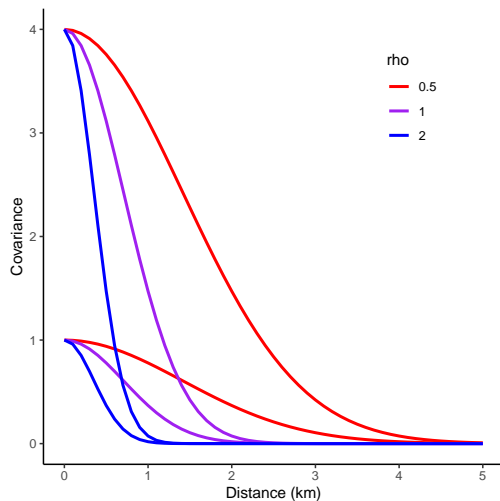
- We can model covariance between things as a function of *distance* and use it to predict what other points in between might be (a.k.a. **Kriging**)
- Squared-exponential is fairly common²:

Σ = covariance

Σ = variance \times correlation

$\Sigma = \sigma^2 \times e^{-\rho^2 \text{Dist}^2}$

- Instead of finding a single σ value, R now looks for σ (maximum covariance) and ρ (decay with distance)



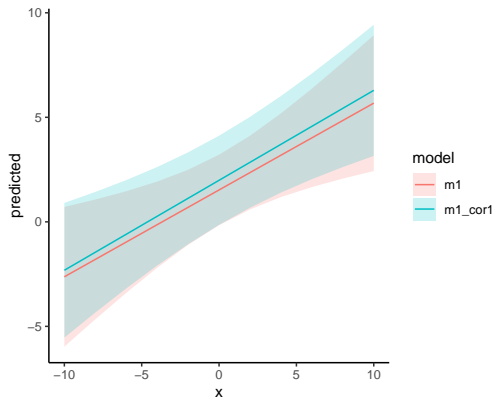
²Also common: AR-1 (temporal processes), Matérn (spatial processes)

How do we do this in R?

- For simpler models, you can use a generalized linear system (glms)
- Here I've used corGaus to specify a Gaussian (squared-exponential) kernel

```
corForm <- corGaus(form = ~ lat + lon)
m1_cor <- gls(y ~ x, correlation = corForm,
              dat = dat2_small)
```

```
## Generalized least squares fit by REML
## Model: y ~ x
## Data: dat2_small
##      AIC      BIC    logLik
## 141.0709 145.7831 -66.53545
##
## Correlation Structure: Gaussian spatial correlation
## Formula: ~lat + lon
## Parameter estimate(s):
##      range
## 0.5602678
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept) 0.7480258 0.7422872   1.007731  0.3236
## x           0.3719829 0.0362915  10.249862  0.0000
##
## Correlation:
## (Intr)
## x 0.089
##
```



Not too bad in this case, but in general,
high spatial correlation effectively means
your sample size is smaller

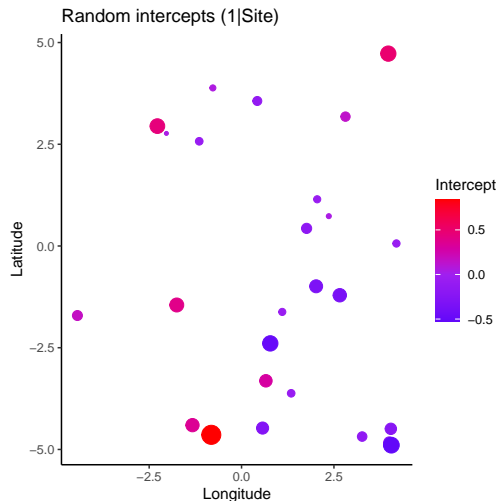
Random effects: standard

- Say that we collected data at 26 sites, and we're interested in the effect of y on x
- Let's first fit a model with a random intercept for site

