Bivariate and Spatial Smoothing

Seminar: Modern Regression Analysis

Sándor Daróczi

July 7, 2023

Agenda

1. Motivation

2. Bivariate Smoothing with P-Splines

3. Spatial Smoothing with Kriging

4. Spatial Smoothing for Discrete Locations

1. Motivation

1. Motivation

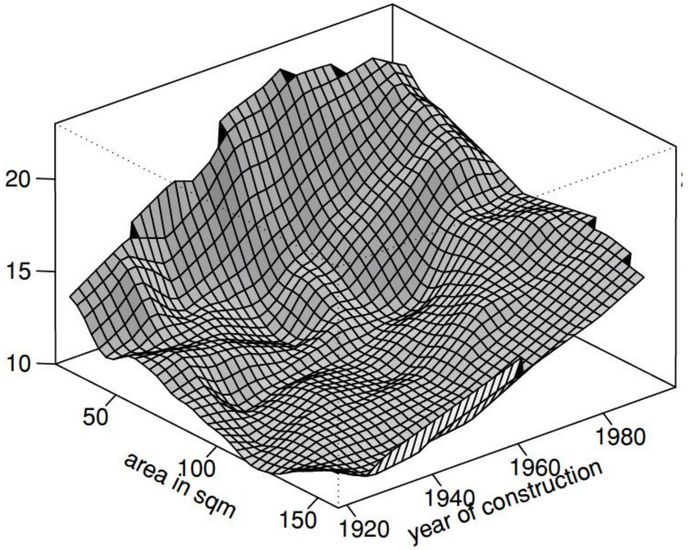
1.1 Bivariate Smoothing

Instead of just one, we have two continuous regressor variables which might interact with each other.

We assume that there is a smooth function f which describes the effect of the variables on the outcome as follows:

$$y_i = f(z_{1i}, z_{2i}) + \varepsilon_i$$

The function f is also called **interaction surface**.



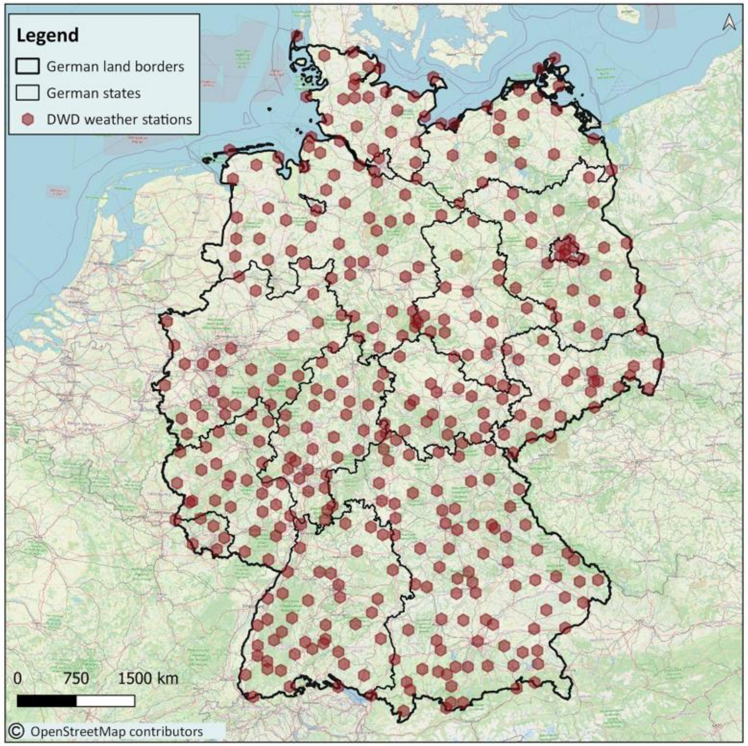
1. Motivation

1.2 Spatial Smoothing

We have 2D spatial data with **spatially correlated** response y and coordinate values z_{1i} and z_{2i} (e.g. longitude, latitude). Data is missing at some locations, and the task is to predict these missing values using the model

$$y_i = f(z_{1i}, z_{2i}) + \varepsilon_i$$

where f is a smooth function.



Source: https://www.sciencedirect.com/science/article/pii/S0013935122023891

Figure: map of DWD weather stations in Germany

2. Bivariate Smoothing with P-Splines

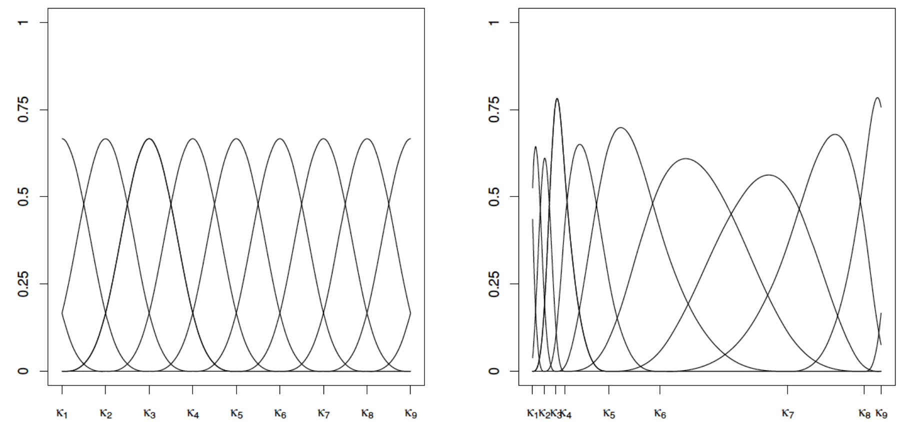
2. Bivariate Smoothing with P-Splines

2.1 One Dimensional Smoothing - Recap

In one dimensional smoothing, we would like to estimate f in the model $y = f(z) + \varepsilon$ using a linear combination of basis functions $B_1(z), B_2(z), \ldots, B_d(z)$:

$$y_i = \sum_{j=1}^{d} \gamma_j B_j(z_i) + \varepsilon_i,$$

where $d = m + l - 1$.



B-spline bases of degree $l = 3$, with equidistant knots (left) and unevenly distributed knots (right)

2. Bivariate Smoothing with P-Splines

2.1 One Dimensional Smoothing - Recap

With the design matrix defined as

$$\boldsymbol{Z} = \begin{pmatrix} B_1(z_1) & \cdots & B_d(z_1) \\ \vdots & \ddots & \vdots \\ B_1(z_n) & \cdots & B_d(z_n) \end{pmatrix}$$

we can write $\boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$, from which the least square estimate can be determined analogously to standard regression:

$$\widehat{\boldsymbol{\gamma}} = (\boldsymbol{Z}'\boldsymbol{Z})^{-1}\boldsymbol{Z}'\boldsymbol{y}$$

2. Bivariate Smoothing with P-Splines

2.2 Tensor Product Bases

Now, in order to estimate the function f in the bivariate model $y = f(z_1, z_2) + \varepsilon,$ we first construct the univariate bases

$$B_j^{(1)}(z_1), \mathrm{j} = 1, \dots, d_1 \quad \text{and} \quad B_r^{(2)}(z_2), \mathrm{j} = 1, \dots, d_2$$

Then, the tensor product basis consists of all functions of the form

$$B_{jr}(z_1, z_2) = B_j^{(1)}(z_1) \cdot B_r^{(2)}(z_2), \qquad \mathrm{j} = 1, \dots, d_1, \mathrm{r} = 1, \dots, d_2$$

The representation of f is

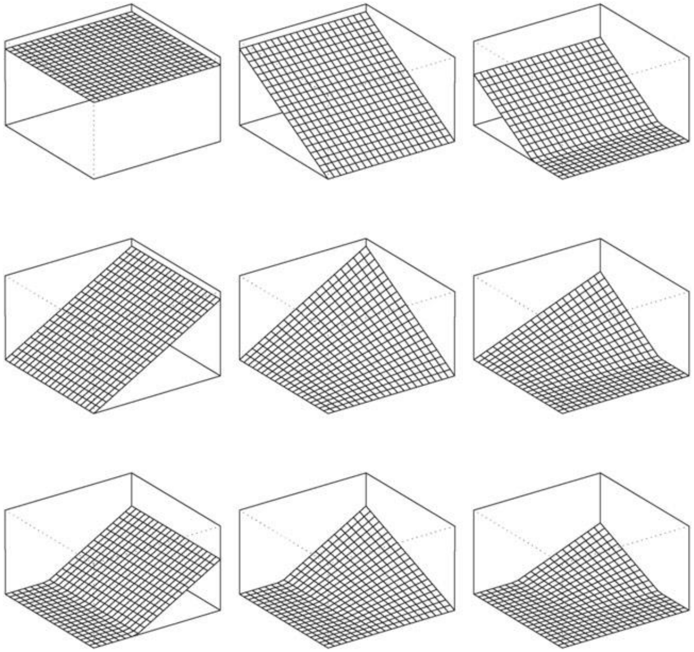
$$f(z_1, z_2) = \sum_{j=1}^{d_1} \sum_{r=1}^{d_2} \gamma_{jr} B_{jr}(z_1, z_2)$$

