Statistical inference for spatio-temporal models

Variogram-based and Likelihood-based

Denis Allard, Liliane Bel, Édith Gabriel, **Thomas Opitz**, Éric Parent

Workshop "An introduction to geostatistical analysis of spatio-temporal data with R"

METMA IX, Montpellier (France)

12 June 2018

Why statistical inference of parametric models ?

- characterize data generating process through a **small to moderate number of interpretable parameters**
- **predict** over space and time
- assess prediction **uncertainty**

Scope of this part of the tutorial

Key aspect will be **dependence of data over space (S) and time (T)**. For implementation, focus on small number of R packages with powerful ST-specific features.

- **stationary ST models**
  - Gaussian ST random field
  - parametric
  - non separability of S and T ?
    separable $\Rightarrow Z(s_1, t_1)$ independent of $Z(s_2, t_2)$ conditional on $Z(s_1, t_2)$
- **parameter inference**
  - least-squares-based, contrasting empirical and model ST variogram
  - (composite) likelihood-based

Unfortunately, time is limited ...

- ~~non-gaussian dependence~~
- ~~hierarchical modeling~~
- ~~Bayesian approaches~~

Pretransformation to stationary Gaussian setup

In the following,
we will assume a **stationary Gaussian model** with **stationary mean** $\mu(s) \equiv \mu$.

- pre-transform margins to make data more **"Gaussian-like"** :
  log-transform positive data, or Box–Cox transform for better symmetry around mean,...
  Example : `boxcox` in `MASS` package
- use stationary model for **residuals** $\tilde{Z}(s,t)$ of a **regression model**, e.g.

$$Z(s,t) \sim \texttt{covariates}, \quad \tilde{Z}(s,t) = Z(s,t) - \hat{Z}(s,t)$$

1. fit regression with `R` functions such as `lm(...)`, `glm(...)`, `gam(...)`
2. extract fitted values $\hat{Z}(s,t)$
3. calculate residuals $\tilde{Z}(s,t)$ (+ make them more Gaussian-like)
4. fit stationary model to $\tilde{Z}(s,t)$

$\Rightarrow$ final model : combine

regression component (trend) + stationary component (dependence)

Spatio-temporal Gaussian model

Gaussian ST random field

$$Z(s,t) = \mu(s,t) + \sigma(s,t)Z^*(s,t), \quad s \in \mathbb{R}^2, t \geq 0$$

with

- **mean surface** $\mu(s,t)$, here $\mu(s,t) \equiv \mu$
- **variance surface** $\sigma^2(s,t)$, here $\sigma(s,t) \equiv \sigma > 0$
- standard Gaussian space-time field $Z^*(s,t)$
  - $\mathbb{E}Z^*(s,t) = 0$, $\mathbb{V}(Z^*(s,t)) = 1$
  - **space-time correlation function** $\mathrm{Cor}((s_1,t_1),(s_2,t_2))$, here $\mathrm{Cor}(s,t)$ with $s = s_2 - s_1$, $t = t_2 - t_1$
    $\rightsquigarrow$ must be **nonnegative definite :**
    $\sum_{1 \leq i,j \leq d} a_i a_j \mathrm{Cor}((s_i,t_i),(s_j,t_j)) \geq 0$ for any vector $\boldsymbol{a} = (a_1,\ldots,a_d)^T$
  - model parameters govern dependence strength (S,T), measurement errors and nugget effects, anisotropy, nonseparability, ...

Observations and densities

- **observation** $\boldsymbol{z} = (z(s_1, t_1), \ldots, z(s_d, t_d))^T$
- **typical ST scenario :** $d = \#\{\text{observation sites}\} \times \#\{\text{observation times}\}$
- d-variate Gaussian density of $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$:

$$\varphi(\boldsymbol{z}; \boldsymbol{\mu}, \Sigma) = |\Sigma|^{-1/2} (2\pi)^{-d/2} \exp\left(-0.5(\boldsymbol{z} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{z} - \boldsymbol{\mu})\right),$$

  with
  - mean vector $\boldsymbol{\mu} = (\mu, \ldots, \mu)^T$
  - covariance matrix Σ based on **stationary ST covariance function** C :
    $\Sigma_{i_1, i_2} = C(s_{i_2} - s_{i_1}, t_{i_2} - t_{i_1}) = \sigma^2 \operatorname{Cor}(s_{i_2} - s_{i_1}, t_{i_2} - t_{i_1})$

Parametric ST covariance models

Usually based on **combining standard stationary covariance models** :

$$C_S(s) = \sigma_S^2 \mathrm{Cor}_S(s), \quad C_T(t) = \sigma_T^2 \mathrm{Cor}_T(t)$$

exponential $\mathrm{Cor}(x) = \exp(-\|x\|/\rho)$, powered exponential, Matérn, Cauchy, ...

- **separable** space-time models :

$$C_{ST}(s,t) = C_S(s) \times C_T(t) = \sigma_{ST}^2 \, \mathrm{Cor}_S(s) \times \mathrm{Cor}_T(t)$$

- **product-sum** (nonseparable) :

$$C_{ST}(s,t) = k \, C_S(s) C_T(t) + C_S(s) + C_T(t)$$

  parameters σ_S, σ_T, k are identifiable and $\sigma_{ST}^2 = k\sigma_S^2\sigma_T^2 + \sigma_S^2 + \sigma_T^2$
- specific nonseparable ST model classes : Gneiting, Iaco-Cesare, Porcu, ...
- **nugget effects** in space and/or time :
  $\mathrm{Cor}(x) = (1 - \mathrm{nugget}/\sigma^2)\,\mathrm{Cor}_0(x)$ for $\|x\| > 0$, with Cor_0 continuous in 0
- many types of **anisotropy** are possible

The Gneiting model [Gneiting, 2002]

$$C_{ST}(s,t) = \psi_T(t^2)^{-\delta/2} C_S\left(\|s\|^2/\psi_T\left(t^2\right)\right)$$

- $\psi_T(t) = c + \gamma_T(t)$ with $c > 0$ and variogram $\gamma_T(\cdot)$
- $C_S(\cdot)$ covariance of Gaussian scale mixture type (Matérn, Cauchy, ...)
- parameter δ ($\geq$ spatial dimension)

A flexible subclass (implemented in `CompRandFld`) :

$$C_{ST}(s,t) = \sigma^2 g_T(t)^{-1} \exp\left(-\frac{d_S(s)}{g_T(t)^{0.5\eta\kappa_S}}\right)$$

- $0 \leq \eta \leq 1$ **nonseparability parameter** (separable if $\eta = 0$)
- **power variograms** $d_T(t) = (|t|/\tau_T)^{\kappa_T}$ and $d_S(s) = (\|s\|/\tau_S)^{\kappa_S}$
- $g_T(t) = 1 + d_T(t)$ and $g_S(s) = 1 + d_S(s)$

R packages : How to create models ?