#Same syntax as lmer models:

```
lmm2 <- glmmTMB(y~x+(1|site),data=dat2)
```

- If we plot the intercepts for each site, we see that they are clustered
- Moran's I is a commonly test for spatial autocorrelation (see `ape` and `spdep` packages)



Random effects: spatial

- Re-fit model with a spatial (exponential) random effect

```
#Coordinates
```

```
dat2$coords <- numFactor(dat2$lon,dat2$lat)
```

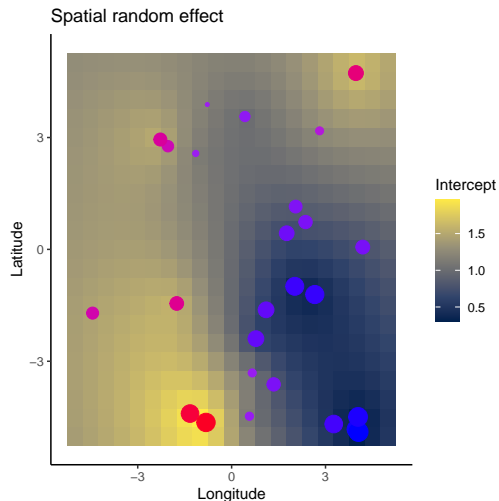
```
#Group factor (only 1 here)
```

```
dat2$group <- factor(rep(1,nrow(dat2)))
```

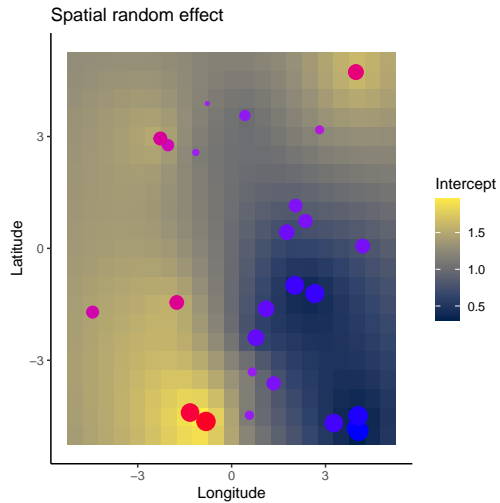
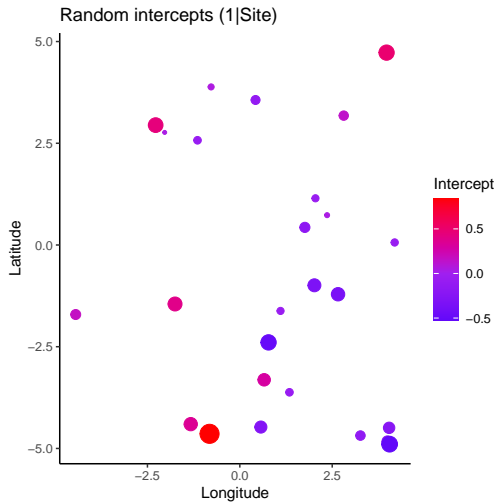
```
#Fit model with spatial random effect
```

```
lmm3 <- glmmTMB(y~x+exp(coords+0|group),data=dat2)
```

- The spatial random *field* is shown in the background, along with individual draws from the field (sites)

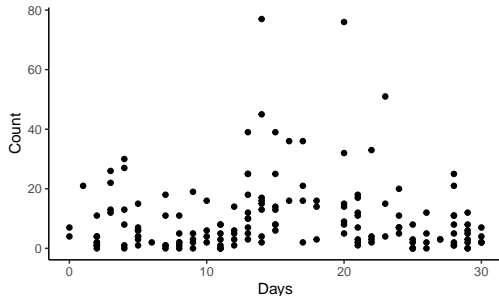


How do these differ?



