Environmental Data Analysis
Part II: Time series and Spatial Statistics

Denis Allard[1]

Biostatistique et Processus Spatiaux (BioSP), INRA, Avignon
http://informatique-mia.inra.fr/biosp/content/homepage-denis-allard

Doctoral program in Environmental Sciences
Università Ca' Foscari Venice
2016-2017

[1] With the help of Carlo Gaetan (Ca' Foscari) who allowed me to use some material from his lecture notes

Unit 3
An introduction to time series and their analysis

What is a time series ?

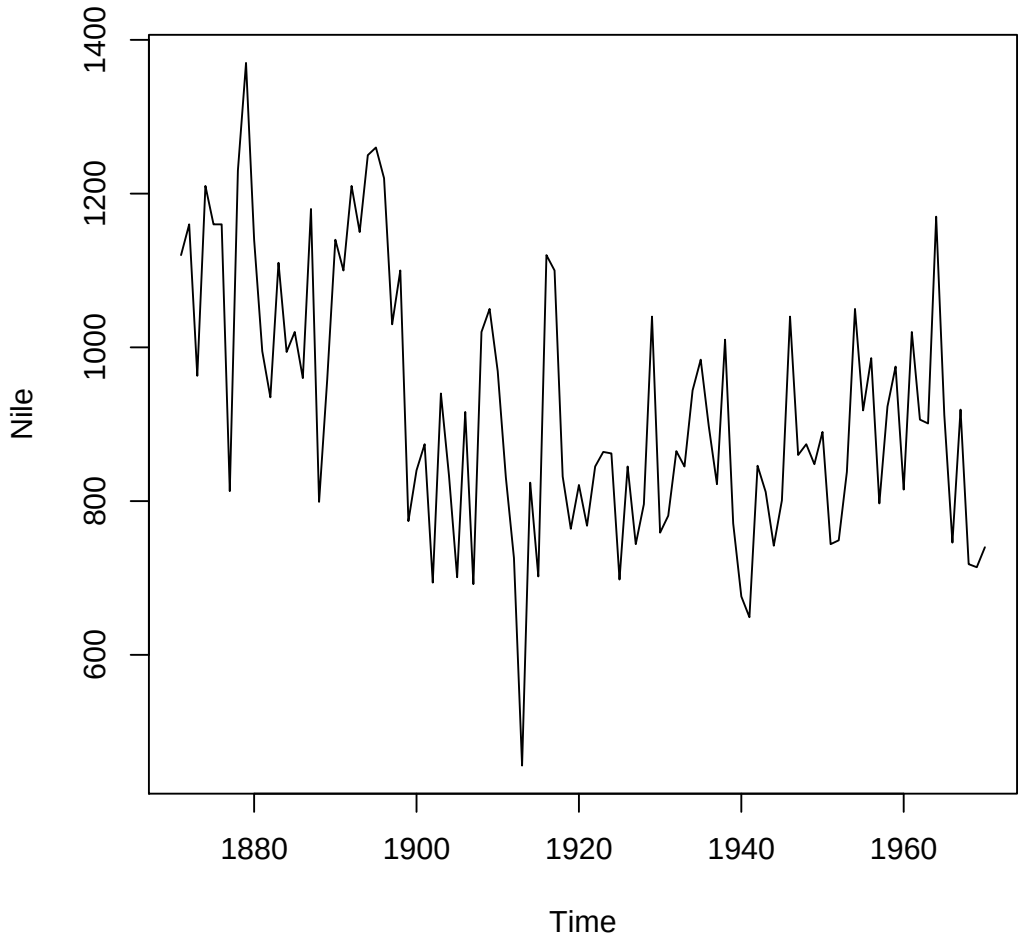
- A time series (ts) is a set of observations taken sequentially in time
- A ts can be represented as a set of pairs

$$\{(t, y_t) \text{ with } t \in \{t_1, t_2, t_3 \ldots, t_n\} \text{ and } y \in \{y_1, y_2, y_3, \ldots, y_n\}$$

- Time can be:
  - Equally spaced; $t = \{1, 2, 3, 4, 5\}$
  - Equally spaced with missing values; $t = \{1, 2, 4, 5, 6\}$
  - Unequally spaced; $t = \{2, 3, 4, 6, 9\}$

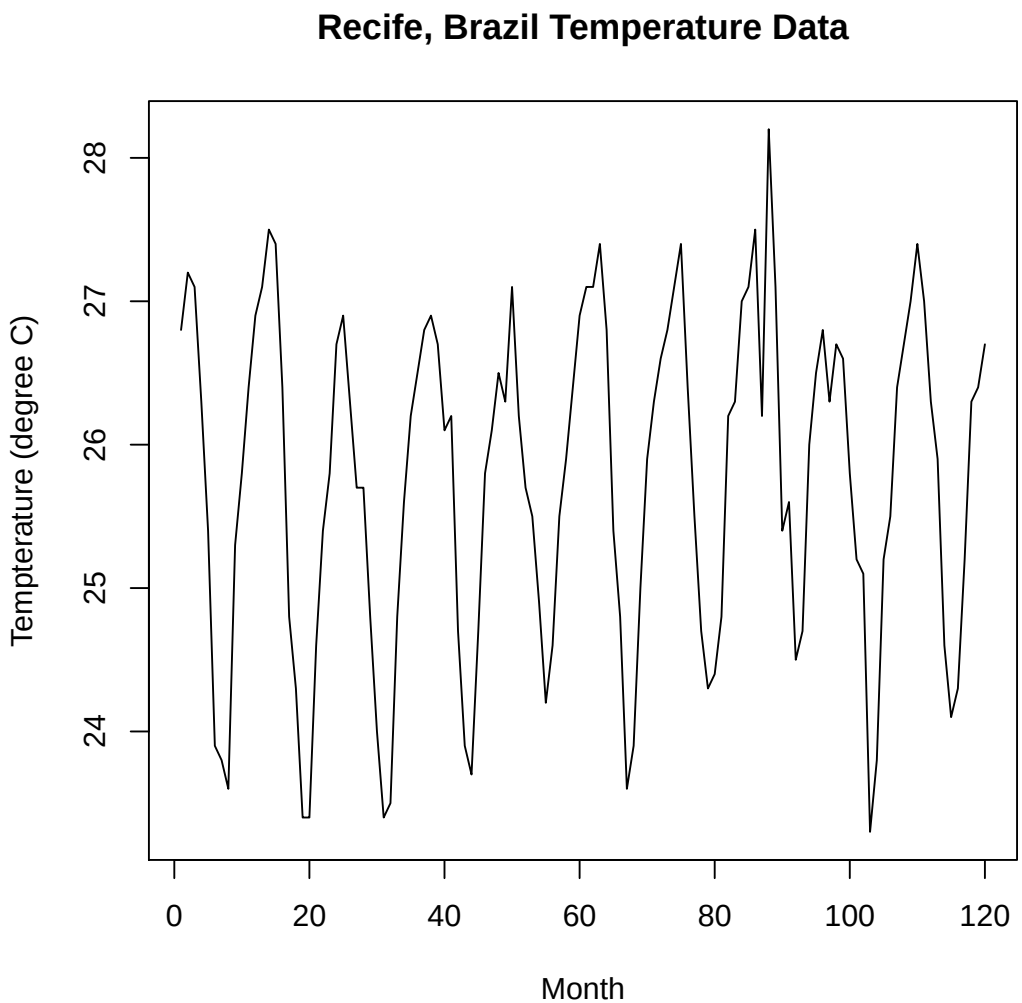
Examples

Measurements of the annual flow of the river Nile at Ashwan 1871–1970. (`Nile` data set part of the R package)



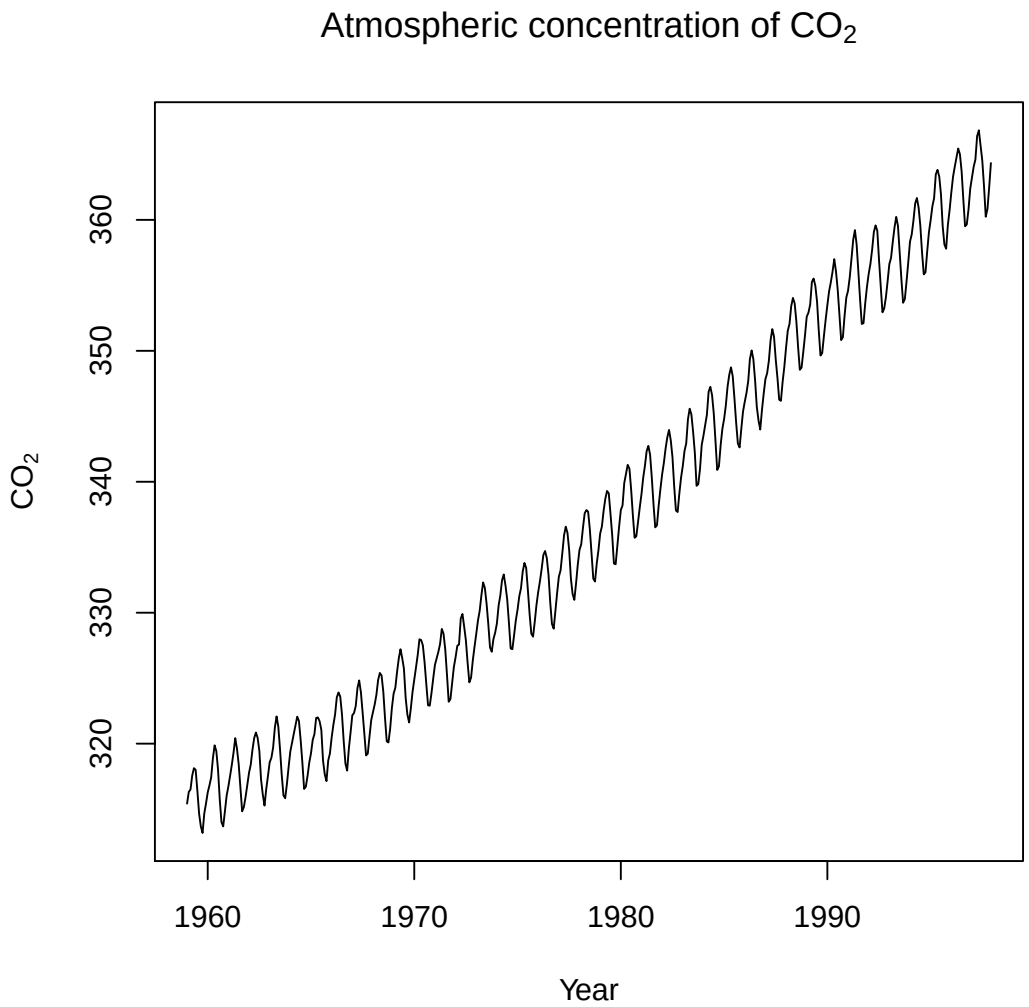
Examples

Average monthly air temperatures (in Celsius) at Recife, Brazil over the period from 1953 to 1962 (Chatfield 2004).



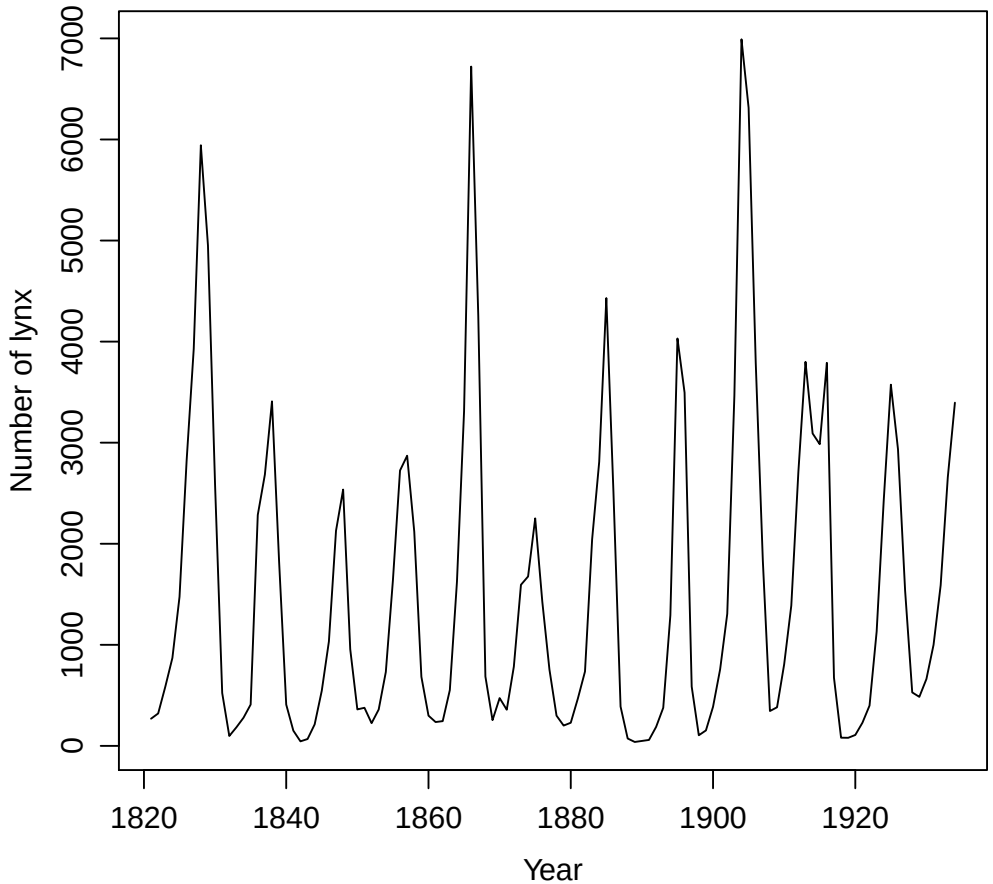
Examples

Atmospheric concentrations of CO2 (ppm) as reported in a 1997 Scripps Institution of Oceanography (SIO) publication (`CO2` data set part of the R package)



Examples

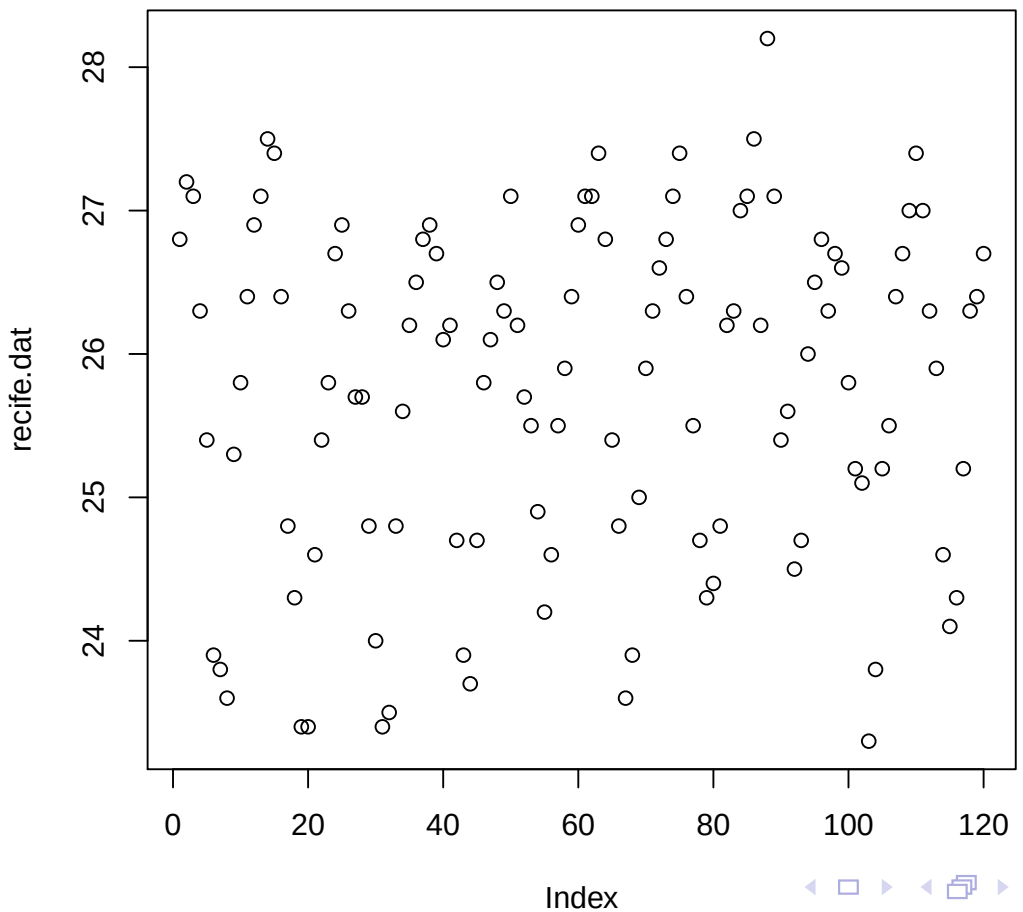
Annual numbers of lynx trapped in Canada from 1821-1934 (`lynx` data set part of the R package)



The `ts` class

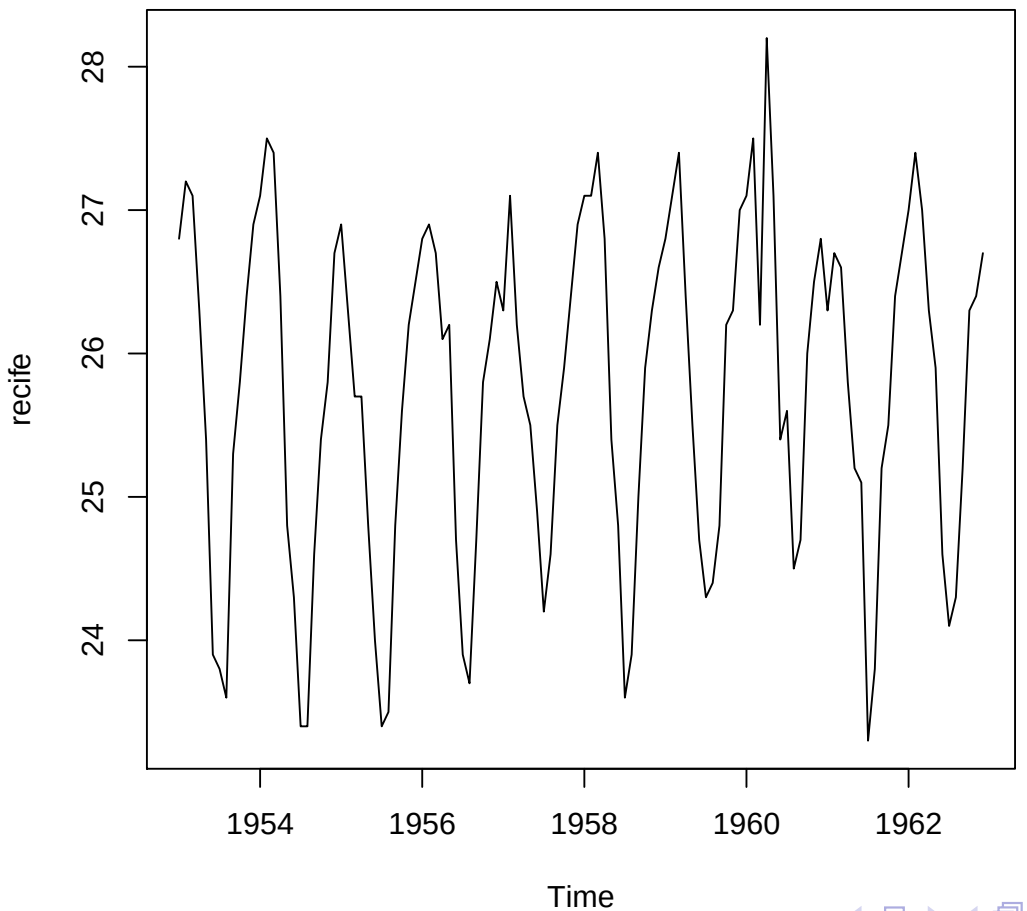
- The basic time series object in R is a `ts` object
- The `ts()` function is used to create a `ts` object `ts()` takes several arguments

```
> recife.dat <- scan("recife.txt")
> plot(recife.dat)
```



The `ts` class

```
> recife.ts <- ts(recife.dat, start = c(1953, 1),
+                 end = c(1962, 12),frequency = 12)
> plot(recife)
```



Goal of the time series analysis

- Is there a trend in the data over time ?
- Is there seasonal variation in the data over time ?
- Is there remaining temporal correlation ?
- Can we use the data for forecast future observations ?

Classical decomposition of time series

- Classical decomposition of an observed time series is a fundamental approach in time series analysis
- The idea is to decompose a time series $\{y_t\}$ into a deterministic part (f_t), a trend (m_t), a seasonal component (s_t), and a remainder (ε_t)

$$\begin{aligned} y_t &= f_t + \varepsilon_t \\ &= m_t + s_t + \varepsilon_t \end{aligned}$$

- The trend and the seasonal component are deterministic
- The remainder is random. Usually,

$$\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$$

- Can be independent, but usually temporal correlation is considered
- We first ”extract” the deterministic part, and then we analyze the random part

Regression methods (with no seasonality)

- polynomial trend (e.g. second order):

$$m_t = b_0 + b_1 t + b_2 t^2$$

- polynomial regression

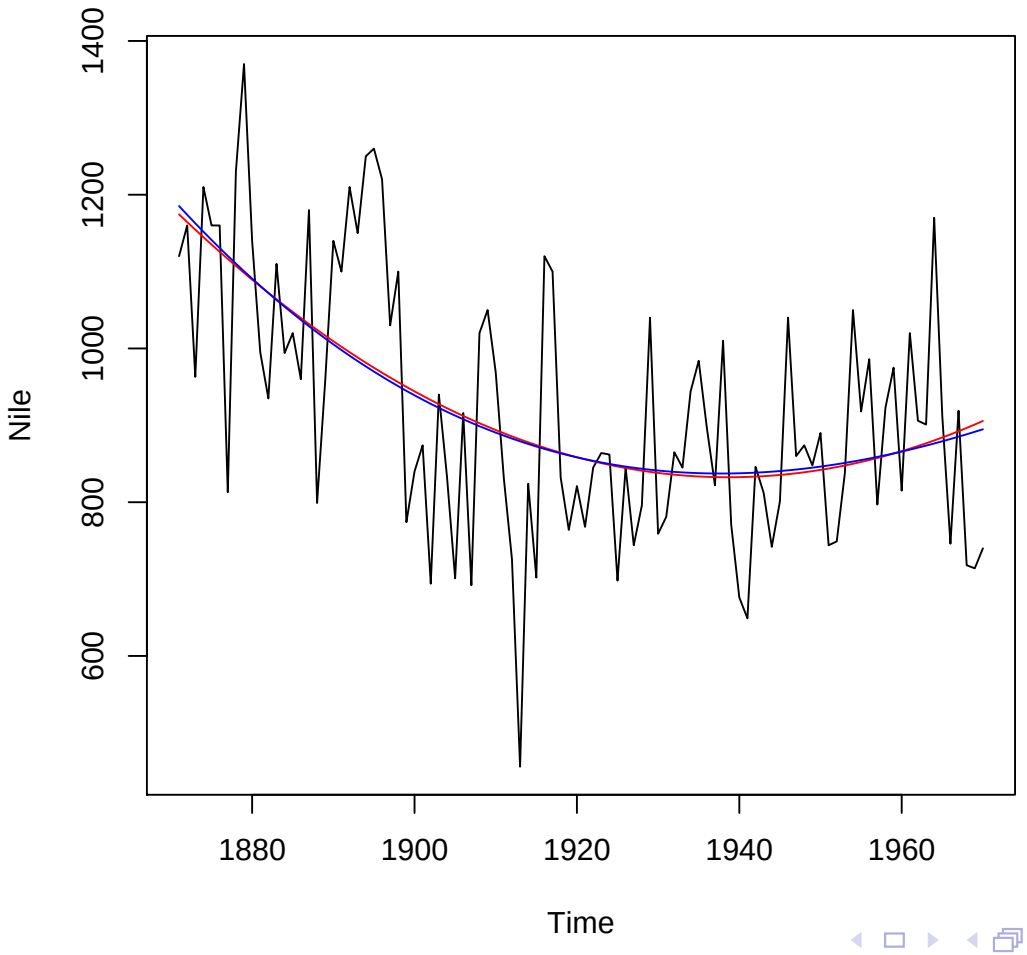
$$y_t = b_0 + b_1 t + b_2 t^2 + \varepsilon_t$$

- we fit the unknown parameters using least squares

```
> tt <- as.numeric(time(Nile))
> fit <-lm(Nile~poly(tt,degree=2,raw=TRUE))
```

Regression methods (with no seasonality)

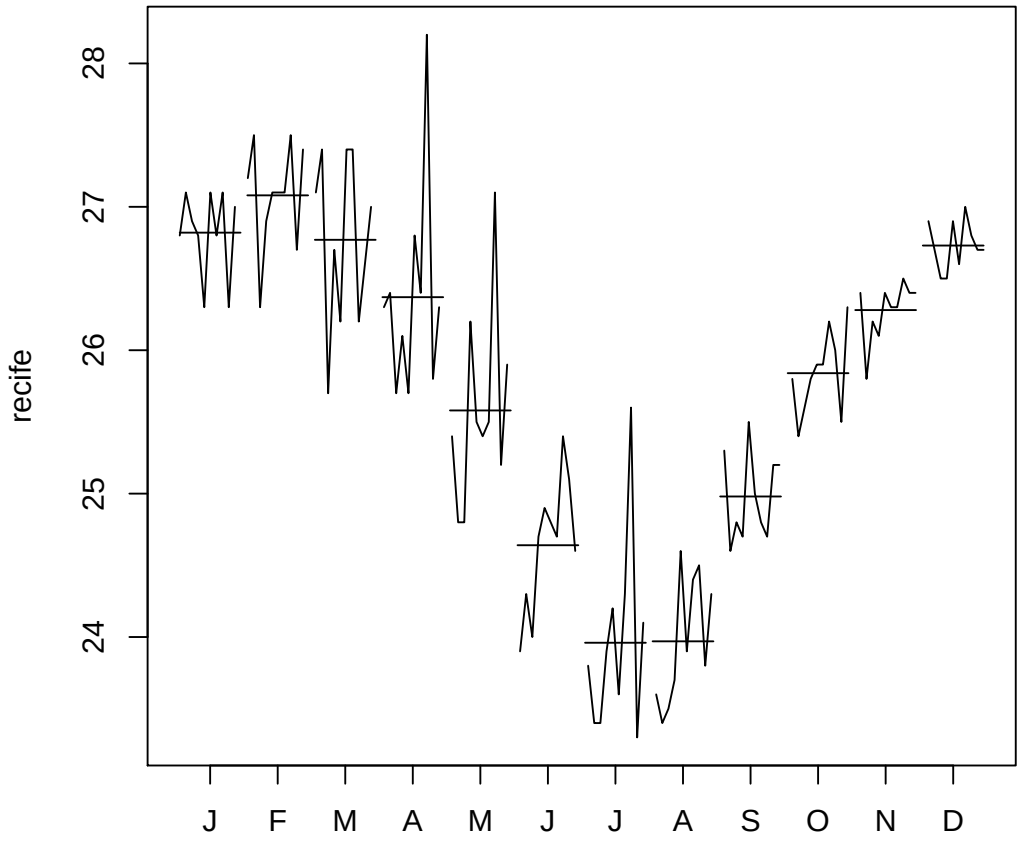
```
> plot(Nile)
> lines(tt,predict(fit2),col='red')
> lines(tt,predict(fit4),col='blue')
```



Regression methods (with seasonality)

- Seasonal effects in Recife data

```
> monthplot(recife)
```



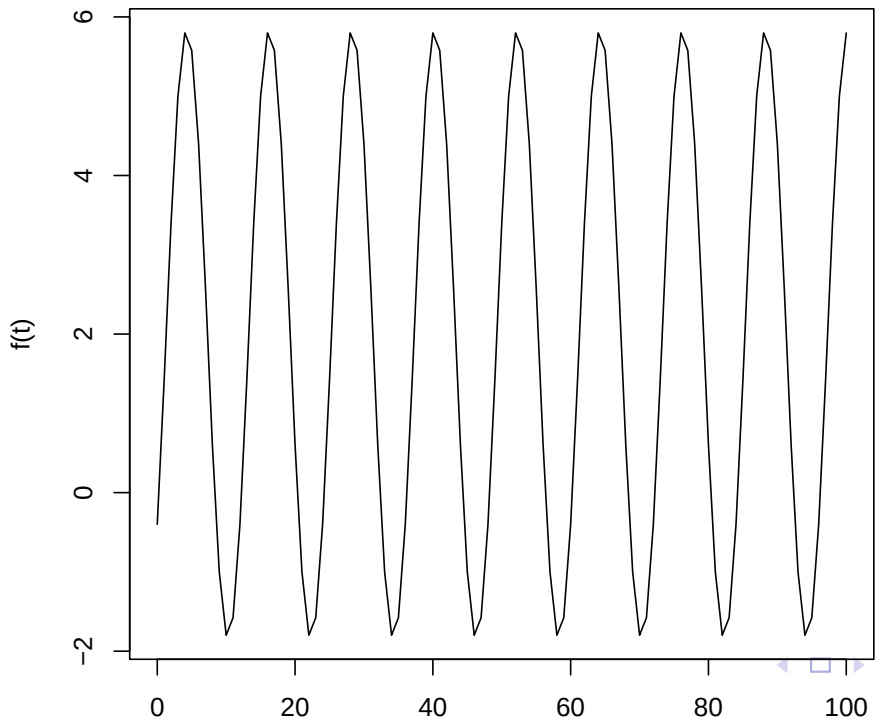
Regression methods: periodic functions

- We can use a periodic function i.e a linear combination of sinus and cosinus functions of period 12.
- For example

$$f(t) = 2 + 3 \times sin((2\pi/12)t) - 2.4 \times cos((2\pi/12)t),$$

when $t = 1, 2, \ldots, 12$.

```
> curve(2+3*sin(2*pi*x/12)-2.4*cos(2*pi*x/12),
+ from=0,to=100,xlab="t",ylab="f(t)")
```



Regression methods: periodic functions

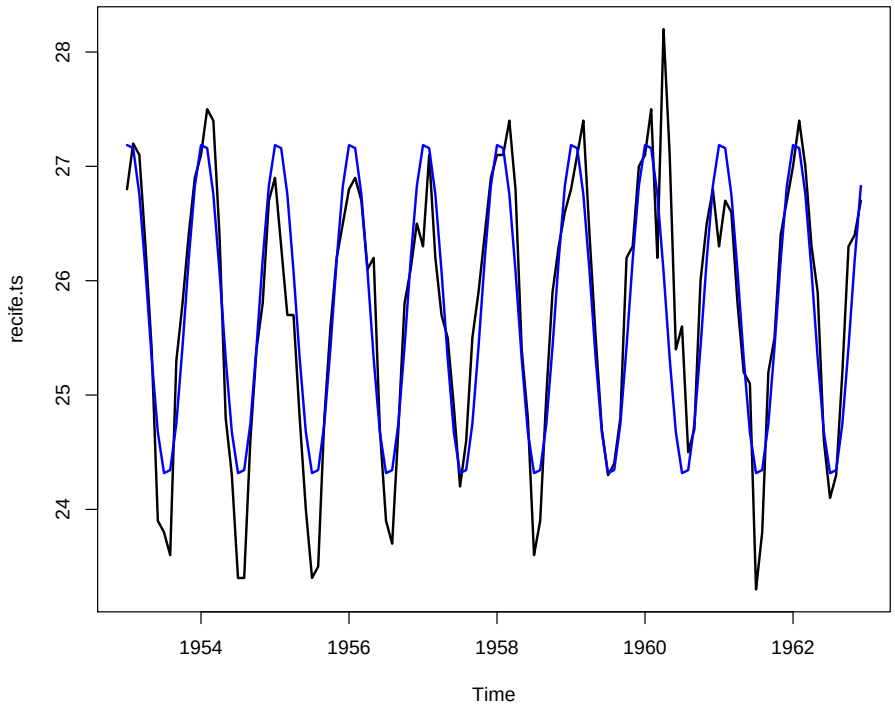
We can fit the following model:

$$y_t = d_0 + d_1 sin((2\pi/12)t) + d_2 cos((2\pi/12)t) + \varepsilon_t$$

```
> tt<-1:length(recife)
> s1<-sin(tt*2*pi/12)
> s2<-cos(tt*2*pi/12)
> fit.periodic<-lm(recife~s1+s2)
> summary(fit.periodic)

Call:
lm(formula = recife ~ s1 + s2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.05539 -0.34025  0.00019  0.24647  2.11767

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 25.75167    0.04808  535.62   <2e-16 ***
s1           1.00372    0.06799   14.76   <2e-16 ***
s2           1.07718    0.06799   15.84   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5267 on 117 degrees of freedom
Multiple R-squared:  0.8003,Adjusted R-squared:  0.7969
F-statistic: 234.5 on 2 and 117 DF,  p-value: < 2.2e-16
```

Regression methods: periodic functions

```
pred.periodic<-predict(fit.periodic)
plot(recife.ts,lwd=2)
lines(as.numeric(time(recife.ts)),pred.periodic,col='blue',lwd=2)
```



Regression methods: trend+ seasonality

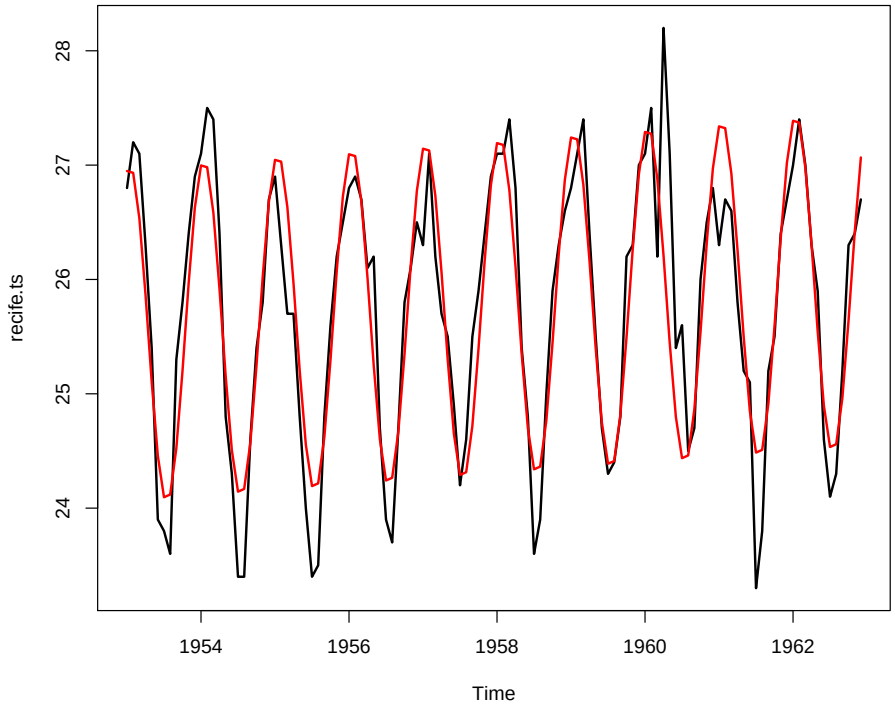
- We can combine linear trend and periodic function for seasonal effect
- we fit the whole model

$$y_t = b_0 + b_1 t + d_1 sin((2\pi/12)t) + d_2 cos((2\pi/12)t) + \varepsilon_t$$

```
> fit.complete <- lm(formula = recife.ts ~ poly(tt, 1, raw = TRUE) + s1 + s2)
> summary(fit.complete)
Call:
lm(formula = recife.ts ~ poly(tt, 1, raw = TRUE) + s1 + s2)

Residuals:
Min       1Q   Median       3Q      Max
-1.18583 -0.30197  0.00491  0.25628  1.99057

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)              25.505460   0.093782 271.965  < 2e-16 ***
poly(tt, 1, raw = TRUE)   0.004070   0.001346   3.023  0.00308 **
s1                        1.018910   0.065937  15.453  < 2e-16 ***
s2                        1.073106   0.065759  16.319  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5093 on 116 degrees of freedom
Multiple R-squared:  0.8149,Adjusted R-squared:  0.8101
F-statistic: 170.2 on 3 and 116 DF,  p-value: < 2.2e-16
```