- RandomFields
  - near endless range of available models (?RMmodel, ?RMmodelsSpacetime)
  - syntax may be somewhat technical owing to very large functionality
  - example : Gneiting model with exponential C_S and power variogram ψ_T

```
#create model object:
model=RMnsst(phi=RMexp(scale=.5,var=1),psi=RMfbm(alpha=1),delta=3)
plot(model,dim=2) #visualize model (x=space, y=time)
#define coordinates: x,y, t (25 sites, observed at 100 time points)
xcoor=runif(25);ycoor=runif(25);tcoor=seq(from=0,to=10,by=.1)
#simulate one realization:
z=RFsimulate(x=xcoor,y=ycoor,T=tcoor,model=model,n=1)
length(z$variable1)
dim(z@coords)
head(z@coords)
tail(z@coords)
```

  - product-sum model :

```
model=RMexp(proj="space",scale=5,var=2)*RMgauss(proj="time",scale=3)+
                RMmatern(proj="space",nu=1,scale=22)+
                RMexp(proj="time",scale=10,var=3)
```

  - nugget effect : RMnugget

R packages : How to create models ?

- `CompRandFld`
  - ST models are identified through character strings :
    - for instance, `"exp_cauchy"` : separable model (S exponential, T Cauchy)
    - `"gneiting"`, `"iacocesare"`, `"porcu"`, ...
    - `?Covmatrix` for a description of implemented models
  - internal parameter names : `CorrelationParam("gneiting")`
  - simulation (interface to `RandomFields`) :

```
z=RFsim(xcoor,ycoor,tcoor,corrmodel="exp_exp",
    param=list(nugget=0,mean=0,scale_s=0.3,scale_t=0.5,sill=1))$data
dim(z) # 100 x 25 here
```

R packages : How to create models ?

- gstat
  - models are R objects (but no simulation facility)
  - vgm() to get list of available standard models
  - available constructions : product-sum, metric, sum-metric...
    but no specific ST classes
  - separable model

```
> sepmod=vgmST("separable", space=vgm(psill=0.9,"Exp", range=147,nugget=0.1),
+              time =vgm(psill=0.9,"Exp", range=3.5,nugget=0.1),sill=40)
> extractParNames(sepmod)
[1] "range.s"  "nugget.s" "range.t"  "nugget.t" "sill"
```

  - product-sum model

```
> vgmST("productSum",space=vgm(0.9,"Exp",1,1), time=vgm(0.9,"Gau",1,1),k=1)
```

Covariance function C and semi-variogram γ

Variograms are more general than covariance functions since they allow handling intrinscally stationary processes (with stationary increments).

We assume **stationary**, leading to **equivalent representation** in terms of the variogram or the covariance function.

Semi-variogram of stationary ST field :

$$\gamma(s,t) \stackrel{def}{=} 0.5\mathbb{E}\left(Z(s,t) - Z(0,0)\right)^2 = \sigma^2 - C(s,t)$$

where $\sigma^2 = C(0,0)$.

Typical parameters

Assumption : ST process is stationary

Parameters are either to estimate or to be fixed a priori.

- mean μ
- variance σ^2_{ST}
- ST nugget or measurement errors : $\theta_{\text{nugget}} = \sigma^2_{ST} - \lim_{t\to 0, \|s\|\to 0} C(s,t)$
- spatial geometric anisotropy $\Delta s \rightsquigarrow \begin{pmatrix} b & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{pmatrix} \Delta s$
  - rotation $a \in [0, \pi)$
  - dilation $b > 0$
- correlation function : scale and shape parameters

Here : empirical estimation of μ and σ^2_{ST} :

- $\hat{\mu} = \frac{1}{d} \sum_{(s_i,t_i)} z(s,t)$, where $d = \#\{(s_i,t_i)\}$ (`mean` function of `R`)
- $\hat{\sigma}^2_{ST} = \frac{1}{d} \sum_{(s_i,t_i)} (z(s,t) - \hat{\mu})^2$ (`var` function of `R`)

Estimating covariance parameters : variogram-based or likelihood-based

- **variogram-based** :
  - **requires empirical ST variogram**
  - **weighted least squares** (WLS) between empirical and parametric model variogram
- **likelihood-based**
  - **maximum likelihood** (ML) : estimate parameter vector $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{z})$$

    with likelihood $\boldsymbol{\theta} \mapsto \ell(\boldsymbol{\theta}; \boldsymbol{z}) = \varphi(\boldsymbol{z}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$
  - **tapering** [Furrer et al., 2006, Kaufman et al., 2008, Stein, 2013] for reducing computational cost
    - use $\tilde{C}(s,t) = C_(s,t) \times C_0(s,t)$ with C_0 compactly supported (Wendland, ...) $\rightsquigarrow$ "sparse" Σ
    - performance may by disappointing in practice
  - **maximum composite likelihood**, e.g. **pairwise likelihood** (PL)