Second challenge

```
##   count temperature site days
## 1     7    5.310173    i     0
## 2     4    7.442478    e     0
## 3    21   11.457067    m     1
## 4     2   18.164156    n     2
## 5     0    4.033639    a     2
## 6     4   17.967794    c     2
```

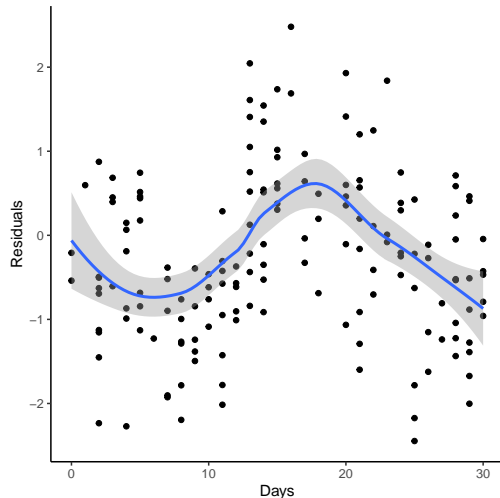


- Things can vary in *time* as well as space
- There is a dataset found [here](#) (timeDat.csv) with count data recorded at each location, along with temperature and a day variable
- How does count change with temperature? Is there temporal autocorrelation between the data?
- Fit a model that accounts for location, as well as any temporal autocorrelation

Second challenge results

```
dat <- read.csv('./timeDat.csv') %>%  
  mutate(site=factor(site))  
  
#Naive random effect model  
m2 <- glmmTMB(count~temperature+(1|site),  
              data=dat,  
              family='nbinom2')  
  
m2Res <- residuals(m2,'deviance')  
  
dat %>% mutate(m2Res) %>%  
  mutate(days=as.numeric(days)) %>%  
  ggplot(aes(x=days,y=m2Res))+  
  geom_point()+  
  geom_smooth(method='loess',formula=y~x)+  
  labs(x='Days',y='Residuals')
```

Looks like a fairly strong pattern in residuals



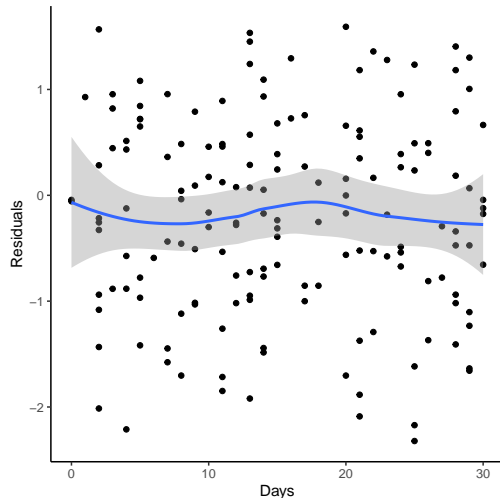
Second challenge results (cont)

```
#Days as "numeric factor"
dat$dayF <- numFactor(dat$days)
dat$group <- factor(rep(1,nrow(dat)))

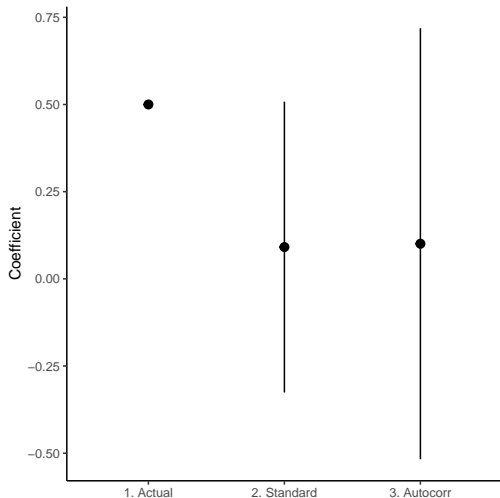
#Fit model with temporal random effect
m2_ac <- glmmTMB(count~temperature+(1|site)+
  exp(dayF+0|group),
  data=dat,
  family='nbinom2')
m2acRes <- residuals(m2_ac,'deviance')

dat %>% mutate(m2acRes) %>%
  mutate(days=as.numeric(days)) %>%
  ggplot(aes(x=days,y=m2acRes))+
  geom_point()+
  geom_smooth(method='loess',formula=y~x)+
  labs(x='Days',y='Residuals')
```