Regression methods: trend+ seasonality

```
> summary(fit.complete)
> pred.complete<-predict(fit.complete)

> plot(recife.ts,lwd=2)
> lines(as.numeric(time(recife.ts)),pred.complete,col='red',lwd=2)
```



Regression methods: trend+ seasonality

Time series decomposition

- Trend

$$\hat{m}_t = \hat{b}_0 + \hat{b}_1 t$$

```
> fit <- fit.complete
> trend<- ts(coef(fit)[1]+ coef(fit)[2]*tt,
+       start = start(recife), frequency = frequency(recife))
```

- Seasonality

$$\hat{s}_t = \hat{d}_1 sin((2\pi/12)t) + \hat{d}_2 cos((2\pi/12)t)$$

```
> season<-ts(coef(fit)[3]*s1+ coef(fit)[4]*s2,
+         start = start(recife),frequency = frequency(recife))
```

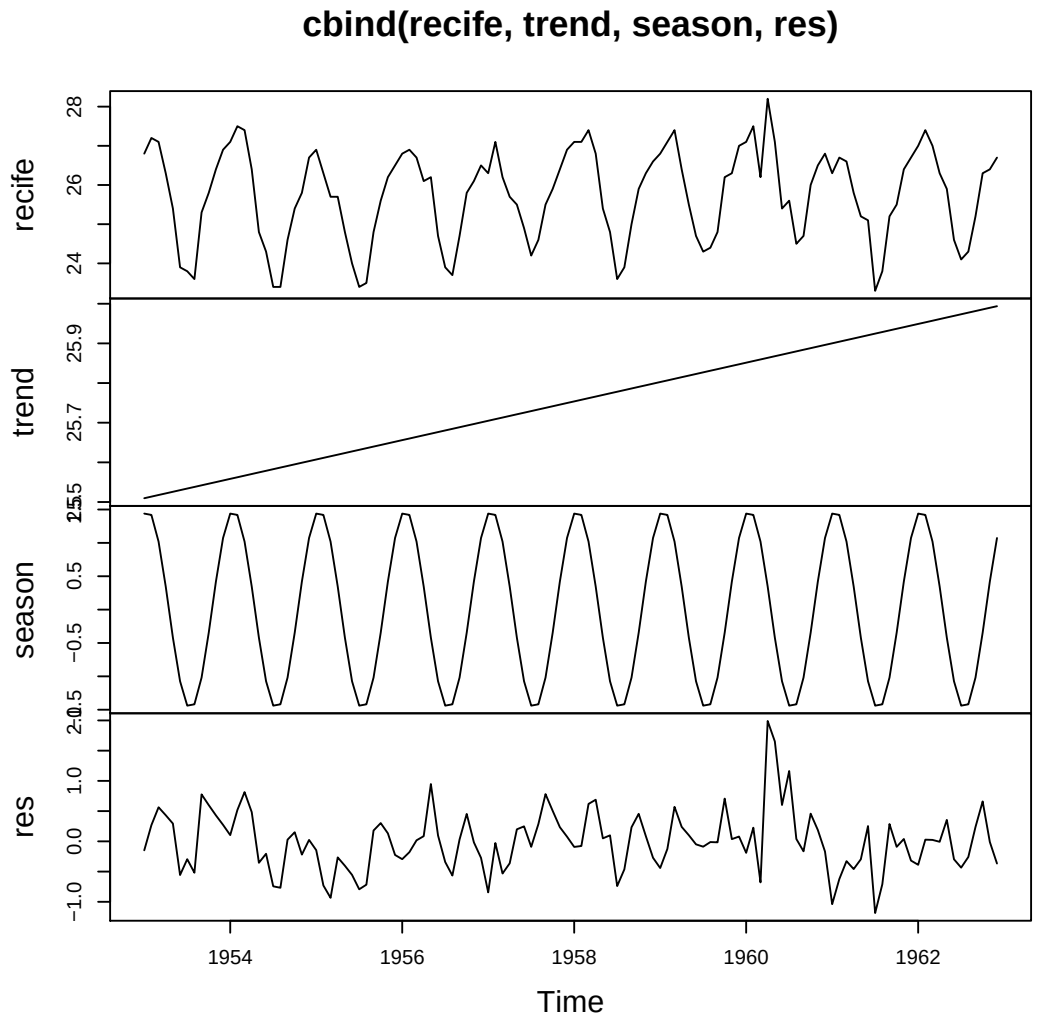
- residuals

$$\hat{\varepsilon}_t = y_t - \hat{y}_t$$

```
> res<-ts(residuals(fit),
+      start = start(recife), frequency = frequency(recife))
```

Regression methods: trend+ seasonality

```
> plot.ts(cbind(recife,trend,season,res))
```



Other trend modeling approaches

We have seen the fitting of parametric functions There are other possibilities

- Moving averages methods, e.g.

$$m_t = \frac{y_t + y_{t-1} \cdots + y_{t-p+1}}{p}$$

- Removing trends by differences. Example: we suppose that the series has a linear trend

$$y_t = b_0 + b_1 t + \varepsilon_t$$

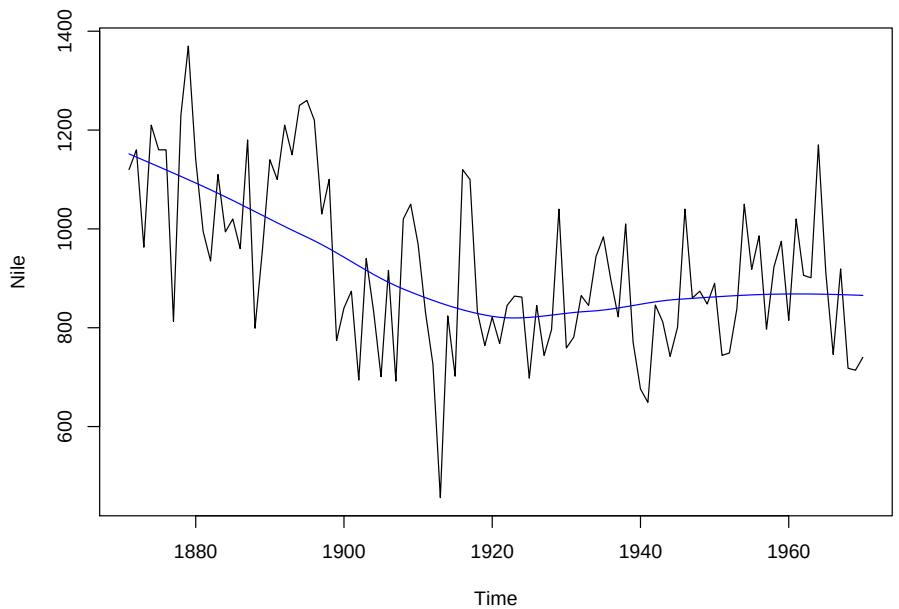
then

$$\begin{aligned} \nabla y_t &= y_t - y_{t-1} \\ &= (b_0 + b_1 t + \varepsilon_t) - (b_0 + b_1 (t-1) + \varepsilon_{t-1}) \\ &= b_1 + \varepsilon_t - \varepsilon_{t-1} \end{aligned}$$

- Non linear regression (kernels, Nadaraya-Watson, splines), see e.g. `loess`

Non parametric trends

```
tt       <- as.numeric(time(Nile))
Nile.np <- loess(Nile~tt)
summary(Nile.np)
plot(Nile,xlab="Time",ylab="Nile")
Nile.loess.pred <- predict(Nile.np,data.frame(tt))
lines(tt,Nile.loess.pred,col="blue")
```



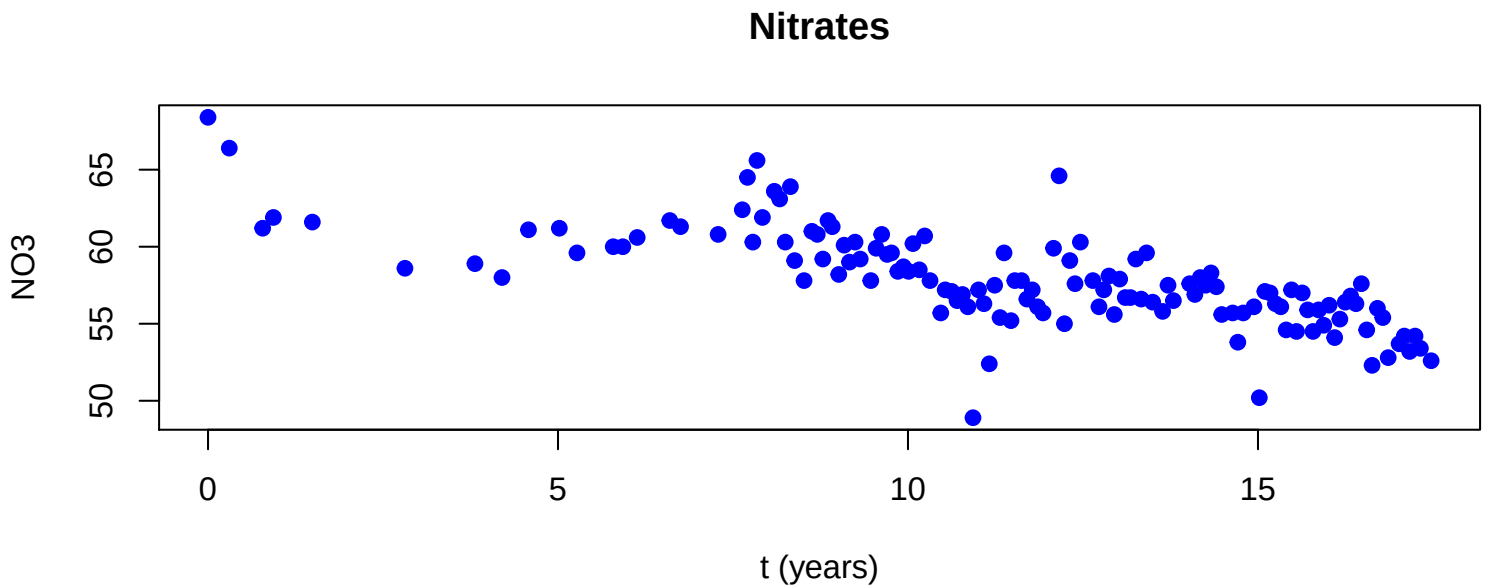
Context

Water Framework Directive

- Directive 2000/60/EC
- Establishing a framework for Community action in the field of water policy
- The Directive aims for 'good status' for all ground and surface waters
- Groundwater must achieve "good quantitative status" and "good chemical status" (i.e. not polluted) by 2015
- River basin (the spatial catchment area of the river) as a natural geographical and hydrological unit
- They are managed according to River Basin Management Plans

Context

- Nitrate content in groundwater
- Government must prove that chemical status must either be below limit or must improve
- Time series from de-identified data and region
- Is there a trend ?
- Is there a change process ?



Formalization

Three competing models

- Model M_0: constant mean; constant variance: 1 parameter
- Model M_1: linear trend; constant variance: 2 parameters
- Model M_2: two linear trends; one change point; continuity is imposed: 4 parameters

1. Testing M_1 vs. M_0 is straightforward
2. Testing M_2 vs. M_0 is OK. It is a particular linear model
3. Testing M_2 vs. M_1 is not obvious. M_1 is not nested within M_2. Therfore, in theory, we cannot use F statistics.

Other statistical issues

- Short series
- Isolated data
- Long interruption in the time series
- Detecting outliers

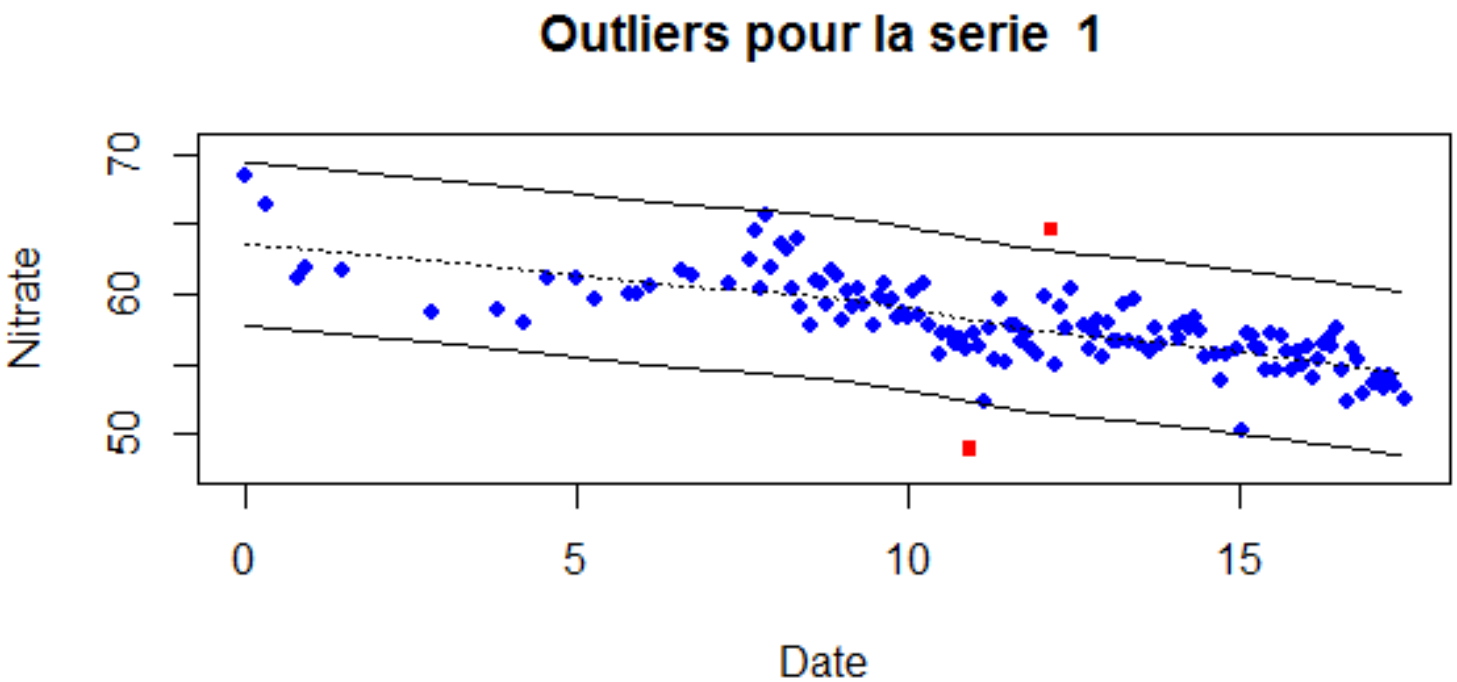
Detecting outliers

- Do a non parametric estimation of Nitrate vs. time
- Compute a pointwise CI corresponding to a very high level, i.e.

$$\mathbb{P}\{ N(t) \notin CI(t) \} = \alpha$$

  with $\alpha = 0.005$.
- Remove all points outside the CI

Detecting outliers



Model selection

1. Check residuals
2. F test when possible (M_1 vs M_0; M_2 vs M_0)
3. p-value of slopes
4. Use Bayesian Information Criterion (BIC) otherwise

The Bayesian Information Criterion

Definition

$$\text{BIC} = -2 \ln \hat{L} + p \ln n,$$

where p is the number of parameters.

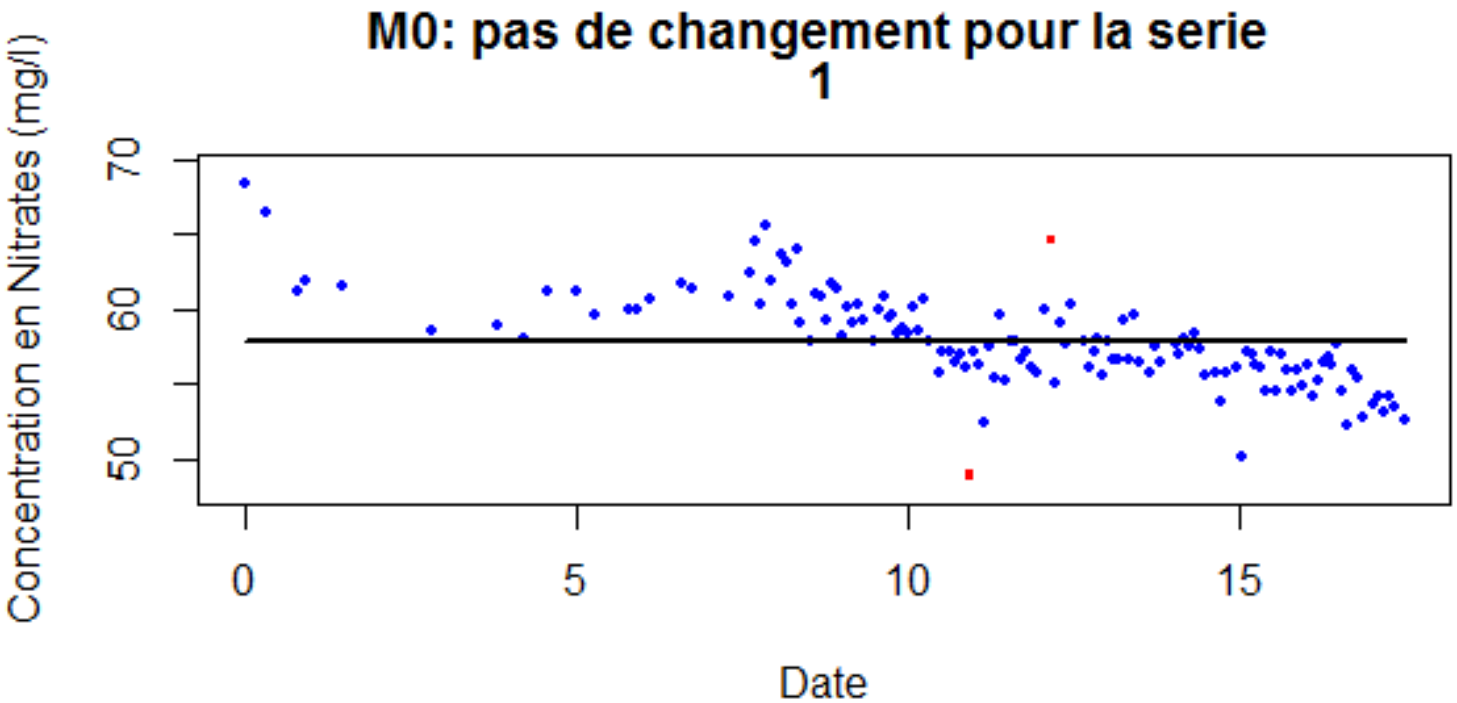
In the Gaussian case, with i.i.d. errors,

$$\text{BIC} = n \ln \hat{\sigma}_\epsilon^2 + p \ln n$$

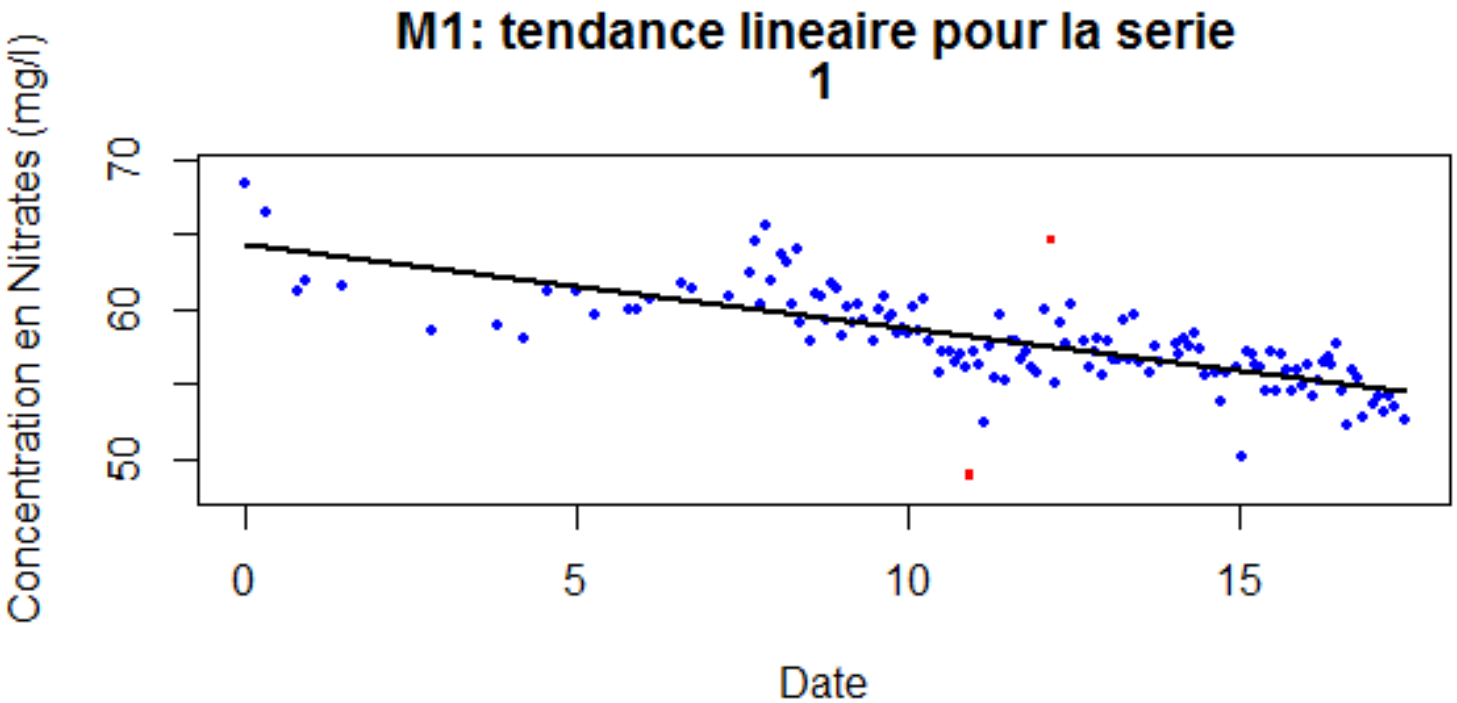
or

$$\text{BIC} = n \ln(SS_R/n) + p \ln n$$

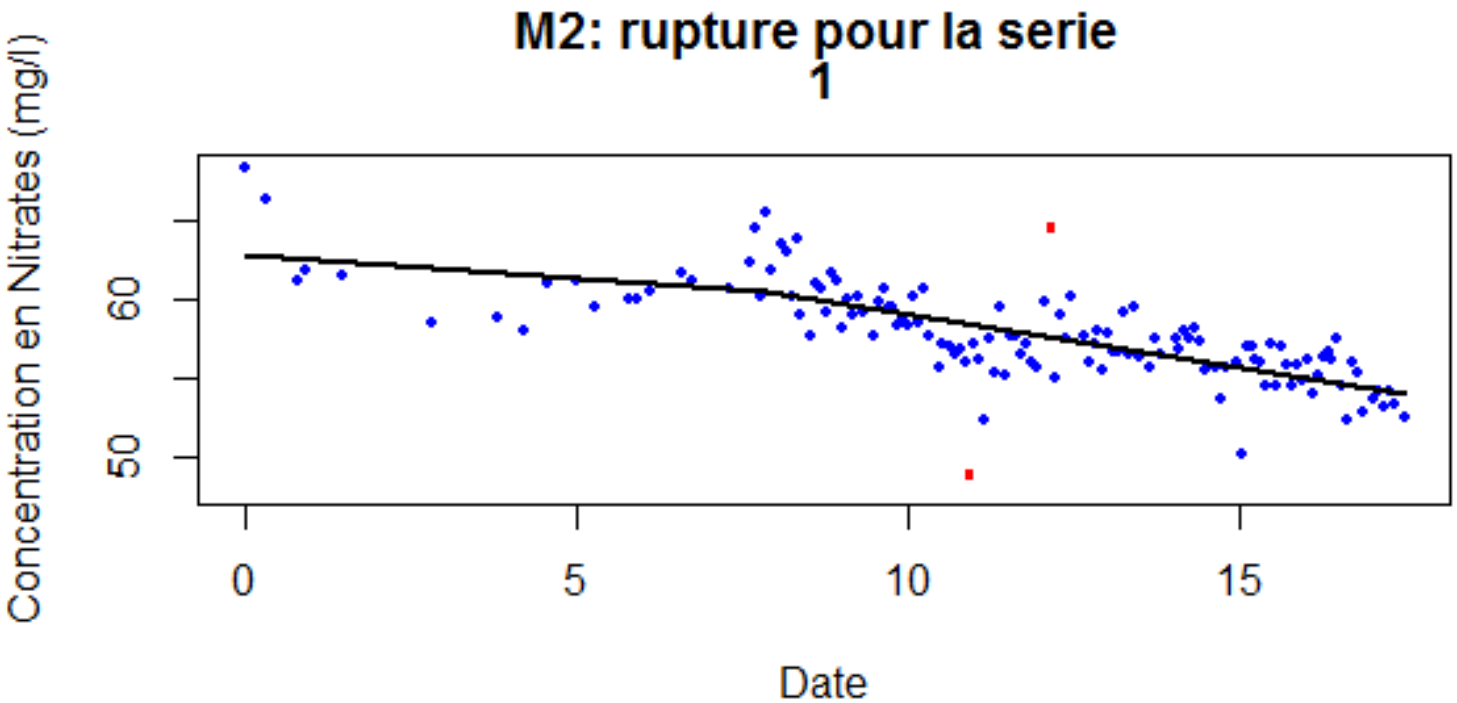
Model M_0


M0: pas de changement pour la serie 1
Concentration en Nitrates (mg/l)
50
60
70
0
5
10
15
Date

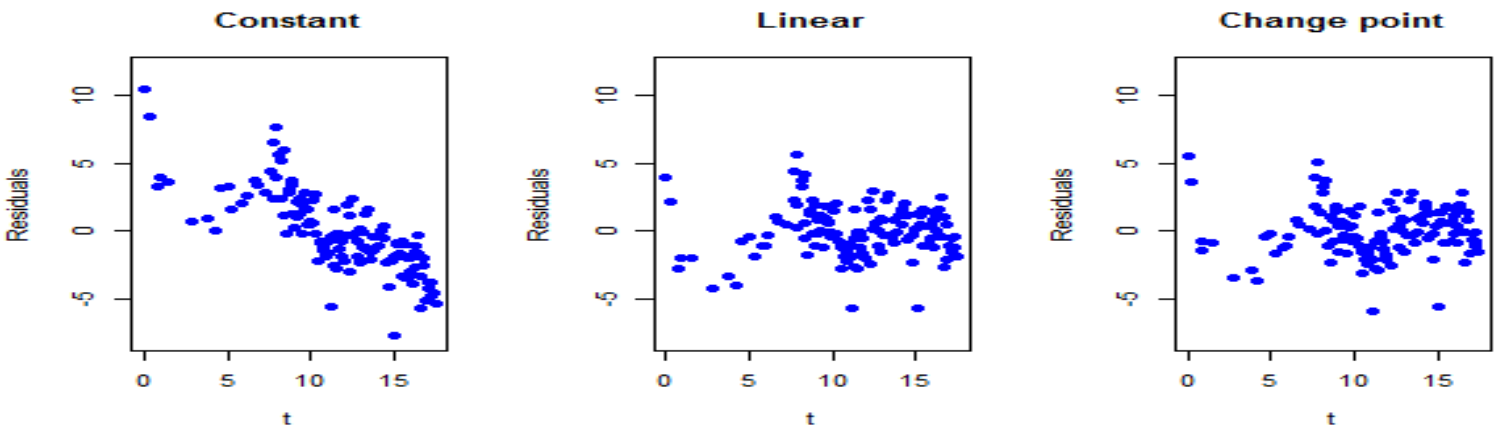
Model M_1


M1: tendance lineaire pour la serie 1
Concentration en Nitrates (mg/l)
50
60
70
0
5
10
15
Date

Model M_2


M2: rupture pour la serie
1
Concentration en Nitrates (mg/l)
50
60
0
5
10
15
Date

Results



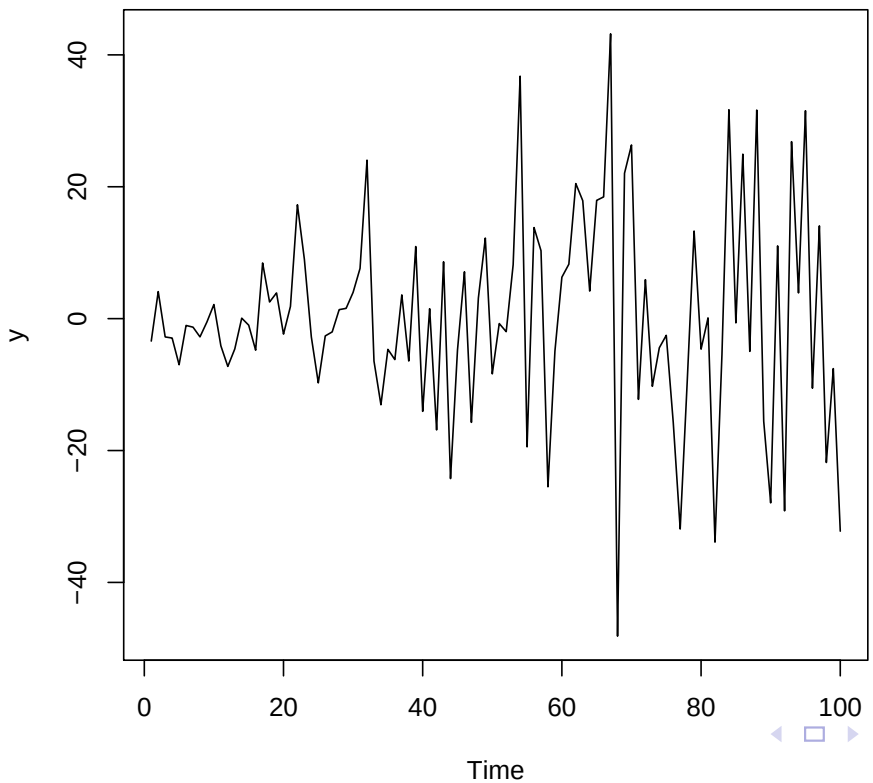
	M_0	M_1	M_2
SS_R	1136.8	450.9	430.1
BIC	926.6	810.3	809.0
p-value	—	10^{-27}	0.05

Stationary Time Series

- Suppose that we observe a time series values $y_1, \ldots, y_n$
- These values can be interpreted as realization of a sequence of random variables
- Each variable y_t has
  - a mean $E(y_t)$
  - a variance $Var(y_t)$

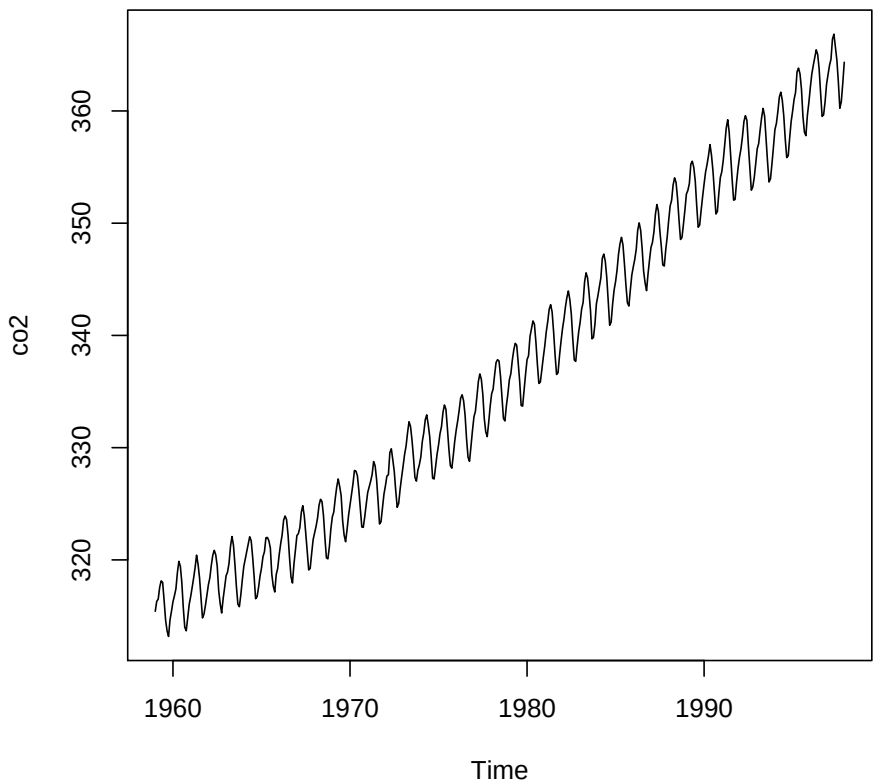
  that can depend on t

Synthetic example:



Stationary Time Series

Atmospheric concentrations of CO2 are expressed in parts per million (ppm) and reported in the preliminary 1997 SIO manometric mole fraction scale.

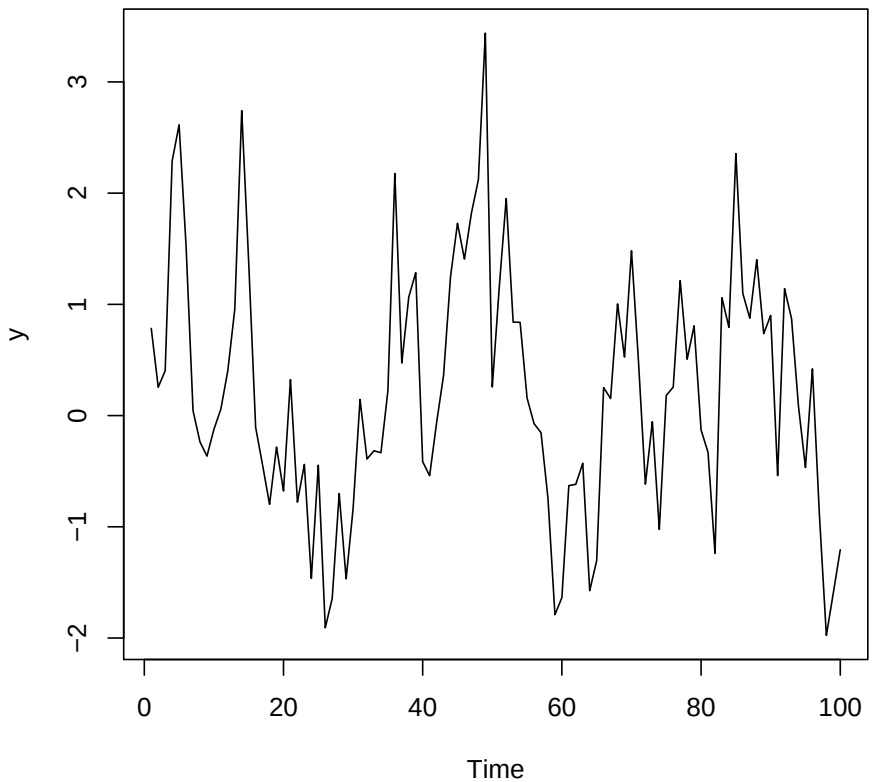


This plot show an evident periodic behaviour and a notable trend.