Estimating covariance parameters : variogram-based or likelihood-based

- **variogram-based** :
  - **requires empirical ST variogram**
  - **weighted least squares** (WLS) between empirical and parametric model variogram
- **likelihood-based**
  - **maximum likelihood** (ML) : estimate parameter vector $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{z})$$

    with likelihood $\boldsymbol{\theta} \mapsto \ell(\boldsymbol{\theta}; \boldsymbol{z}) = \varphi(\boldsymbol{z}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$
  - **tapering** [Furrer et al., 2006, Kaufman et al., 2008, Stein, 2013] for reducing computational cost
    - use $\tilde{C}(s,t) = C_(s,t) \times C_0(s,t)$ with C_0 compactly supported (Wendland, ...) $\rightsquigarrow$ "sparse" Σ
    - performance may by disappointing in practice
  - **maximum composite likelihood**, e.g. **pairwise likelihood** (PL)

Weighted least squares between empirical and model variogram

Define set of **distance classes**

$$D_k = \{(s_i, t_i) : \mathrm{dist}((s_i, t_i), (\tilde{s}_k, \tilde{t}_k)) \leq \varepsilon_k\}, \quad k = 1, \ldots, K$$

(typically a space-time grid of class barycenters)

WLS estimator :

$$\hat{\boldsymbol{\theta}}_{WLS} = \arg\min_{\theta} \sum_{k=1}^{K} \omega_k \left|\gamma_{\boldsymbol{\theta},k} - \hat{\gamma}_k\right|^2$$

- $\omega_k = 1$ ordinary least squares
- $\omega_k = |D_k|/\gamma^2_{\boldsymbol{\theta},k}$: correct for variance of $\hat{\gamma}_k$
- overweight small ST distances, ... (similar to tapering, weighted PL)

Maximum composite likelihood

[Lindsay, 1988, Varin et al., 2011]

Pairwise likelihood if based on bivariate distributions :

$$\mathrm{PL}(\boldsymbol{\theta}; \boldsymbol{z}) = \prod_{i_1, i_2} f_{\boldsymbol{\theta}, i_1, i_2}(z_{i_1}, z_{i_2})$$

maximum pairwise likelihood : $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \Theta} \mathrm{PL}(\boldsymbol{\theta}; \boldsymbol{z})$

- $f_{\theta, i_1, i_2} \in$ density of $(Z(s_1, t_1), Z(s_2, t_2))^T$, $Z(s_1, t_1) - Z(s_2, t_2)$ or $Z(s_1, t_1) \mid Z(s_2, t_2)$
- **good asymptotic properties** : consistency, normality, unbiasedness $\Rightarrow$ calculation of standard errors, hypothesis testing ...
- reduce computational burden by **using only pairs that are "close" in space and time**, which may further reduce bias in certain cases

Model selection

Often, we have to **choose between several (fitted) covariance models**.

Techniques :

- visual comparison **"fitted vs. empirical" for covariance function**
- **cross-validation**
- **information criteria such as CLIC** (Composite Likelihood Information Criterion)
- nested models : likelihood-based **statistical hypothesis tests** (with adaptations to composite likelihood)

Selected R packages at a glance

	RandomFields	CompRandFld	gstat
Strengths	**models, simulation**	**statistical inference**	**data structures, exploration**
spacetime R objects	–	–	yes
simulation	RFsimulate	RFsim	(predict.gstat)
Σ_θ	RFcovmatrix	Covmatrix	(variogramLine)
emp. ST variogram	RFempiricalvariogram	EVariogram	variogramST
ML	RFfit(?)	FitComposite	–
PL	–	FitComposite	–
WLS	RFfit(?)	WLeastSquare	fit.StVariogram
standard errors	RFfit(?)	FitComposite	–
information criteria	AIC	AIC, CLIC	–
CV	RFcrossvalidate	–	krige.cv
kriging	RFinterpolate	Kri	krigeST
test nested models	–	HypoTest	–
separable	yes	yes	yes
product-sum, ...	(yes)	–	yes
Gneiting	RMnsst	yes	–
spatial anisotropy	yes	–	(partial)
tapering	–	yes	–

Other packages with less general functionality or different approaches : rgeos (spatial), SpatioTemporal ...

⚠ **weak interpackage compatibility** (data structures, models)
except for spacetime/gstat

Data structure

- **AirBase daily observation data** (PM10, O3, ...) on **irregularly spaced observation sites**
  - Europe, with $\approx$ 400 stations in France
  - ⚠ many missing data
  - station types : "background", "traffic", "industry", ...
- daily **predictions of chemical-physical CHIMERE model** (forecasts and hindcasts) over regular grid
- no other covariates (weather, urbanisation, industries, ...), but already partially integrated into the CHIMERE values

To avoid handling strong nonstationarities, we **keep only background stations**.

Modeling approach : PM10

We focus on modeling the **particulate matter concentrations** PM10.

ST Gaussian model for error correction for

$$\text{observation}(s,t) - \texttt{prediction CHIMERE}(s,t)$$

- space-time variation of mean surface is modeled through the CHIMERE values
- ⚠ different spatial supports require interpolating CHIMERE values to irregular observation sites ⇒ use bilinear spatial interpolation (`interp.surface`-function from `fields` package)

Load and select data (1/2)

- load previously created data objects : daily observations, CHM predictions, stations
- select French background stations

```
> load(paste0(DATA, 'OBS_jour.Rdata'))
> load(paste0(DATA, 'CHM.Rdata'))
> load(paste0(DATA, 'stations.Rdata'))
> IDs_bg   <- stations$station_european_code[stations$type_of_station=="Background"]
> IDs_fr   <- stations$station_european_code[stations$country_name=="France"]
> IDs_bgfr <- intersect(IDs_bg,IDs_fr)
> idx_stations <- which(stations$station_european_code %in% IDs_bgfr)
> coord    <- cbind(stations$station_longitude_deg,stations$station_latitude_deg)[idx_stat
> dim(coord) #around 700 French background stations
[1] 705   2
```

Load and select data (2/2)

- choose 3-month period (Jan-Mar 2014) for fitting models
- transform from latitude-longitude to Lambert93 projection