2. Bivariate Smoothing with P-Splines

2.2 Tensor Product Bases

$$B_1^{(1)}(z_1) = 1,$$
$$B_2^{(1)}(z_1) = z_1,$$
$$B_3^{(1)}(z_1) = (z_1 - \kappa_1)_+$$

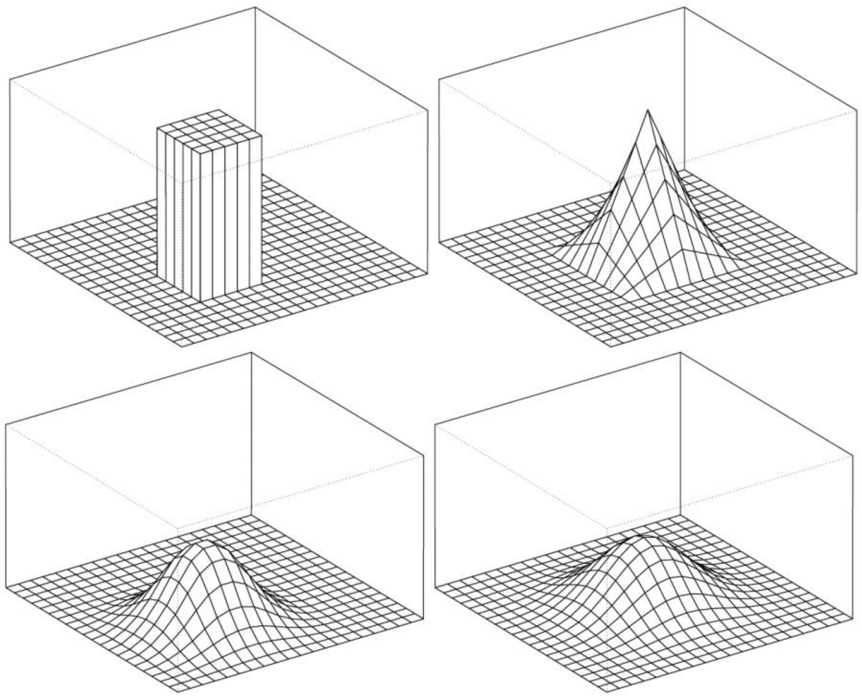
$$B_1^{(2)}(z_2) = 1,$$
$$B_2^{(2)}(z_2) = z_2,$$
$$B_3^{(2)}(z_2) = (z_2 - \kappa_2)_+$$



Example: Tensor product splines obtained from univariate TP bases

2. Bivariate Smoothing with P-Splines

2.2 Tensor Product Bases



Another example: tensor product basis functions obtained from univariate B-splines of degrees $l = 0, 1, 2, 3$

2. Bivariate Smoothing with P-Splines

2.2 Tensor Product Bases

The regression equation obtained from tensor product splines is

$$\boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon},$$

where the design matrix Z has rows

$$\boldsymbol{z}_i' = (B_{11}(z_{i1}, z_{i2}), \ldots, B_{d_1 1}(z_{i1}, z_{i2}), \ldots, B_{1 d_2}(z_{i1}, z_{i2}), \ldots, B_{d_1 d_2}(z_{i1}, z_{i2}))$$

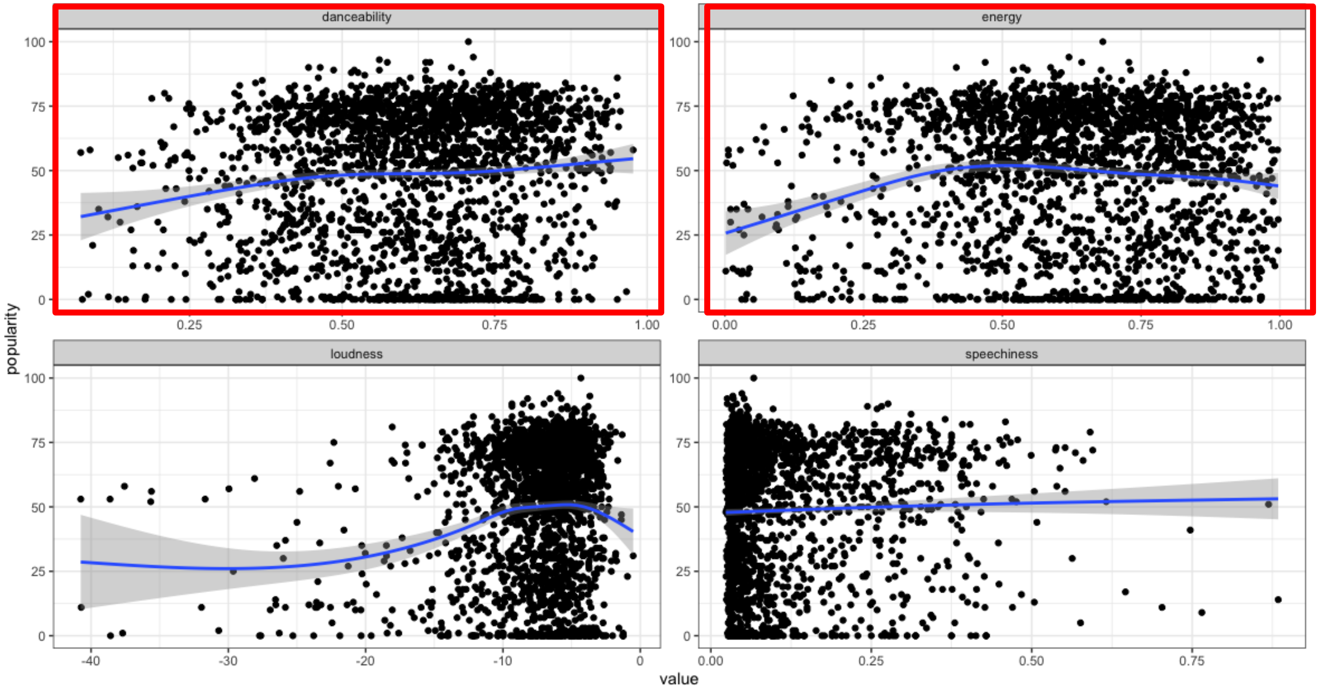
and the vector of regression coefficients is $\boldsymbol{\gamma} = (\gamma_{11}, \ldots, \gamma_{d_1 1}, \ldots, \gamma_{1 d_2}, \ldots, \gamma_{d_1 d_2})'$.
The least square estimate is

$$\widehat{\boldsymbol{\gamma}} = (\boldsymbol{Z}'\boldsymbol{Z})^{-1}\boldsymbol{Z}'\boldsymbol{y}$$

2. Bivariate Smoothing with P-Splines

2.3 Bivariate Smoothing for the Spotify Dataset

Goal: Find two continuous variables from the Spotify dataset with possible nonlinear effects, and apply bivariate smoothing to predict popularity.



2. Bivariate Smoothing with P-Splines

2.3 Bivariate Smoothing for the Spotify Dataset

Without interactions

```
> bivariate_model <- lm(z ~ x + y, data = df)
> summary(bivariate_model)

Call:
lm(formula = z ~ x + y, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-54.291 -22.011   8.224  22.473  49.925 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   38.581      2.470  15.622  < 2e-16 ***
x             13.373      3.122   4.283 1.92e-05 ***
y              3.090      2.761   1.119    0.263    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26.87 on 2299 degrees of freedom
Multiple R-squared:  0.009041,	Adjusted R-squared:  0.008179 
F-statistic: 10.49 on 2 and 2299 DF,  p-value: 2.923e-05
```

With interactions

```
> bivariate_model.inter <- lm(z ~ x * y, data = df)
> summary(bivariate_model.inter)

Call:
lm(formula = z ~ x * y, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-54.106 -22.246   8.294  22.256  50.544 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   22.130      4.716   4.692 2.86e-06 ***
x             46.215      8.613   5.365 8.88e-08 ***
y             32.443      7.688   4.220 2.54e-05 ***
x:y          -56.891     13.913  -4.089 4.48e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26.78 on 2298 degrees of freedom
Multiple R-squared:  0.0162,	Adjusted R-squared:  0.01491 
F-statistic: 12.61 on 3 and 2298 DF,  p-value: 3.539e-08
```