Stationary Time Series

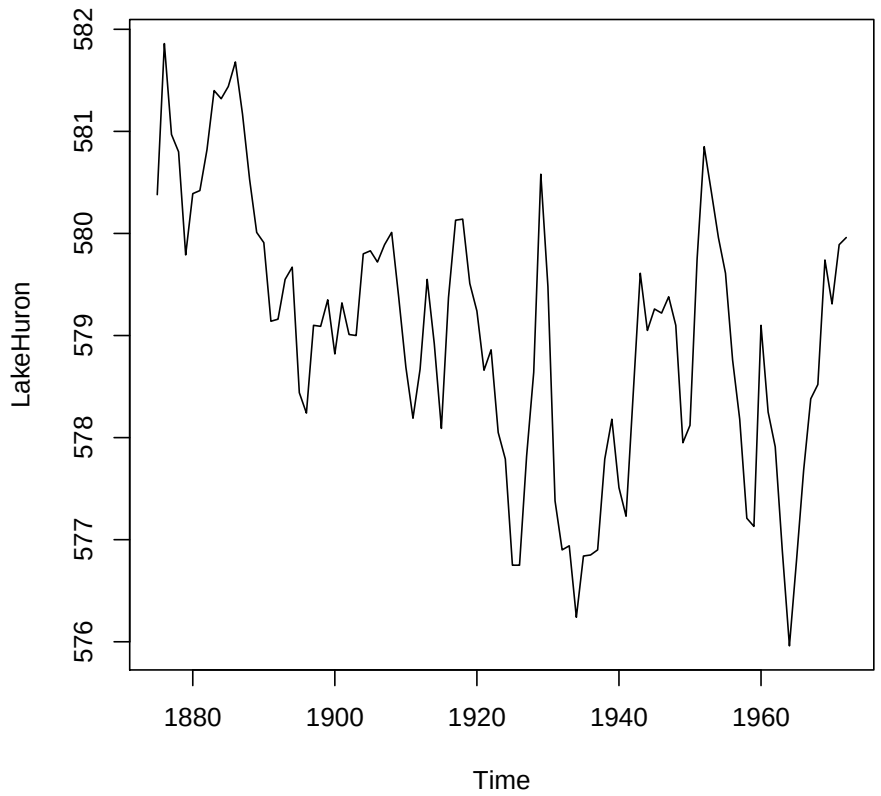
- Roughly speaking, a stationary time series is one that has the same mathematical properties at any given time point.
- Mean and variance does change with time
- and it does not have periodic variations

Synthetic example



Stationary Time Series

Annual measurements of the level, in feet, of Lake Huron 1875-1972 were measured.



This plot does not show any evident periodic behaviour, nor does it indicate a notable trend.

Correlation and autocorrelation

- The 'usual' correlation between n pairs of observations on two variables x and y

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

- Given n observations, under stationarity assumption we can form $n - 1$ pairs

$$(y_1, y_2), (y_2, y_3), \ldots, (y_{n-1}, y_n)$$

the correlation between y_t and y_{t+1}

$$r_1 = \frac{\sum_{t=1}^{n-1}(y_t - \overline{y}^{(1)})(y_{t+1} - \overline{y}^{(2)})}{\sqrt{\sum_{t=1}^{n-1}(y_t - \overline{y}^{(1)})^2 \sum_{t=1}^{n-1}(y_{t+1} - \overline{y}^{(2)})^2}}.$$

Correlation and autocorrelation

- Since $\overline{y}^{(1)} = \sum_{t=1}^{n-1} y_t/(n-1)$ and $\overline{y}^{(2)} = \sum_{t=1}^{n-1} y_{t+1}/(n-1)$ are approximately equal, a simplification is given by

$$r_1 = \frac{\sum_{t=1}^{n-1}(y_t - \overline{y})(y_{t+1} - \overline{y})}{\sqrt{\sum_{t=1}^{n}(y_t - \overline{y})^2}}$$

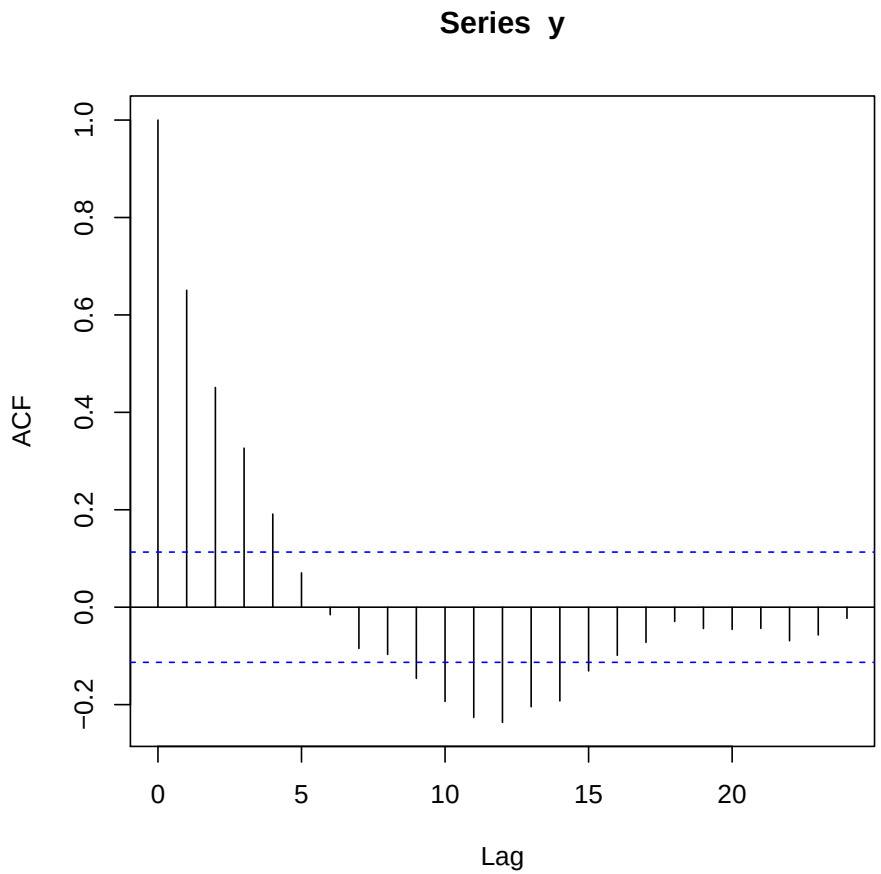
- for y_t and y_{t+k} we get

$$r_k = \frac{\sum_{t=1}^{n-k}(y_t - \overline{y})(y_{t+k} - \overline{y})}{\sqrt{\sum_{t=1}^{n}(y_t - \overline{y})^2}}$$

- This is called the **autocorrelation coefficient** at lag k.

The correlogram

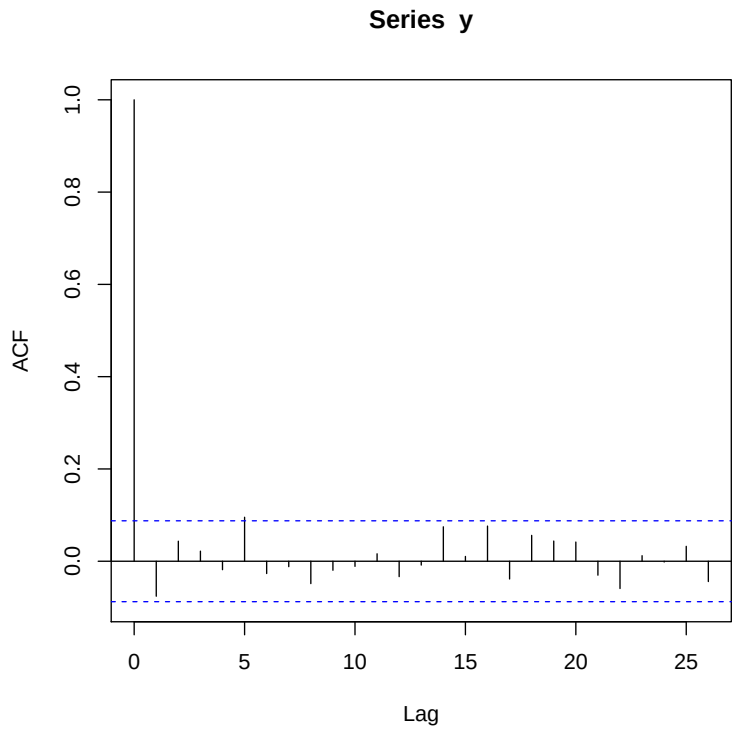
- The correlogram is a graph where r_k is plotted against the lag k.
- In R the command is `acf(y)` for a time series `y`.
- Only for regularly spaced time series



The correlogram: examples

- A random series (or white noise)
- For a time series completely random, for large n, $r_k \simeq 0$ for all non-zero values of k.

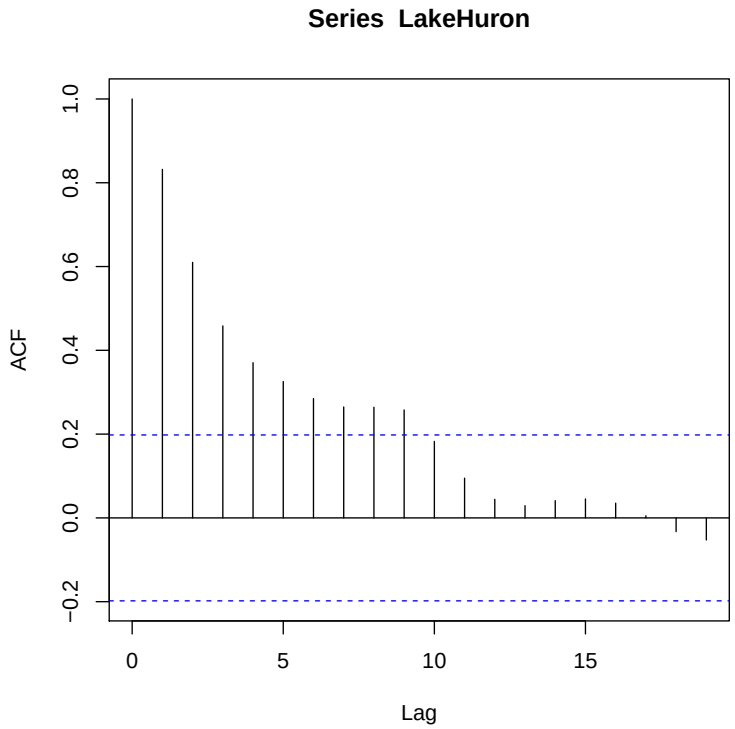
```
> set.seed(19)
> y <-rnorm(500)
> acf(y)
```



The correlogram: examples

- Short-term correlation
- Fairly large value of r_1 followed by a few further coefficients which, while greater than zero, tend to get successively smaller.

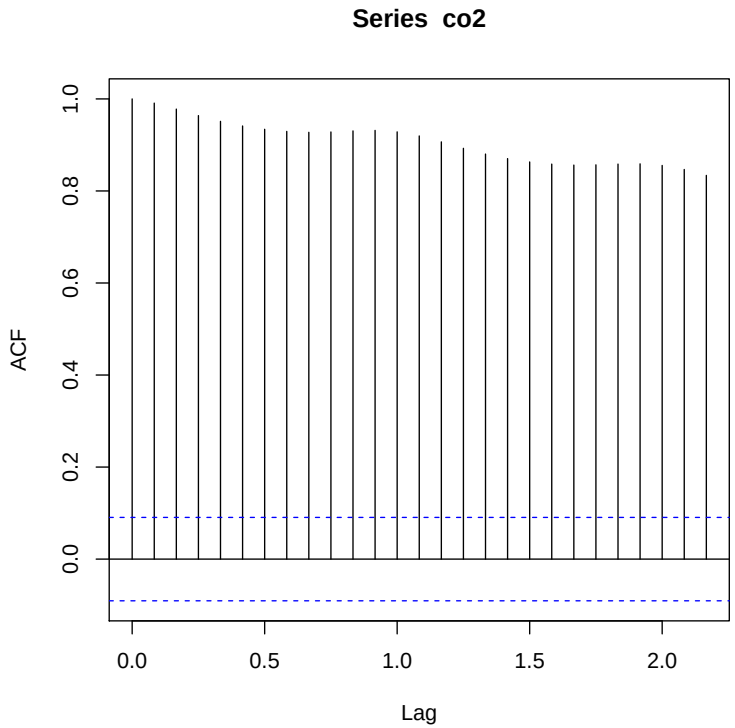
```
> acf(LakeHuron)
```



The correlogram: examples

- Non-stationary series with trend
- the values of r_k will not come down to zero except for very large values of the lag. This is because large (small) values tend to be followed by a large number of further large (small) values.

```
> acf(co2)
```

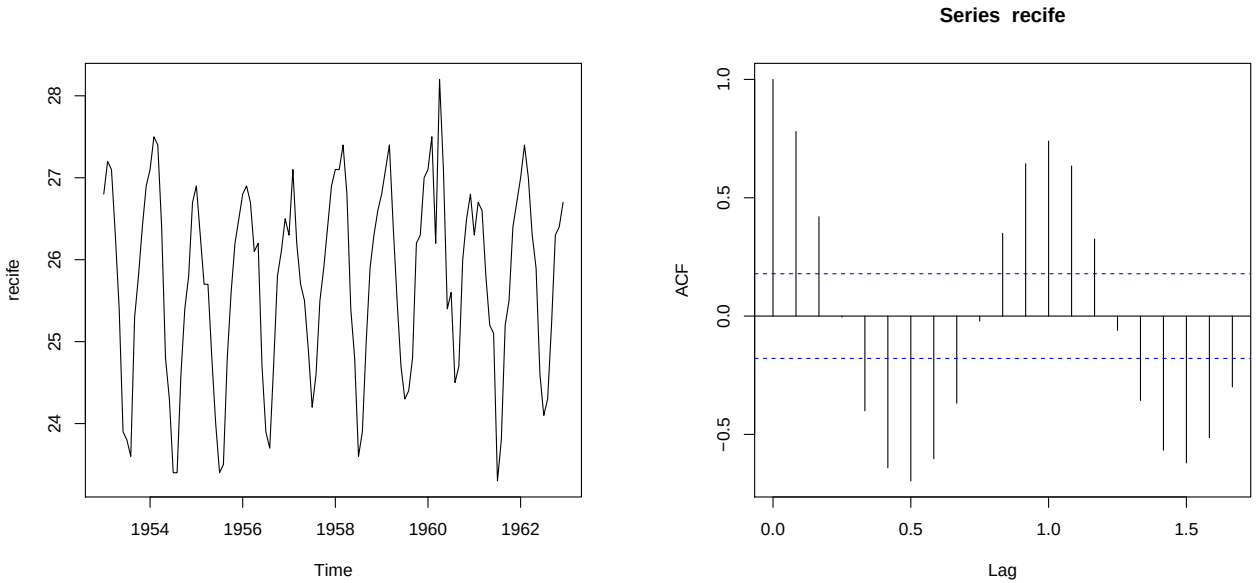


- Note that in theory, because of the non stationarity, one should note use `acf` on these data

The correlogram: examples

- Seasonal fluctuations
- the correlogram exhibit an oscillation at the same frequency as the seasonality

```
> acf(recife)
```



Verification of the randomness

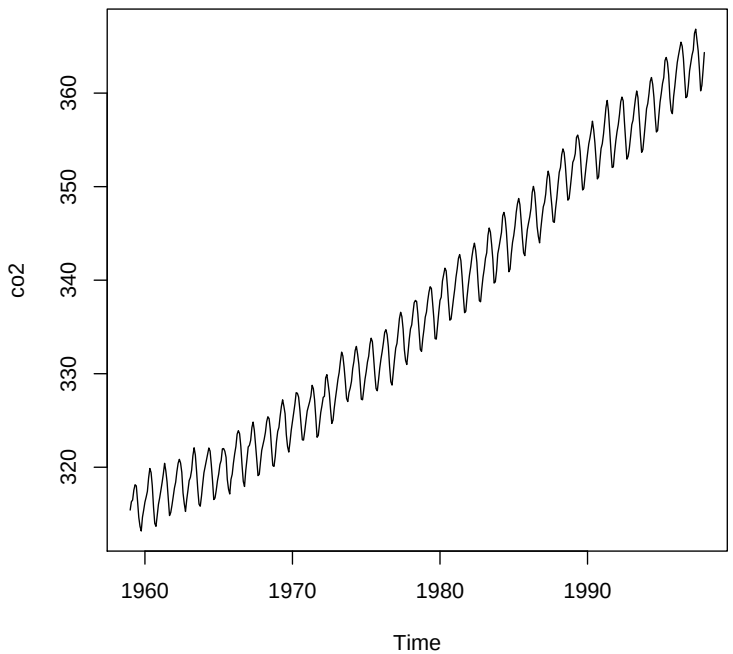
- For a large number of observation and for a time series completely at random, the autocorrelation $r_k \simeq 0$
- Probability theory shows that $r_k \simeq \mathcal{N}(0, 1/n)$
- So that, if a time series is random, 19 out of 20 (95%) of the values of r_k can be expected to lie between $\pm \frac{2}{\sqrt{n}}$, the blue dashed lines in the correlogram.

Randomness in the residuals

- Recall the classical decomposition

$$y_t = m_t + s_t + \varepsilon_t$$

- The trend component and the seasonal component are components that explain the main pattern of the time series with respect to the time



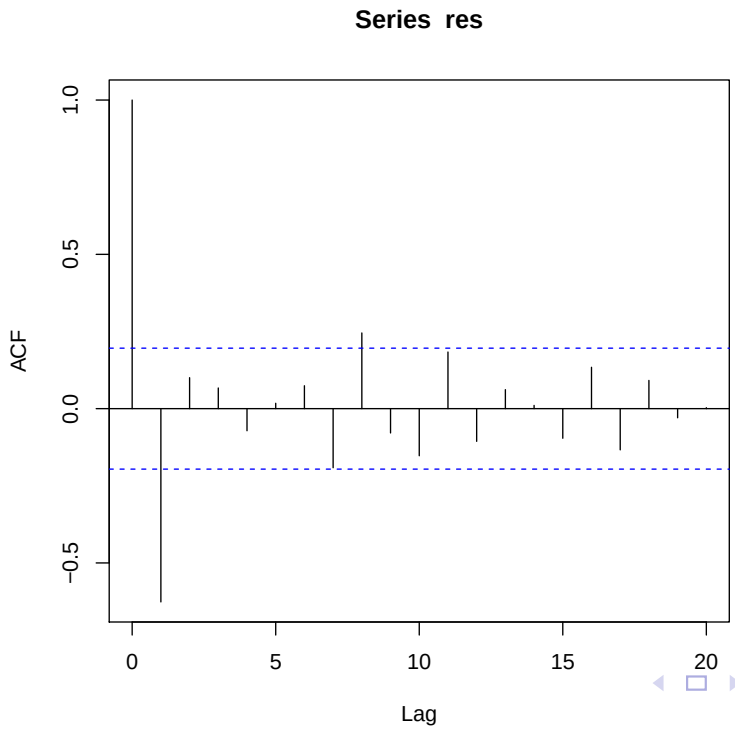
Randomness in the residuals

- after removing these components, we expect that the residuals

$$\hat{\varepsilon}_t = y_t - (\hat{m}_t + \hat{s}_t)$$

  loose any particular relationship with the time
- a way of checking this is to consider the correlogram of the residuals
- because we use a symmetric linear filter, values are missing at the beginning and at the end.
- Nile time series with

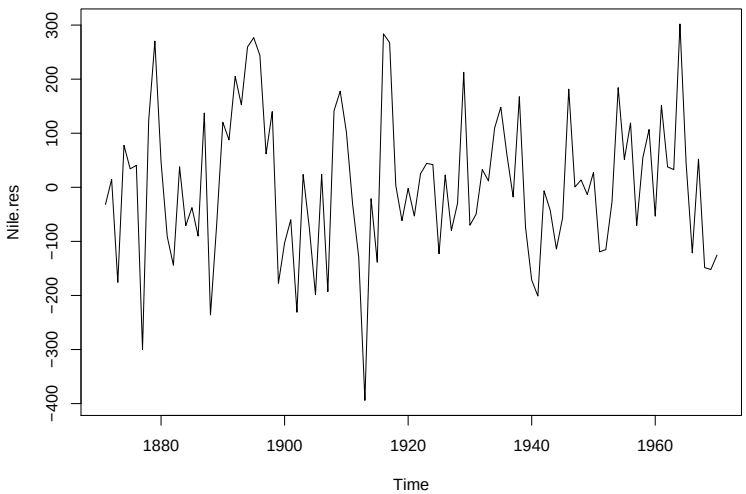
$$\hat{m}_t = \frac{y_{t-1} + y_t + y_{t+1}}{3}$$

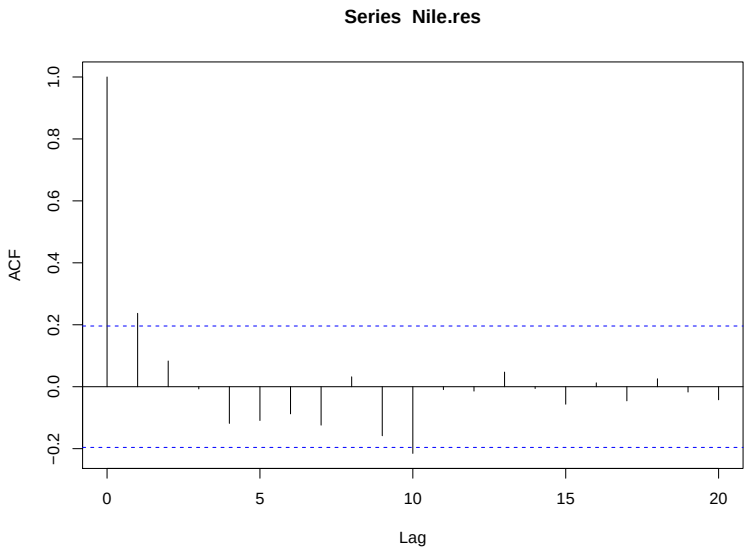


Randomness in the residuals

Nile residuals from Loess fit

```
> Nile.res <- ts(as.numeric(Nile - Nile.loess.pred),
                 start=start(Nile),end=end(Nile))
> plot(Nile.res)
> acf(Nile.res)
```





Time series models

- Can we use the data for forecast future observations ?
- For doing this, we need a (stochastic) model that relates future observations $\{y_{n+1}, \ldots, y_{n+k}\}$ to the observed data $\{y_1, \ldots, y_n\}$
- Time series fall into the general field of Stochastic Processes which can be described as random phenomenon that evolve over time.
- We have already encountered one example: the random time series which consists of a sequence of random variables $y_1, y_2, \ldots$ that are independent and have the same distribution.
- This model is called **white noise** provided that the mean is equal to zero and the variance is equal to 1

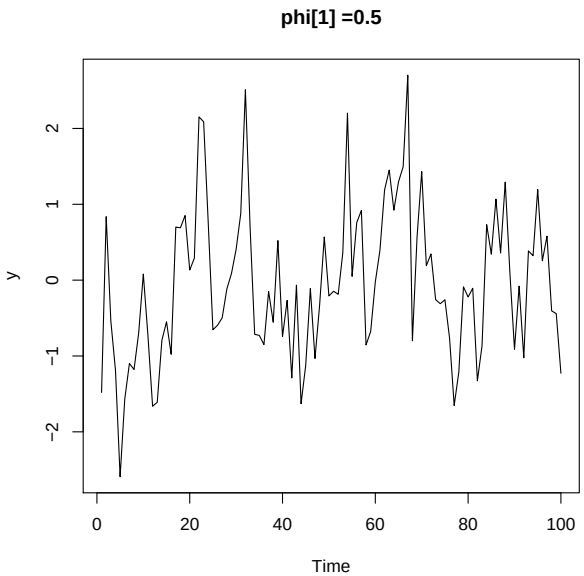
Autoregressive Processes

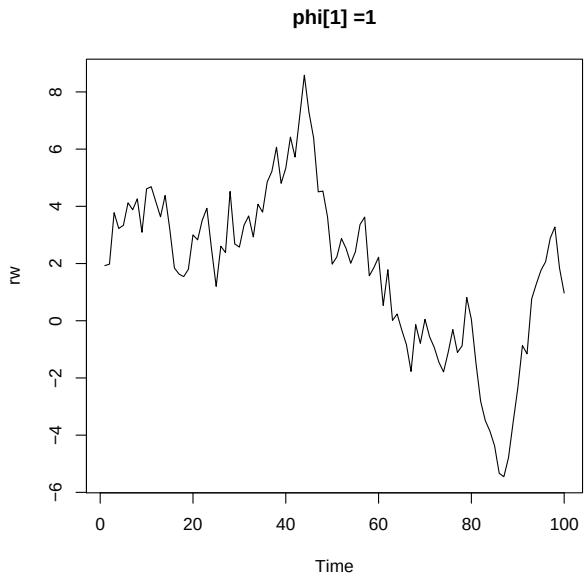
- The current value y_t depends on the previous one only:

$$y_t = \phi_1 y_{t-1} + \varepsilon_t$$

  where ε_t is a white noise, i.e. a sequence of i.i.d $\mathcal{N}(0, 1)$
- The processes is called autoregressive process of order 1, **AR(1)**.
- For $|\phi_1| < 1$, we have a stationary process.
- Autocorrelation:

$$r_k = \phi^k$$





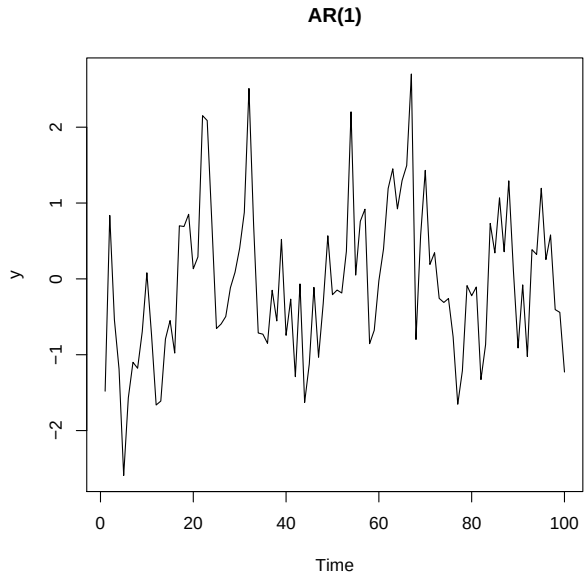
Autoregressive Processes

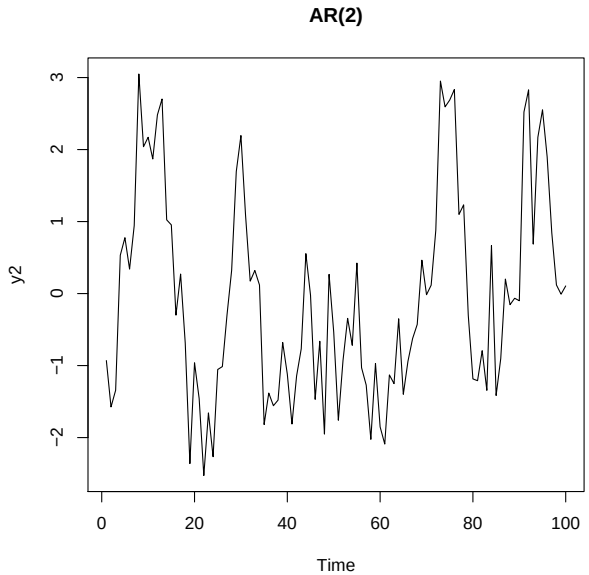
- Autoregression process of order $p > 0$, **AR(p)**

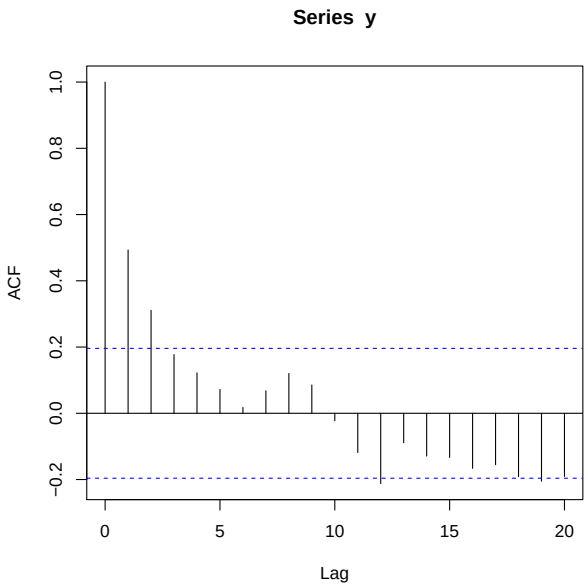
$$y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

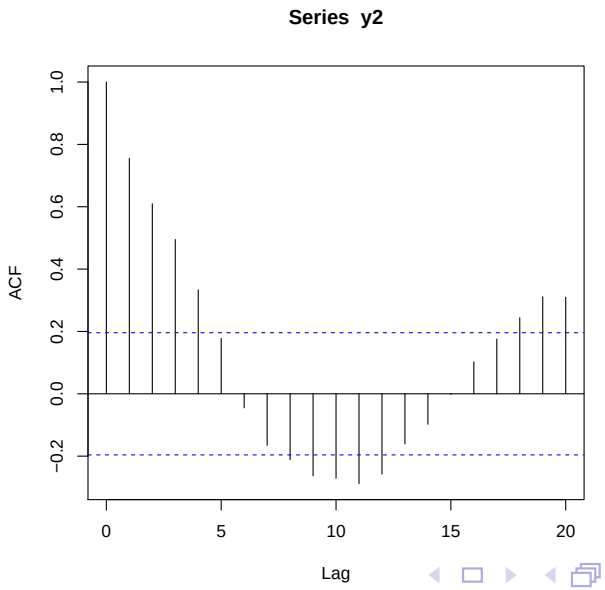
- Like a multiple regression model, except that the regressors are just the past values of the series.
- Autoregressive series are stationary processes provided. the variance of the terms are finite and this will depend on the value of the ϕ's
- The autocorrelation r_k decays to zero quickly for stationary process as a wave.

Autoregressive Processes









Moving Average Processes

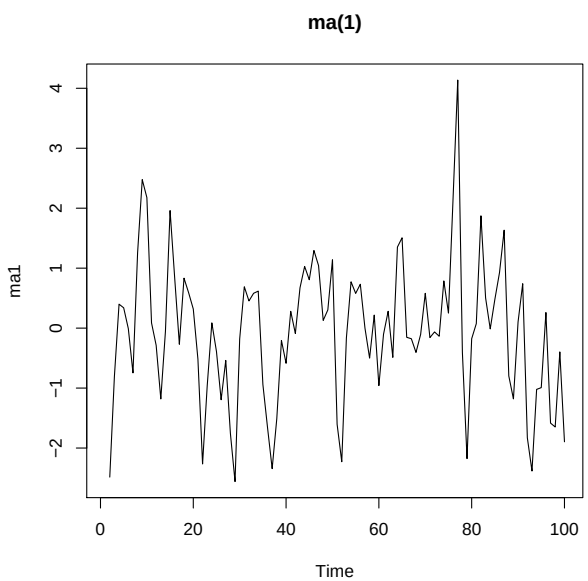
- Moving average processes can be useful for modeling series that are affected by a variety of unpredictable events where the effect of these events have an immediate effect as well as possible longer term effects.
- Let $\{\varepsilon_t\}_{t=1,2,\ldots}$ be a sequence of i.i.d random variables (usually $\mathcal{N}(0,1)$)
- Moving average of order 1: **MA(1)**

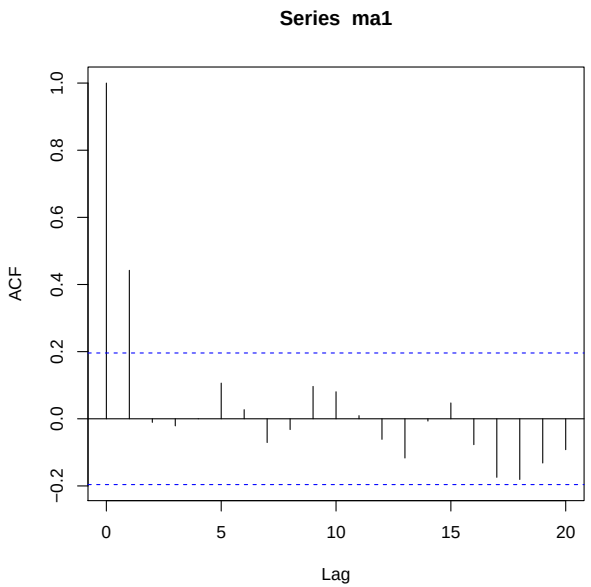
$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$

- The autocorrelation r_1 is different from zero and the other values r_2, r_3, ... are in theory equal to 0; in practice very small.

Moving Average Processes

Moving average of order 1, **MA(1)**:



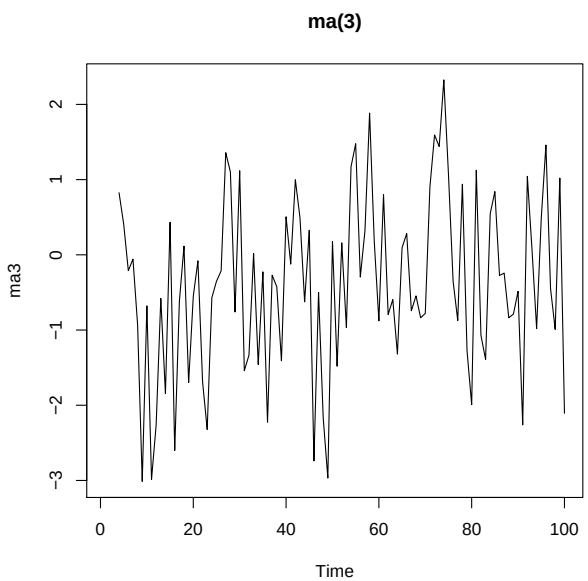


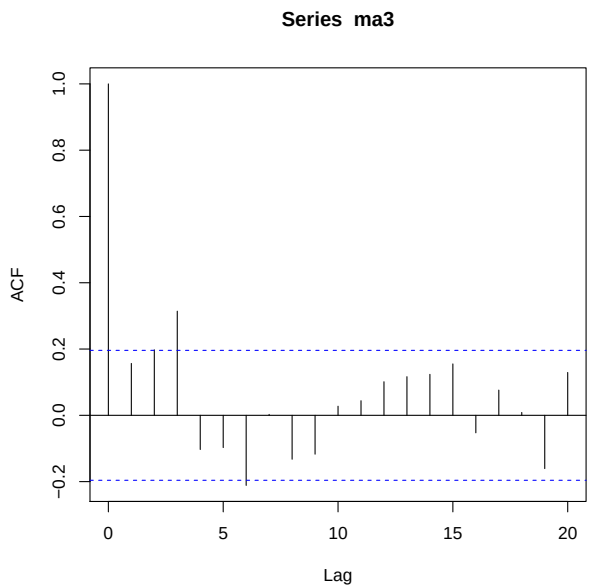
Moving Average Processes

Moving average of order q, **MA(q)**

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}.$$

- Stationary process without any restriction for θ_i.
- the autocorrelation r_q is different from zero and the other values r_{q+1}, r_{q+2}, ... are in theory equal to 0; in practice very small.