(using STFDF object `OBS_sel`)

```
> tstart   <- "2014-01-01"
> tend     <- "2014-03-31"
> OBS_jour <- OBS_jour[OBS_jour$ID %in% IDs_bgfr,]
> OBS_sel  <- stConstruct(OBS_jour,space=c('long','lat'),time='date',
+                         SpatialObj=SpatialPoints(OBS_jour[,c('long','lat')]))
> OBS_sel  <- as(OBS_sel,"STFDF")
> proj4string(OBS_sel)="+init=epsg:4326"
> OBS_sel  <- OBS_sel[,paste0(tstart,"::",tend)]
> OBS_sel@sp <- spTransform(OBS_sel@sp, CRS("+init=epsg:2154"))
```

Interpolate CHIMERE predictions

- interpolate from prediction grid to observation sites through simple bilinear interpolation : CHM_sites
- save residuals PM10res=(PM10 − CHM_sites) into OBS_sel

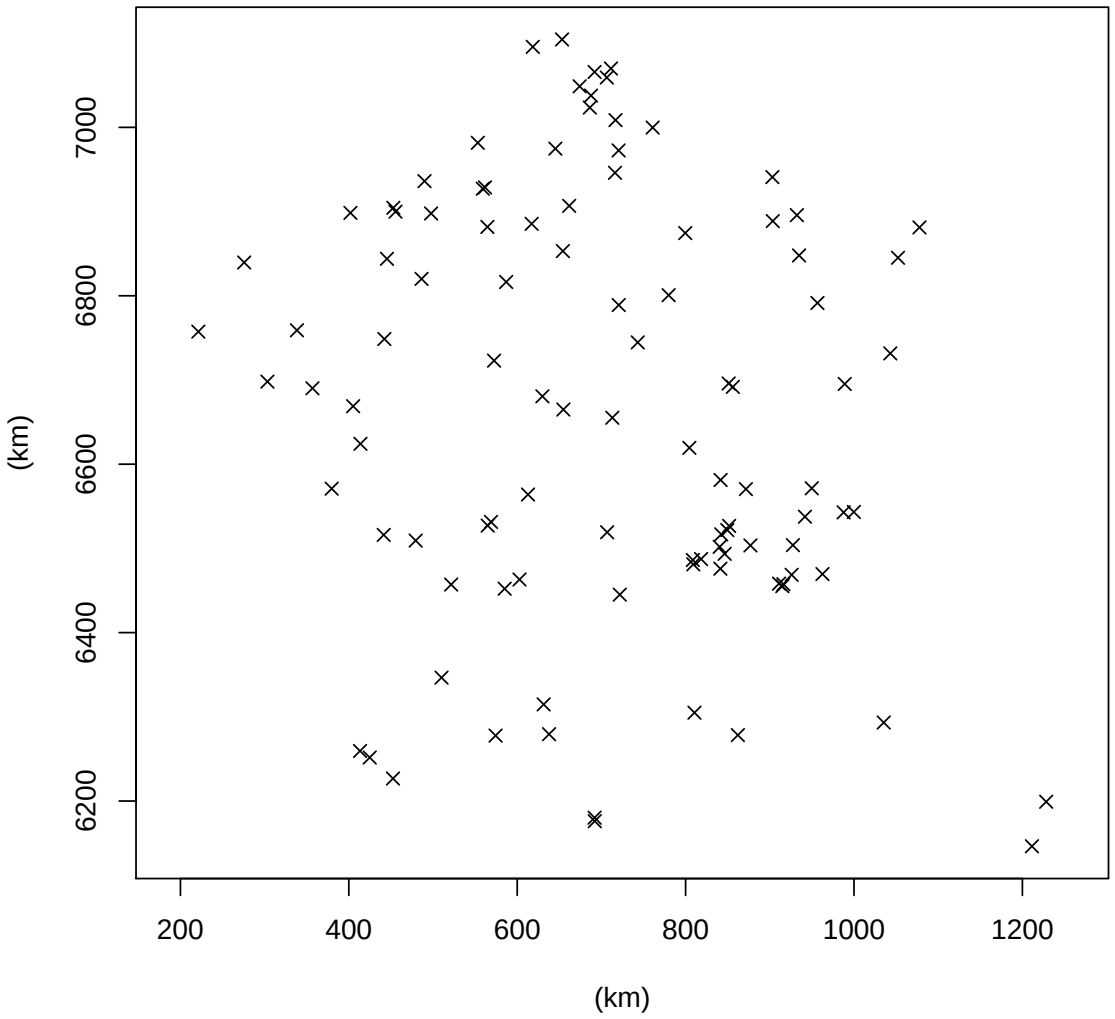
```
> library(fields)
> CHM_sel  <- CHM[as.character(CHM$time)>=tstart&as.character(CHM$time)<=tend,]
> grid_CHM <- unique(cbind(CHM_sel$lon,CHM_sel$lat))
> CHM_mat  <- matrix(CHM_sel$PM10, nrow=nrow(grid_CHM))
> dim(CHM_mat)
[1] 11211    90
> lon_grid <- sort(unique(CHM_sel$lon))
> lat_grid <- sort(unique(CHM_sel$lat))
> idx_sites_obs <- match(unique(OBS_sel@data$ID),IDs_bgfr)
> fun2interp <- function(i){
+   tmp <- list(x=lon_grid,y=lat_grid,
+               z=matrix(CHM_mat[,i],length(lon_grid),length(lat_grid)))
+   interp.surface(tmp, coord[idx_sites_obs,])
+ }
> CHM_sites <- sapply(1:ncol(CHM_mat),fun2interp)
> OBS_sel@data$CHM <- as.numeric(CHM_sites)
> OBS_sel@data$PM10res <- OBS_sel@data$PM10-as.numeric(CHM_sites)
```