2. Bivariate Smoothing with P-Splines

2.3 Bivariate Smoothing for the Spotify Dataset

Fitting bivariate smoothing model with P-Splines

```
> model.te1 <- gam(z ~ te(x, y, bs = 'ps', sp = c(0.05, 0.05)),
+                  family=gaussian,
+                  method = "REML",
+                  data=df)
> summary(model.te1)

Family: gaussian
Link function: identity

Formula:
z ~ te(x, y, bs = "ps", sp = c(0.05, 0.05))

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  48.5456     0.5547   87.52   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
          edf Ref.df     F p-value
te(x,y) 22.35  23.26 3.683  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.0272   Deviance explained = 3.66%
-REML =  10852  Scale est. = 708.29    n = 2302
```

x: danceability

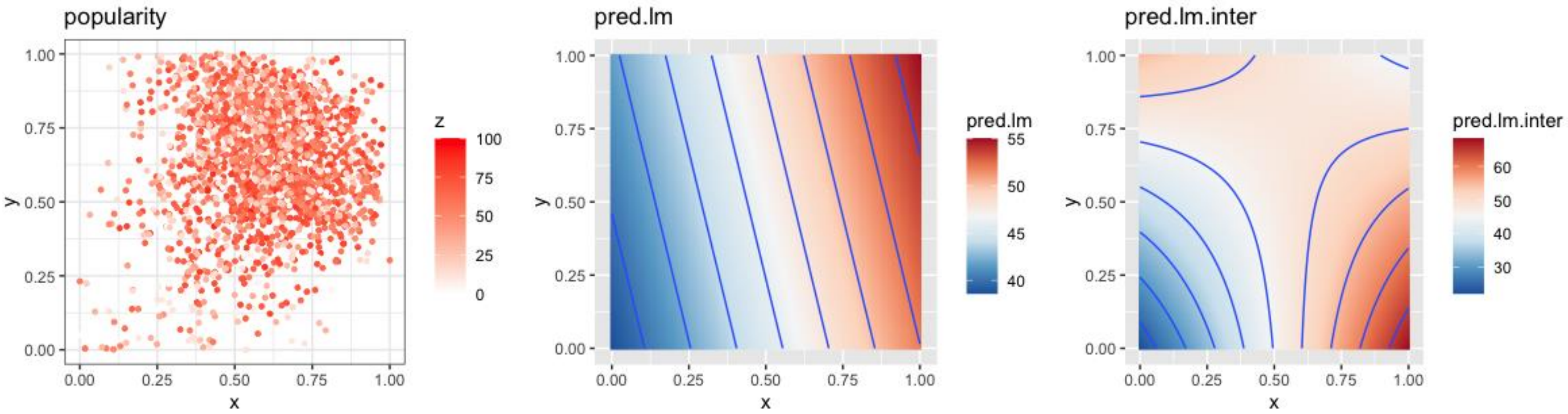
y: energy

z: popularity

sp: penalization coefficients

2. Bivariate Smoothing with P-Splines

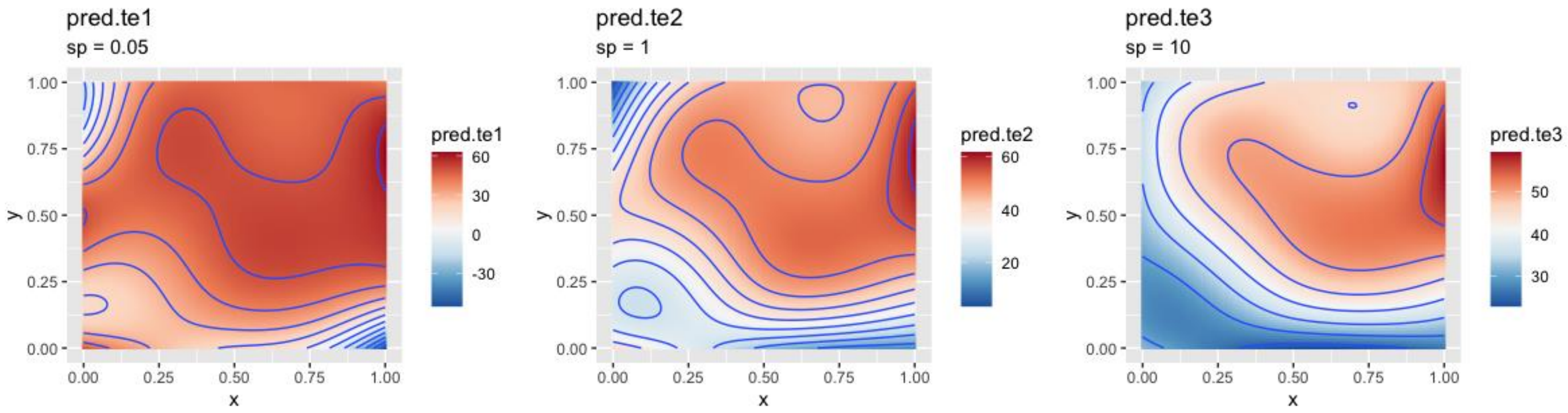
2.3 Bivariate Smoothing for the Spotify Dataset



Predictions of the linear models (without and with interaction term)

2. Bivariate Smoothing with P-Splines

2.3 Bivariate Smoothing for the Spotify Dataset



Predictions of the P-Spline smoothers on $[0,1]^2$ with $sp = 0.05,\ sp = 1$ and $sp = 10$

2. Bivariate Smoothing with P-Splines

2.3 Bivariate Smoothing for the Spotify Dataset

	$sp = 0.05$	$sp = 1$	$sp = 10$	$lm(z \sim x * y)$
Deviance explained	3.66%	3.57%	3.37%	1.62%
R-sq. (adj)	0.0272	0.0282	0.0286	0.01491

➡ Smoothing results in a more increased fit compared to linear models

Drawback: difficult interpretability

3. Spatial Smoothing with Kriging

3. Spatial Smoothing with Kriging

3.1 Radial Basis Functions

Radial basis functions are scalar valued functions $B_\kappa\colon \mathbb{R}^2 \to \mathbb{R}$ such that there is a $B\colon \mathbb{R}^+ \to \mathbb{R}$ with

$$B_\kappa(z) = B(\|z-\kappa\|) = B(r),$$

where $r \coloneqq \|z-\kappa\|$. I.e. the value of B_κ only depends on the distance from the knot $\kappa = (\kappa_1, \kappa_2)$.

Typically, we choose each knot κ_j from the set of all observation points $\{z_1, \ldots, z_n\}$

➡ Distribution adapts better to the data

3. Spatial Smoothing with Kriging

3.1 Radial Basis Functions

Most well-known example: **Thin plate splines**

$$f(z_1, z_2) = \beta_0 + \beta_1 z_1 + \beta_2 z_2 + \sum_{j=1}^{n} \gamma_j B_j(z_1, z_2),$$

where

$$B_j(z_1, z_2) = \left\|z - z_j\right\|^2 \log\left(\left\|z - z_j\right\|\right)$$

The corresponding radial basis function is

$$B(r) = r^2 \log(r)\,.$$

3. Spatial Smoothing with Kriging

3.1 Radial Basis Functions

Thin plate splines are obtained via minimizing the bivariate analogue of the integrated square second derivative penalty:

$$\int\int\left[\left(\frac{\partial^2}{\partial^2 z_1}+2\frac{\partial^2}{\partial z_1\partial z_2}+\frac{\partial^2}{\partial^2 z_2}\right)f(z_1,z_2)\right]^2 dz_1dz_2$$

In this sense, thin plate splines are generalizations of natural cubic splines to the bivariate case.