ARMA processes

- We can combine the moving average (MA) and the autoregressive models (AR) processes to form a mixed autoregressive/moving average process.
- **ARMA(1,1)** model
$$y_t = \phi_1 y_{t-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1}.$$
- ARMA may adequately model a time series with fewer parameters than using only an MA process or an AR process.
- In general, we can define **ARMA(p,q)** model
- Goal of statistical modelling: use the simplest model possible that still explains the data (**principle of parsimony**).

Identification of a times series model

Two steps:

1. Choosing p and q
   The correlogram can greatly help in determining the appropriate value of p and q for time series data.
2. Fitting the parameters of the model $(\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)$

- Consider for instance a **AR(2)** model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t$$

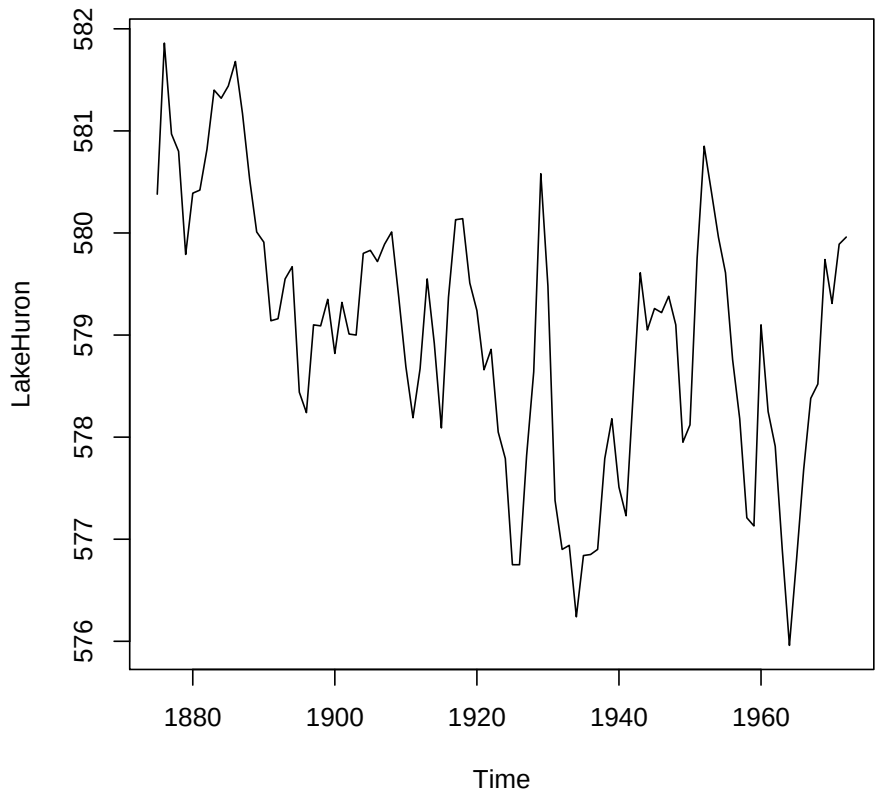
- Sum of squared residuals

$$\text{SSR}(\phi_1, \phi_2) = \sum_{t=3}^{n} \{\underbrace{y_t - (\phi_1 y_{t-1} + \phi_2 y_{t-2})}_{\text{residual}}\}^2$$

- The pair $(\hat{\phi}_1, \hat{\phi}_2)$ that minimizes $\text{SSR}(\phi_1, \phi_2)$ identifies the best (auto)regression in terms of the method of least squares
- Actually there are several estimation methods (see `help(ar)`)
- We consider the annual measurements of the level of Lake Huron 1875:

Fitting an Autoregressive Model

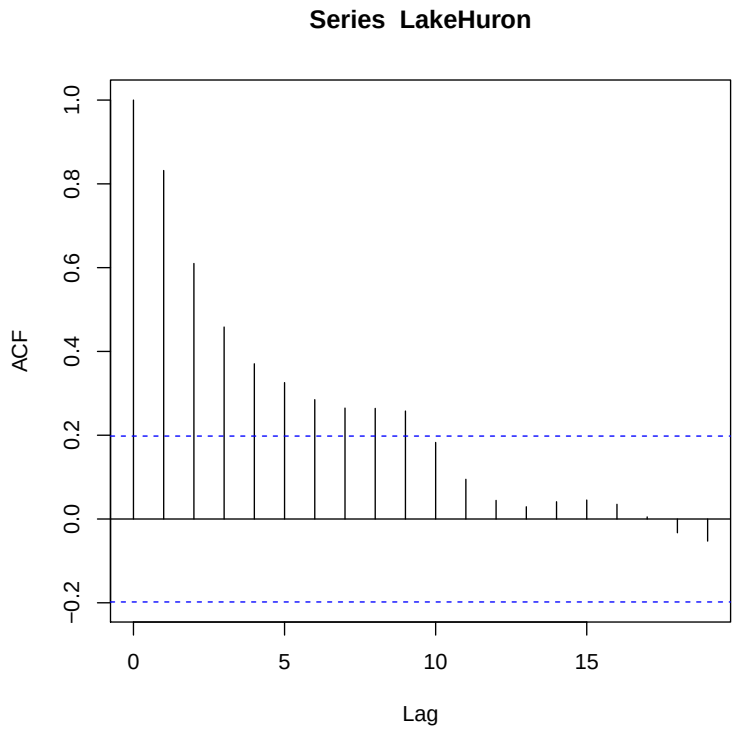
```
> plot(LakeHuron)
```



The correlogram suggests a stationary time series

Fitting an Autoregressive Model

```
> acf(LakeHuron)
```



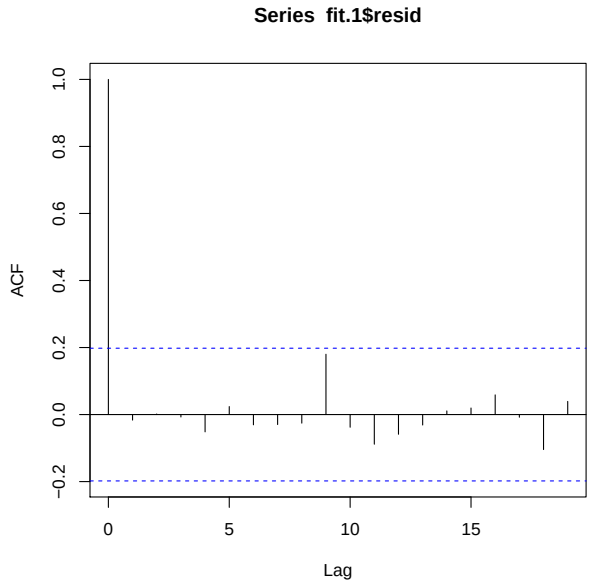
Fitting an Autoregressive Model

- We fit an **AR(3)**

```
> fit.1<-ar(LakeHuron,aic=FALSE,order=3)
```

- and we inspect the correlogram of the residuals

```
> acf(fit.1$resid,na.action=na.pass)
```

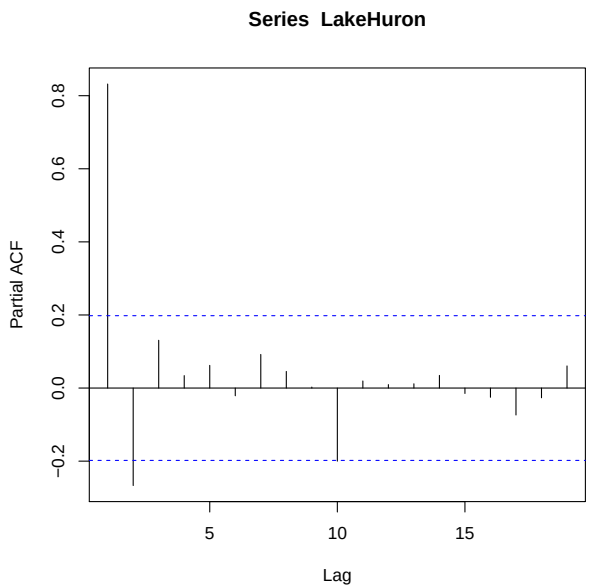


- Determining the order p of an AR process is difficult.
- Correlogram for AR(p) processes for higher orders p can have complicated behaviours

The partial autocorrelation function

- Use the partial autocorrelation function (PACF).
- (Roughly speaking) The partial autocorrelation is the last coefficient ϕ_p in an **AR(p)** model
- It measures the excess correlation at lag p that is not accounted for by an **AR(p - 1)** model.
- Plot of the estimates $\hat{\phi}_k$ of the last coefficient ϕ_k, for $k = 1, 2, \ldots$

```
> pacf(LakeHuron)
```



The plot suggests an AR(2) model

Information Criteria (AIC/BIC)

- In order to choose a model from several competing other models, a popular criterion for making the decision is to use a penalization of the likelihood.
- For a fitted AR time series of length n, the IC are defined to be

$$\begin{aligned} IC &= \text{goodness of fit} + \text{penalty for the complexity} \\ AIC(p) &= n \times \text{logarithm(Sum of square of residuals)} + 2p \\ BIC(p) &= n \times \text{logarithm(Sum of square of residuals)} + p \log n \end{aligned}$$

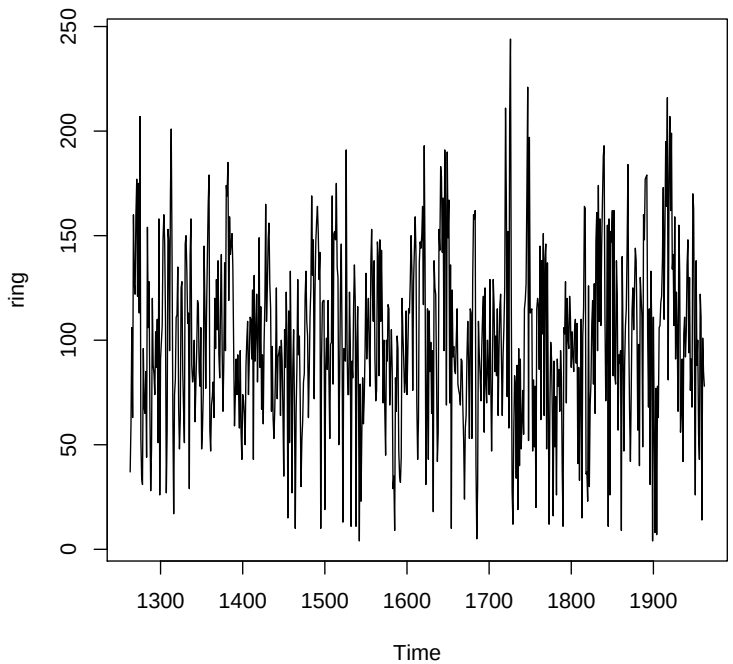
- We choose the minimum AIC/BIC

```
> fit.aic<-ar(LakeHuron)
> fit.aic
Call:
ar(x = LakeHuron)

Coefficients:
1        2
1.0538  -0.2668

Order selected 2  sigma^2 estimated as  0.5075
```

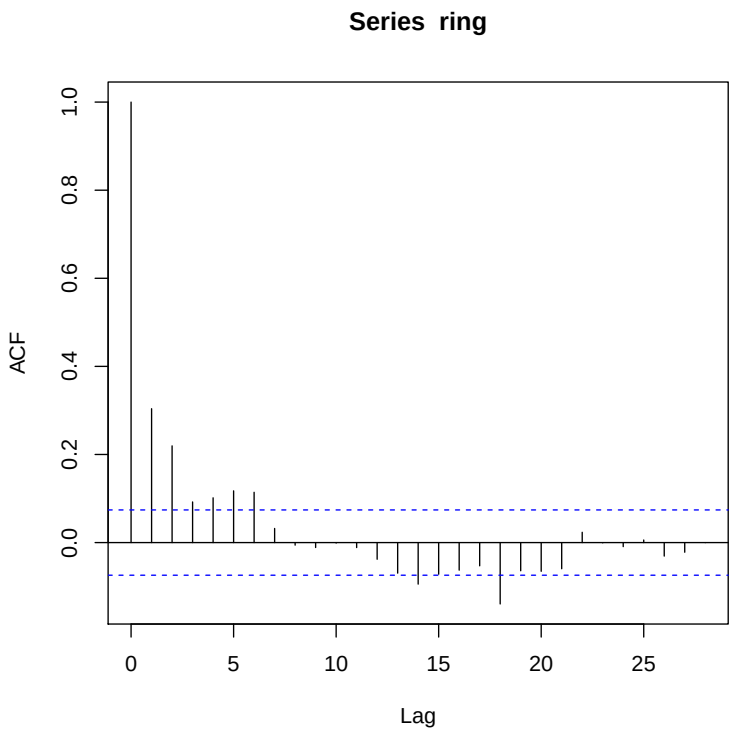
Model identification: example

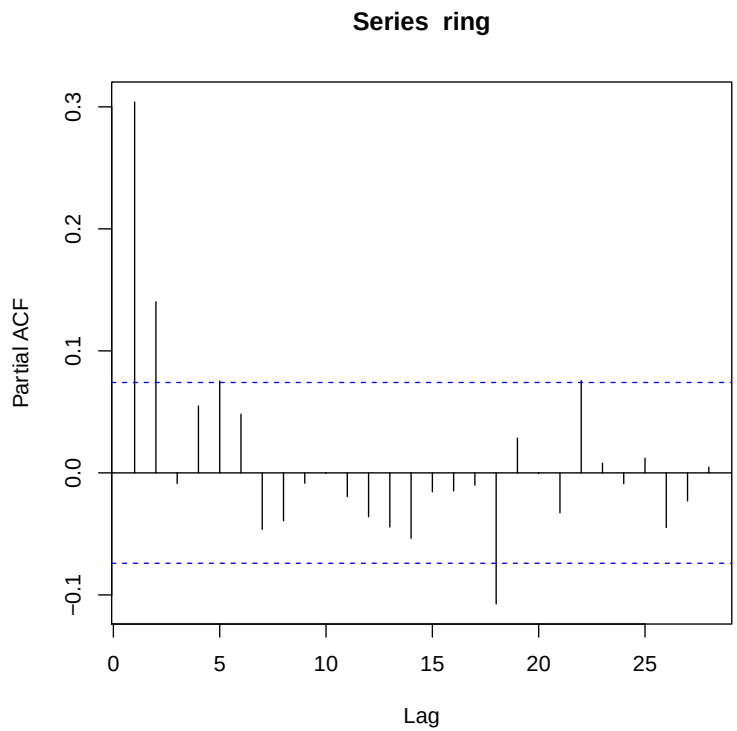
- Time series of 700 tree ring indices for Douglas fir at the Navajo National Monument in Arizona. This data is available from 1263 to 1962 and is listed in a report by Stokes et al. (1973).
- Stationary time series

```
> ring<-ts(scan('navajo.txt'),
+ start=1263,frequency=1)
> plot(ring)
```



Model identification: example





An **ARMA(1,1)** model may adequately model the data

Model identification: example

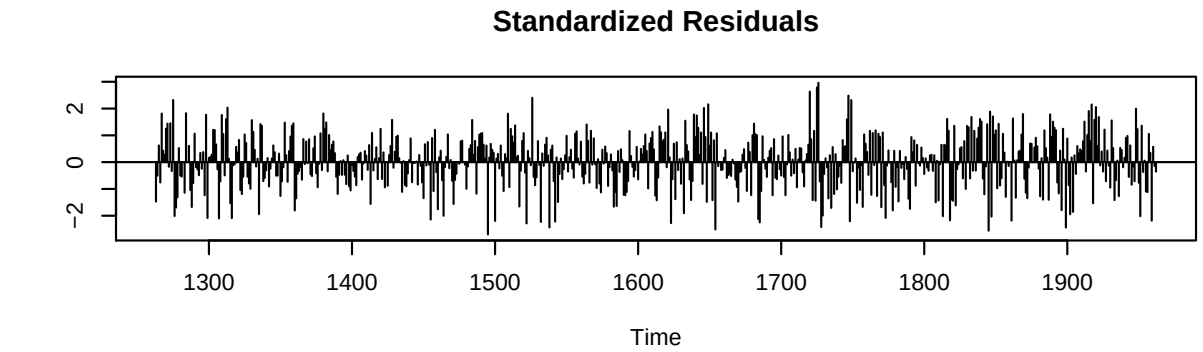
- We fit the model using the function `arima`
- General syntax: `arima(x, order = c(p, d, q))`
- Here

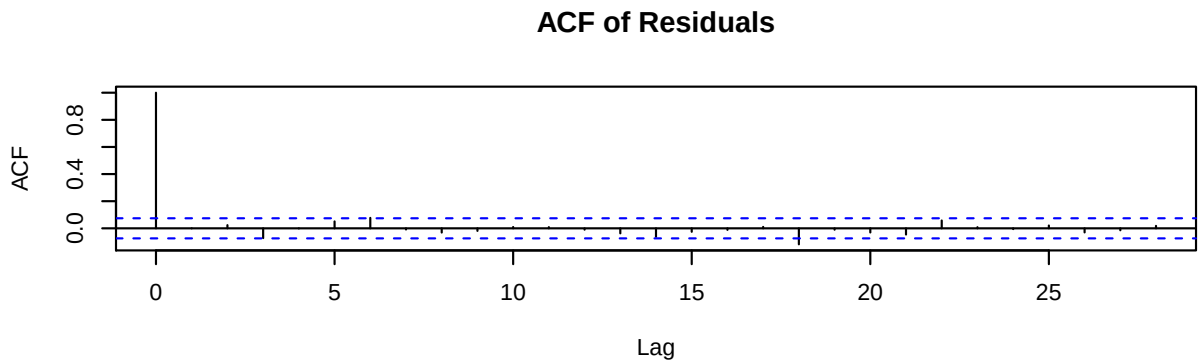
```
> ring<-ts(scan('navajo.txt'),start=1263,frequency=1)
> plot(ring)
> ring.fit<-arima(ring,order=c(1,0,1))
> print(ring.fit)
Call:
arima(x = ring, order = c(1, 0, 1))

Coefficients:
ar1      ma1  intercept
0.6809  -0.4232    99.3762
s.e.  0.0824   0.1031     2.7280

sigma^2 estimated as 1601:  log likelihood = -3575.67,  aic = 7159.34
```

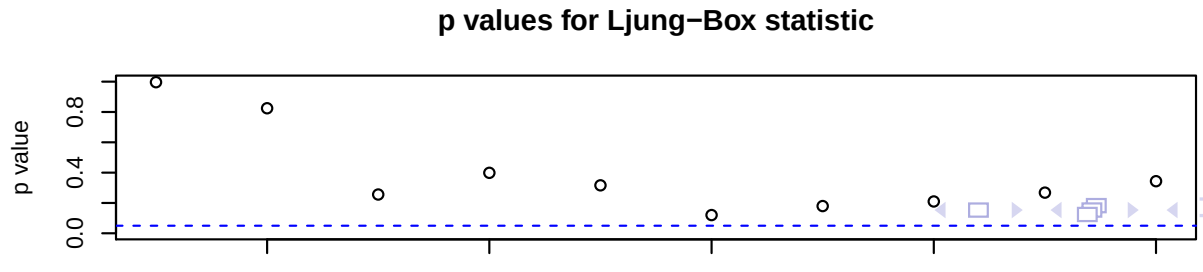
Model identification: example

Let us look at the residuals

```
> tsdiag(ring.fit)
```







Model identification: example

Main features in the three plots

- No **outliers** (standardized residuals are in the interval [-3,3])
- No autocorrelation of the standardized residuals
- Observed P-value of the **Liung-Box (LB) statistics** indicates no autocorrelation for the residuals
  - LB is based on the sum of the first k squared autocorrelation coefficients for the residuals
  - P-values at different values of k are indicators of the randomness of the standardized residuals
  - a p-value greater 0.05 points out that the time series of the standardized residuals looks like a realisation of a white-noise

Model identification: example

A competitive model could be a **MA(2)**

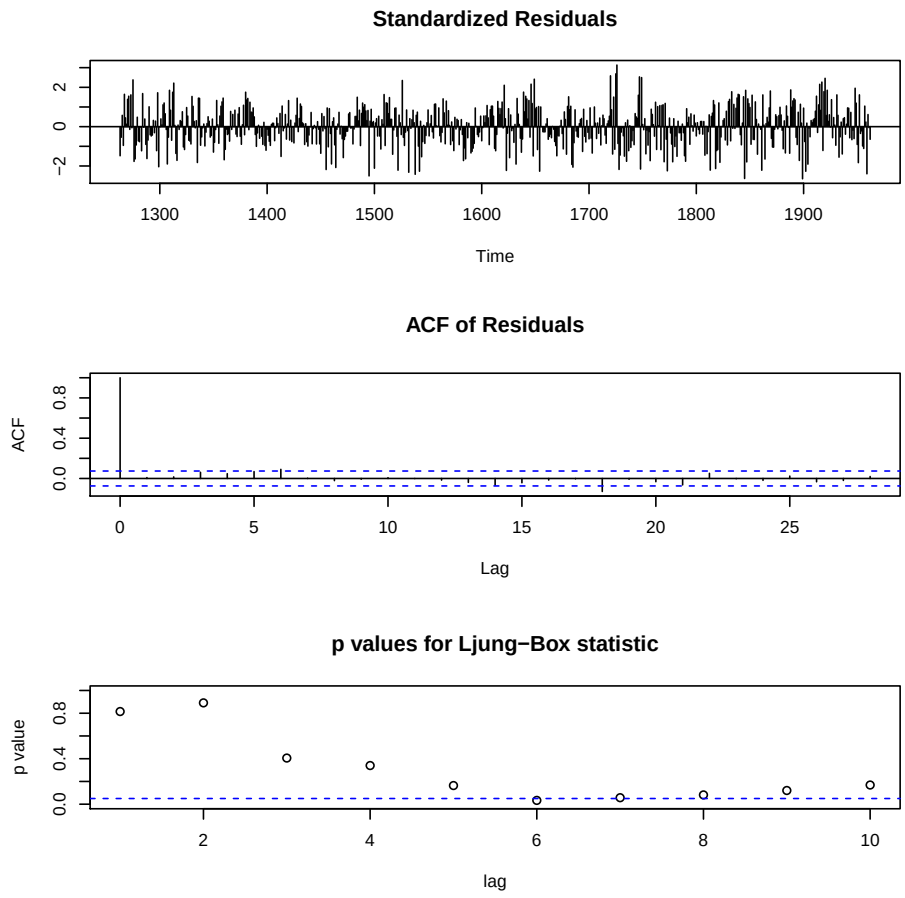
```
> ring.fit2<-arima(ring,order=c(0,0,2))
> print(ring.fit2)
Call:
arima(x = ring, order = c(0, 0, 2))

Coefficients:
ma1     ma2  intercept
0.2601  0.1969     99.449
s.e.  0.0369  0.0366      2.207

sigma^2 estimated as 1608:  log likelihood = -3577.3,  aic = 7162.59
```

Let us we look at the residuals

Model identification: example

Residual plot:

```
> tsdiag(ring.fit2)
```



- both models pass the diagnostic check.
- for contrasting a set of models consider the AIC criterion and choose the minimum.

Forecasting

Once a model has been identified and its parameters have been estimated, one important purpose is to predict future values of a time series.

Reminders

- Forecast for linear regression model. The model for a new value (not necessarily in the future) is

$$y_{i+1} = a + bx_{i+1} + \varepsilon_{i+1}$$

- we assume the knowledge of x_{i+1} and the prediction (**plug-in prediction**), after estimating the parameters, is

$$\hat{y}_{i+1} = \hat{a} + \hat{b}x_{i+1}$$

Use the same approach in time series

Forecasting **AR(1)** time series: one-step-ahead

- We know the values $y_1, \ldots, y_t$ (present and past) and we want to predict y_{t+1}
- From now the forecast value will be indicated by $\hat{y}_t$
- Forecast for an **AR(1)** model:

$$y_{t+1} = \phi_1 y_t + \varepsilon_{t+1}$$

- we know the present and past values, so the plug-in prediction is

$$\hat{y}_{t+1} = \hat{\phi}_1 y_t$$

- the **one-step-ahead forecasting error** (residual) is

$$\hat{\varepsilon}_t = y_t - \hat{y}_t$$

Forecasting **MA(1)** time series: one-step-ahead

- Model **(MA(1))** is
$$y_{t+1} = \varepsilon_{t+1} + \theta_1 \varepsilon_t$$
- Since we know the present past values, we can compute previous values of the residuals
$$\hat{\varepsilon}_t = y_t - \hat{y}_t;\ \hat{\varepsilon}_{t-1} = y_{t-1} - \hat{y}_{t-1}; \cdots$$
- The forecast is thus
$$\hat{y}_{t+1} = \hat{\theta}_1 \hat{\varepsilon}_t$$
- Forecasts are calculated recursively.

Forecasting **ARMA(1,1)** time series: one-step-ahead

- The model (ARMA(1,1)) is

$$y_{t+1} = \phi_1 y_t + \varepsilon_{t+1} + \theta_1 \varepsilon_t$$

- Since we know the present past values, we can compute previous values of the residuals
- The forecast is thus

$$\hat{y}_{t+1} = \hat{\phi}_1 y_t + \hat{\theta}_1 \hat{\varepsilon}_t$$

Important

- We expect that the mean of $\hat{\varepsilon}_t$ is approximately equal to zero
- A measure of forecast accuracy: **prediction mean square** error (PMSE) which is the mean of the square of one-step-ahead forecasting errors.

Important Remarks

- We can similarly do k-step ahead forecast: predicting y_{t+k} using the present and the past.
- An interval forecast usually consists of an upper and lower limit between which a future value is expected to lie with a prescribed probability.
- The length of the interval is related to the variability of the k-step ahead forecast errors; it increases with k
- a general espression is

$$[\hat{y}_t(k) - cv(k), \hat{y}_t(k) + cv(k)]$$

where c is a constant related to the prescribed probability and $v(k)$ is standard deviation of of the k-step ahead forecast error [For a (approximate) 95% interval forecast one sets $c = 2$.]

Forecasting in R

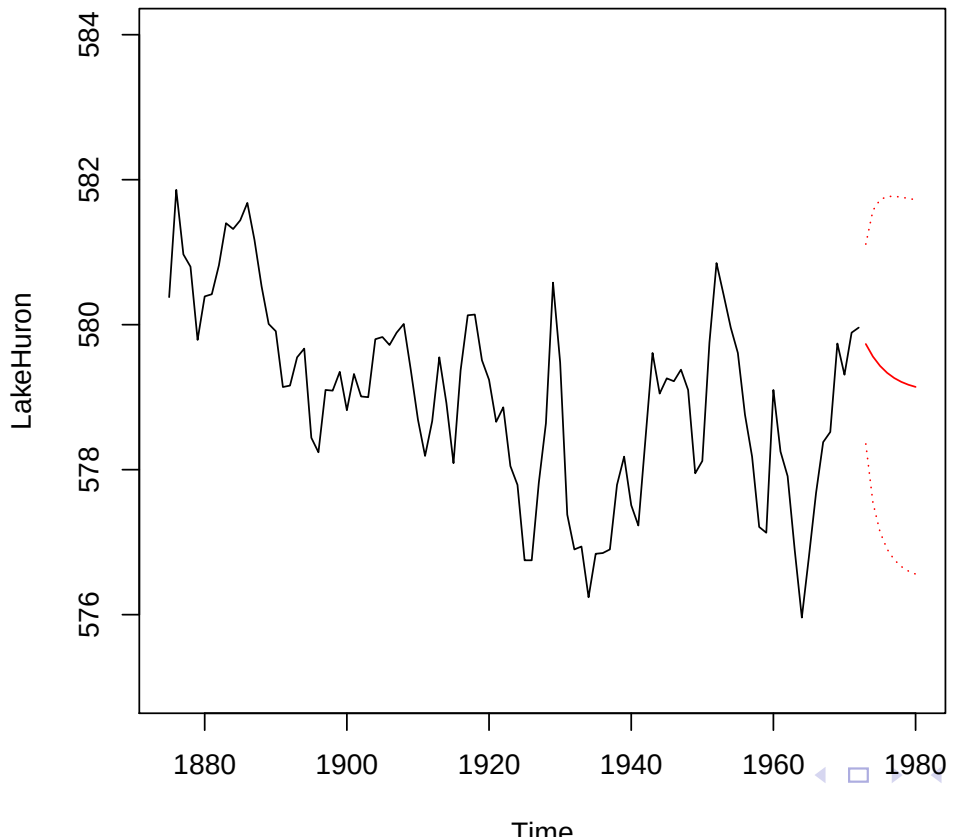
- Assume, that we are satisfied with the fit of an **ARIMA(1,0,1)**–model to the Lake Huron–data:
- We wish to prediction for the next 8 years

```
> fit<-arima(LakeHuron,order=c(1,0,1))
> LH.pred<-predict(fit,n.ahead=8)
> LH.pred
$pred
Time Series:
Start = 1973
End = 1980
Frequency = 1
[1] 579.7334 579.5604 579.4316 579.3357 579.2642 579.2109 579.1713 579.1417

$se
Time Series:
Start = 1973
End = 1980
Frequency = 1
[1] 0.6891588 1.0070366 1.1459938 1.2162684 1.2535636 1.2737869 1.2848710
[8] 1.2909802
```

Forecasting in R

Plot of the results

```
> plot(LakeHuron,xlim=c(1875,1980),ylim=c(575,584))
> LH.pred<-predict(fit,n.ahead=8)
> lines(LH.pred$pred,col="red")
> lines(LH.pred$pred+2*LH.pred$se,col="red",lty=3)
> lines(LH.pred$pred-2*LH.pred$se,col="red",lty=3)
```



Unit 4
Introduction to Spatial Statistics

Spatial Data

- Observations with explicit information about location (coordinates)
- Spatial structure is important for the observed variables: Tobler's law

  "observations taken at sites close together tend to be more alike than observations taken at sites far apart"

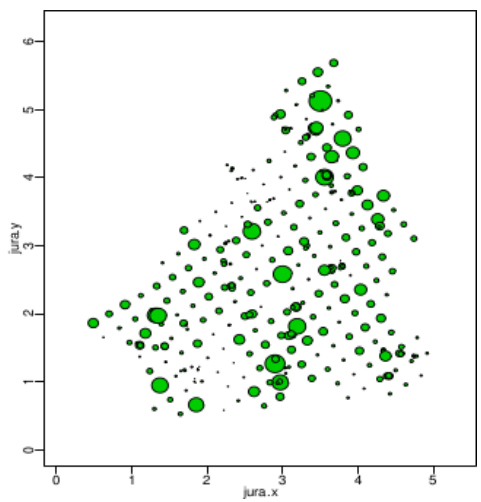
- Non-spatial analyses of spatial data may yield incorrect statistical results
- Regression analysis that "forgets" dependence (any type of dependence: time series, spatial, . . .) may suggest that some predictors are important while they are not

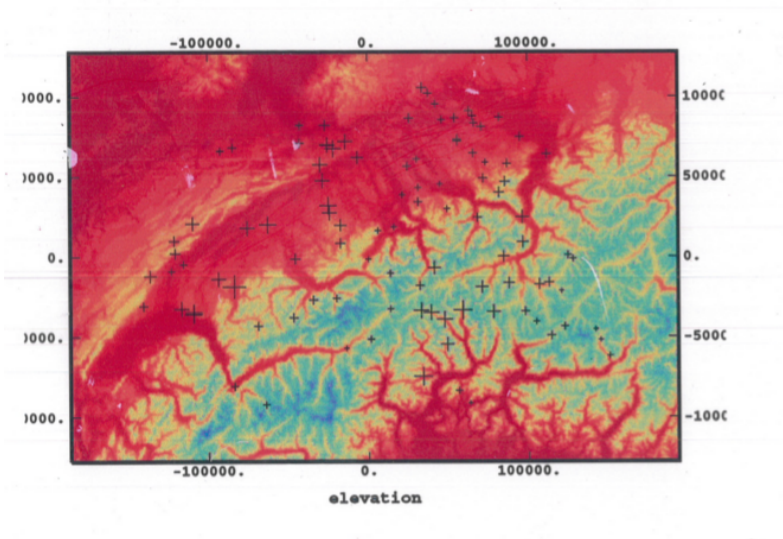
Geostatistical Data

- The variable is defined at any location of the domain
- But it is measured at some limited number of points

e.g., temperatures, precipitations, soil data, air pollution, ...