Estimation with CompRandFld – Remove missing data

- need observation matrix (time × sites) **without missing values** CHM_sites

```
> data_mat <- matrix(OBS_sel@data$PM10res,ncol=dim(OBS_sel)[1],nrow=dim(OBS_sel)[2],byrow=T
> dim(data_mat)
[1]  90 378
> mean(is.na(data_mat)) #around 44 per cent of missing data
[1] 0.4398001
> data2keep=list(data_mat=data_mat,times=1:nrow(data_mat),stations=1:ncol(data_mat))
> cleandata=function(data2keep,propNA){
+   idx2keep=apply(is.na(data2keep$data_mat),1,mean) < propNA
+   data2keep$times=data2keep$times[idx2keep]
+   data2keep$data_mat  <- data2keep$data_mat[idx2keep,]
+   idx2keep=apply(is.na(data2keep$data_mat), 2, mean) < propNA
+   data2keep$stations=data2keep$stations[idx2keep]
+   data2keep$data_mat  <- data2keep$data_mat[,idx2keep]
+   data2keep
+ }
> data2keep=cleandata(data2keep,.9)
> data2keep=cleandata(data2keep,.2)
> data2keep=cleandata(data2keep,.1)
> data2keep=cleandata(data2keep,.05)
> data2keep=cleandata(data2keep,.025)
> data2keep=cleandata(data2keep,.01)
> data2keep=cleandata(data2keep,.10^{-10})
> data_mat=data2keep$data_mat
> dim(data_mat) #we keep 70 stations and 103 days
[1]  70 103
```

Estimation with `CompRandFld` – Selected sites



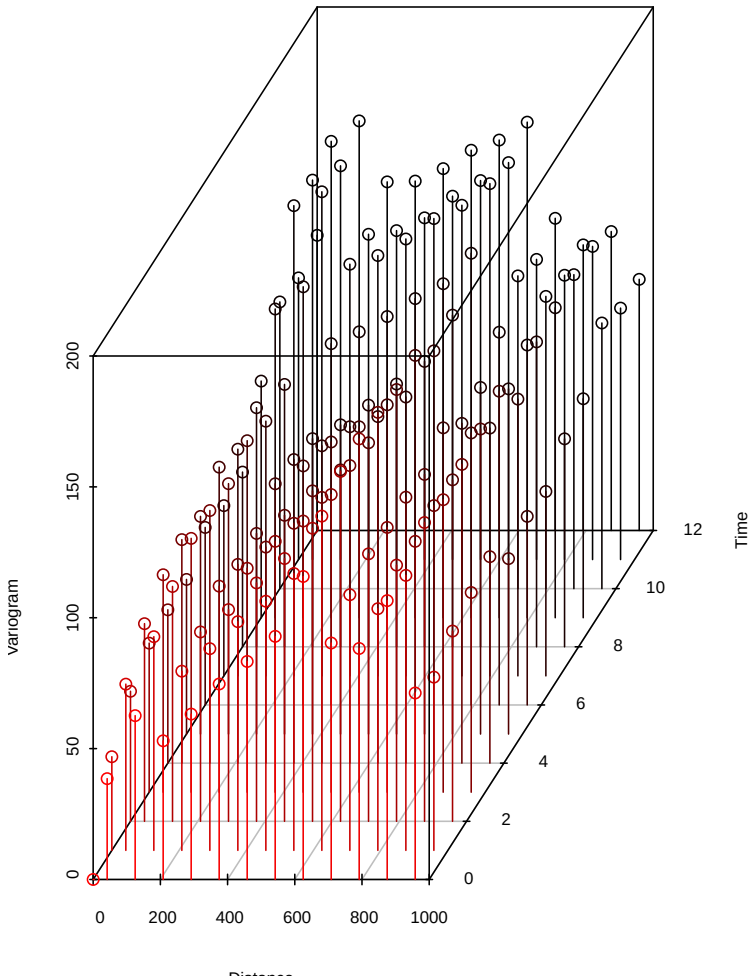
Estimation with CompRandFld – Define covariance models

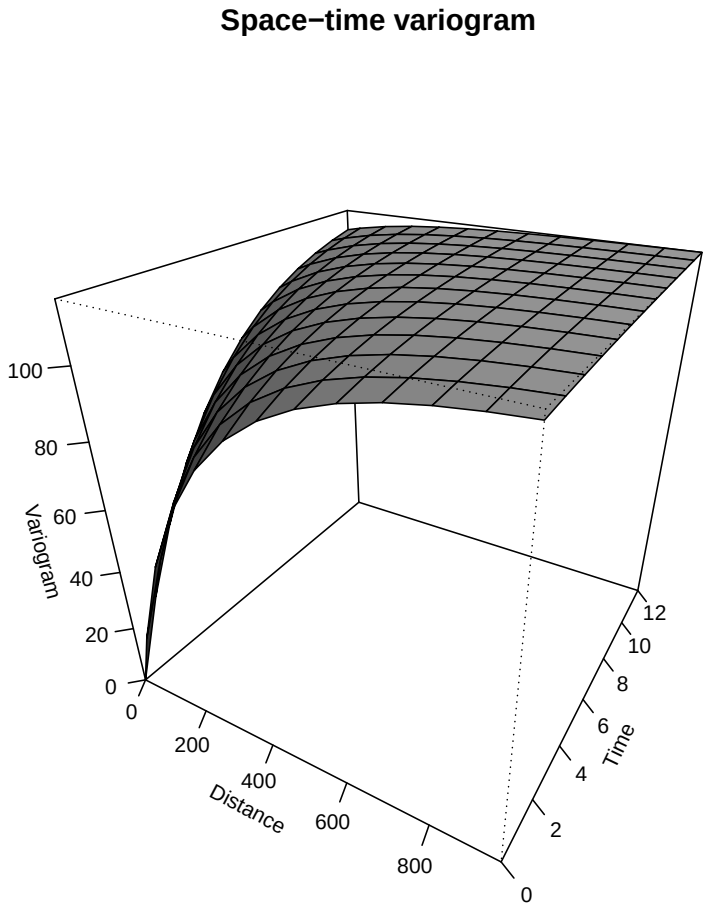
```
> mean_est <- mean(data_mat); mean_est
[1] 8.024326
> var_est  <- var(as.numeric(data_mat)); var_est
[1] 116.0013
> cormod1 <- "exp_exp"
> fixed1  <- list(mean=mean_est,nugget=0,sill=var_est)
> start1  <- list(scale_s=200,scale_t=2)
> cormod2 <- "exp_exp"
> fixed2  <- list(mean=mean_est,sill=var_est)
> start2  <- list(scale_s=200,scale_t=2,nugget=0)
> cormod3 <- "gneiting"
> fixed3  <- list(sill=var_est,mean=mean_est,nugget=0,power_s=1,power_t=1)
> start3  <- list(scale_s=200,scale_t=2,sep=.5)
> cormod4 <- "gneiting"
> fixed4  <- list(sill=var_est,mean=mean_est,power_s=1,power_t=1)
> start4  <- list(scale_s=200,scale_t=2,sep=.5)
> cormod5 <- "gneiting"
> fixed5  <- list(sill=var_est,mean=mean_est,nugget=0,power_s=.5,power_t=.5)
> start5  <- list(scale_s=200,scale_t=2,sep=.5)
> cormod6 <- "gneiting"
> fixed6  <- list(sill=var_est,mean=mean_est,power_s=.5,power_t=.5)
> start6  <- list(scale_s=200,scale_t=2,sep=.5)
> cormod7 <- "gneiting"
> fixed7  <- list(sill=var_est,mean=mean_est,nugget=0)
> cormod8 <- "gneiting"
> fixed8  <- list(sill=var_est,mean=mean_est)
```