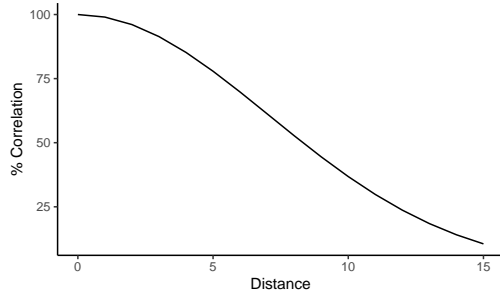
Pattern has largely disappeared



Second challenge results (cont. . .)

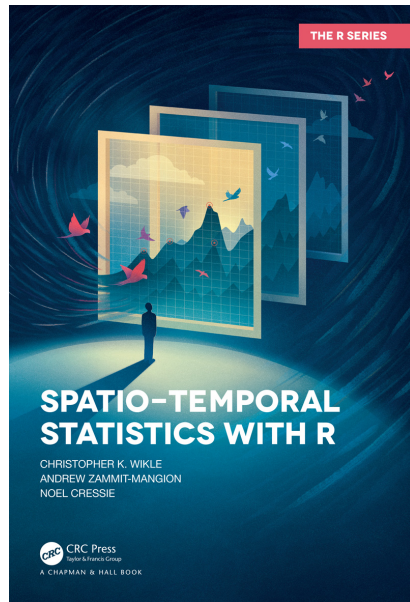


- Model with autocorrelation has greater uncertainty in prediction
- Unfortunately, neither model could separate the effect from zero
- Covariance function can be used to *plan future sampling*



What if my data varies over space AND time?

- You can jointly estimate covariance functions for different types of distances, however. . .
- Make sure that these can actually be estimated! Did you actually gather data at all combinations of space and time?
- In the literature these can be *separable* or *non-separable* covariance functions (see [here](#) or [here](#) for more details)

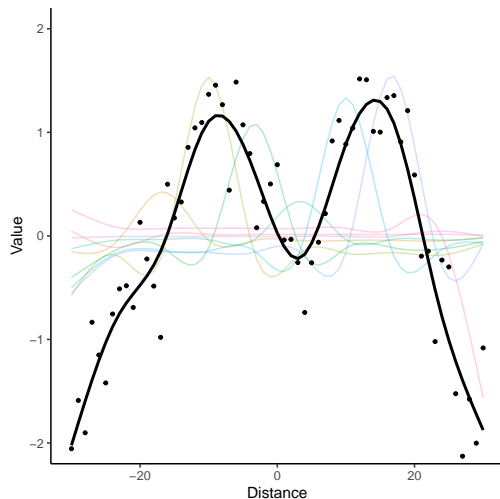


Problem: hard for large datasets

- You'll notice that I've only been doing this for correlated random intercepts
- OK because the covariance matrix for intercepts is small (26×26), but the covariance matrix for the entire dataset is large (260×260)
- If data points are all from different times, this can start using a lot of memory and can take a *long time*
- In general, *covariance-based methods* do not scale well to larger datasets, so we need a better way of doing things

Solution: basis functions

- There are quicker ways to deal with spatial and temporal effects, including spatial partial differential equations (R-INLA and TMB, see [here](#)) or Gaussian predictive process (plgp) models
- We're going to use an approach you've already used: additive models!
- GAMs are a common way to deal with this problem, as they move the ST variation from the covariance matrix into a random effect, which speeds up estimation