- Characterizing spatial variations: variograms
- Predicting (interpolating) the variable at unmeasured locations: kriging
- Evaluating the prediction error, and the sampling design
- Simulating random processes with similar spatial variations

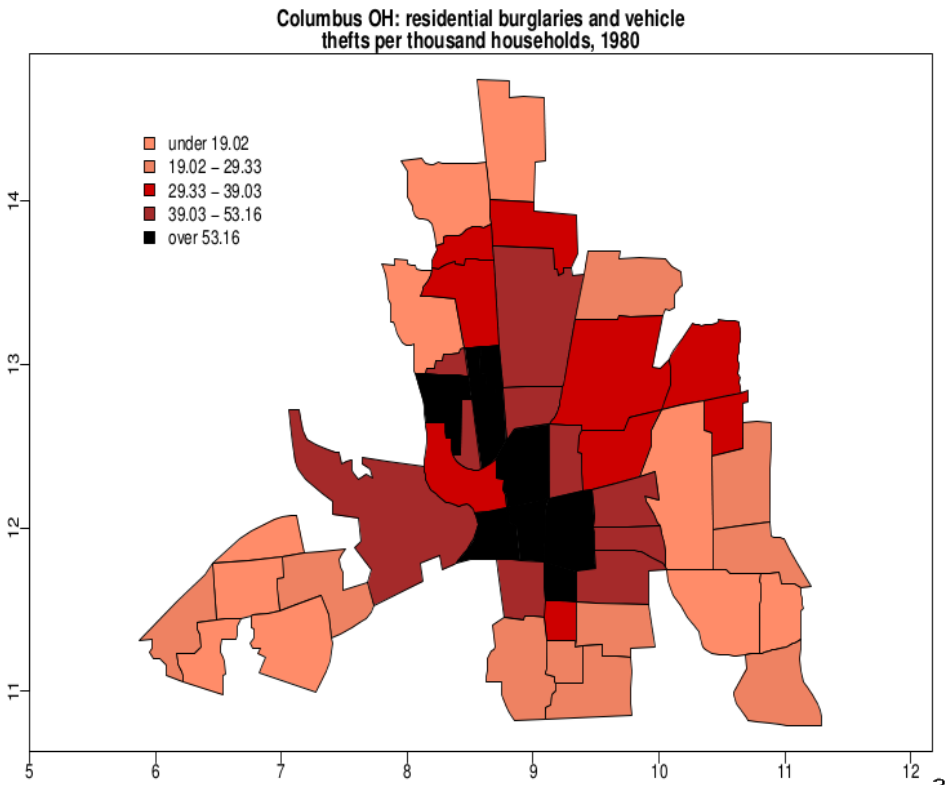




Lattice Data

- Population related data; epidemiology ...
- data collected on administrative units; spatial econometrics
- Remote sensing data

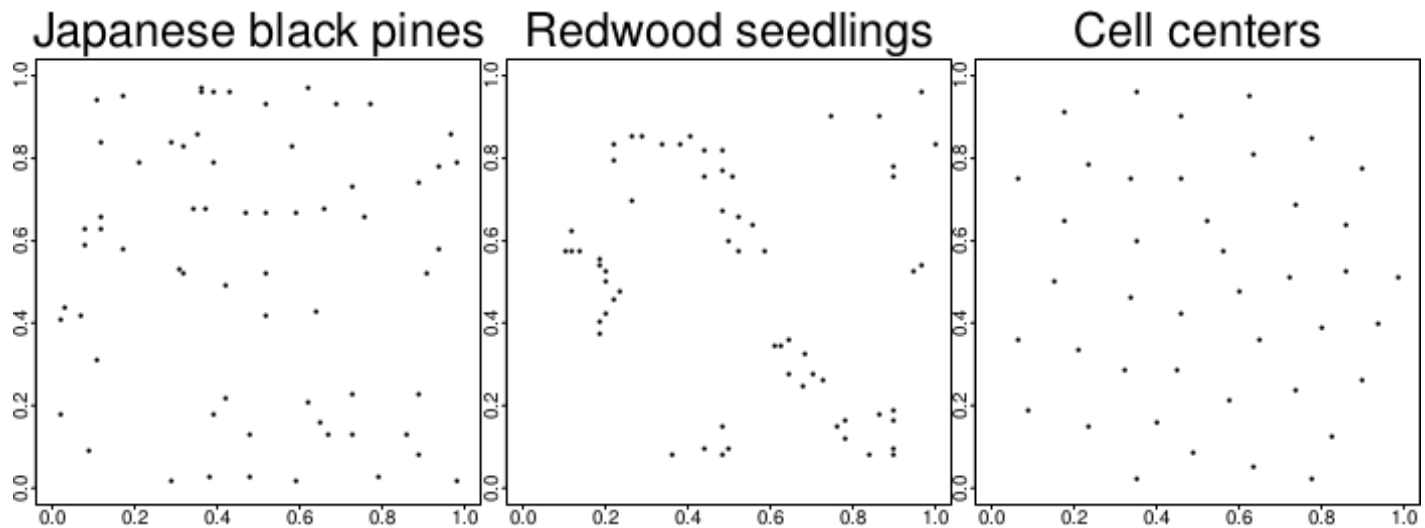
- Characterizing spatial variations
- Testing independence between neighbours; residual analysis



Point and object processes

- Location of trees (and other, kind of plants) in a forest; location of animals
- Location of earthquakes; avalanches; any type of natural hazards

- Characterizing the spatial distribution; complete randomness, regularity, clustering?
- Relating the density of points, objects with respect to available co-variates
- Simulating spatial distributions of points and objects



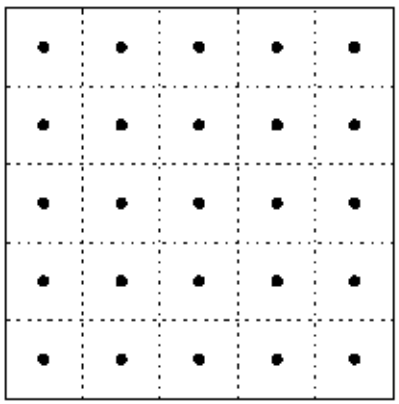
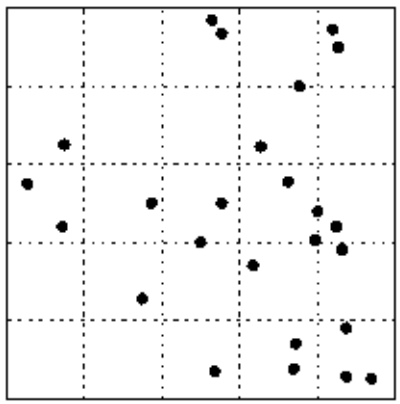
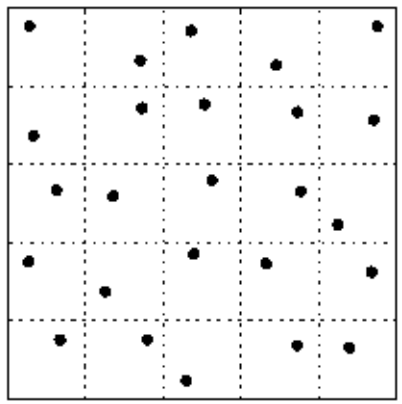
Difficulties

- Quite often, unique realization, with no replicates. For example: one pollution event, one geological site, etc...
  Hence several theoretical problems, since usually statistics is based on replicates
- Sometimes the studied area is clearly defined: limits of a country, of a region, ...
  Sometimes it is part of the variable under investigation: soil pollution, orebody, fish stocks, ...
- Possible bias: samples in ”interesting areas”

Unbiased sampling

Three sampling approaches are always unbiased

- *Random sampling*
  Inefficient coverage of the area, with redundancies and voids
  but samples efficiently short distances
- *Regular sampling*
  Good coverage of the area,
  but no information at distances smaller than the mesh of the grid
- *Stratified random sampling*
  Good balance between coverage and information at all distances

Specifics difficulties with spatial data

- Non independent data
- No ordering when $d \geq 2$
- Two types of asymptotics (i.e., when $N \to \infty$). Increasing the domain, or densifying the data in fixed area
- Likelihood methods not always adapted
- Sometimes: border effects, string dependences,...

Organization of the course

- Introduction to two main concepts in geostatistics: variograms, kriging
- with many illustrations
- ”Swiss Jura” data-set
- R Scripts

Geostatistics

- The term geostatistics was coined by G. Matheron (1962)
- Matheron and his colleagues (in Fontainebleau, France) used this term for prediction problems in the mining industry
- The prefix 'geo' concerns data related to earth
- D. G. Krige (1919–2013)and Matheron (1930 – 2000) formulated the theory of geostatistics and kriging in the 1960s
- Today, geostatistical methods are applied in many areas beyond mining, such as
  - soil science
  - epidemiology
  - ecology
  - forestry
  - meteorology
  - astronomy
  - social sciences
  - ...
- and, more in general, in all applications where data are collected at geographical locations

Terminology

- Domain D, here, part of $\mathbb{R}^2$
- Arbitrary point in D, $\mathbf{s} \in D$.
- Observed spatial locations $\mathbf{s}_1, \ldots, \mathbf{s}_n$, $\mathbf{s}_i \in D$
- Observed values $z(\mathbf{s}_1), \ldots, z(\mathbf{s}_n)$
- The function $z(\cdot)$ is called a regionalized variable
- We assume $z(\cdot)$ is one realization of a random field $Z(\cdot)$

$\rightsquigarrow$ We will need Stochastic models for random fields

Distance

- 1D: coordinates s on a line w.r.t. some origin (0)
- 2D: coordinates $\boldsymbol{s}$ on a grid w.r.t. some origin $(0, 0)$, $\boldsymbol{s} = (s_1, s_2) = (x, y) = (E, N)$
- 3D: coordinates $\boldsymbol{s}$ are grid and elevation from a reference value, $\boldsymbol{s} = (s_s, s_2, s_3) = (x, y, z) = (E, N, H)$
- In the analysis of spatial data the distance between the data points is very important
- In this course, focus only on Euclidean distances: 2D distance between points $\boldsymbol{s}$ and $\boldsymbol{s}'$ is

$$d(\boldsymbol{s}, \boldsymbol{s}') = \sqrt{(s_{i1} - s'_{i1})^2 + (s_{i2} - s'_{i2})^2}$$

- Many other types of distances, *e.g.*
  - great-circle distance
  - azimuth distance
  - travel distance from point to point
  - time needed to get from point to point
- Latitude-longitude coordinates needed to be transformed to grid coordinates in some 2D projection

Function `spDists()` from package **sp** useful to transform distances from longitude-latitude system to Euclidean system

spDistsN1 {sp} R Documentation

Euclidean or Great Circle distance between points

Description

The function returns a vector of distances between a matrix of 2D points, first column longitude, second column latitude, and a single 2D point, using Euclidean or Great Circle distance (WGS84 ellipsoid) methods.

Usage

```
spDistsN1(pts, pt, longlat = FALSE)
spDists(x, y = x, longlat = FALSE)
```

Arguments

`pts`	A matrix of 2D points, first column x/longitude, second column y/latitude, or a SpatialPoints or SpatialPointsDataFrame object
`pt`	A single 2D point, first value x/longitude, second value y/latitude, or a SpatialPoints or SpatialPointsDataFrame object with one point only
`x`	A matrix of n-D points with row denoting points, first column x/longitude, second column y/latitude, or a Spatial object that has a coordinates method
`y`	A matrix of n-D points with row denoting points, first column x/longitude, second column y/latitude, or a Spatial object that has a coordinates method
`longlat`	if FALSE, Euclidean distance, if TRUE Great Circle distance

Value

`spDistsN1` returns a numeric vector of distances in the metric of the points if longlat=FALSE, or in kilometers if longlat=TRUE.

`spDists` returns a full matrix of distances in the metric of the points if longlat=FALSE, or in kilometers if longlat=TRUE; it uses `spDistsN1` in case points are two-dimensional. In case of `spDists(x,x)`, it will compute all n x n distances, not the sufficient n x (n-1).

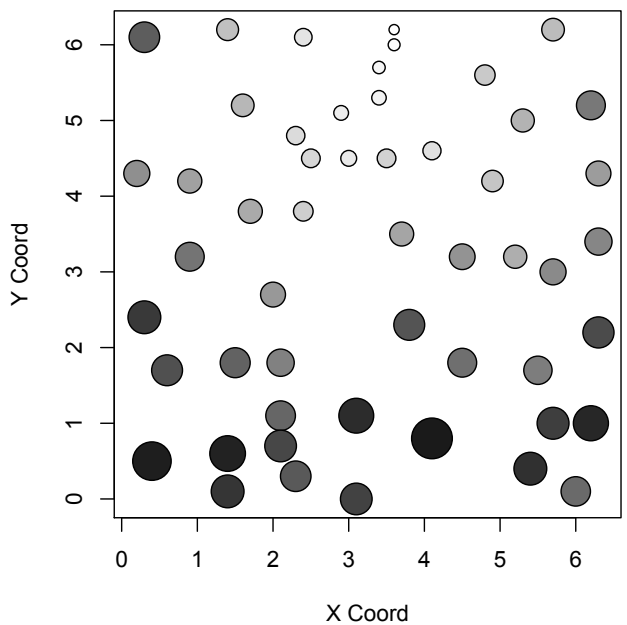
Elevation Data

- Data $(z_i, \boldsymbol{s}_i), i = 1, \ldots, 52$
- z_i is the surface elevation with one unit corresponding to 10 feet ($\sim$ 3.05 meters) of elevation
- $\boldsymbol{s}_i$ locations within square $\mathcal{A}$
- Unit distance is 50 feet ($\sim$ 15.24 meters)
- Target: construction of a continuous elevation map for the *whole* square region
- Source: Davis (1972). *Statistics and Data Analysis in Geology*. Wiley
- Available in **geoR** as the `elevation` data

Elevation Data

]

```
> install.packages("geoR")
> library("geoR")
> data(elevation)
> help(elevation)
> points(elevation)
> help(points.geodata)
> points(elevation, cex.min=1, cex.max=4, col="gray")
```

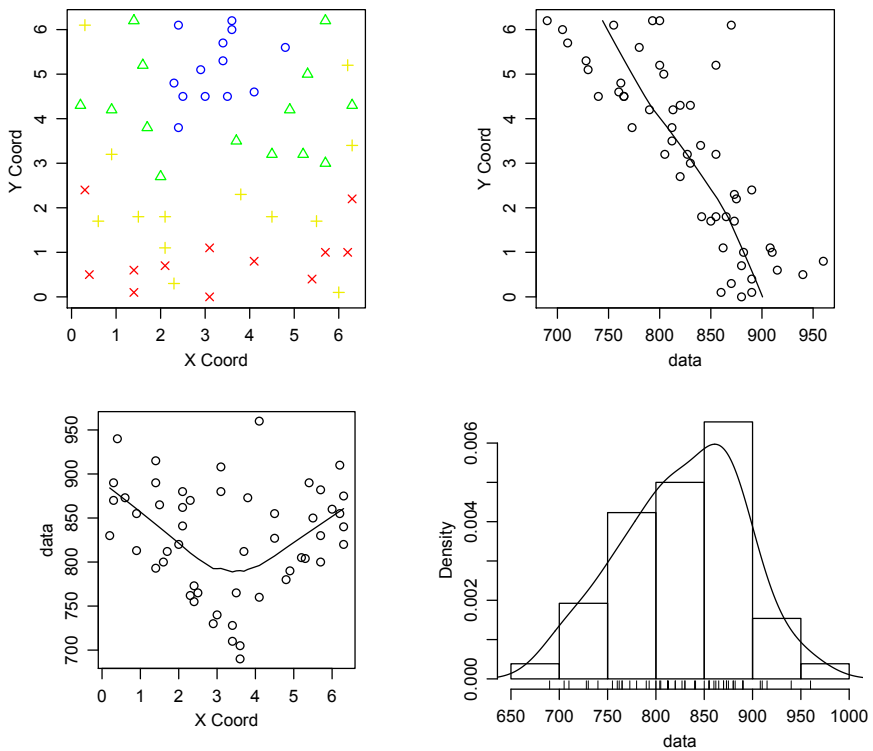


Elevation Data

```
> summary(elevation)
$n
[1] 52

$coords.summary
x   y
min 0.2 0.0
max 6.3 6.2

$distances.summary
min      max
0.200000 8.275869

$data.summary
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
690.0   787.5   830.0   827.1   873.0   960.0

attr(,"class")
[1] "summary.geodata"
```

Elevation Data

```
> plot(elevation, lowess=T)
```



`geodata` Objects

- `geodata` is a list with obligatory and optional components
- Obligatory components:
  - `coords` matrix with 2D coordinates
  - `data` vector with measurements (responses) at the locations corresponding to `coords`
- Optional components:
  - `borders` matrix with coordinates defining the boundary of the study area
  - `covariate` matrix with covariates
  - `...`

`geodata` Objects

- Swiss Jura
- ASCII file `jura.dat`

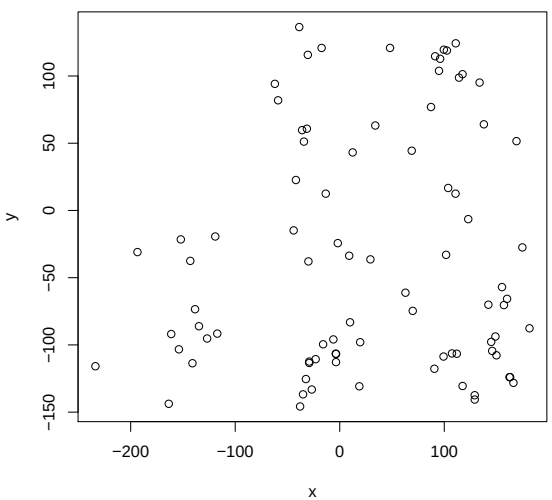
```
> jura <- read.geodata("jura.dat", header = T,data.col=3:11, skip = 22)
> names(jura)
$coords
[1] "X" "Y"

$data
[1] "Rock" "Land" "Cd"   "Cu"   "Pb"   "Co"
[7] "Cr"   "Ni"   "Zn"
```

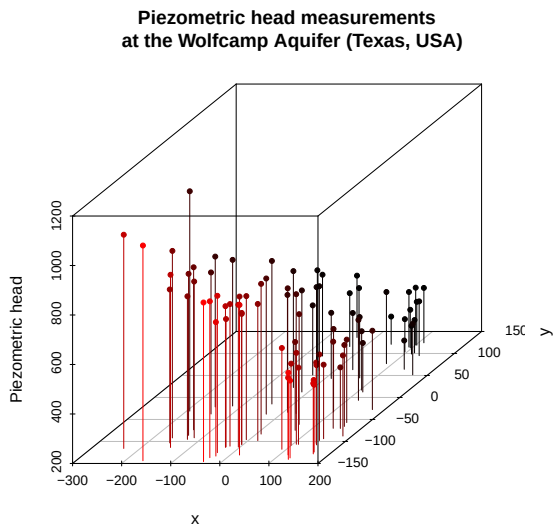
Spatial structure

- The observations are suspected of having a coherent spatial structure, the characterization of which may be important.
- Spatial variations can be decomposed into two components:

$$\begin{array}{lcccc} \text{Data} & = & \text{large-scale variation} & + & \text{small-scale variation} \\ z(s) & = & \mu(s) & + & \varepsilon(s) \end{array}$$

- Example Piezometric head measurements taken at the Wolfcamp Aquifer, Texas, USA.





Exploring the large scale variation

Exploration of the large scale variation can be considered as a usual regression problem. Available tools include

- simple interpolation
- trend surface analysis: linear regression with spatial coordinates (or, e.g., their polynomial functions, as well as other attributes measured at observation points) acting as covariates
- spatial moving averages (including nearest neighbour methods)
- non parametric regression, as wih `loess`

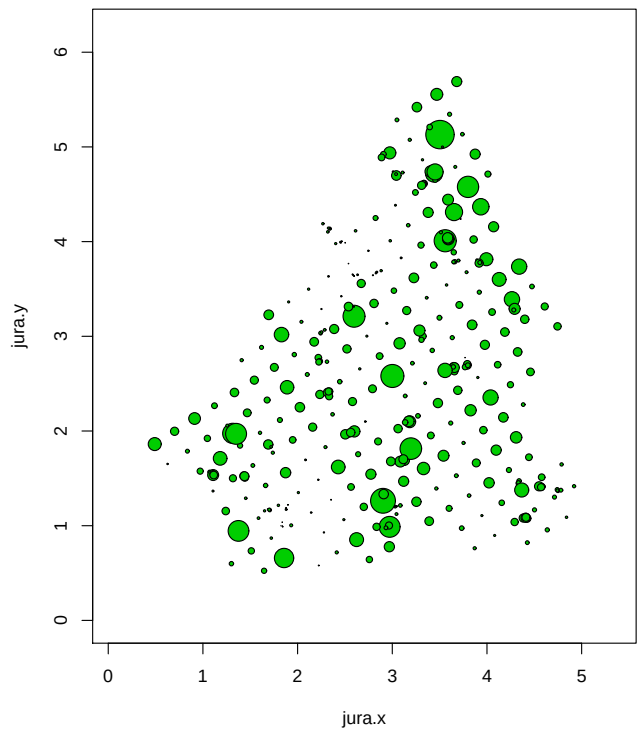
Quite similar to trends in times series. Not further detailed in this class

Exploring spatial dependence

Tobler's first law of geography[a]

[a](Tobler, 1970). Economic Geography, 46(20):234-240

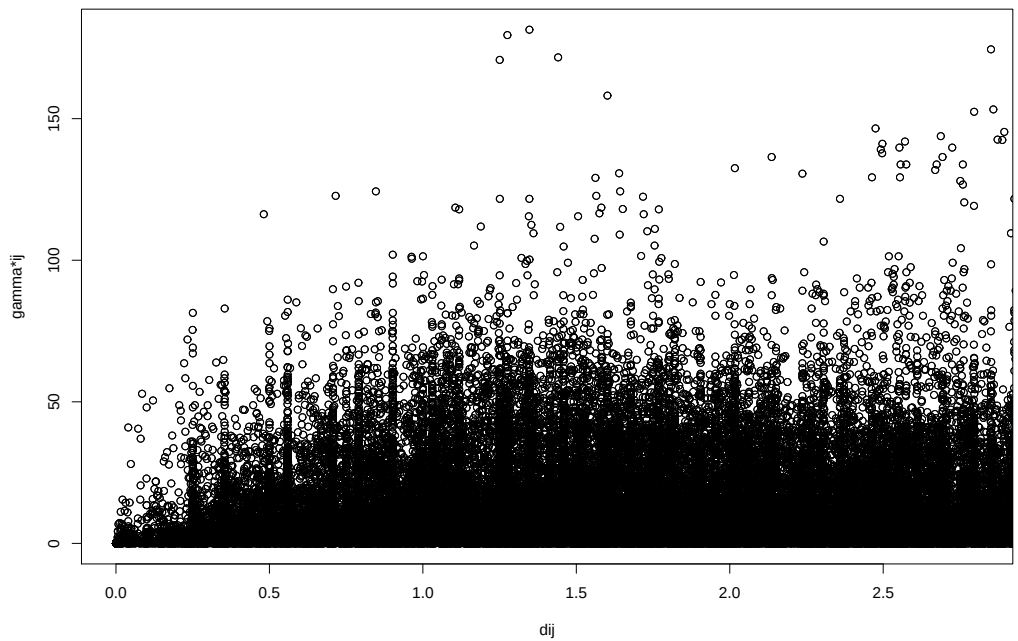
"Everything is related to everything else, but near things are more related than distant things"



Variogram cloud

Semi squared variation:

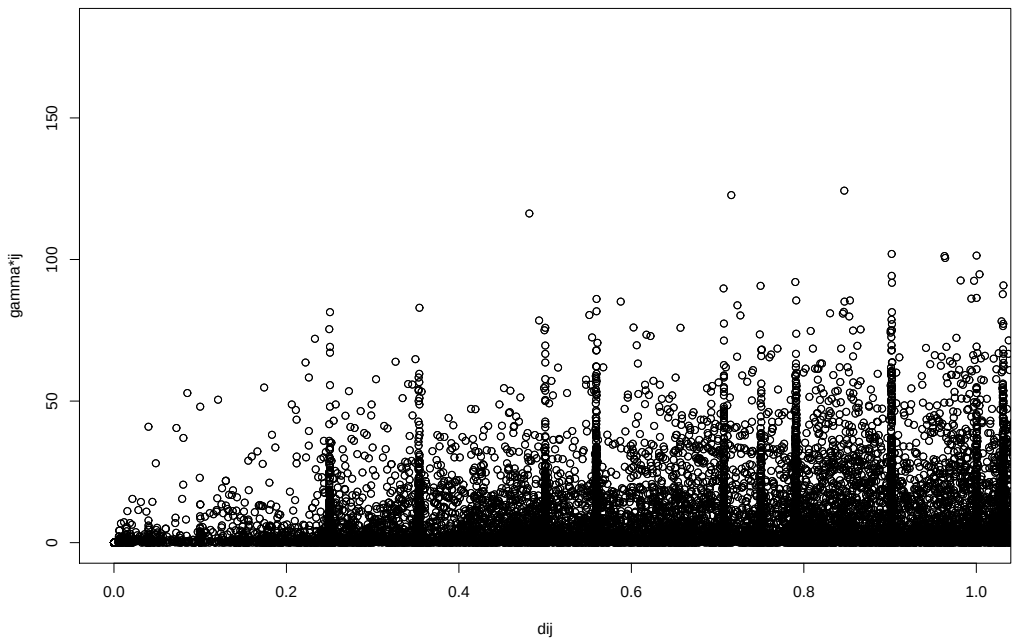
$$\gamma_{ij}^{*} = \frac{(z(s_i) - z(s_j))^2}{2} \qquad d_{ij} = \|s_i - s_j\|$$



Variogram cloud

Semi squared variation:

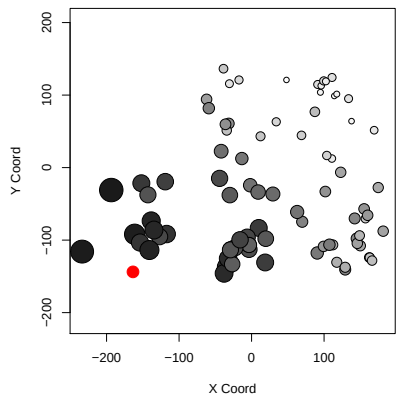
$$\gamma_{ij}^* = \frac{(z(s_i) - z(s_j))^2}{2} \qquad d_{ij} = \|s_i - s_j\|$$

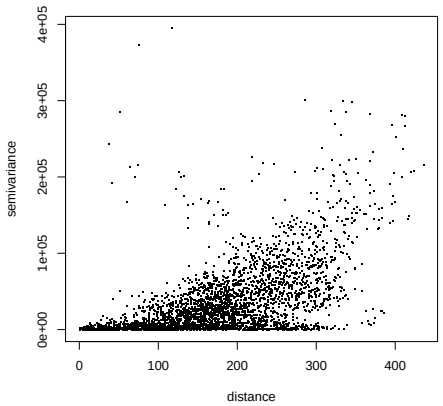


Zoom between $|h| = 0$ et $|h = 1|$.

The variogram cloud

- Shows the variation within all pairs of points as a function of their separation distance;
- Too many points: hard to interpret;
- Allows the identification of outliers; Shows which point-pairs do not fit the general pattern (outliers)





The empirical variogram

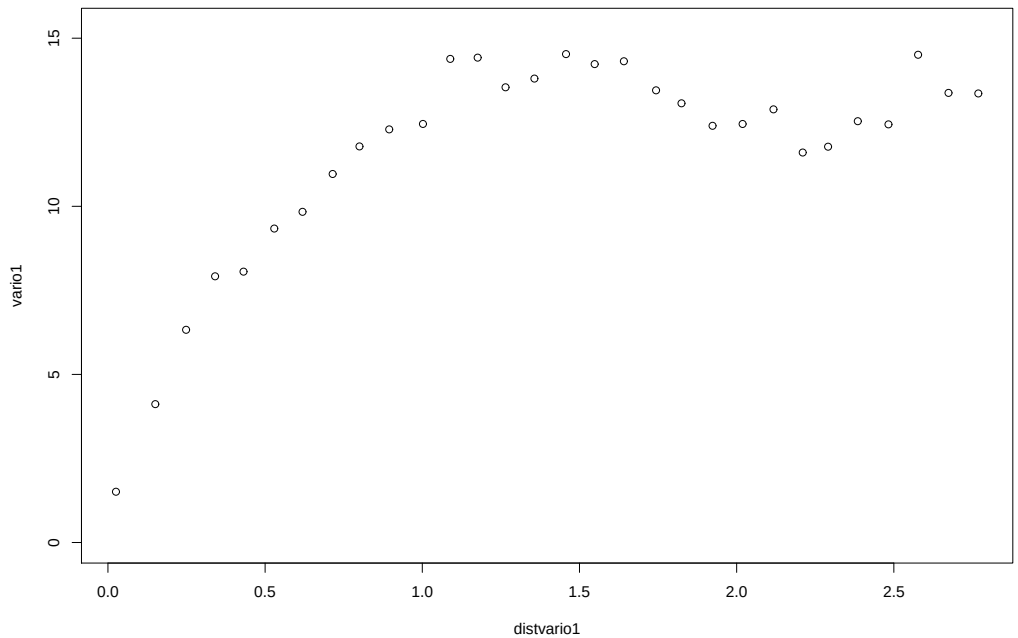
- To summarize the variogram cloud, we group the distances into lags (separation bins, like a histogram)
- We then compute the average of the semi squared variation of all the point-pairs in the bin. This defines the semi-variance
- The empirical variogram is the graph of semi-variance as a function of distance lags:

$$\hat{\gamma}(d_k) = \frac{1}{2n_k} \sum_{i,j;d_{ij} \simeq d_k} (Z(s_i) - Z(s_j))^2$$

where n_k is the number of point pairs separated by distance d_k (up to some tolerance)

The empirical variogram

$$\hat{\gamma}(d_k) = \frac{1}{2n_k} \sum_{i,j;d_{ij} \simeq d_k} (Z(s_i) - Z(s_j))^2$$



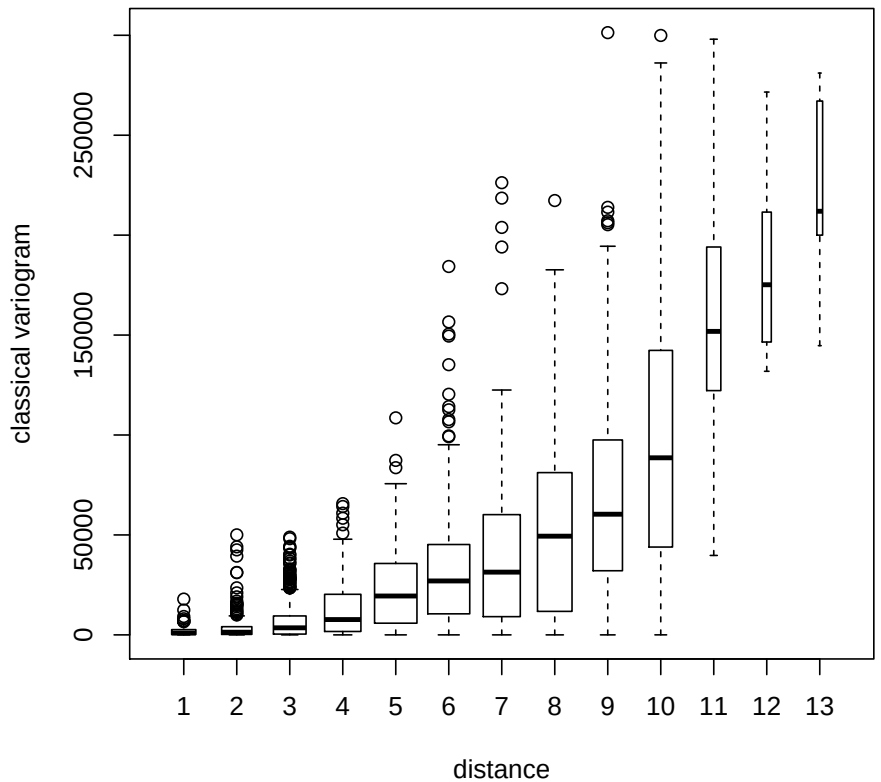
Defining the bins

Some practical considerations for defining the bins:

- Each bin should have enough points to give an accurate estimate of the semi-variance; otherwise the variogram is erratic;
- If a bin is too wide, we do not represent the variation of the (theoretical) variogram with the distance
- The largest separation should not exceed half the longest separation in the dataset;
- Local spatial dependence which is the most interesting, not long distance values
- All computer programs that compute variograms use some defaults for the largest separation and number of bins; `variog` uses the longest separation, and divides this into 13 equal-width bins.

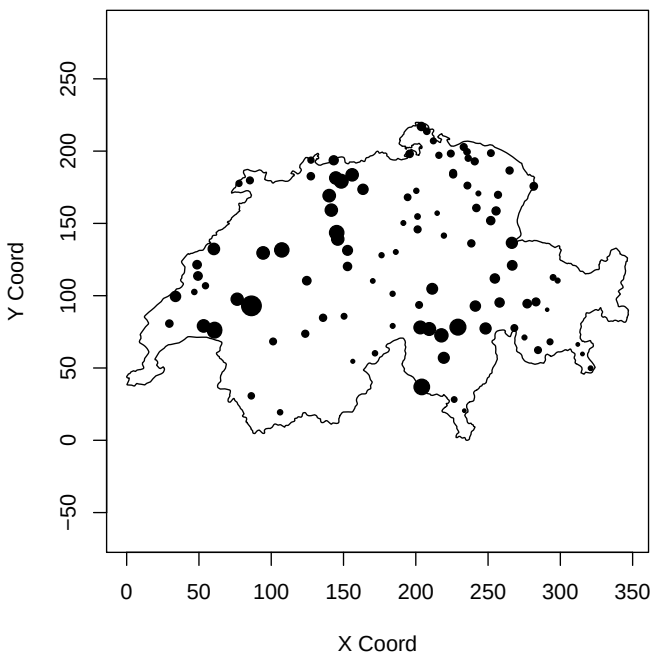
Defining the bins

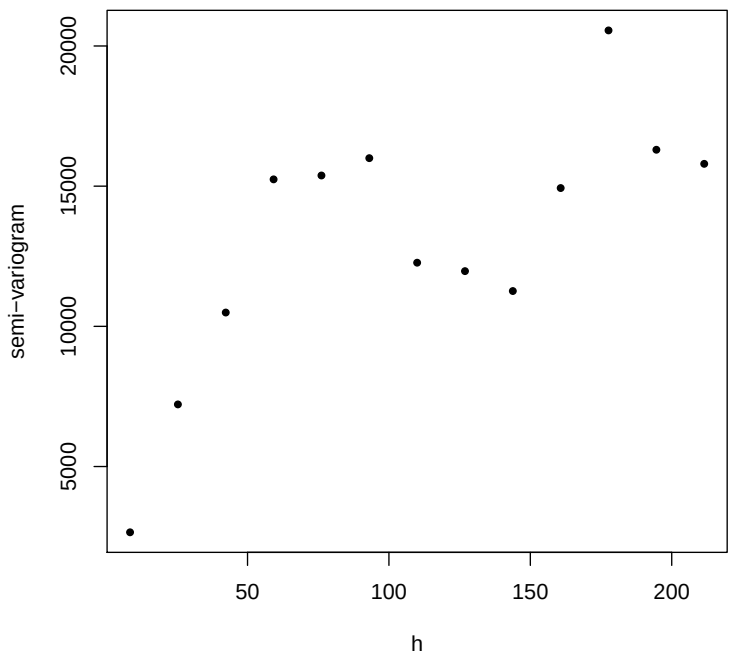
```
> wolf.bin<-variog(wolfcamp,option="bin",bin.cloud=TRUE)
> plot(wolf.bin,bin.cloud=TRUE)
```



Precipitation in Switzerland

- We consider one daily cumulative rainfall data from the Swiss meteorological service measured on May 8, 1986
- First precipitation event after Chernobyl's radioactive cloud traveled across Europe (dataset `sic.100` in the `geoR` package).
- As daily rainfall is a good indicator of the effect of radioactive fallout, these data allowed contamination risk to be evaluated after the Chernobyl disaster (26th April, 1986)





Precipitation in Switzerland

```
> data(SIC)
> points(sic.100, borders = sic.borders, pch=20, cex.max=3)
> max.dist <- 220
> sic.bin<-variog(sic.100,option="bin",estimator.type="classical",
                  max.dist=max.dist)
> plot(sic.bin$u,sic.bin$v,pch=20,col=1,ylab="semi-variogram",xlab='h')
```

Some properties of the empirical variogram

$$\hat{\gamma}(d_k) = \frac{1}{2n_k} \sum_{i,j;d_{ij} \simeq d_k} (Z(s_i) - Z(s_j))^2$$

- Positive
- $\hat{\gamma}(0) = 0$, but $\hat{\gamma}(\epsilon) > 0$ when $\epsilon > 0$
- In general, increasing function

- The empirical variogram is the most popular tool for characterizing the spatial variability
- Need theoretical models for doing optimal spatial interpolation, the kriging

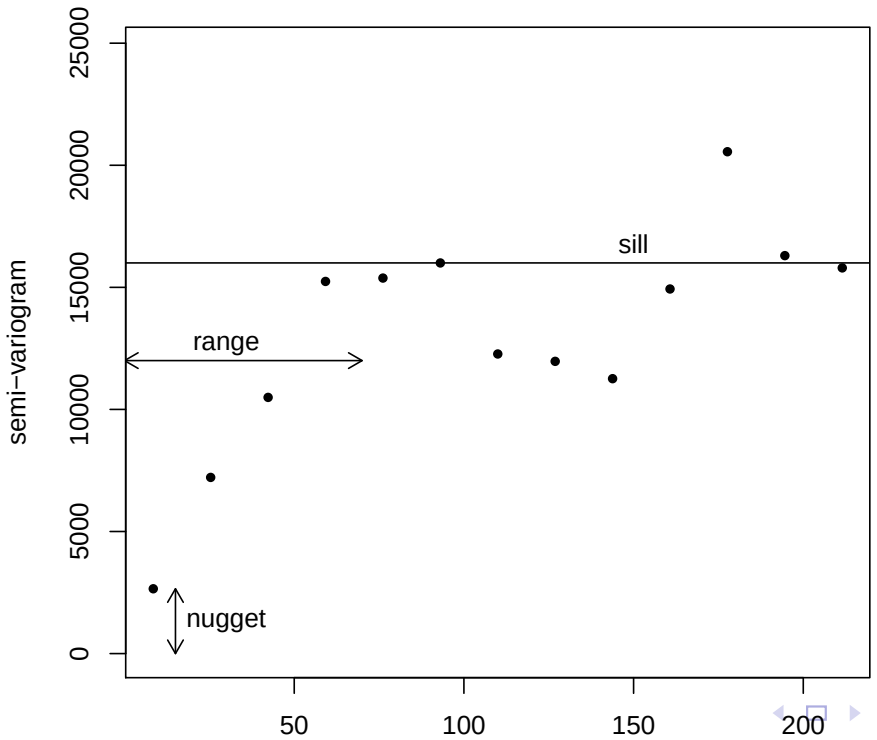
Features of the empirical variogram

Main features characterizing the spatial dependence (only qualitatively at this stage)

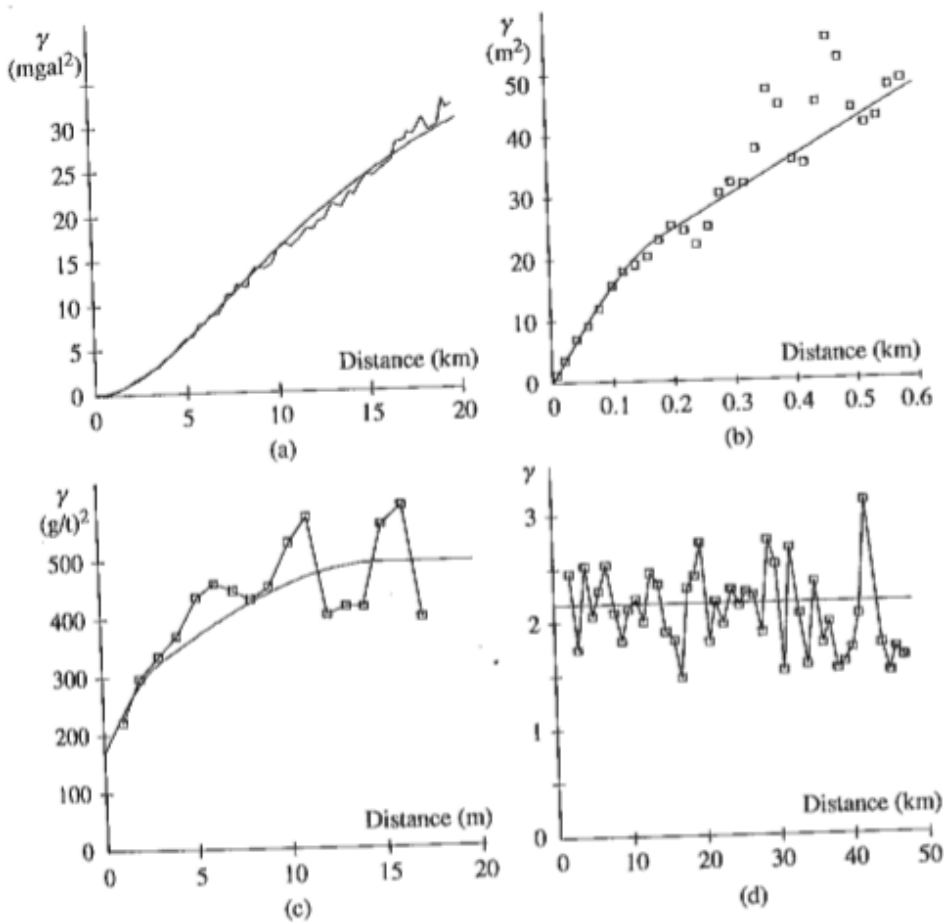
Sill: maximum semi-variance represents variability in the absence of spatial dependence

Range: separation between point-pairs at which the sill is reached distance at which there is no evidence of spatial dependence

Nugget: semi-variance as the separation approaches zero represents variability at a point that can't be explained by spatial structure



Empirical variograms and regualrity



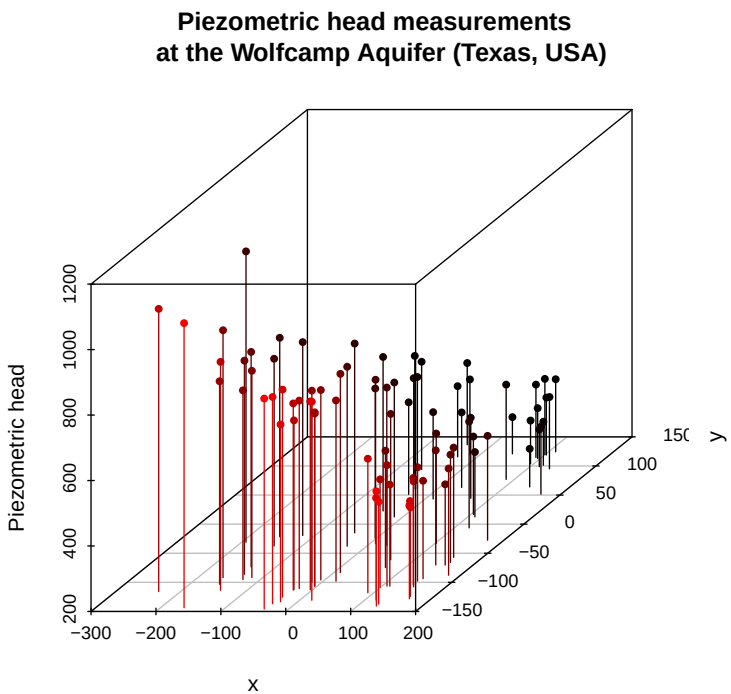
From top to bottom, and from left to right: micrgravity; depth of a geological layer (Paris basin); gold grade in a gold mine (Salsigne, France); log-permeanility (Paris basin); From Chilès et Delfiner (2012).