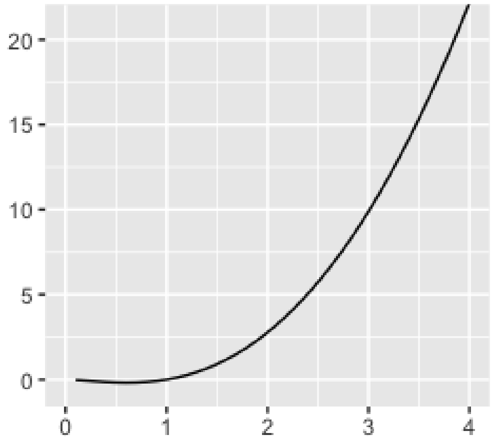
3. Spatial Smoothing with Kriging

3.1 Radial Basis Functions

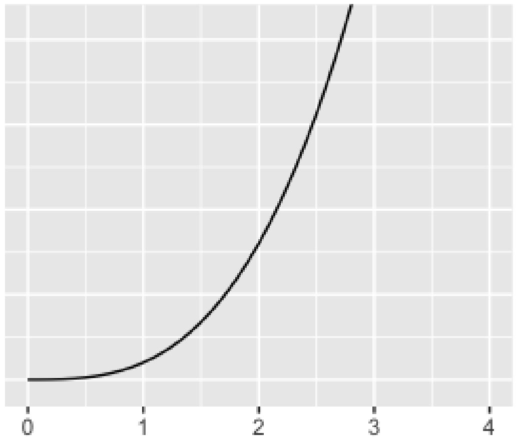
Other examples:

$$B(r) = r^l, \qquad l\ odd$$

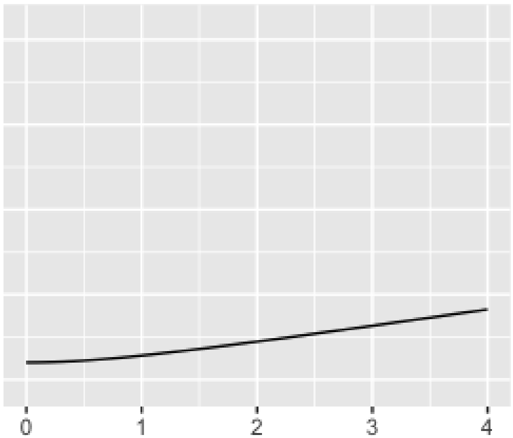
$$B(r) = \sqrt{r^2 + c^2}, \qquad c > 0\ constant$$



$B(r) = r^2 \log(r)$



$B(r) = r^3$



$B(r) = \sqrt{r^2 + 1}$

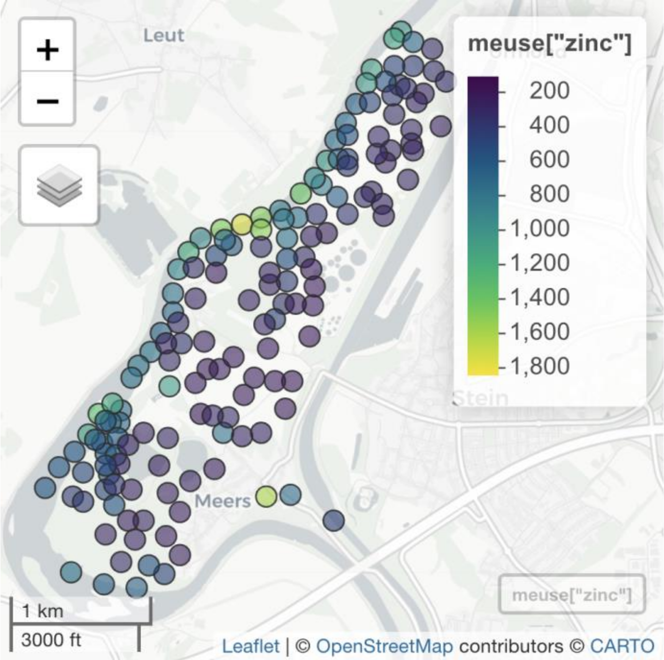
3. Spatial Smoothing with Kriging

3.2 Classical Geostatistical Model

We would like to define a smoothing approach for modelling interaction surfaces which have spatial correlations and spatial trends.

Example: the *meuse* dataset contains measurements of four heavy metals sampled from the top soil in a flood plain along the river Meuse. The goal is to model the spatial distribution of zinc concentration values for locations with no data.

```
> head(meuse)
       x      y cadmium copper lead zinc  elev       dist   om ffreq soil lime landuse dist.m
1 181072 333611    11.7     85  299 1022 7.909 0.00135803 13.6     1    1    1      Ah     50
2 181025 333558     8.6     81  277 1141 6.983 0.01222430 14.0     1    1    1      Ah     30
3 181165 333537     6.5     68  199  640 7.800 0.10302900 13.0     1    1    1      Ah    150
4 181298 333484     2.6     81  116  257 7.655 0.19009400  8.0     1    2    0      Ga    270
5 181307 333330     2.8     48  117  269 7.480 0.27709000  8.7     1    2    0      Ah    380
6 181390 333260     3.0     61  137  281 7.791 0.36406700  7.8     1    2    0      Ga    470
```



Source: https://pages.cms.hu-berlin.de/EOL/gcg_quantitative-methods/Lab14_Kriging.html

3. Spatial Smoothing with Kriging

3.2 Classical Geostatistical Model

The classical geostatistical model is for this purpose and has the form

$$y(s) = \mu(s) + \gamma(s) + \varepsilon(s), \qquad s \in \mathbb{R}^2$$

where

1) $\mu(s) = \boldsymbol{x}(s)'\boldsymbol{\beta}$ is the spatial trend for covariates x

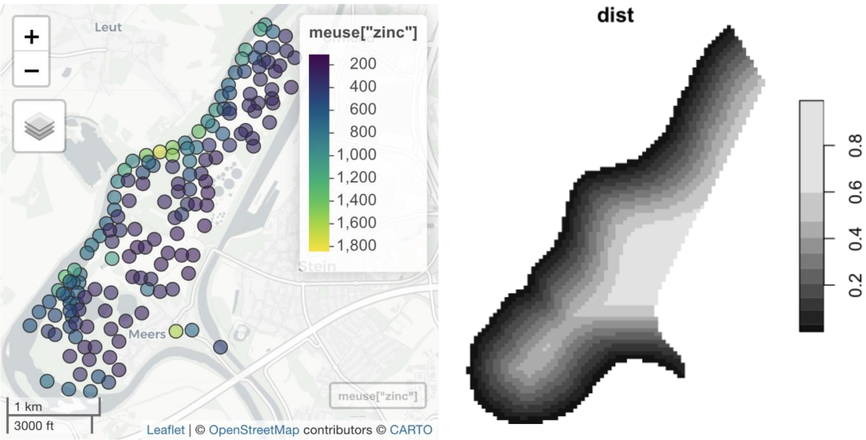
2) $\gamma(s)$ is a stationary Gaussian field with $E\big(\gamma(s)\big) = 0$, $Var\big(\gamma(s)\big) = \tau^2$ and $Corr\big(\gamma(s), \gamma(t)\big) = \rho(s,t) = \rho(\|s - t\|) = \rho(h)$

3) $\varepsilon(s) \sim N(0, \sigma^2)$ is an i.i.d. error term

3. Spatial Smoothing with Kriging

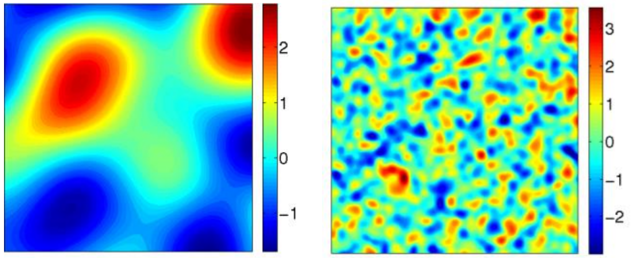
3.2 Classical Geostatistical Model

1) The spatial trend $\mu(s) = \boldsymbol{x}(s)'\boldsymbol{\beta}$ can be for example the distance from the river:



Source: https://rpubs.com/nabilabd/118172

2) Some possible plots of the Gaussian field $\gamma(s)$:



Source: https://www.semanticscholar.org/paper/Generating-Realisations-of-Stationary-Gaussian-by-Powell/3e940aa314d07db1157a8683a3d42189b97c1b48/figure/2

3. Spatial Smoothing with Kriging

3.2 Classical Geostatistical Model

In matrix notation, we can write

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{Z\gamma} + \boldsymbol{\varepsilon}$$

where $\boldsymbol{\gamma} = (\gamma(s_{(1)}), \ldots, \gamma(s_{(d)}))'$ are the values of the Gaussian process at the d unique observed locations, and

$$\boldsymbol{Z}[i,j] = \begin{cases} 1 & if\ y_i\ is\ observed\ at\ point\ s_{(j)} \\ 0 & otherwise \end{cases}$$

3. Spatial Smoothing with Kriging

3.3 Kriging as a Basis Function Approach

To obtain a compact form that can be used for predictions, we can use a different parametrization of the model using the basis functions $B_j(s) = \rho(s, s_{(j)})$. The reparametrized model is defined as

$$\boldsymbol{y} = \boldsymbol{X\beta} + \widetilde{\boldsymbol{Z}}\widetilde{\boldsymbol{\gamma}} + \boldsymbol{\varepsilon},$$

where the design matrix is

$$\widetilde{\boldsymbol{Z}}[i, j] = B_j(s_i) = \rho(s_i, s_{(j)})$$

3. Spatial Smoothing with Kriging

3.3 Kriging as a Basis Function Approach

For a single observation, this corresponds to

$$y(s) = x(s)'\beta + f_{geo}(s) + \varepsilon(s), \qquad f_{geo}(s) = \sum_{j=1}^{d} \tilde{\gamma}_j \, \rho(s, s_{(j)})$$

To penalize $\widetilde{\boldsymbol{\gamma}}$**,** we can define $\boldsymbol{R}[i,j] = \rho(s_{(i)}, s_{(j)})$ and use the penalty function

$$\lambda \widetilde{\boldsymbol{\gamma}}' \boldsymbol{K} \widetilde{\boldsymbol{\gamma}} = \frac{\sigma^2}{\tau^2} \widetilde{\boldsymbol{\gamma}}' \boldsymbol{R} \widetilde{\boldsymbol{\gamma}}$$

From this, the least square estimate can be easily computed.