Covariance vs Random Effects

Estimate covariance explicitly

$$\hat{y} = X\beta$$

$$y \sim \text{Normal}(\hat{y}, \Sigma)$$

$$\Sigma = \begin{pmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,j} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{i,1} & \sigma_{i,2} & \cdots & \sigma_{i,j} \end{pmatrix}$$

$$\sigma_{i,j} = \alpha^2 \exp\left(-\frac{1}{2\rho^2} D_{i,j}^2\right)$$

Approximate covariance with basis functions

$$\hat{y} = X\beta + Zu$$

$$y \sim \text{Normal}(\hat{y}, \sigma)$$

$$u \sim \text{Normal}(0, \lambda S)$$

GAM approach

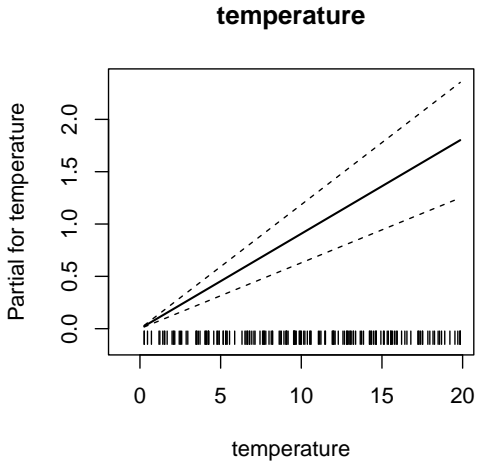
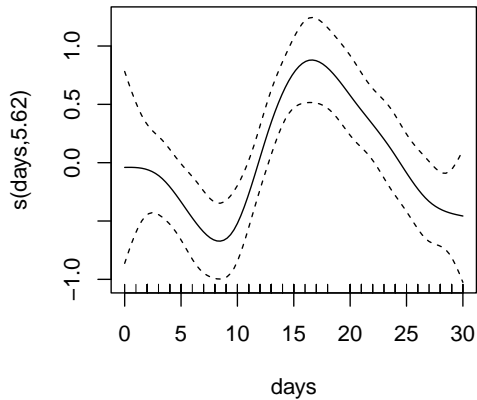
```
m3 <- gam(count~temperature+
           s(days,bs='cr'),
           data=dat,family='nb')
summary(m3)

##
## Family: Negative Binomial(1.413)
## Link function: log
##
## Formula:
## count ~ temperature + s(days, bs = "cr")
##
## Parametric coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.17059    0.16275   7.193 6.36e-13 ***
## temperature  0.09075    0.01390   6.530 6.58e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df Chi.sq p-value
## s(days) 5.625  6.746  45.02  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.271   Deviance explained = 34.3%
## -REML = 514.13   Scale est. = 1           n = 160
```

- Continuous variables (time) can be specified as a smoother $s(\text{time})$
- 2D distances (e.g. E & N) also possible: $s(E,N)$
 - Be careful that units are on the same length scale!
- Same caveats as other GAMs apply: do you have enough basis dimensions (k)? See [Lecture 6](#)
- Test for residual autocorrelation OR do sensitivity analysis for k

GAM results

Default GAM plots:



Third challenge

- During my PhD, I studied alfalfa leafcutting bees (*Megachile rotundata*) in canola fields. I counted bees visiting flowers at distances away from their shelters ("hives"); some of these data are found [here](#) (seedFieldDat.csv).
- Using a GAM, model how visitation changes with: distance from shelter (sDist), edge of field (eDist), while controlling for observation time (time), day of year (StartTime), and different Fields. You can either do a model for each Year separately, or have Year as a term in the model



Seed canola field



To do for next week

- Find a paper that is important for your field and read it (preferably not a methods paper)
- In the Methods section, pay attention to how they write about their data collection and data analysis
 - How many samples did they gather? Was this easy to figure out?
 - Do they describe any models? If you had to write these models in R, what might they look like?
- How do they show the results of these models in the Results section?
- Was this paper easy to read? What would make it easier to understand?