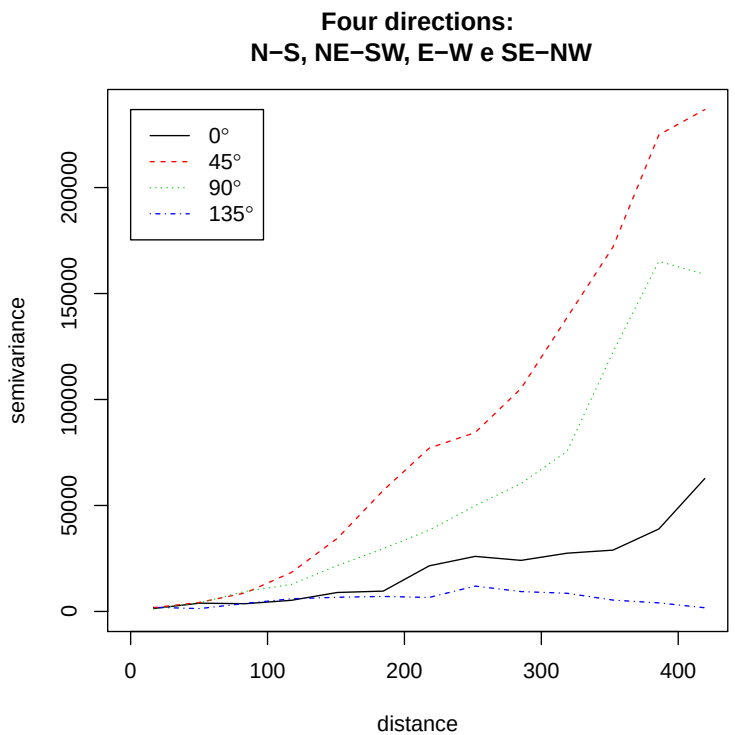
Anisotropy

- We have been considering spatial dependence as if it is the same in all directions from a point (isotropic or omnidirectional).
- Not always true! Variation may depend on direction, not just distance. Examples: prevailing winds; pollution in a valley; horizontal vs. vertical, ...
- We will now refer to the separation vector; up till now this has just meant distance, but now it includes direction

Anisotropy

- To detect anisotropy, one computes variograms along different distance
- We see if they are different
- No formal statistics tests!



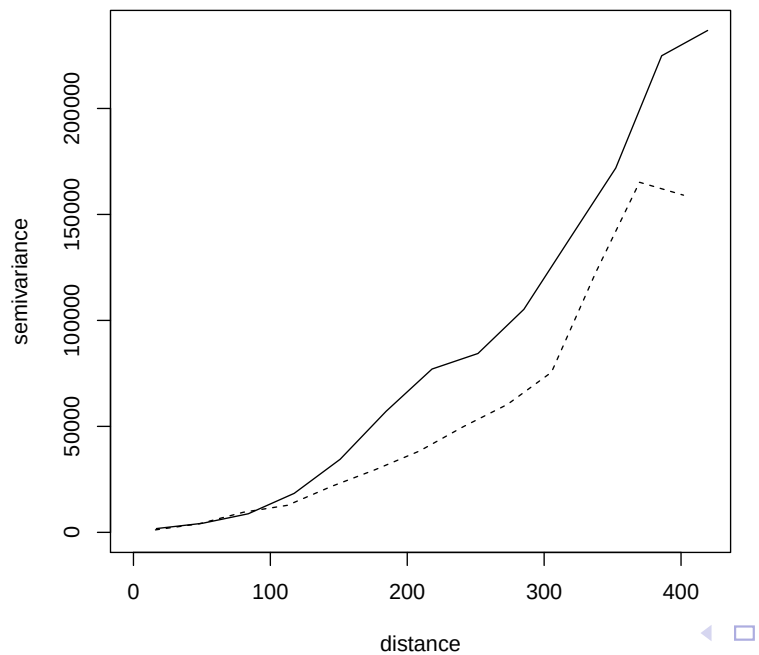


Anisotropy

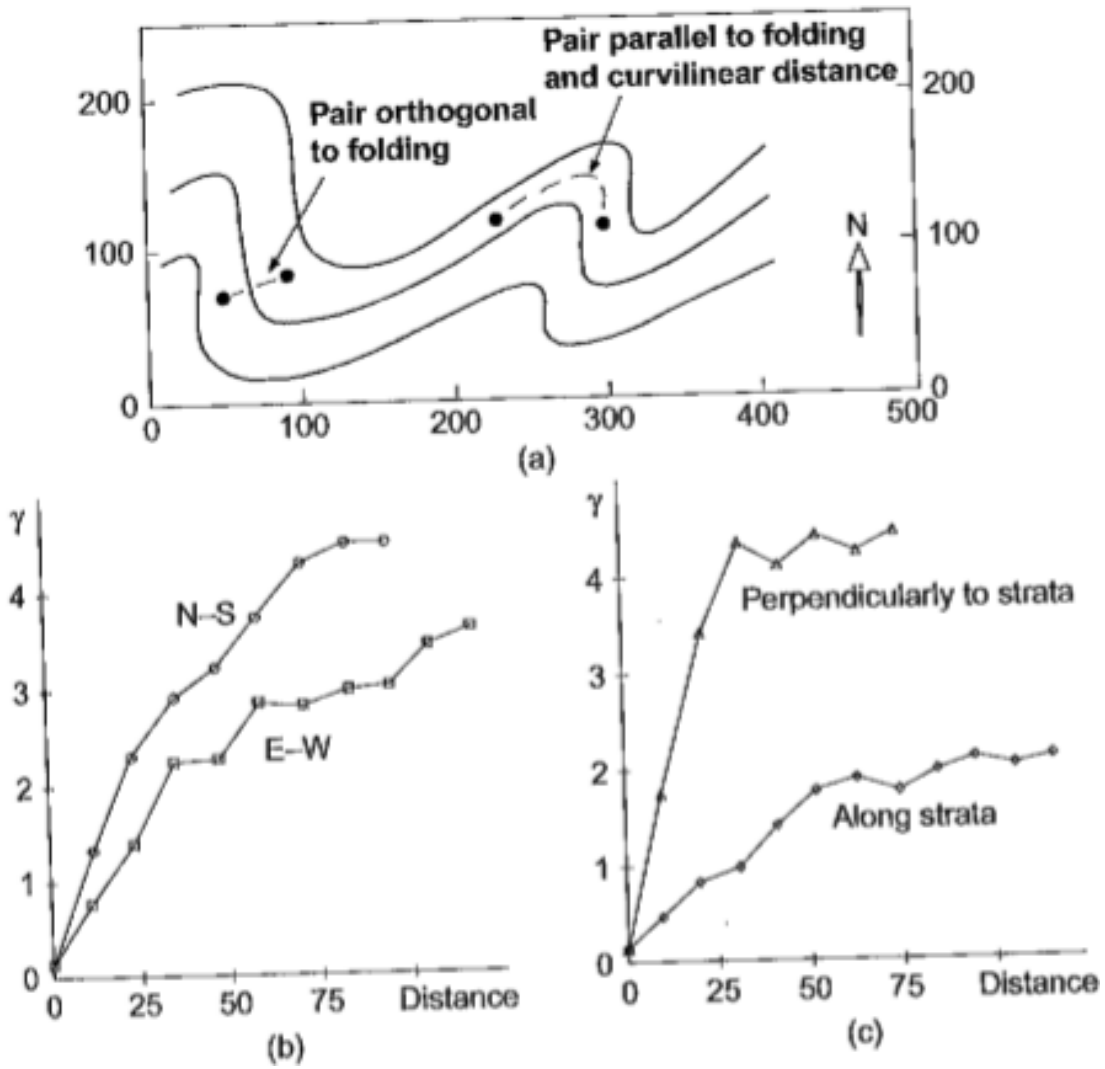
```
> wolf.bin4<-variog4(wolfcamp)
> plot(wolf.bin4)
> title(main='Four directions:\n N-S, NE-SW, E-W e SE-NW',cex.main=2)
```

Plot of specific directions: NE-SW, (45^o degrees, solid line); E-W (90^o degrees, dashed line)

```
> wolf.bin4<-variog(wolfcamp,option="bin", direction=pi/4)
> wolf.bin2<-variog(wolfcamp,option="bin", direction=pi/2)
> plot(wolf.bin4,type='l')
> lines(wolf.bin2$u,wolf.bin2$v,lty=2)
```

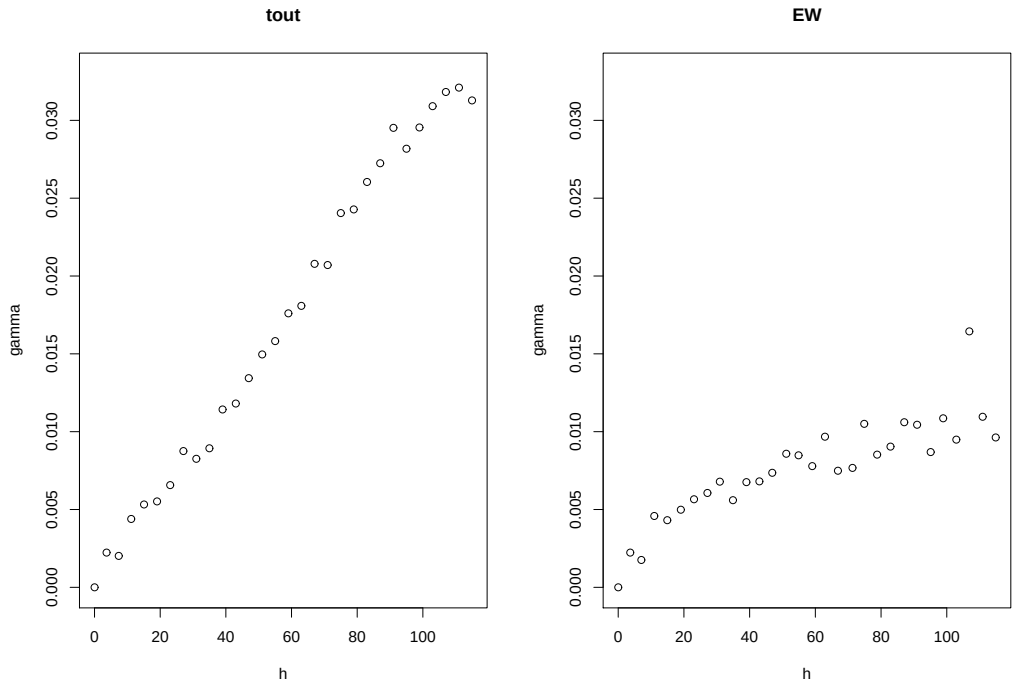


Empirical variogram and Curvilinear coordinates



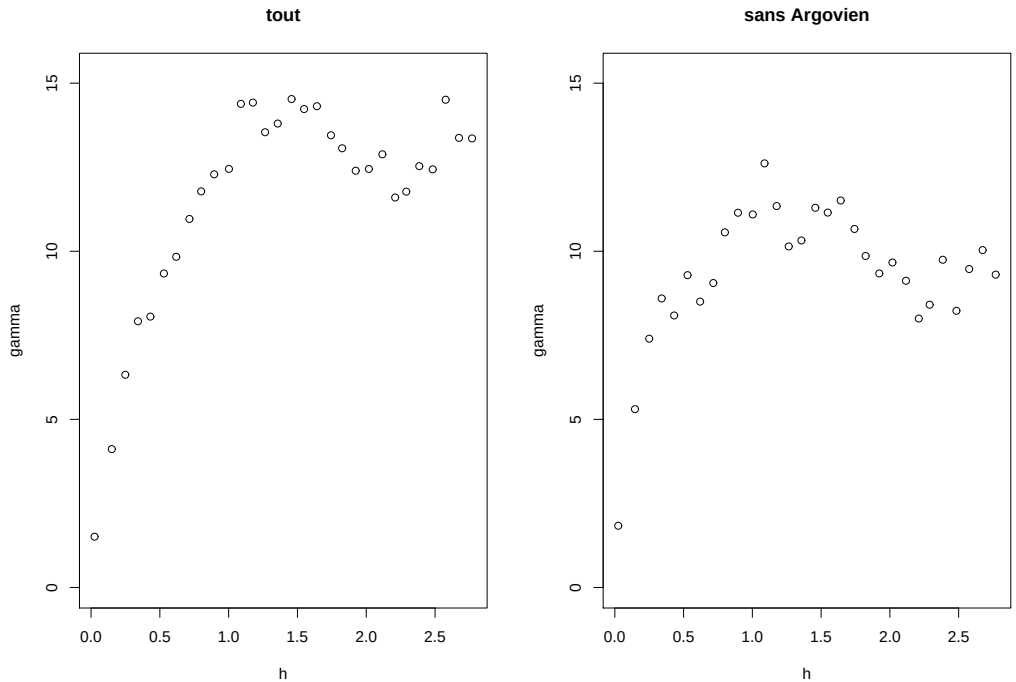
From Chilès et Delfiner (2012).

Empirical variogram and gradient



Chlorophyll content in the Mediterranean See: N-S gradient

Empirical variogram and factors



Co in Swiss Jura: ”Rock Type effect”.

Modelling spatial variation as a random field I

- Idea: The observed values are only one of many possible realizations of a random field (also called a "stochastic" process)
- There is only one reality (which is sampled), but it is one realization of a process that could have produced many realities
- This random process is spatially auto-correlated, so that values are somewhat dependent.
- The (non independent) random values $\{Z(s),\ s \in \mathcal{D}\}$ define a random field.
- A probability model governs the random field; this is where we can model spatial dependence.
- The values $\{z(s),\ s \in \mathcal{D}\}$ are one realization of the random field $Z(\cdot)$

Stationarity

Stationarity

"Probabilities are translation invariant, i.e. they are identical for all $s \in D$."
In most cases, it is sufficient to assume 2nd order stationarity

Second order Stationarity

"Means and (co-)variances are identical for all $s \in D$."

$$\begin{aligned} \mathrm{E}[Z(s+h)] = \mathrm{E}[Z(s)] &= \mu \\ \mathrm{cov}(Z(s), Z(s+h)) &= C(h) \end{aligned}$$

for all s and h in D.

The autocovariance depends on the separation vector h only

Covariance function

Covariance function

$$C(h) = \text{cov}(Z(s), Z(s+h))$$

We drop the prefix auto-

- $C(0) = \sigma^2$
- $C(h) = C(-h)$
- $C(h)$ can be negative, but $|C(h)| \leq C(0)$
- $C(h)$ must be a special function, called positive definite function. Thus, one always has

$$\text{Var}\left(\sum_{i=1}^{n} \lambda_i Z(s_i)\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j C(s_i - s_j) \geq 0$$

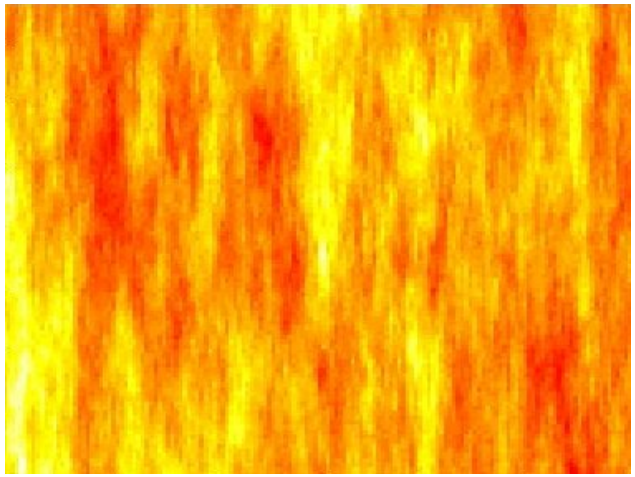
for all n, all $s_1, \ldots, s_n$ and all $\lambda_1, \ldots, \lambda_n$.

Some valid covariance functions

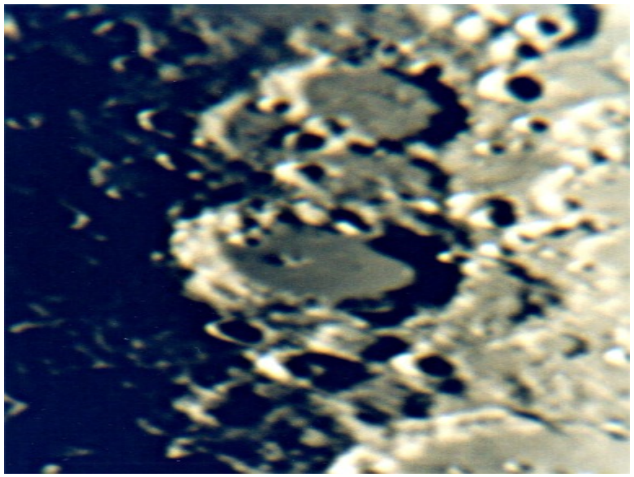
- Spherical : $C(h) = \begin{cases} \sigma^2(1 - \frac{3}{2}\frac{|h|}{a} + \frac{1}{2}\frac{|h|^3}{a^3}), & \text{if} \quad 0 \leq |h| \leq a \\ 0, & \text{if} \quad |h| \geq a \end{cases}$
- Exponential : $C(h) = \sigma^2 \exp(-|h|/a), \quad a > 0$
- Matérn :$C(h) = \sigma^2(\alpha|h|)^\nu \mathcal{K}_\nu(\alpha|h|), \quad \nu, \alpha > 0$
- Gaussian : $C(h) = \sigma^2 \exp(-|h|^2/a), \quad a > 0$
- ...