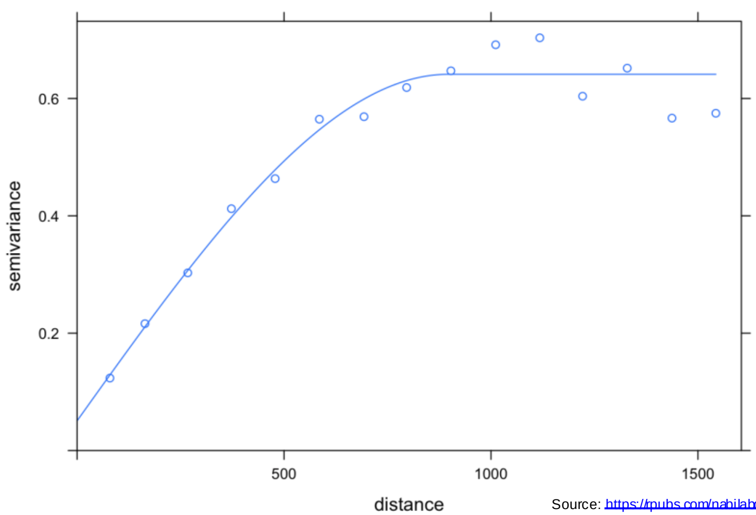
3. Spatial Smoothing with Kriging

3.4 Kriging in R

In order to perform kriging using R, one first has to fit a variogram to model the variability between points with respect to the distance between them.

```
lzn.vgm <- variogram(log(zinc)~1, meuse) # calculates sample variogram values
lzn.fit <- fit.variogram(lzn.vgm, model=vgm(1, "Sph", 900, 1)) # fit model
```

```
plot(lzn.vgm, lzn.fit) # plot the sample values, along with the fit model
```



Source: https://rpubs.com/nabilabd/118172

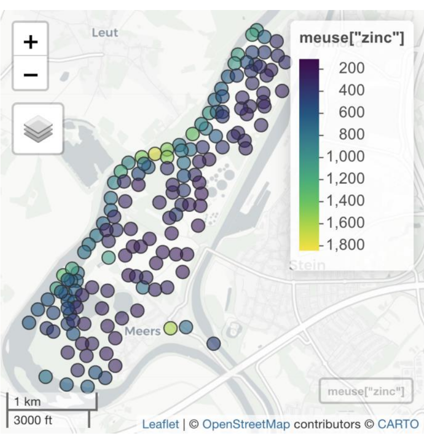
3. Spatial Smoothing with Kriging

3.4 Kriging in R

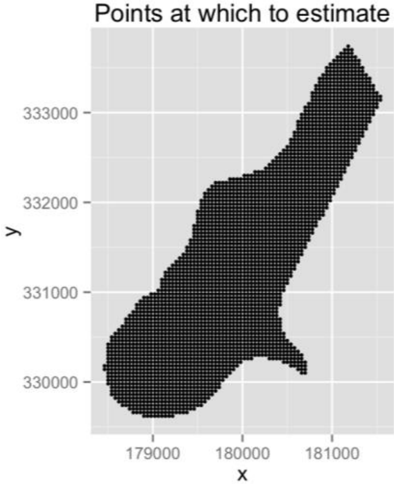
Then, we take our datapoints, the grid which we would like to predict, and the variogram, and use the *krige* function of the *gstat* library to compute our predictions.

```
lzn.kriged <- krige(log(zinc) ~ 1, meuse, meuse.grid, model=lzn.fit)
```

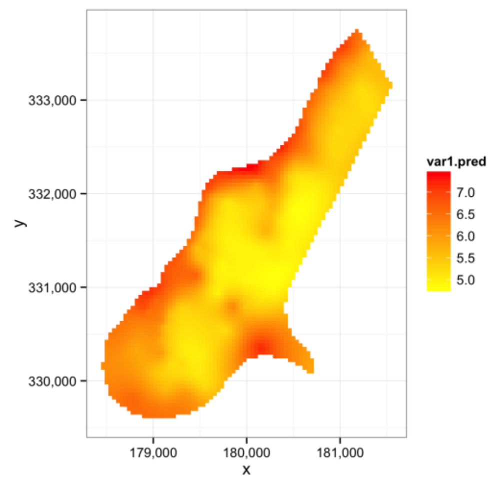
Points with measurements



Points at which to estimate



Source: https://rpubs.com/nabilabd/118172



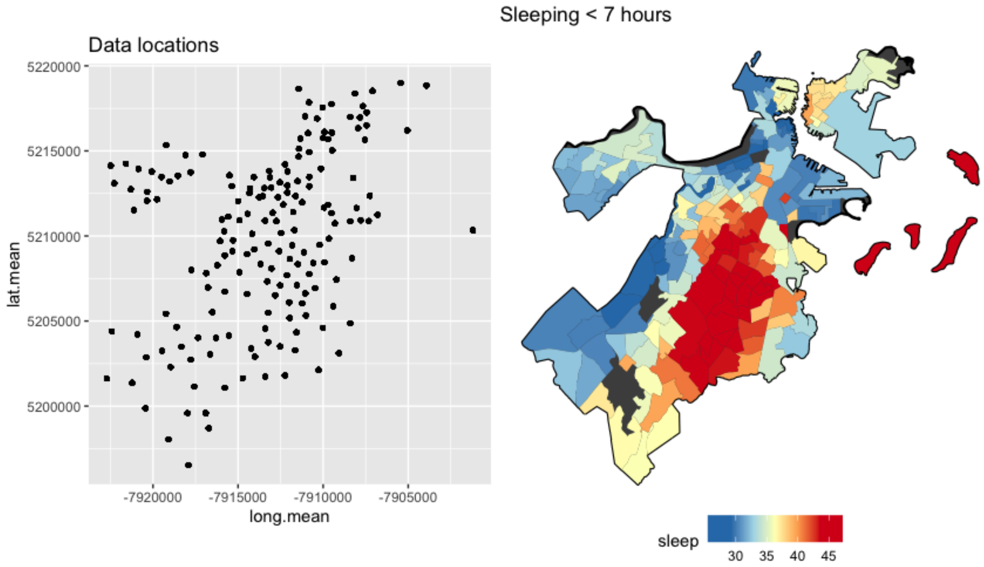
Source: https://rpubs.com/nabilabd/118172

4. Spatial Smoothing for Discrete Locations

4. Spatial Smoothing for Discrete Locations

4.1 Motivation

Sometimes, spatial information is only available at discrete locations / regions.
Methods for continuous spatial data cannot always be applied in this case.

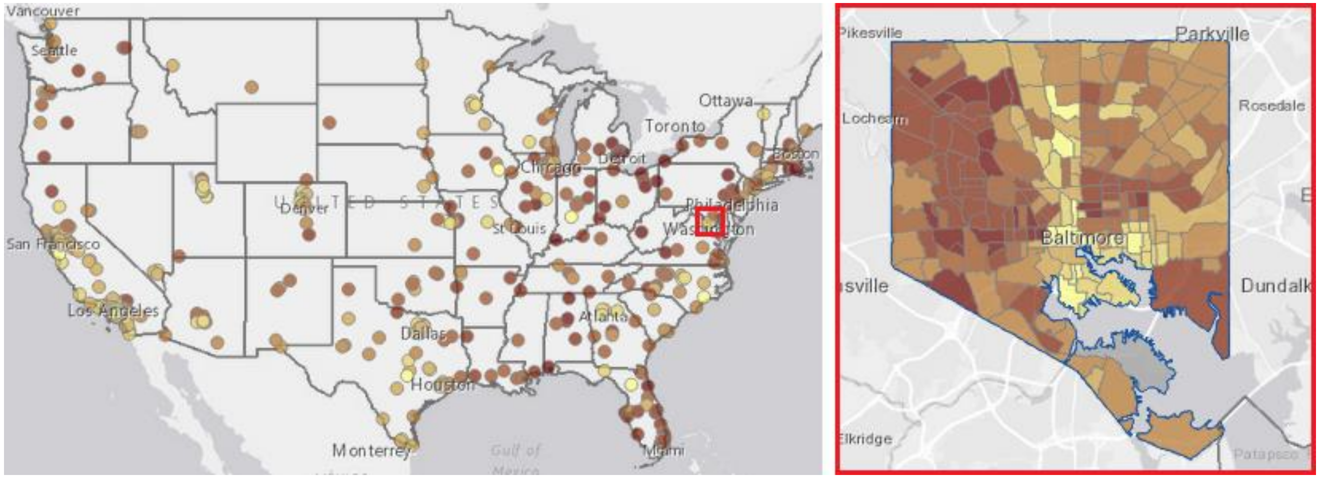


Prevalence of sleeping < 7 hours in the city of Boston. Data obtained from the 500 cities dataset.