Estimation with CompRandFld – Estimate Model 1 (PL / WLS)

```
> fit1PL <- FitComposite(data=data_mat,coordx=coord_fit,coordt=times_fit,maxdist=300,
      maxtime=4,corrmodel=cormod1,likelihood="Marginal",type="Pairwise",fixed=fixed1,
      start=start1,varest=T)
> fit1PL
[...] Maximum log-Composite-Likelihood value: -6577607.64
CLIC : 13155436
Estimated parameters:
scale_s  scale_t
238.015    4.141
Standard errors:
scale_s  scale_t
5.8847   0.1825
Variance-covariance matrix of the estimates:
scale_s    scale_t
scale_s   34.630069   0.007295
scale_t    0.007295   0.033290
> library(scatterplot3d)
> Covariogram(fit1PL,vario=vgm_emp,show.vario=T,pch=20)
> fit1WLS <- WLeastSquare(data=data_mat,coordx=coord_fit,coordt=times_fit,maxdist=300,maxt
> fit1WLS
[...] Estimated parameters:
scale_s  scale_t
306.903    3.131
> (fit1PL$param-fit1WLS$param)/pmax(abs(fit1PL$param),abs(fit1WLS$param))
scale_s    scale_t
-0.2244608  0.2439148
```

Estimation with `CompRandFld` – Plot empirical vs. fitted





Estimation with CompRandFld – Fit and compare all models

- select model with minimal composite likelihood information criterion (CLIC)
- test for nested models

```
> load(file=paste0(DATA,"fitsWLS.Rdata"))
> load(file=paste0(DATA,"fitsPL.Rdata"))
> which.min(unlist(sapply(fitsPL,getElement,"clic")))
[1] 8
> unlist(sapply(fitsPL,getElement,"clic"))
[1] 13155436 13129950 13149224 13130463 13138167 13126159 13137701 13126002
> fit8 <- fitsPL[[8]]
> fit7 <- fitsPL[[7]]
> fit5 <- fitsPL[[5]]
> HypoTest(fit8,fit7,fit5,statistic="Wald")
Num.Par Diff.Par Df     Chisq Pr(>chisq)
fit8       6       NA NA        NA         NA
fit7       5        1  1 0.9344535 0.33370837
fit5       3        2  2 8.7120541 0.01282926
```

Previous model selection criteria lead us to keep one of Models 7 or 8 :

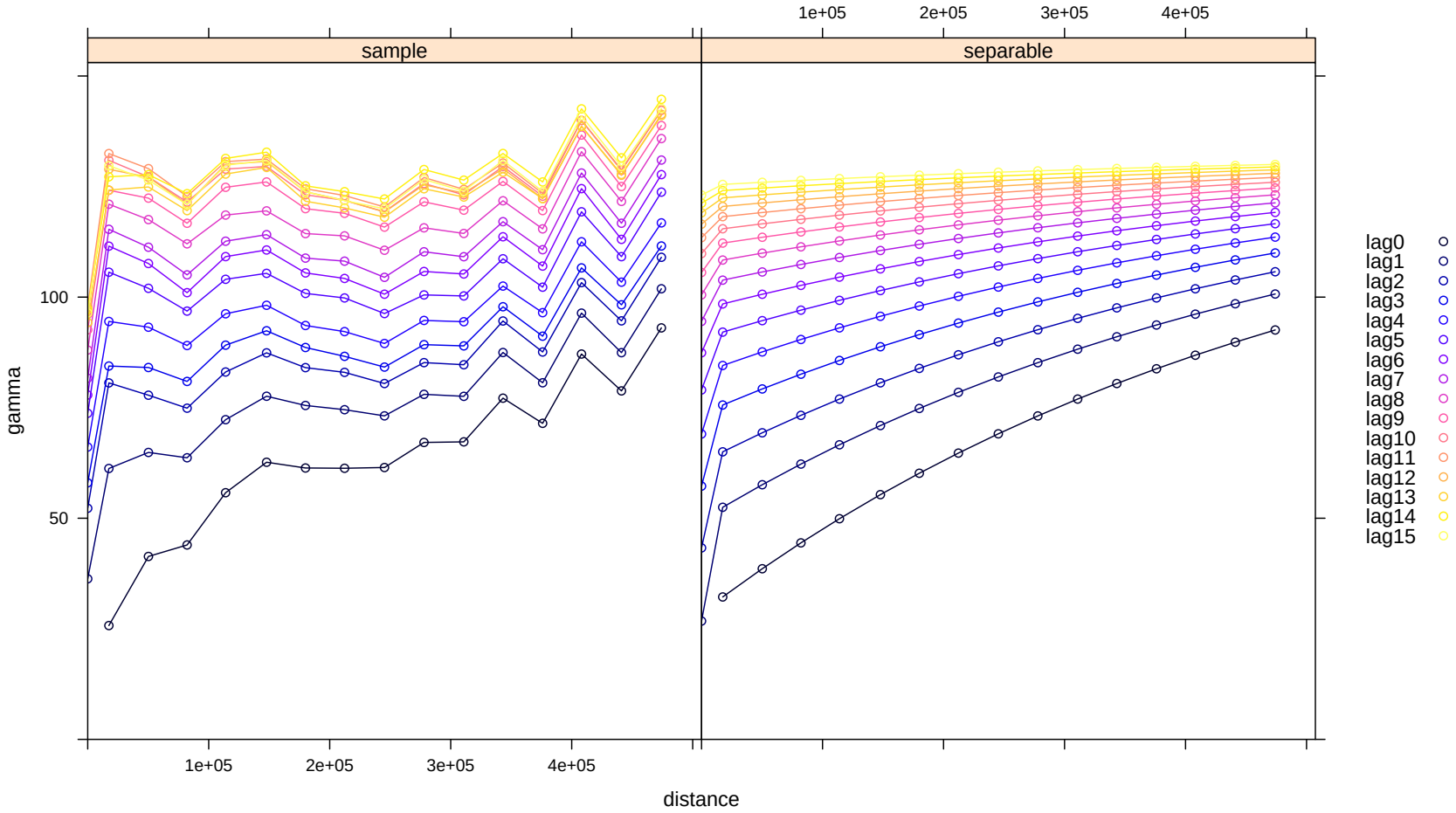
```
> fit7$param
power_s     power_t     scale_s     scale_t         sep
0.4371113   0.7314831 424.7527140   3.6254877   1.0000000
> fit8$param
nugget     power_s     power_t     scale_s     scale_t         sep
8.3616423   0.4290766   0.7774891 632.4262168   4.3469095   1.0000000
```