Examples

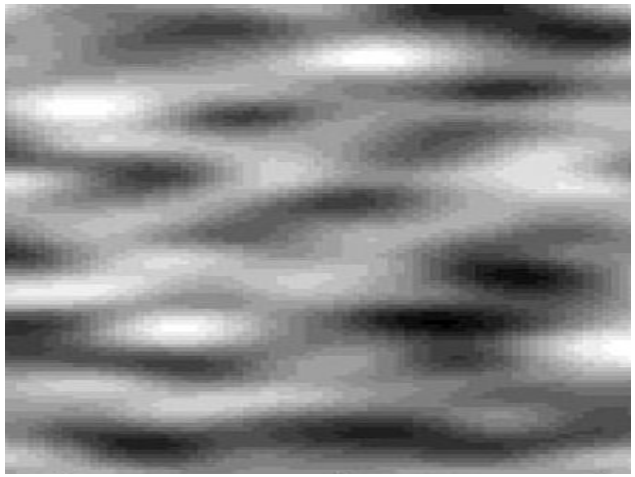
Colza cultivation in Lombardy



Stationary random field

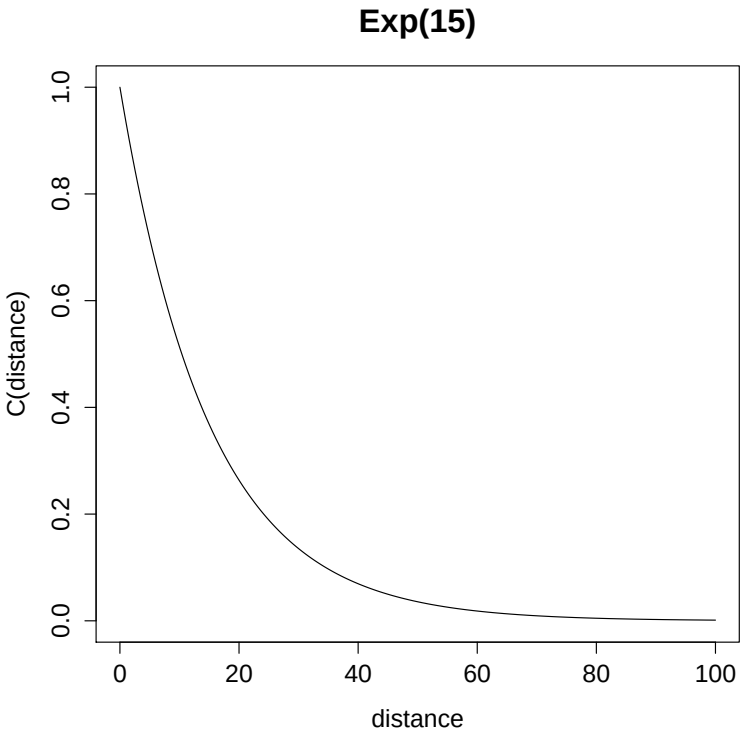


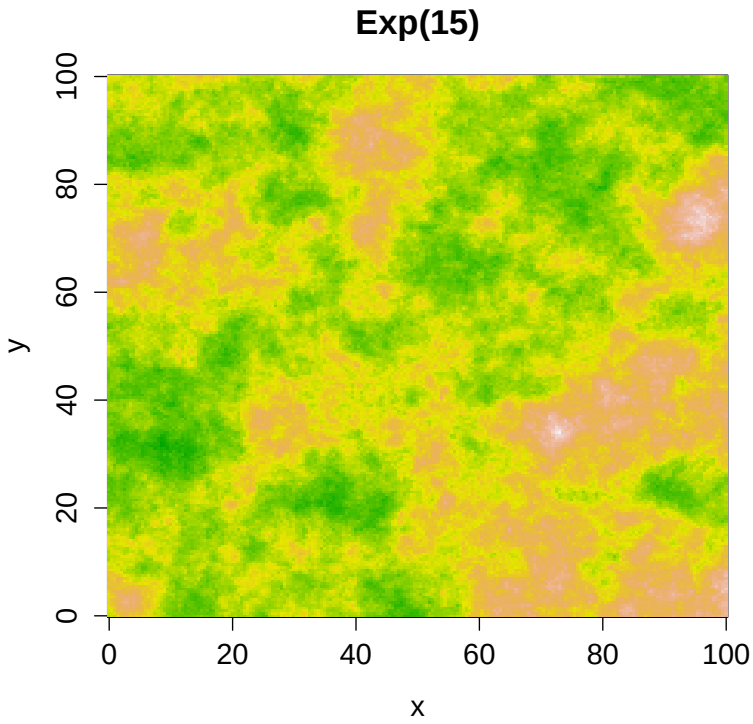
the moon's surface



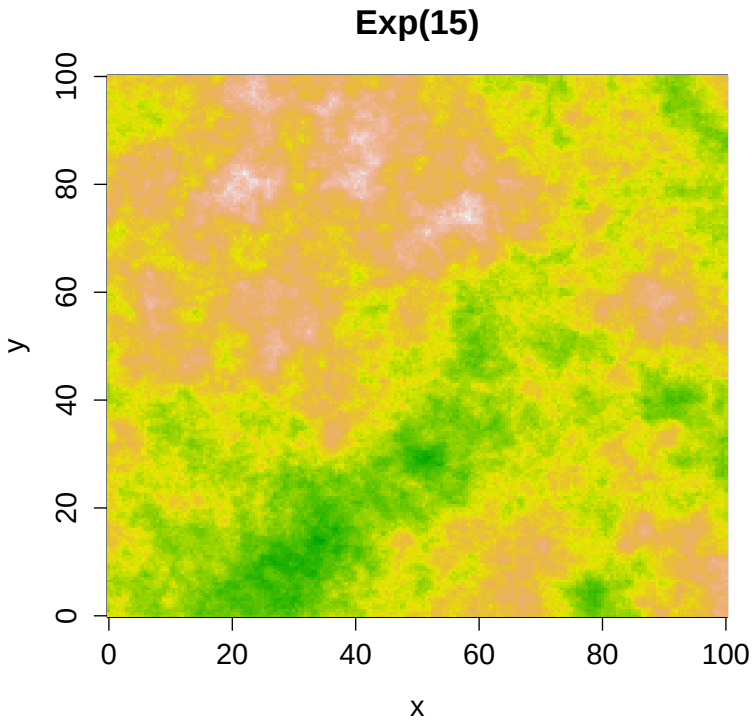
Anisotropic random field

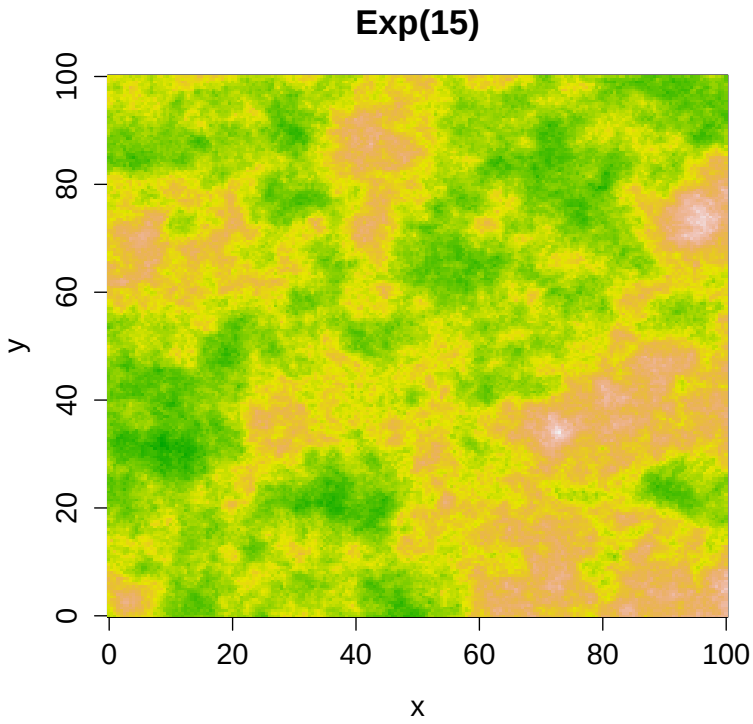
Exponential covariance



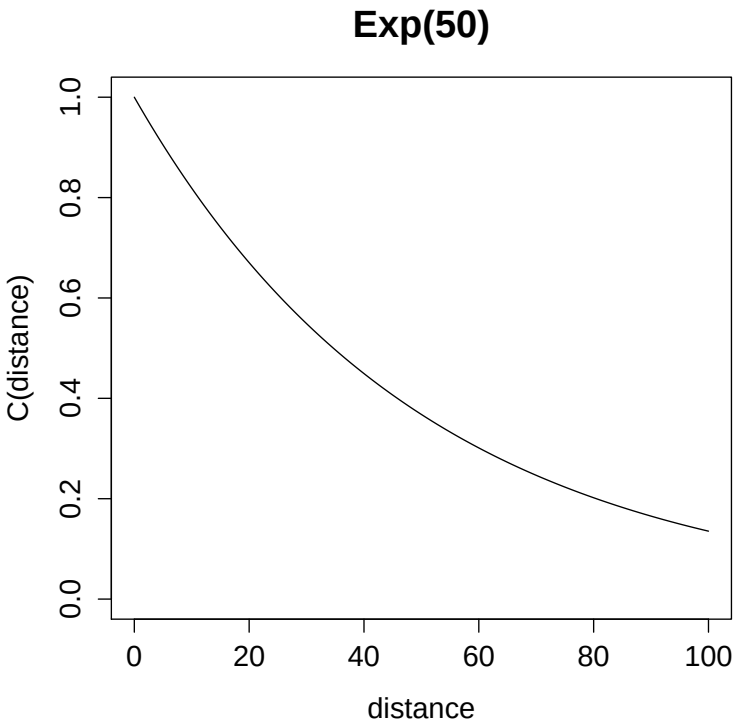


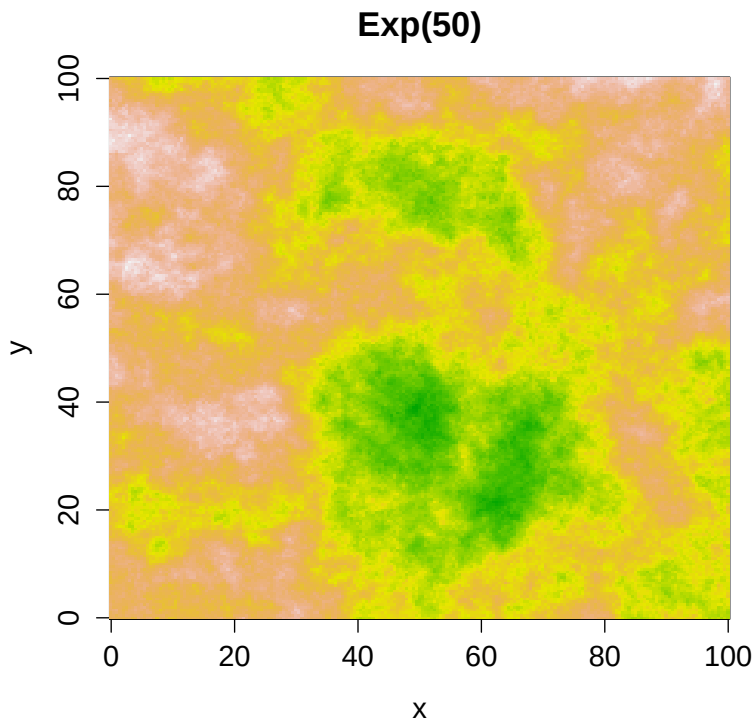
Exponential covariance



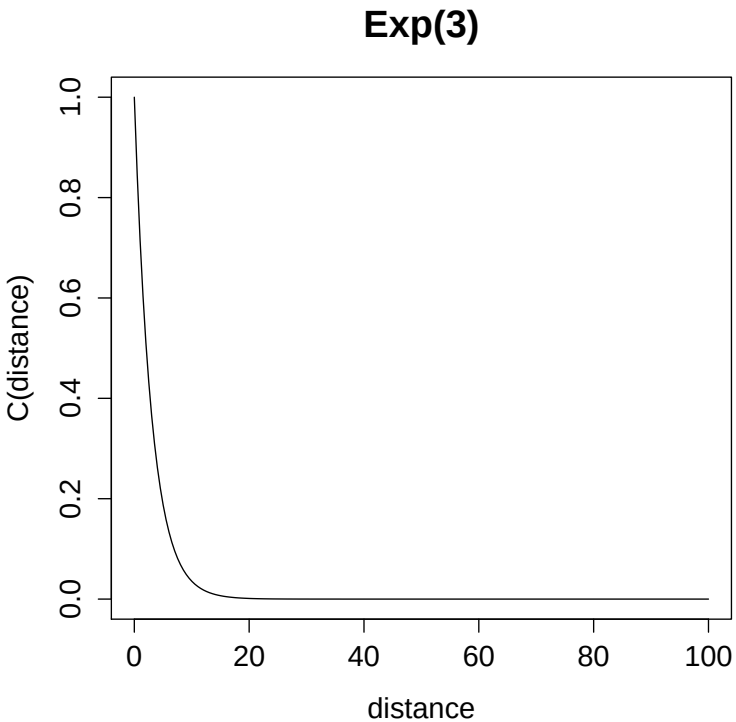


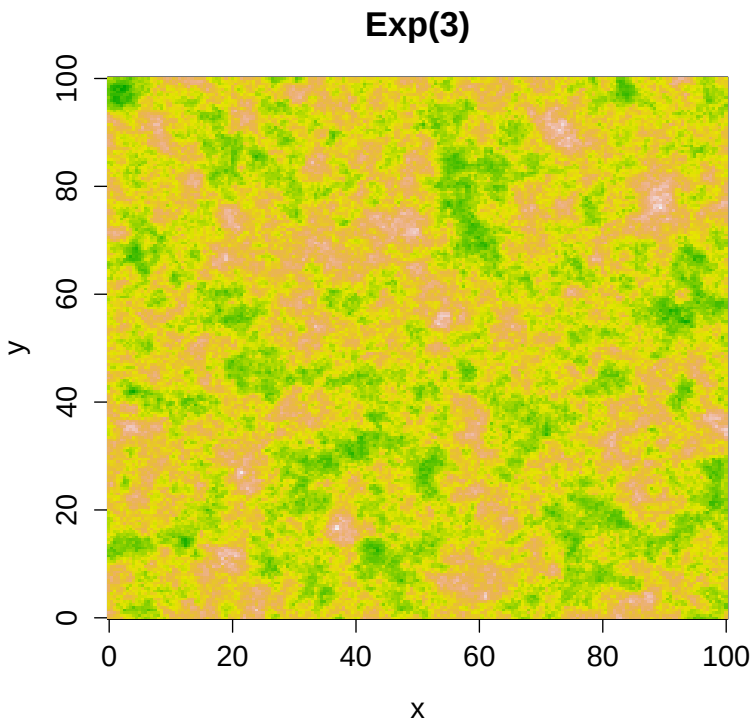
Exponential covariance



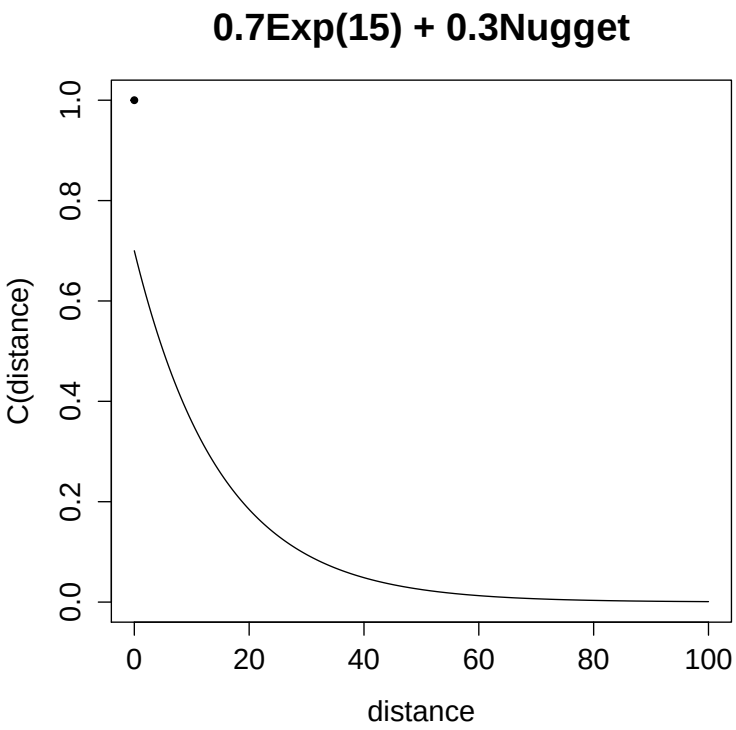


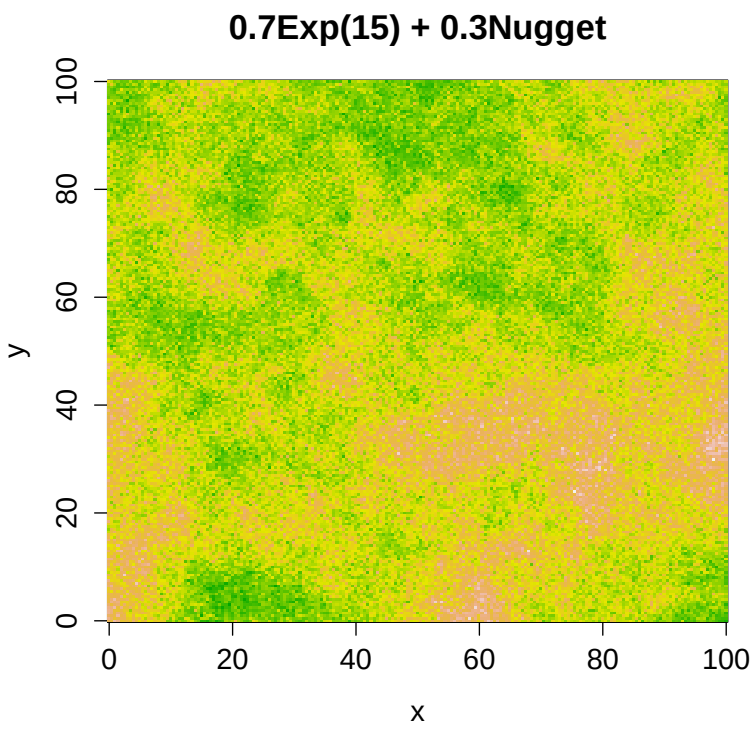
Exponential covariance



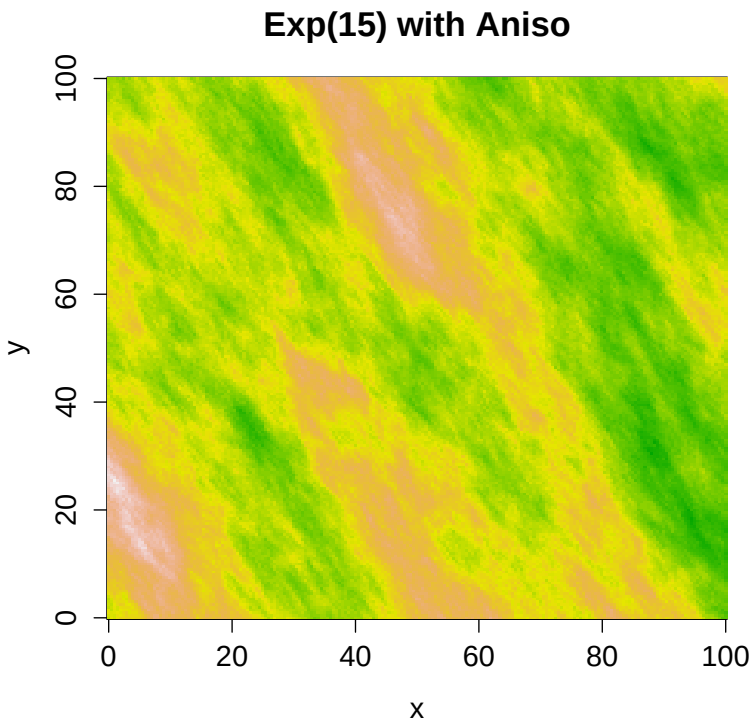


Exponential covariance

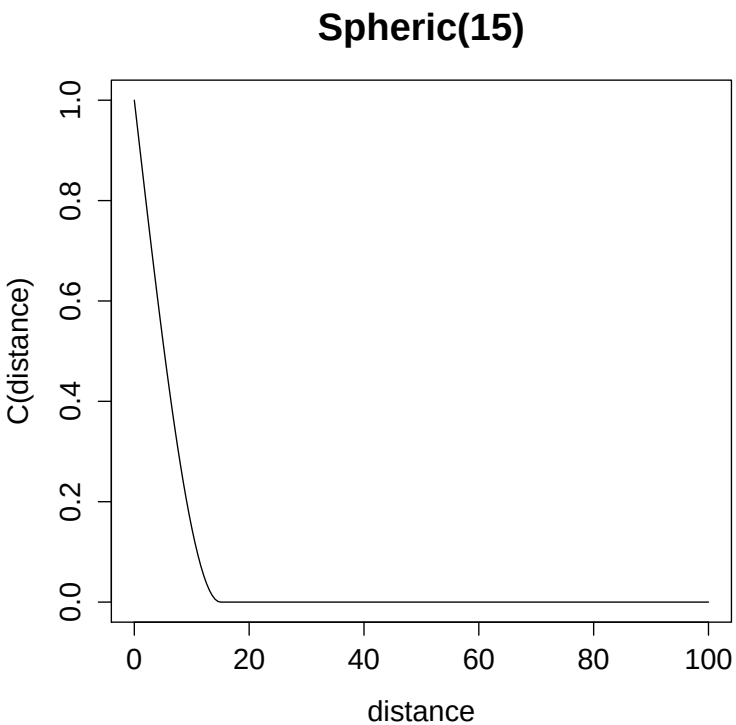


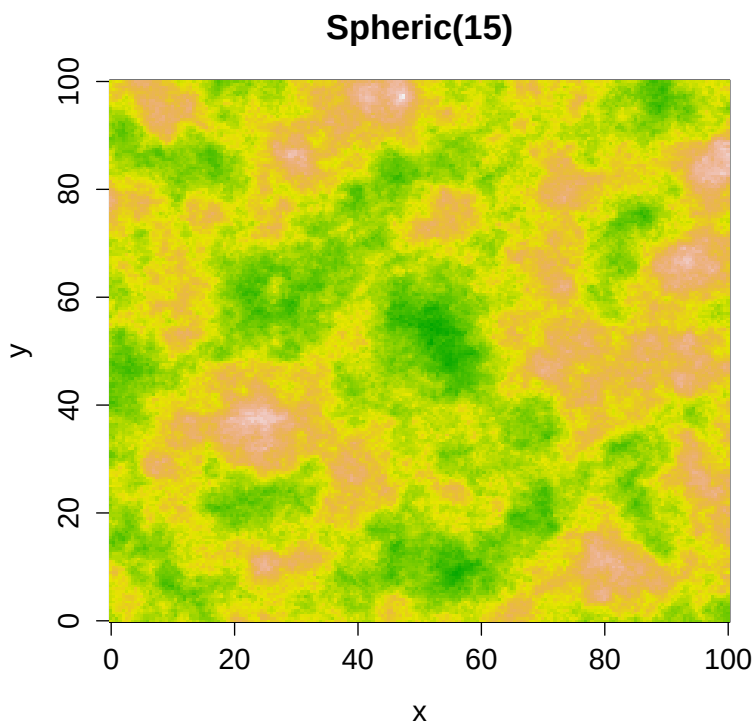


Exponential covariance

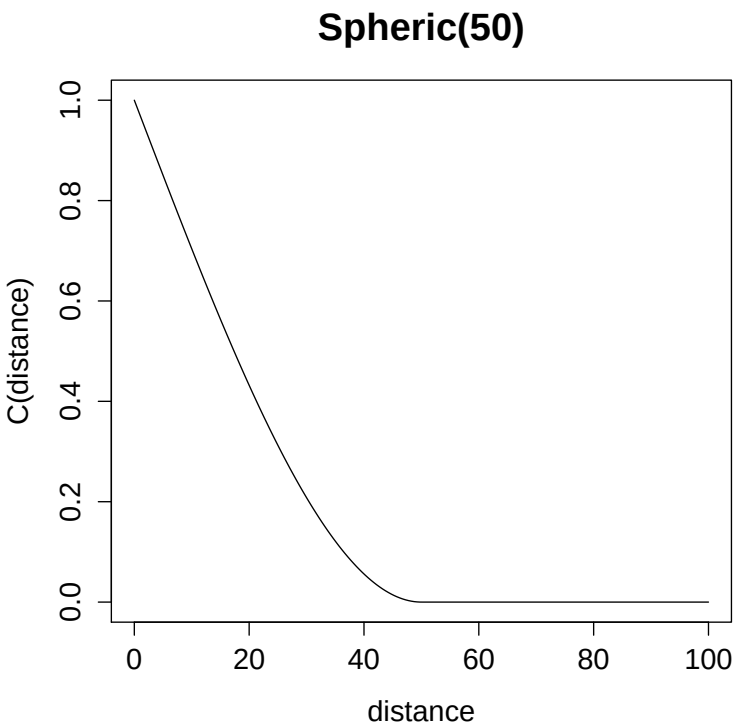


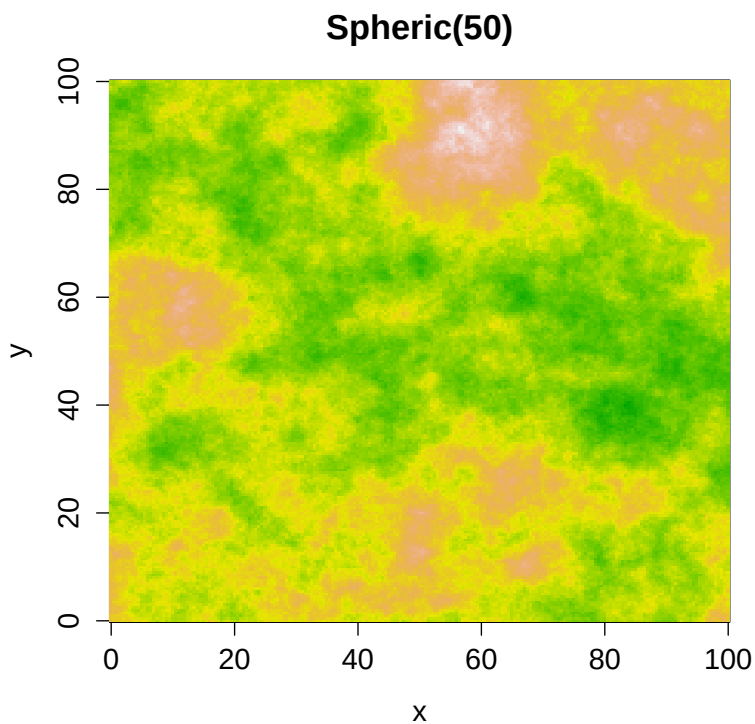
Spherical covariance



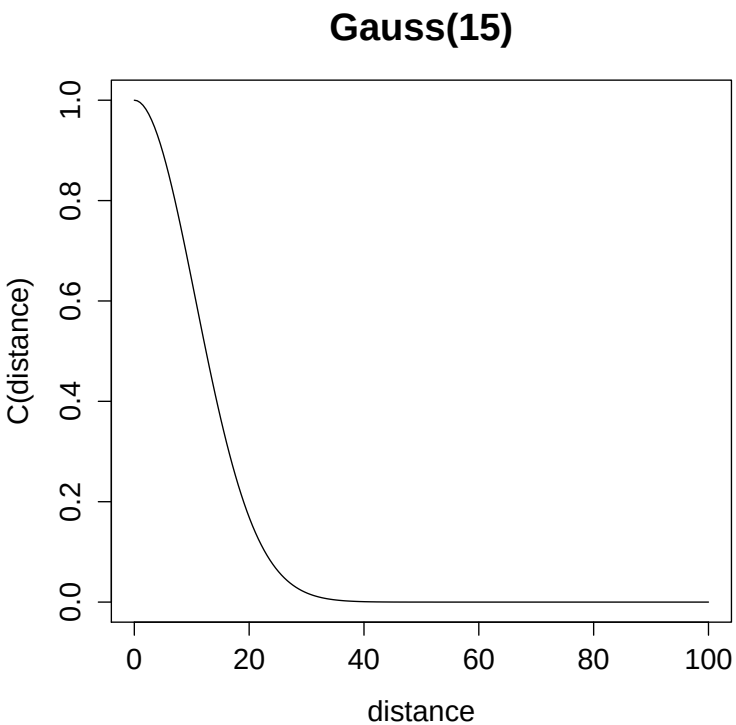


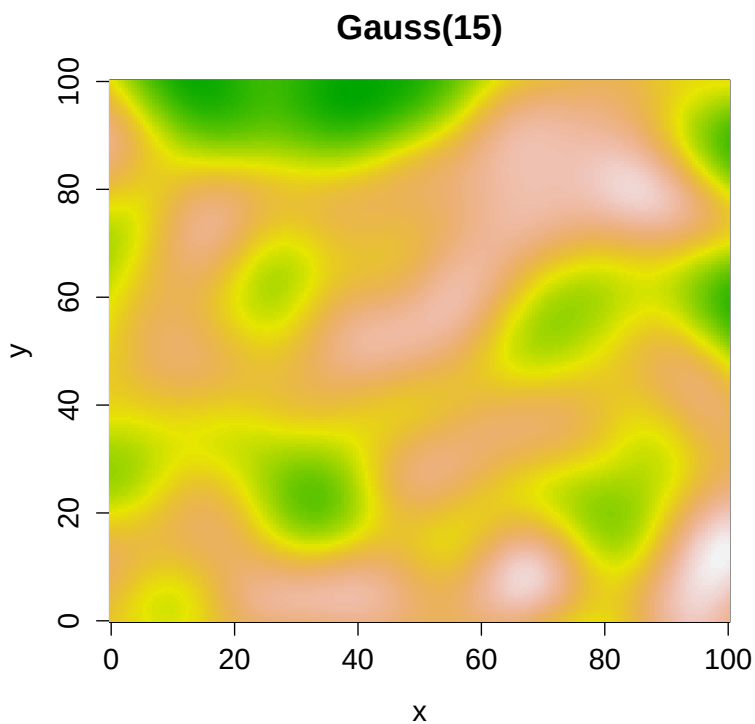
Spherical covariance



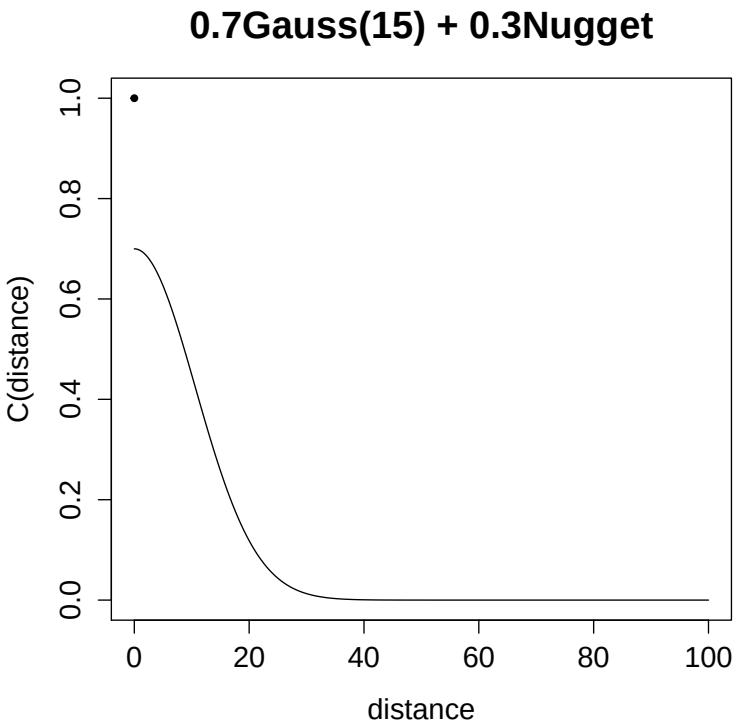


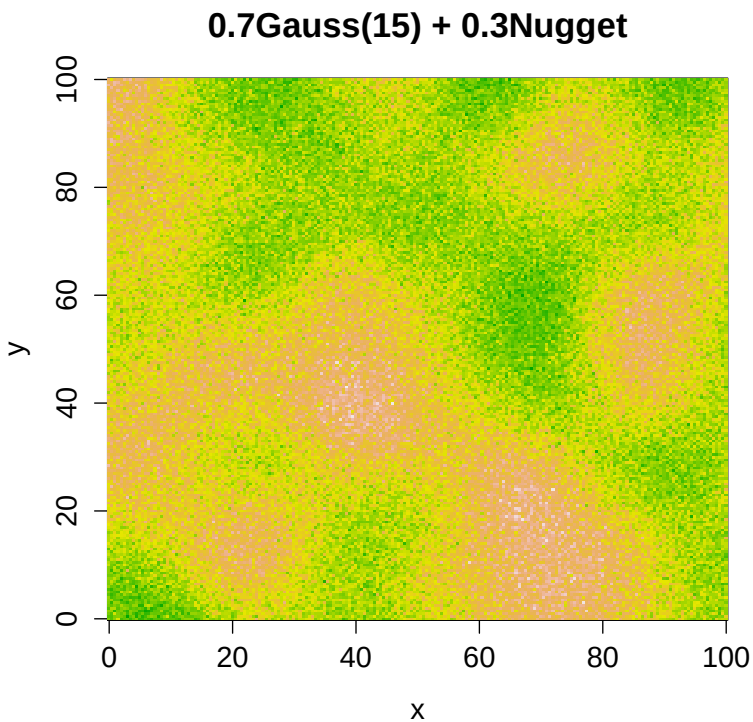
Gaussian covariance



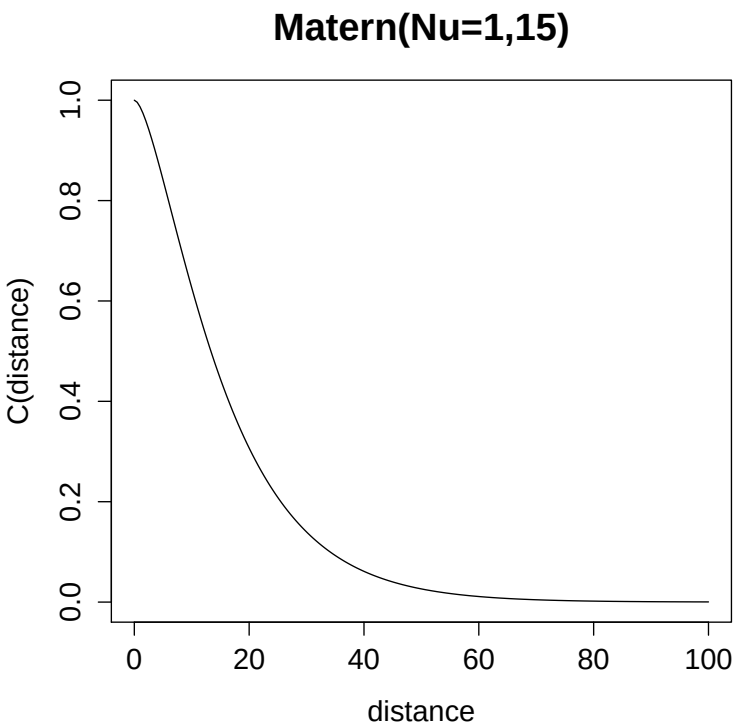


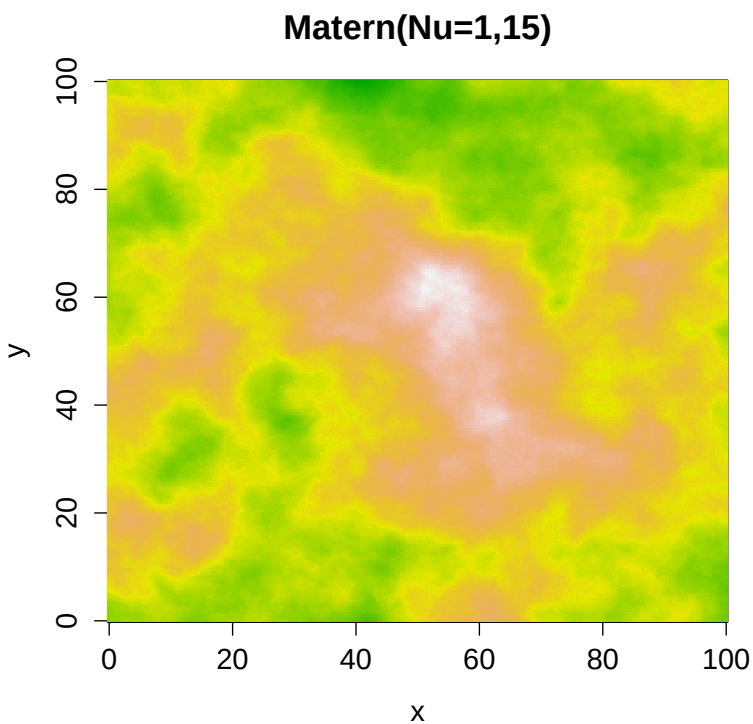
Gaussian covariance



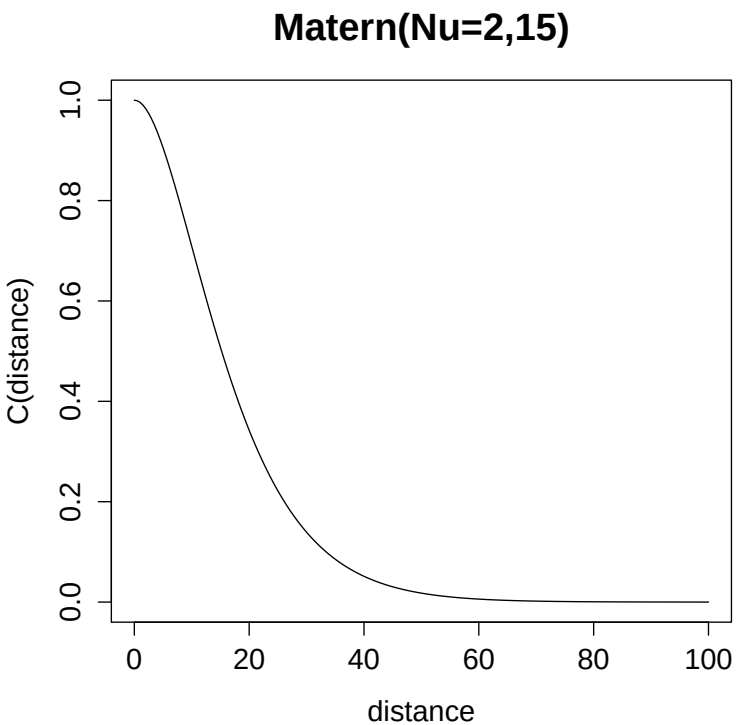


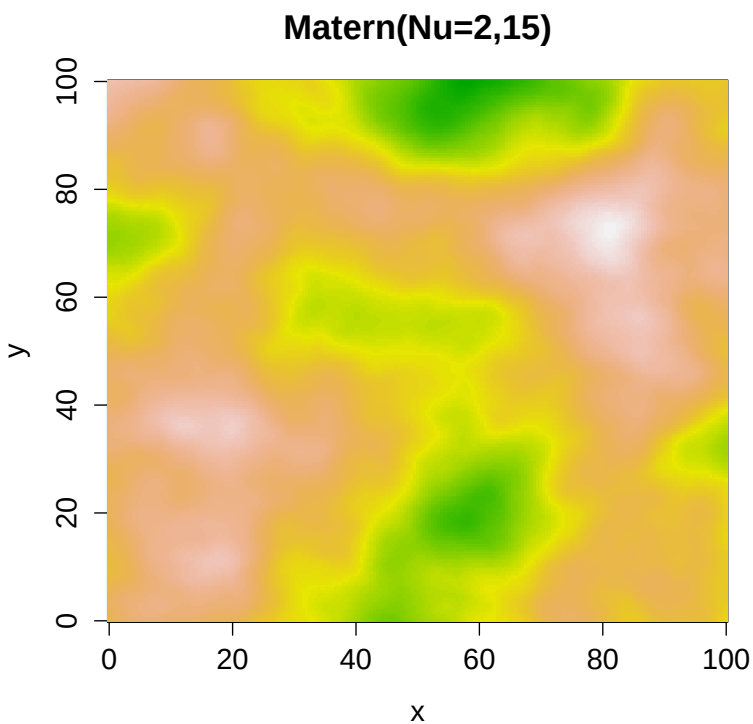
Matérn Covariance



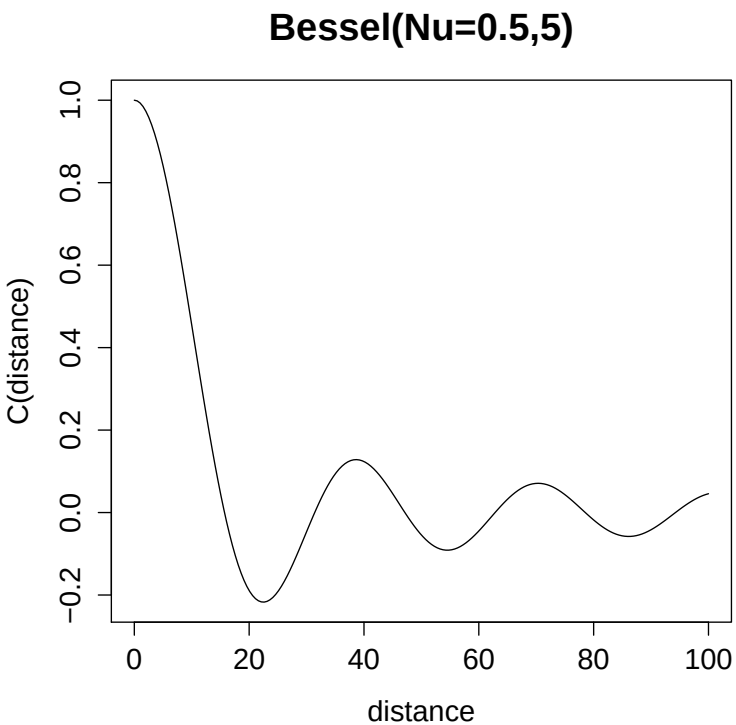


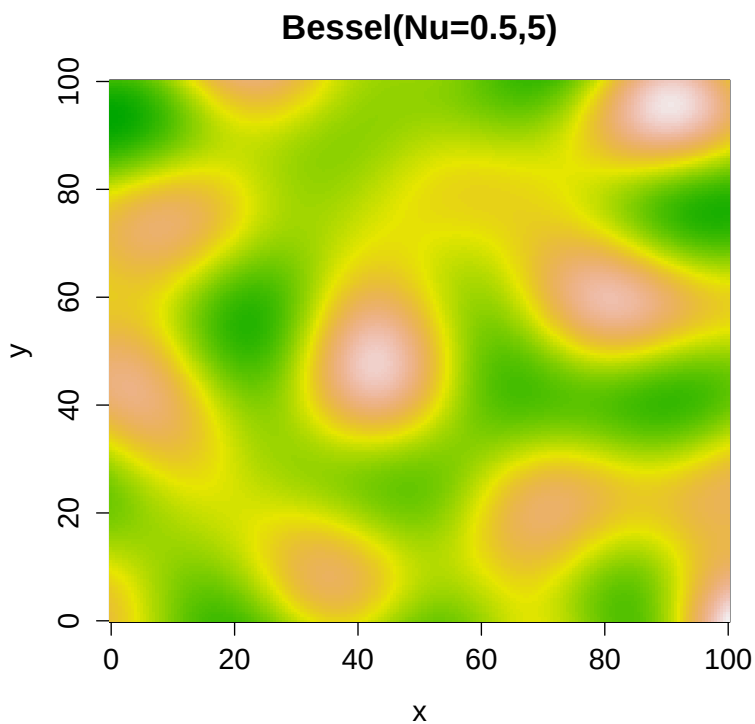
Matérn Covariance



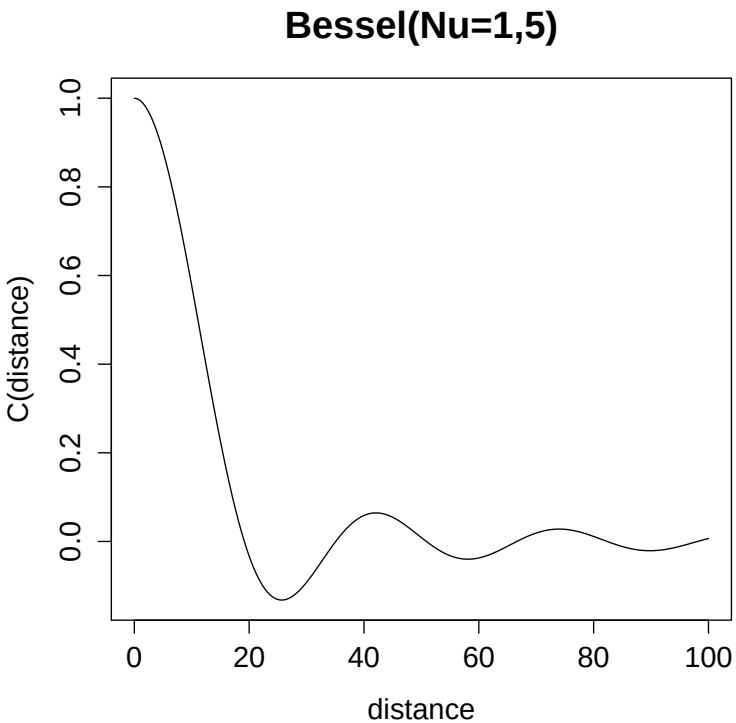


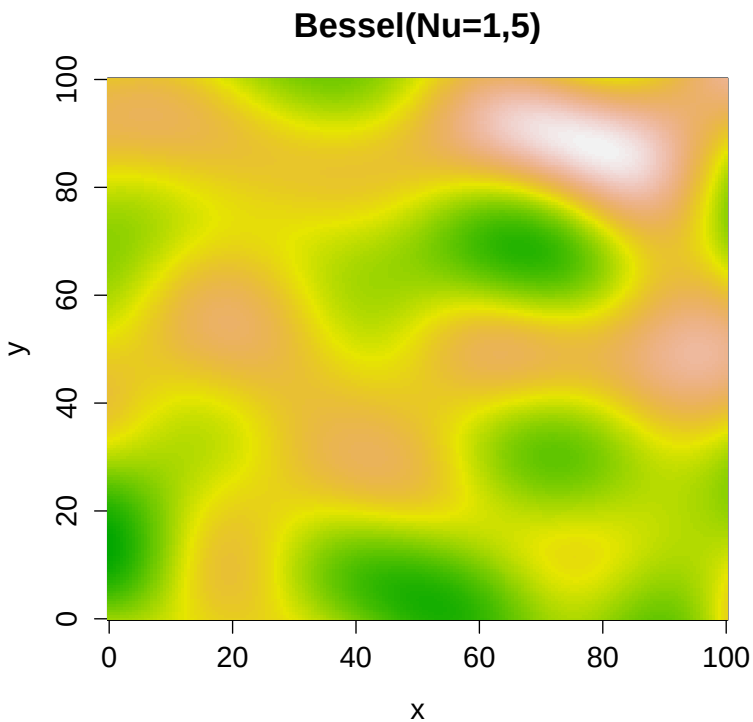
Bessel Covariance





Bessel Covariance





Theoretical variogram

Question

What is the relationship between empirical variogram and covariance functions?

Answer

Theoretical variograms

- Recall the empirical variogram

$$\hat{\gamma}(d_k) = \frac{1}{2n_k} \sum_{i,j;d_{ij} \simeq d_k} (Z(s_i) - Z(s_j))^2$$

- We move from

$$\text{Data} \quad \rightarrow \quad \text{Mathematics}$$

- Hence, we move from

$$\text{"Average"} \quad \rightarrow \quad \text{"Expectation"}$$

Theoretical variogram

- When the semi-variance of variations,

$$\frac{\text{Var}(Z(s) - Z(s'))}{2}$$

depends only on the separation vector $h = s - s'$, we define the (theoretical) semi-variogram

$$\gamma(h) = \frac{\text{Var}(Z(s) - Z(s'))}{2}.$$

- The theoretical variogram cannot be any function
- A function $\gamma(h)$ must be conditionally negative definite to be a valid variogram.
- It can be unbounded: the function

$$\gamma(h) = a.\|h\|^{\alpha}, \qquad 0 < \alpha < 2$$

is a valid variogram.