4. Spatial Smoothing for Discrete Locations

4.2 500 cities dataset

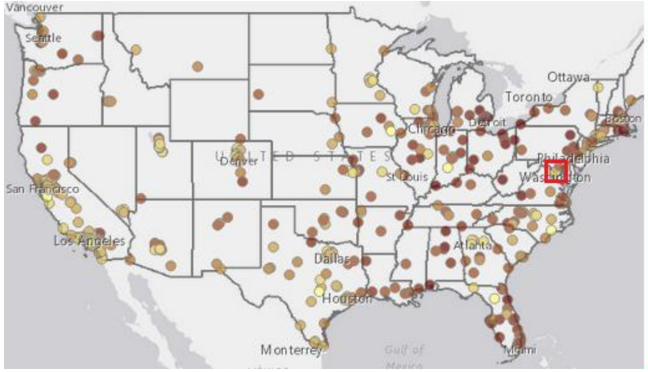
The Spotify dataset alone is not adapted for investigating discrete spatial smoothing algorithms. Therefore, we are going to use the [500 cities dataset] containing health-related statistics at the city level for the 500 largest cities in the US.

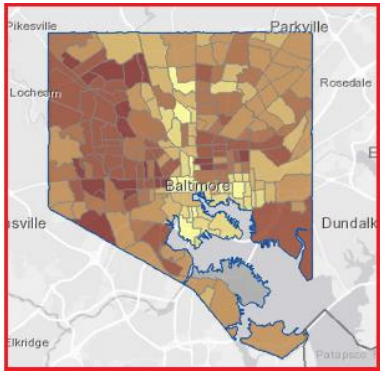


4. Spatial Smoothing for Discrete Locations

4.2 500 cities dataset

- The full dataset has 27,210 rows and 63 columns
- Most important features: Geolocation, StateAbbr, PlaceName, Place_TractID, Population
- Health-related prevalence statistics, including smoking, stroke, cancer, high blood pressure, …
- Data aggregated for small area levels (census tracts)
- Data contains **model-based estimates**



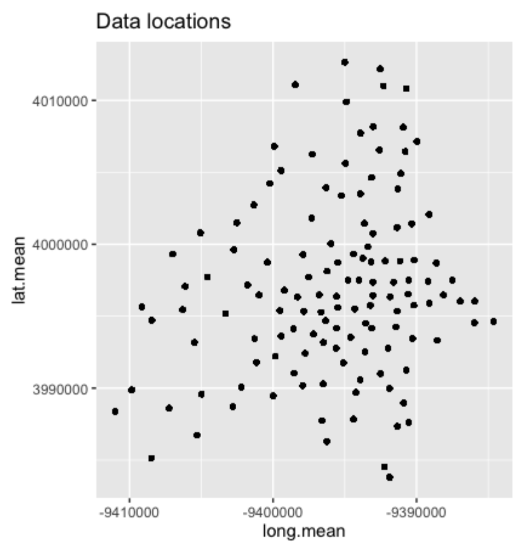


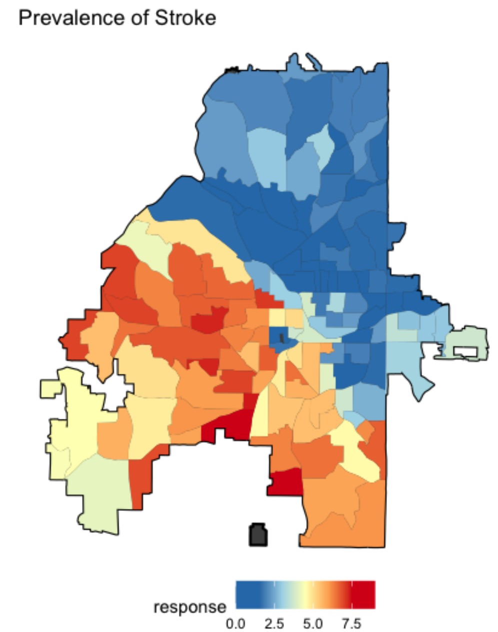
4. Spatial Smoothing for Discrete Locations

4.2 500 cities dataset

- In this example we are looking at the **prevalence of stroke** for the city of Atlanta, Georgia.
- Data has been collected over 137 different census areas.
- Out of this, 7 contain missing data for stroke prevalence, the geolocations of the remaining 130 are plotted below.

Goal: Apply smoothing on the data in order to obtain more robust predictions and clearly visible trends.



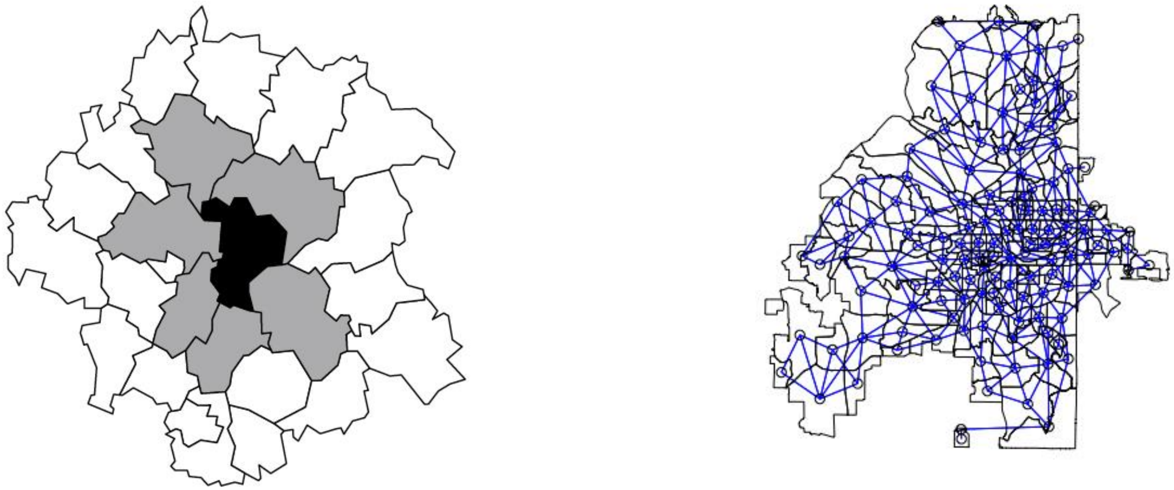


4. Spatial Smoothing for Discrete Locations

4.3 Smoothing with Spatial Neighbourhoods

In case of discrete spatial data, it is not that straightforward to calculate distances between two locations compared to continuous data. Therefore, we use the **neighbourhood structure** of the regions to assign a spatial structure to our model.

Notation: $s \sim r$ or $r \in N(s)$ if the regions s and r are neighbours.



4. Spatial Smoothing for Discrete Locations

4.3 Smoothing with Spatial Neighbourhoods

Every region is assigned its own regression coefficient

$$f_{geo}(s) = \gamma_s, \qquad s = 1, \dots, d.$$

Our model is obtained by minimizing the following PLS criterion:

$$PLS(\lambda) = \sum_{i=1}^{n} (y_i - f_{geo}(s_i))^2 + \lambda \sum_{s=2}^{d} \sum_{\substack{r \in N(s) \\ r < s}} (\gamma_r - \gamma_s)^2$$

4. Spatial Smoothing for Discrete Locations

4.3 Smoothing with Spatial Neighbourhoods

In compact form, we can write

$$\boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{Z}[i,s] = I(y_i\ was\ observed\ in\ region\ s)$, and $\boldsymbol{\gamma} = (f_{geo}(s_1), \dots, f_{geo}(s_d))'$.
Moreover, the penalty term can be written as $\lambda\boldsymbol{\gamma}'\boldsymbol{K}\boldsymbol{\gamma}$, where

$$\boldsymbol{K}[s,r] = \begin{cases} -1 & s \neq r, r \in N(s) \\ 0 & s \neq r, r \notin N(s) \\ |N(s)| & s = r \end{cases}$$

From this, we obtain the PLS estimate $\hat{\boldsymbol{\gamma}} = (\boldsymbol{Z}'\boldsymbol{Z} + \lambda\boldsymbol{K})^{-1}\boldsymbol{Z}'\boldsymbol{y}$.