Estimation with gstat – WLS fitting

- direct use of spacetime data structures
- Weighted Least Squares fitting for a selection of models (separable, product-sum, metric, sum-metric ...)
- ⚠ time unit is 1 hour, space unit is 1m

```
> vario <- variogramST(PM10res~1,data=OBS_sel)
|=================================================================================| 100%
Warning message:
In variogramST(PM10res ~ 1, data = OBS_sel) :
strictly irregular time steps were assumed to be regular
> separableModel <- vgmST("separable",space=vgm(psill=0.9,"Exp",range=500000,nugget=0.1),
+                         time=vgm(psill=0.9,"Exp",range=3*24,nugget=0.1),sill=200)
> extractParNames(separableModel)
[1] "range.s"  "nugget.s" "range.t"  "nugget.t" "sill"
> separable_fit  <- fit.StVariogram(model=separableModel,object=vario)
> plot(vario,separable_fit,all=T,map=F)
```

Estimation with gstat – Plot empirical vs. fitted

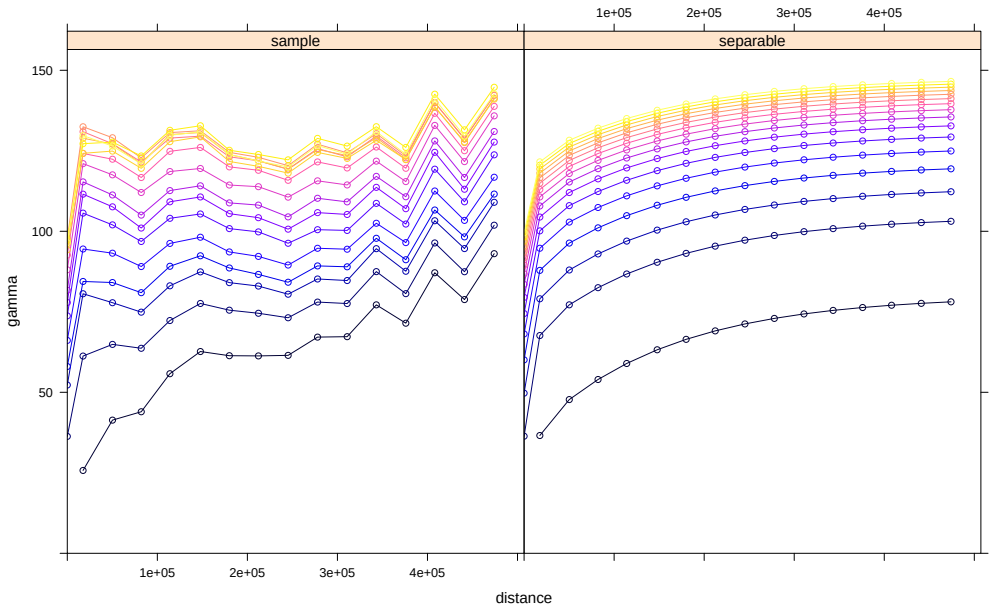


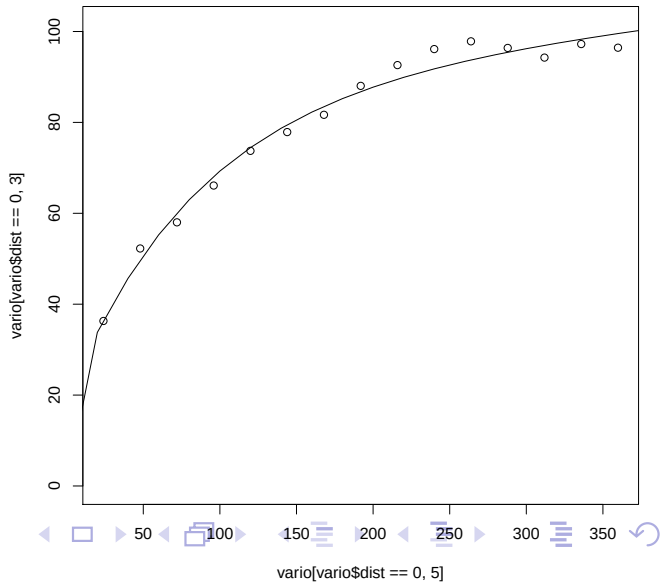
Fit does not seem to be very accurate.