4. Spatial Smoothing for Discrete Locations

4.4 Markov Random Fields

The smoothing model based on spatial neighbourhoods can be also interpreted in a Bayesian context $\Rightarrow$ **Markov Random Fields (RMFs).**

The base idea is that the conditional distribution of γ_s given all other effects $\gamma_r, r \neq s$ should **only depend on its neighbours**. If we assume that all conditional distributions are normal, we arrive at the following formulation:

$$\gamma_s \mid \gamma_r \sim N\left(\frac{1}{|N(s)|}\sum_{r:r\sim s}\gamma_r, \frac{\tau^2}{|N(s)|}\right) \qquad \text{for } r \sim N(s)$$

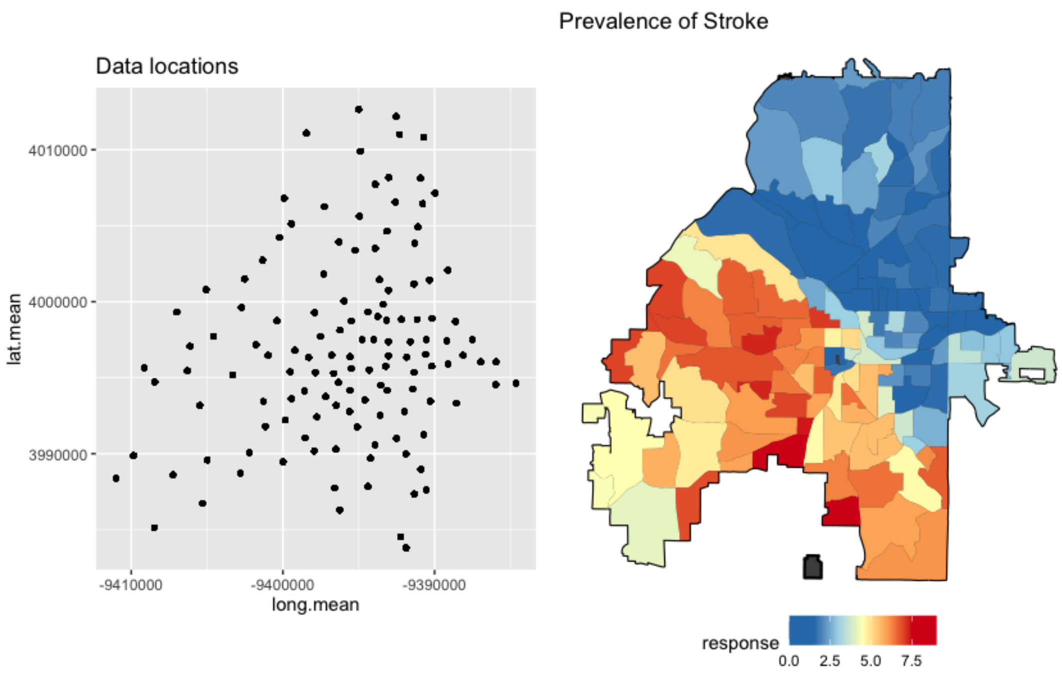
From this, we obtain

$$p(\boldsymbol{\gamma}|\tau^2) \propto \left(\frac{1}{\tau^2}\right)^{(d-1)/2} \exp\left(-\frac{1}{2\tau^2}\boldsymbol{\gamma}'\boldsymbol{K}\boldsymbol{\gamma}\right)$$

4. Spatial Smoothing for Discrete Locations

4.4 R Code for MRFs

- Recap: we would like to apply smoothing on the data containing **prevalence of stroke** for the city of Atlanta, Georgia.
- 137 different census areas, 130 with actual data.



4. Spatial Smoothing for Discrete Locations

4.5 R Code for MRFs

Detailed spatial information about the regions are contained in the **map** object.

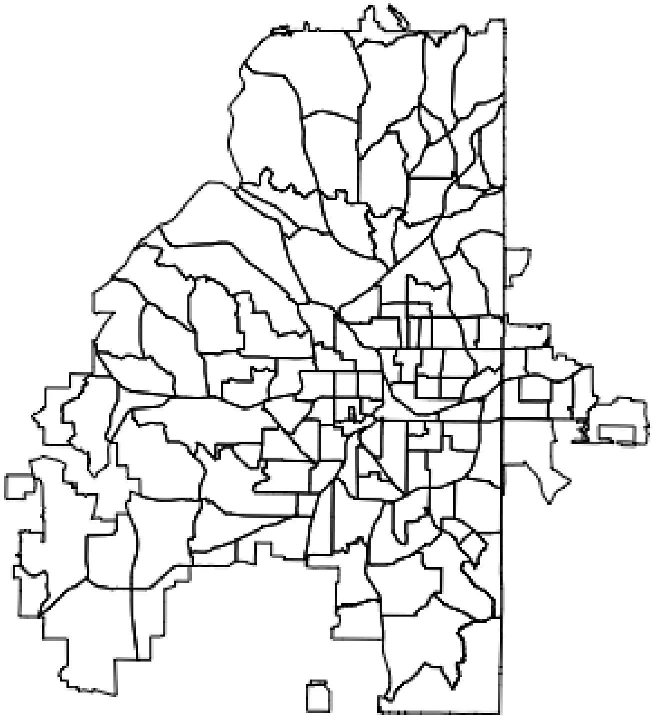
```
> glimpse(map)
Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
  ..@ data       :'data.frame': 137 obs. of  6 variables:
  .. ..$ place2010 : chr [1:137] "1304000" "1304000" "1304000" "1304000" ...
  .. ..$ tract2010 : chr [1:137] "13089020100" "13089020200" "13089020300" "13089020400" ...
  .. ..$ ST        : chr [1:137] "13" "13" "13" "13" ...
  .. ..$ PlaceName : chr [1:137] "Atlanta" "Atlanta" "Atlanta" "Atlanta" ...
  .. ..$ plctract10: chr [1:137] "1304000-13089020100" "1304000-13089020200" "1304000-13089020300"
  .. ..$ PlcTrPop10: chr [1:137] "1492" "1943" "3574" "2376" ...
  ..@ polygons   :List of 137
  .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
  .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
  .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
  .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
  .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
  .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
```

4. Spatial Smoothing for Discrete Locations

4.5 R Code for MRFs

Detailed spatial information about the regions are contained in the **map** object.

```
> plot(map)
```

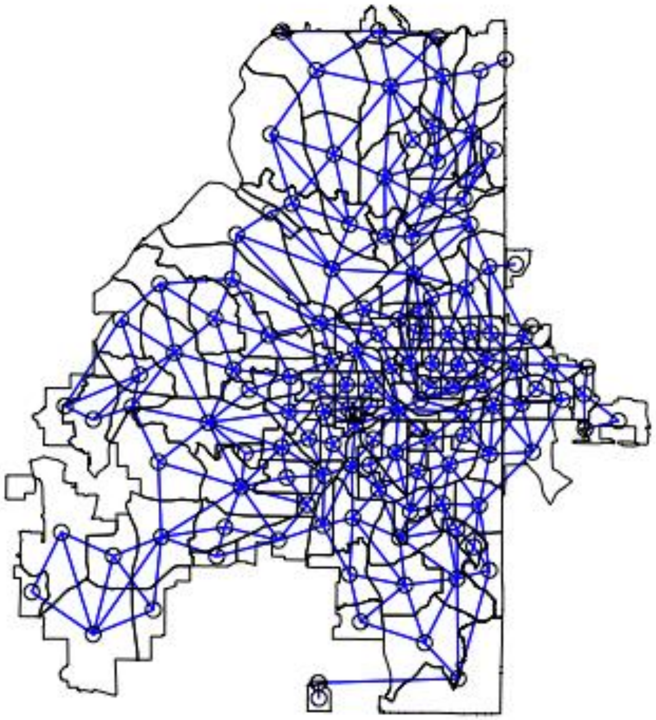


4. Spatial Smoothing for Discrete Locations

4.5 R Code for MRFs

Defining and plotting the neighbourhood graph of the areas.

```
> nb <- poly2nb(map, row.names = 1:N_original
> coords <- coordinates(map)
> plot(nb, coords=coords, col="blue")
> plot(map, add=TRUE)
```



4.5 R Code for MRFs

Fitting MRF using the *gam* function of the *mgcv* library.

```
> library(mgcv)
> mrf.m1 <- gam(response ~ s(id, bs = 'mrf', xt = list(nb = nb), sp = 1), # define MRF smooth
+               data = df_joined_aggr,
+               method = 'REML')
> mrf.m1

Family: gaussian
Link function: identity

Formula:
response ~ s(id, bs = "mrf", xt = list(nb = nb), sp = 1)

Estimated degrees of freedom:
104  total = 105.21

REML score: 203.8385
```

Coefficient of penalty term in

$$PLS(\lambda) = \sum_{i=1}^{n} (y_i - f_{geo}(s_i))^2 + \lambda \sum_{s=2}^{d} \sum_{\substack{r \in N(s) \\ r<s}} (\gamma_r - \gamma_s)^2$$

4. Spatial Smoothing for Discrete Locations

4.5 R Code for MRFs

```
> library(mgcv)
> mrf.m2 <- gam(response ~ s(id, bs = 'mrf', xt = list(nb = nb), k = 30), # define MRF smooth
+               data = df_joined_aggr,
+               method = 'REML')
> mrf.m2

Family: gaussian
Link function: identity

Formula:
response ~ s(id, bs = "mrf", xt = list(nb = nb), k = 30)

Estimated degrees of freedom:
23.8  total = 24.76

REML score: 210.1403
```

Dimension of basis used to represent the smooth terms

4. Spatial Smoothing for Discrete Locations

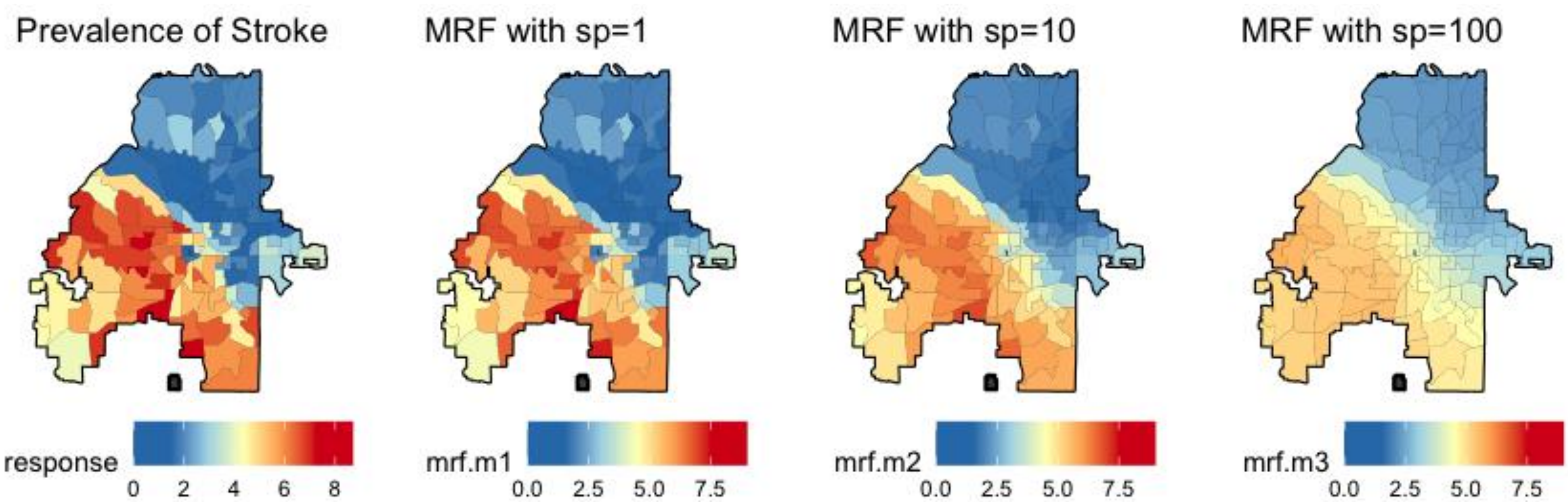
4.5 R Code for MRFs

```
> summary(mrf.m1)

Family: gaussian
Link function: identity

Formula:
response ~ s(id, bs = "mrf", xt = list(nb = nb), sp = 1)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.75149    0.03428   109.4   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
        edf Ref.df     F p-value
s(id) 104.2    129 29.48  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.967   Deviance explained = 99.3%
-REML = 203.84  Scale est. = 0.15749   n = 134
```

4. Spatial Smoothing for Discrete Locations

4.5 R Code for MRFs

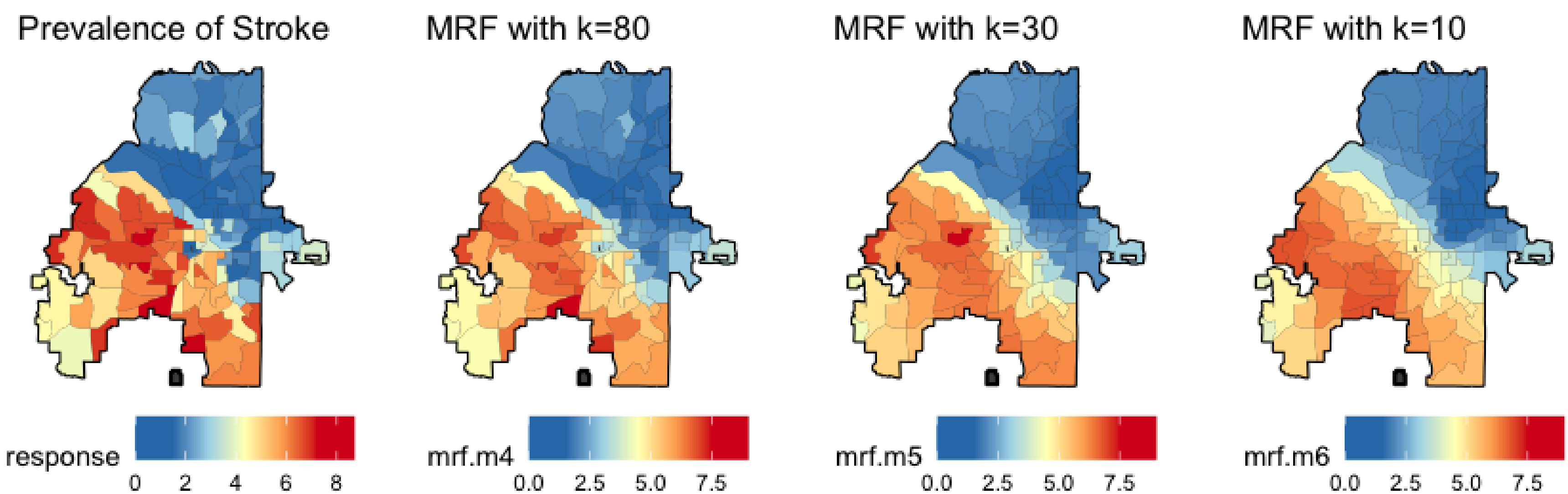
MRF smoothing using the *sp* parameter:



4. Spatial Smoothing for Discrete Locations

4.5 R Code for MRFs

MRF smoothing using the *k* parameter:



4. Spatial Smoothing for Discrete Locations

4.5 R Code for MRFs

sp parameter

	$sp = 1$	$sp = 10$	$sp = 100$
df	104	44.8	9.46
Deviance explained	99.3%	90.3%	67.2%
R-sq. (adj)	0.967	0.854	0.646

k parameter

	$k = 80$	$k = 30$	$k = 10$
df	58.6	23.8	8.6
Deviance explained	94.2%	84.8%	78.3%
R-sq. (adj)	0.895	0.815	0.768

Literature

[1] L. Fahrmeir et.al (2013). Regression – Models, Methods and Applications. Springer Berlin Heidelberg

[2] Earl Duncan (2020). Publicly Available Spatial Data Sets in Health Research, v1.2 (29 March 2020), https://rpubs.com/Earlien/Publicly_Available_Spatial_Data_Sets

[3] Centers for Disease Control and Prevention. Share 500 Cities: Census Tract-level Data (GIS Friendly Format), 2019 release, https://chronicdata.cdc.gov/500-Cities-Places/500-Cities-Census-Tract-level-Data-GIS-Friendly-Fo/k86t-wghb

[4] Spatial interpolation in R, Humboldt-Universität zu Berlin, Geography Department. https://pages.cms.hu-berlin.de/EOL/gcg_quantitative-methods/Lab14_Kriging.html

[5] https://rpubs.com/nabilabd/118172

[6] Nikolaou et.al (2023). High-resolution spatiotemporal modeling of daily near-surface air temperature in Germany over the period 2000–2020, Environmental Research, Volume 219, 2023, 115062, ISSN 0013-9351, https://doi.org/10.1016/j.envres.2022.115062. Figure 1

[7] Powell, C.E. (2013). Generating Realisations of Stationary Gaussian Random Fields by Circulant Embedding. Figure 3

[8] First steps with MRF smooths. Gavin Simpson, 19 October 2017. https://fromthebottomoftheheap.net/2017/10/19/first-steps-with-mrf-smooths/