Estimation with gstat – Manual "eye fit"

- convergence of parameter optimization seems questionable ...
- need to play around with initial parameters
- instead, can try manual "eye fit" :
  visual inspection for various parameter configurations to get good match

```
sill <- 250
separable_man <- vgmST("separable",space=vgm(.18,"Exp",1.5e5,0.01,
add.to=vgm(.12,"Exp",8e3,.01)),time=vgm(80/sill,"Exp",1670,19/sill,
add.to=vgm(66/sill,"Exp",85,0)),sill=sill)
plot(vario,separable_man,all=T,map=F)
variot(ProductSum_fit,vario)
```

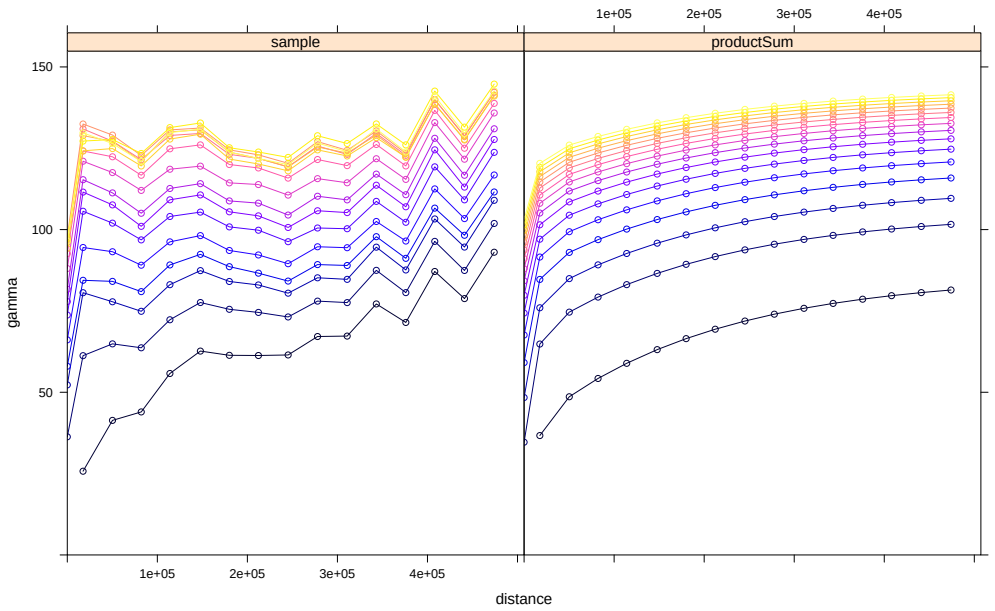


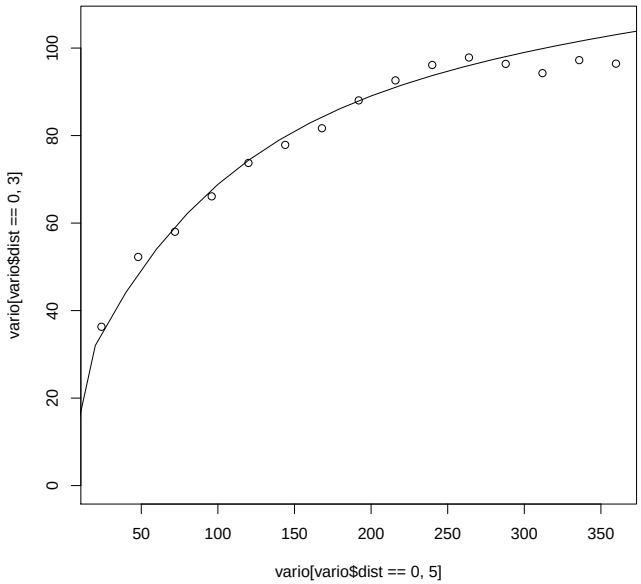


Estimation with gstat – Various other fits

- convergence seems to strongly depend on very good initial values
- well-chosen manual "eye fits" seem to provide better results (?)

```
> ProductSum_man <- vgmST("productSum",space=vgm(12.5,"Exp",2e5,0,add.to=vgm(10,"Exp",9e3,0
+                         time=vgm(35,"Exp",800,9.5,add.to=vgm(36,"Exp",85,0)),k=0.035)
> plot(vario,ProductSum_man,all=T,map=F)
> variot(ProductSum_man,vario)
```





Some take-home messages

- getting the ST dependence right is paramount to **predicting values and uncertainty over space and time**
- an ever-increasing functionality for estimating ST models is available in `R`
- interesting future extensions of packages ?
  (`NA` data in `CompRandFld`, ST estimation facilities in `RandomFields`, ...)
- currently, `CompRandFld` most complete package with respect to ST estimation
- complex ST models make parameter estimation challenging
  $\Rightarrow$ model selection tools use approximations, may sometimes mislead
  $\Rightarrow$ always **check if model fit looks "trustworthy"**
  $\Rightarrow$ compare estimation for different initial values
- there are many ways to **extend an S model to include T dependence**, **there may be several good choices**

Furrer, R., Genton, M. G., and Nychka, D. (2006).
Covariance tapering for interpolation of large spatial datasets.
J. Computnl Graph. Statist., pages 502–523.

Gneiting, T. (2002).
Nonseparable, stationary covariance functions for space–time data.
Journal of the American Statistical Association, 97(458) :590–600.

Gräler, B., Pebesma, E., and Heuvelink, G. (2016).
Spatio-temporal interpolation using gstat.
R Journal, 8(1) :204–218.

Kaufman, C. G., Schervish, M. J., and Nychka, D. W. (2008).
Covariance tapering for likelihood-based estimation in large spatial data sets.
Journal of the American Statistical Association, 103(484) :1545–1555.

Lindsay, B. G. (1988).
Composite likelihood methods.
Contemporary mathematics, 80(1) :221–39.

Padoan, S. A. and Bevilacqua, M. (2015).
Analysis of random fields using CompRandFld.

Stein, M. L. (2013).
Statistical properties of covariance tapers.
Journal of Computational and Graphical Statistics, 22(4) :866–885.

Varin, C., Reid, N., and Firth, D. (2011).
An overview of composite likelihood methods.
Statistica Sinica, pages 5–42.