- When $\gamma(h)$ is bounded, the following relationship between variograms and covariance functions hold:

$$\gamma(h) = C(0) - C(h).$$

Properties of the variogram

- $\gamma(h) \geq 0; \qquad \gamma(0) = 0$
- $\gamma(-h) = \gamma(h)$
- If $\lim_{|h| \to \infty} \gamma(h) = S < +\infty$, then $Z(\cdot)$ is second order stationary and

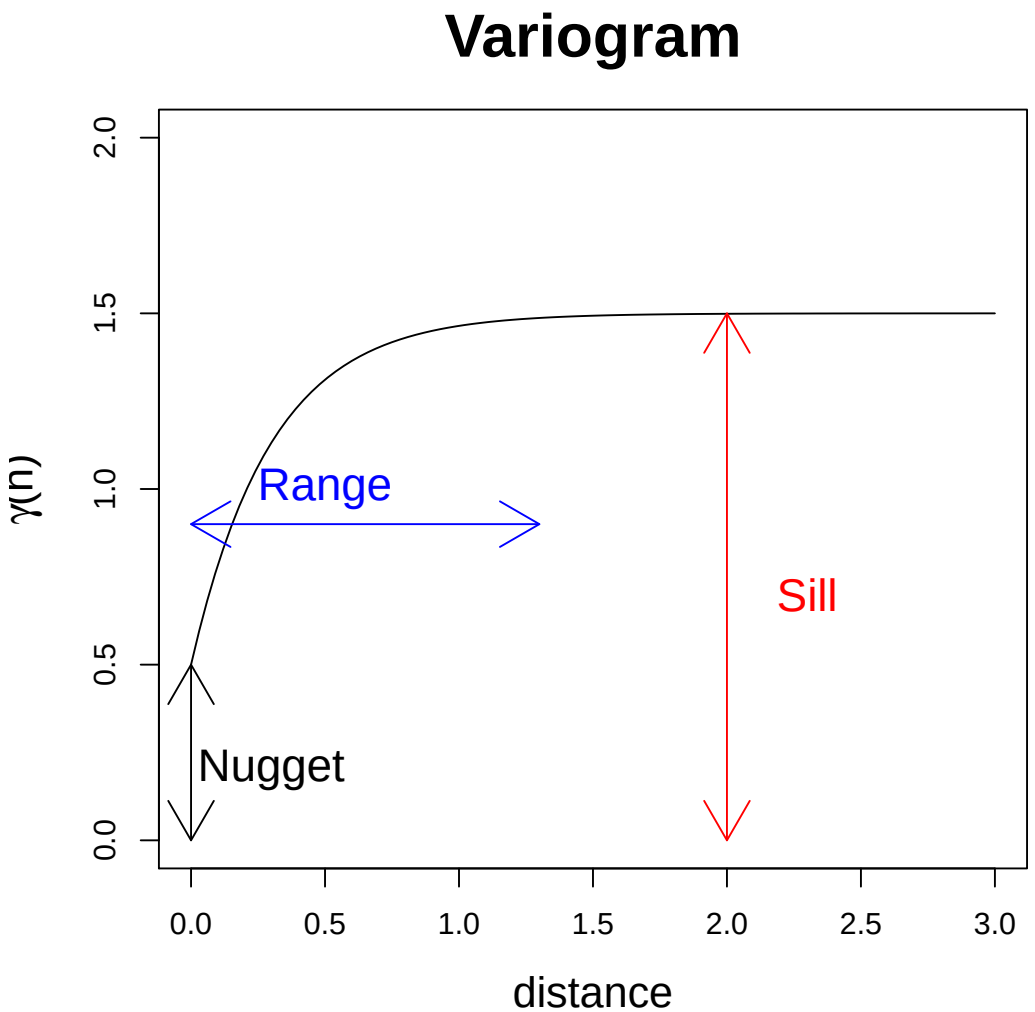
$$\gamma(h) = C(0) - C(h).$$

- It is conditionally negative definite, i.e.

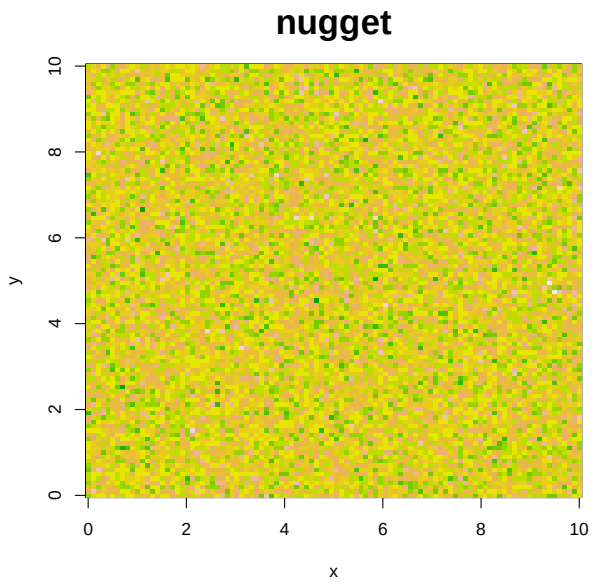
$$\sum_{i=1}^{n} \lambda_i = 0 \quad \sum_{ij} \lambda_i \lambda_j \gamma(s_i - s_j) \leq 0,$$

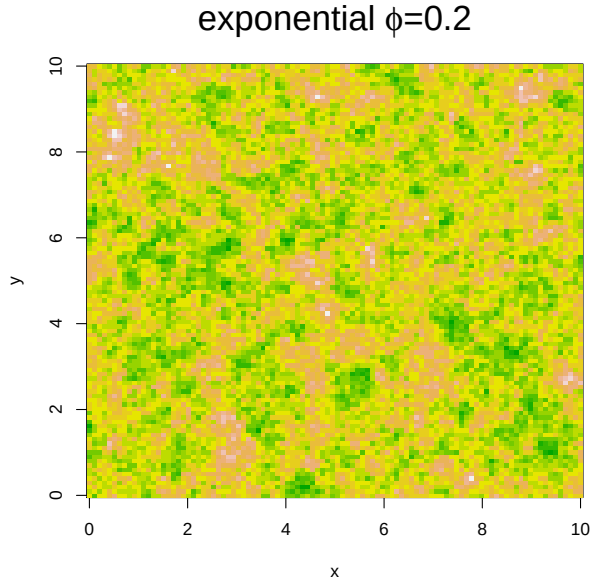
  for all n, all $s_1, \ldots, s_n$, and all $\lambda_1, \ldots, \lambda_n$
- The empirical variogram is an unbiased estimator of the theoretical variogram
- Regularity of the variogram at $h = 0 \iff$ regularity of the random field $Z(\cdot)$
  - Twice differentiable variogram at $h = 0 \iff$ differentiability of $Z(\cdot)$
  - Continuous variogram at $h = 0 \iff$ continuous $Z(\cdot)$
  - Discontinuous variogram at $h = 0$ (i.e. nugget effect) $\iff$ discontinuous $Z(\cdot)$

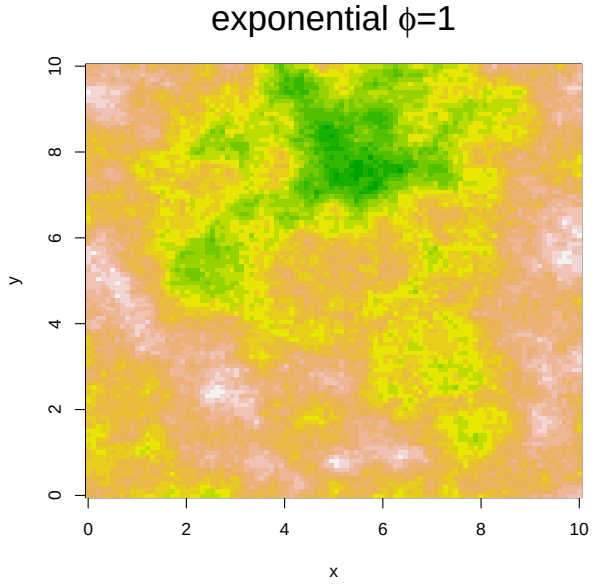
Properties of the variogram

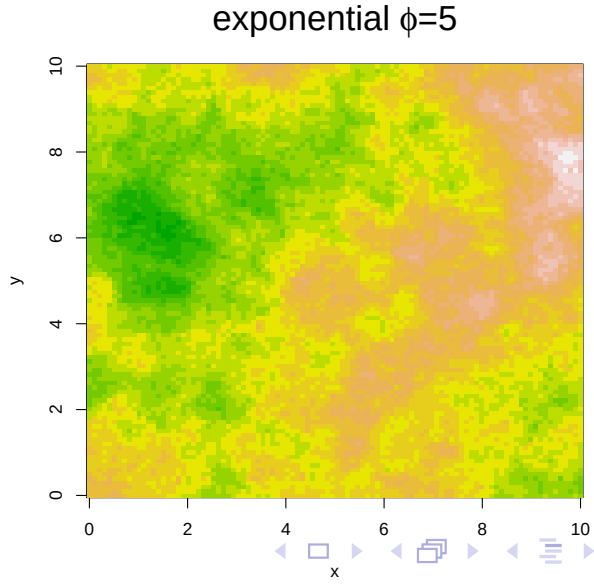


Simulated examples







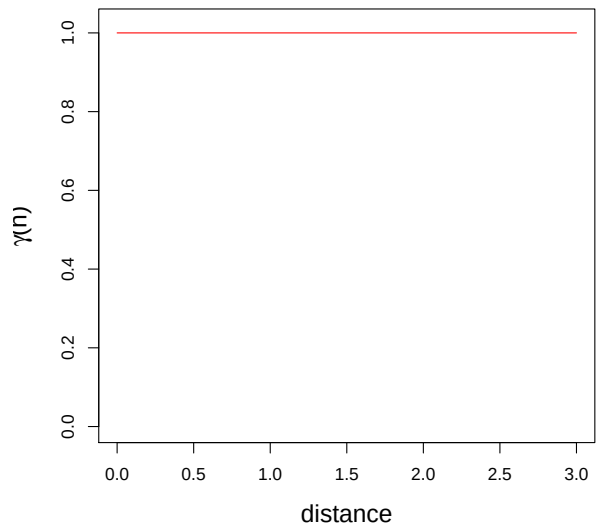


Examples of variograms

Nugget model (aka *white noise*)

$$\gamma(h) = \begin{cases} 0, & h = 0 \\ \sigma^2 & h \neq 0 \end{cases}$$

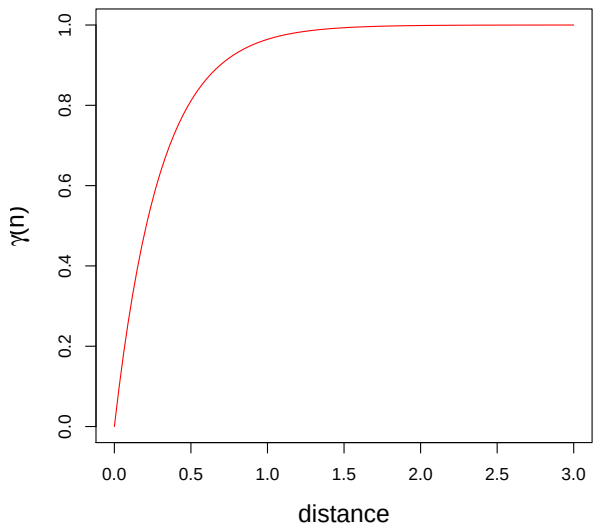
$$\sigma^2 = 1$$



Exponential model

$$\gamma(h) = \sigma^2\{1 - \exp\{-h/\phi\}\}$$

$$\sigma^2 = 1, \phi = 0.3$$

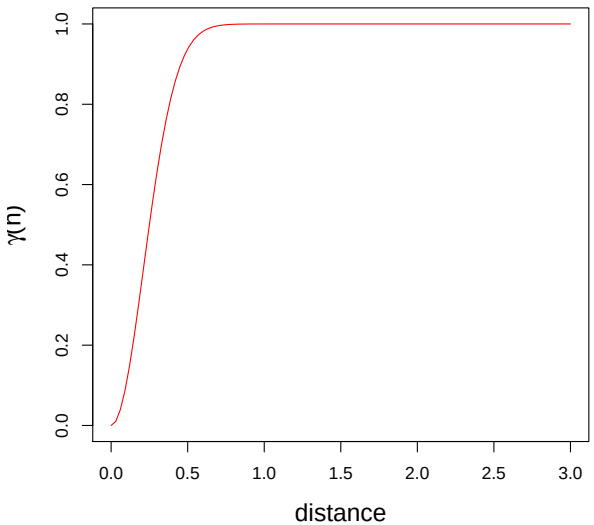


Other examples of variograms

Gaussian model

$$\gamma(h) = \sigma^2\{1 - \exp\{-\frac{h^2}{\phi}\}\}$$

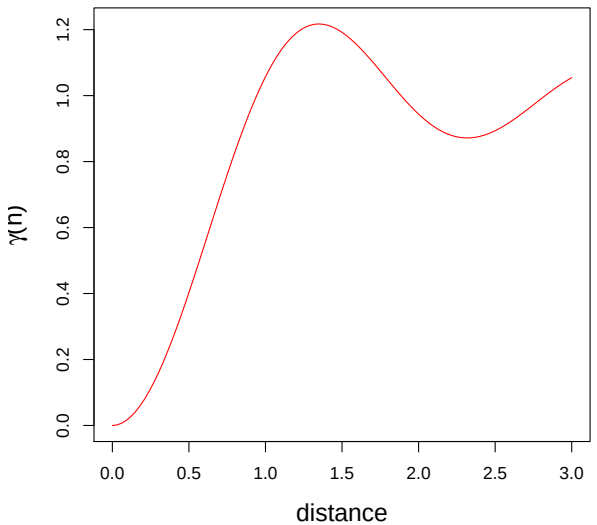
$$\sigma^2 = 1, \phi = 0.3$$



Wave model

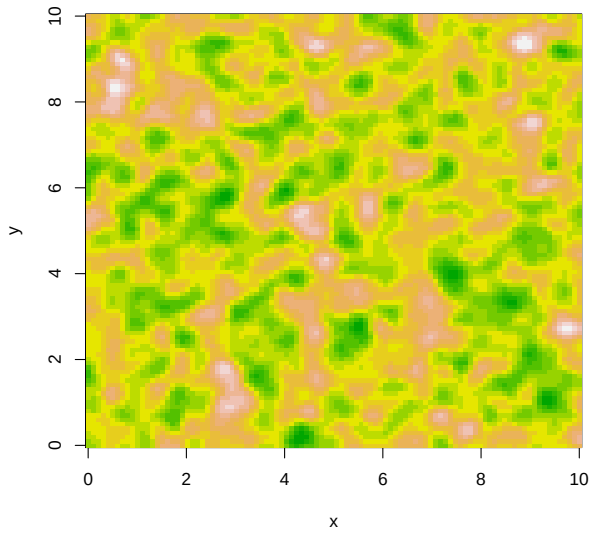
$$\gamma(h) = \sigma^2\{1 - \frac{\phi}{h}\sin(\frac{h}{\phi})\}$$

$$\sigma^2 = 1, \phi = 0.3$$

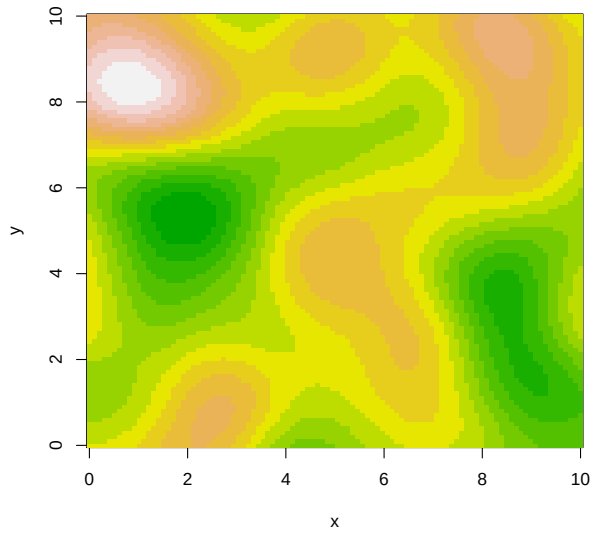


Simulated examples

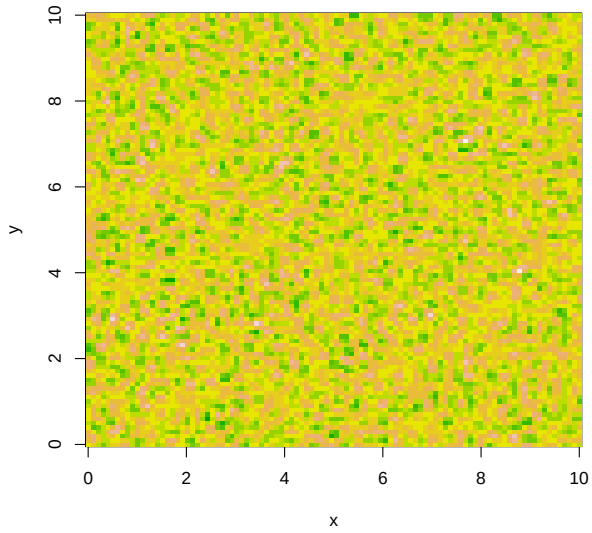
Gaussian, $\phi=0.6/\sqrt{3}$



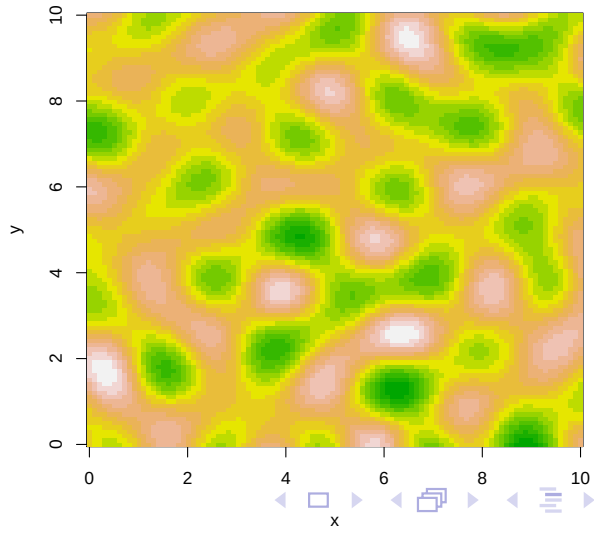
Gaussian, $\phi=2$



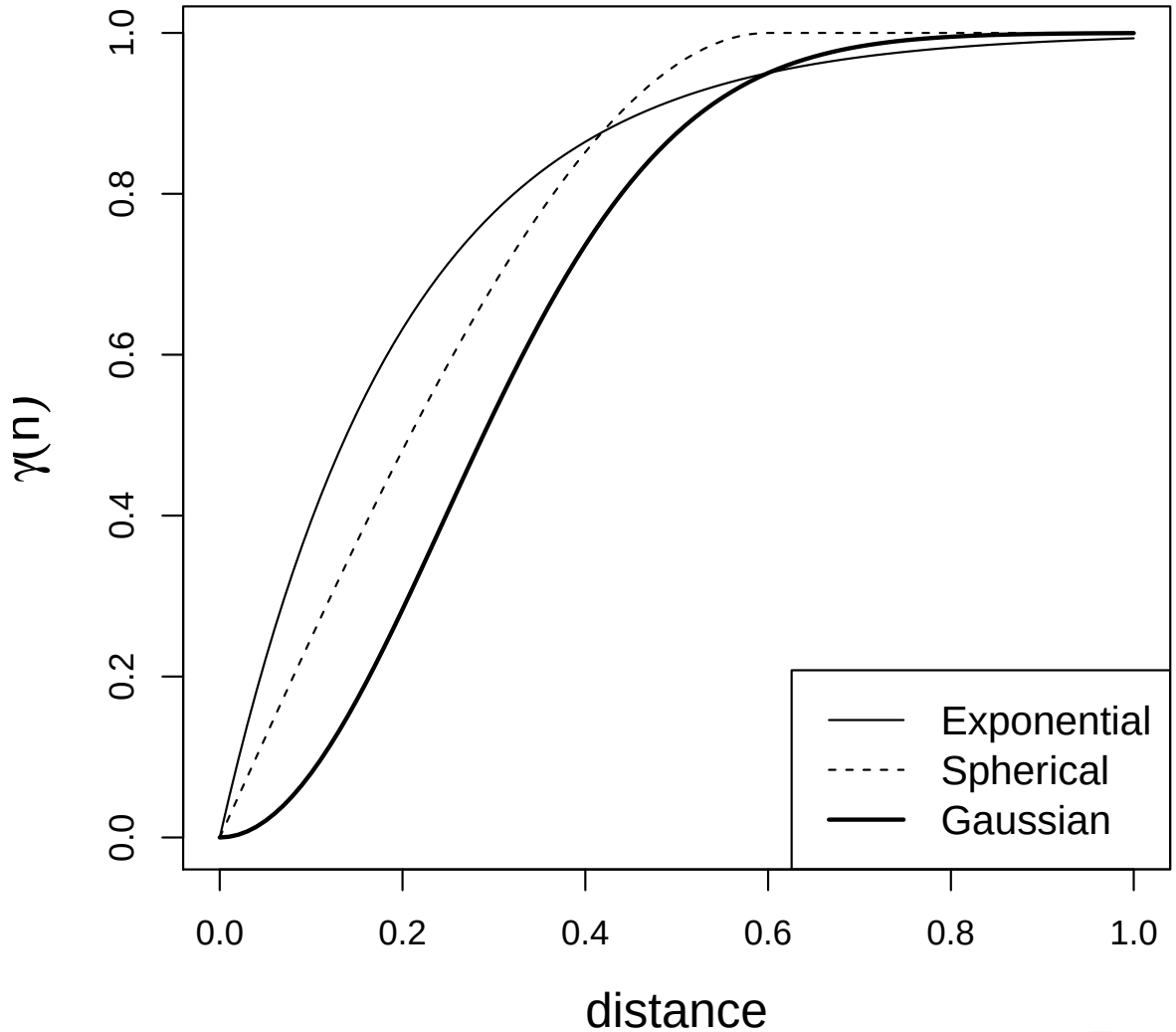
Gaussian, $\phi=0.1$



wave, $\phi=0.3$



Variogram models with the same ”practical” range



An example of simulation with R

The function `grf()` generates simulations of Gaussian random fields for given variogram (covariance) parameters.

- Define the number of spatial locations in each simulations. The locations are taken at random on the unit square $[0, 1]^2$.

```
> n<-100
```

- Define the model (here the exponential model). See the help of the `cov.spatial` function for further details.

```
> cov.model<-"exp"
```

- Define σ^2 (partial sill) and ϕ range parameter.

```
> cov.pars <- c(1, .25)
```

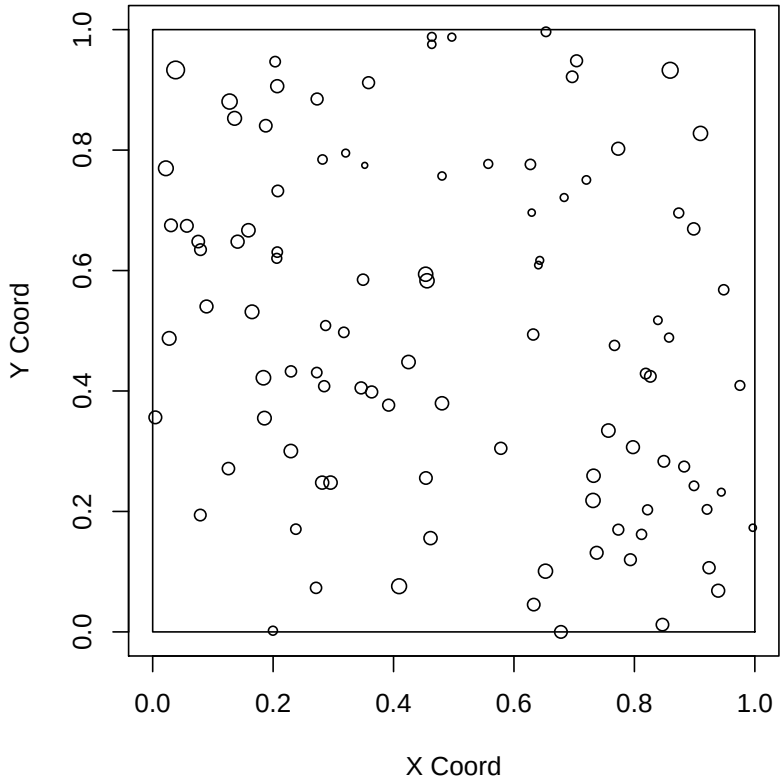
- Do the simulations

```
> mysim <- grf(n = n,cov.model= cov.model,cov.pars = cov.pars)
```

- Display the simulated locations and values

```
> points(mysim)
```

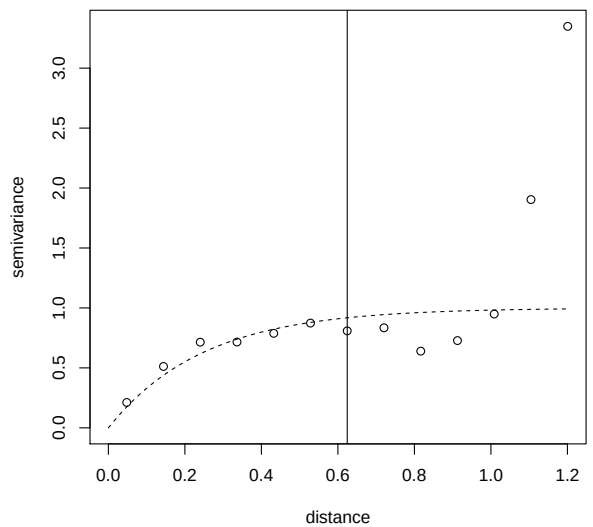
An example of simulation with R



An example of simulation with R

- Compare the empirical vs theoretical variogram

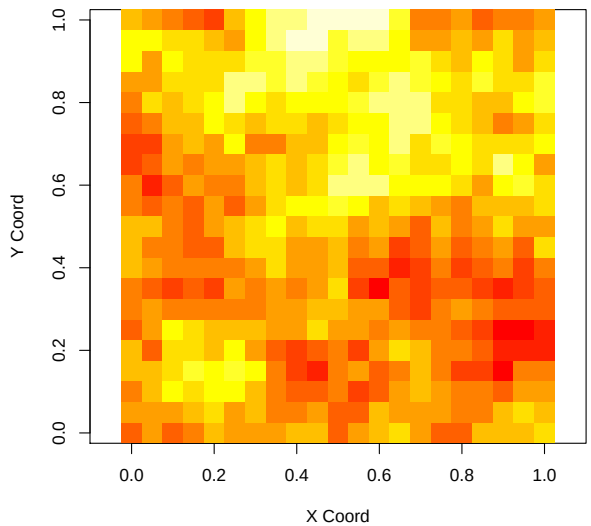
```
> plot(mysim)
> abline(v=max(dist(mysim$coords))/2)
```



An example of simulation with R

- An example on a regular grid

```
> n<-441
> mysim2 <- grf(n=n, grid = "reg",cov.pars = cov.pars, cov.model = cov.model
> image(mysim2)
```



Variogram analysis and model fitting

- For prediction we need a variogram function, that will be used at any possible distance
- Hence, we need a theoretical variogram, estimated using the empirical variogram
- But remember: theoretical variograms are special functions (cond. neg. definite)
- General approach: first choose a valid variogram among the possible ones
- Then estimate the parameters that best fit the theoretical variogram to the empirical one

Estimation methods

Weighted least squares

Find the parameters that minimize

$$\sum_{k=1}^{K} \frac{N(h_k)}{\gamma^2(h_k;\theta)} \{\hat{\gamma}(h_k) - \gamma(h_k;\theta)\}^2$$

where

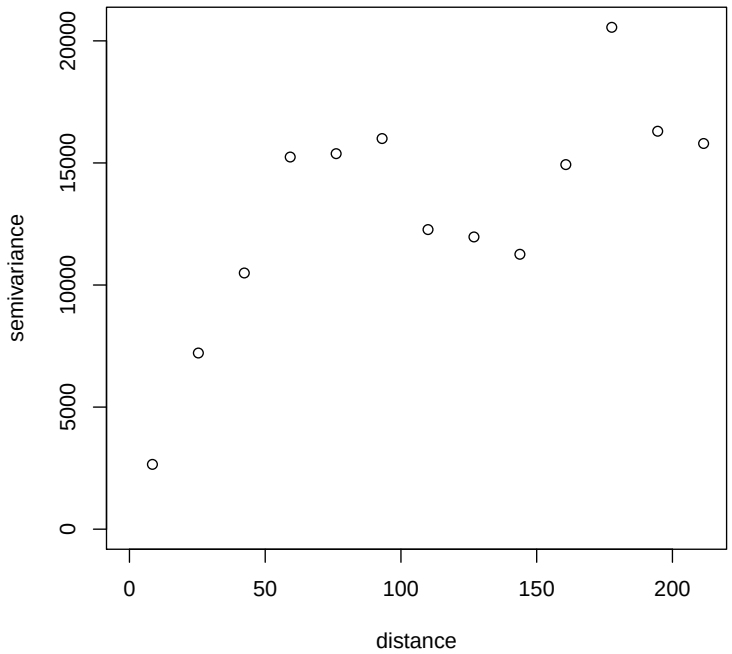
- $\gamma(h; theta)$ is a variogram function with parameters θ
- $\hat{\gamma}(h_k)$ is the empirical variogram
- both are computed at distances h_k, with $k = 1, \ldots, K$

- More emphasize at short distances
- Most popular method
- Quite robust results

Other methods: maximum likelihood; composite likelihood, ...

Example: Switzerland data

```
> data(SIC)
> max.dist<-220
> sic.bin<-variog(sic.100,max.dist=max.dist)
> plot(sic.bin)
```



Example: Switzerland data

- We need starting values for the sill and the range

```
> ini<-c(15000,50)
```

- We choose the model

```
> cov.model <- 'exp'
```

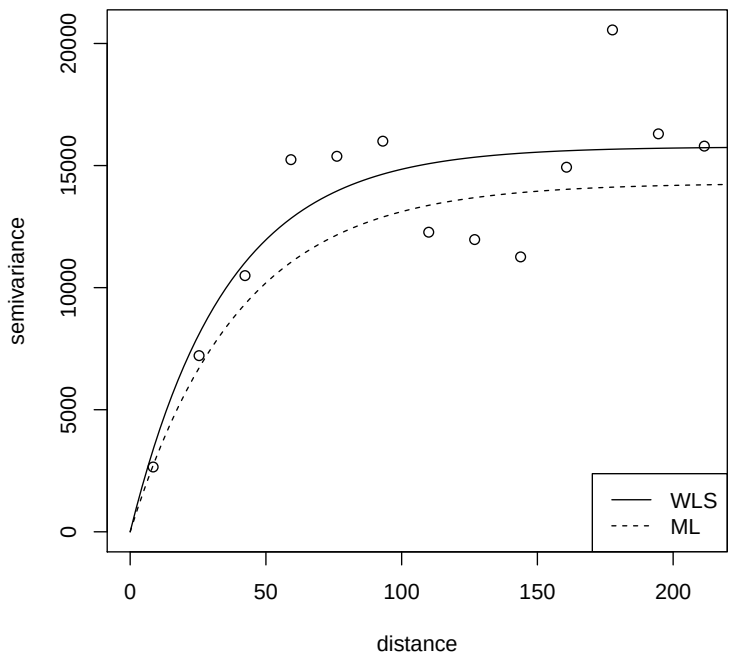
- we fit the model using two different criteria

```
> wls.fit <- variofit(sic.bin, ini = ini, cov.model=cov.model,weights="cress
                      fix.nugget=TRUE)
> ml.fit <- likfit(sic.100, ini = ini, cov.model=cov.model,fix.nugget=TRUE)
```

Example: Switzerland data

- We plot the fitted theoretical variogram and we contrast it with the empirical one

```
> plot(sic.bin)
> lines(wls.fit,lty=1)
> lines(ml.fit,lty=2)
> legend("bottomright", legend=c("WLS","ML"), lty=c(1,2))
```



What sample size to fit a variogram model ?

- Can't use non-spatial formulas for sample size, because spatial samples are correlated, and each sample is used multiple times in the variogram estimate
- Stochastic simulation from an assumed random field with a known variogram suggests:
  - $<$ 50 points: not at all reliable
  - 100 to 150 points: more or less acceptable
  - $>$ 250 points: almost certaintly reliable
- This is very worrying for many environmental datasets (soil cores, vegetation plots, ...) especially from short-term fieldwork, where sample sizes of 40 - 60 observations are typical !

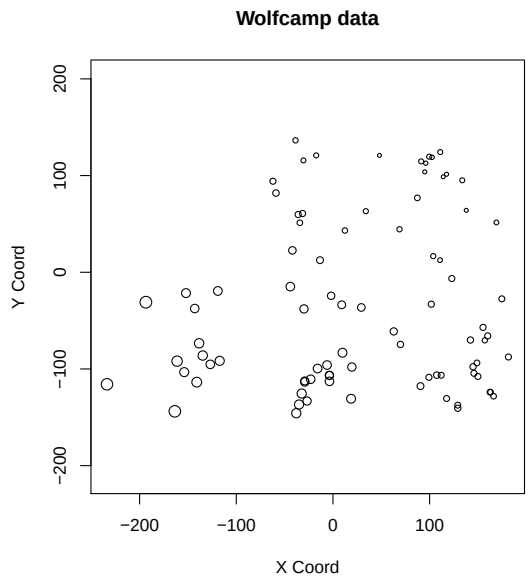
Spatial prediction

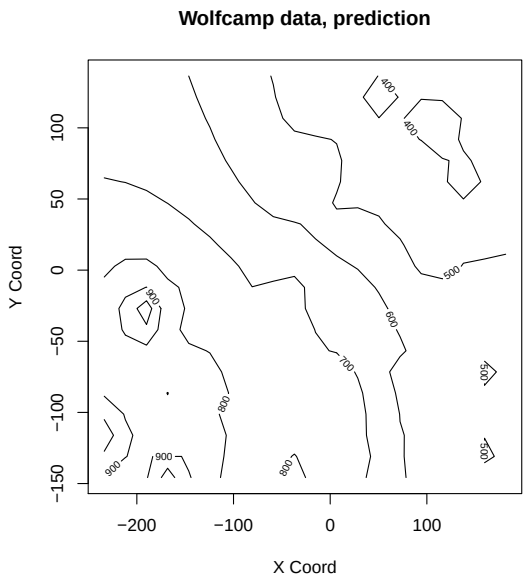
Spatial prediction from point samples is one of the main practical applications of geostatistics

- Objective: we know the value of some attribute at some observation locations, but we need to know it at an unsampled site. Our prediction at a site s will be denoted by $\hat{Z}(s)$.
- Prediction can be at:
  - Selected points of particular interest;
  - All points on a grid; the result is a map of the spatial field at the grid resolution

Spatial prediction: Model-based or not?

- A predictor is called model-based or geostatistical if it requires a model of spatial structure.
- The most common is Kriging; the geostatistical basis is the variogram model
- Otherwise it is called non-parametric and makes no assumption about spatial dependence
  - An example is inverse-distance weighted average
  - Another example is local smoother such as `loess()`





What is kriging ?

- Kriging is a spatial prediction algorithm based on a continuous model of stochastic spatial variation.
- Kriging $=$ prediction of the small scale random component process, using the variogram
- Different types of kriging exist, which pertains to the assumptions about the large scale component (mean structure) of the spatial model

$$E[]Z(s)] = \mu(s)$$

- Taxonomy
  - Simple kriging: The large scale component is a known constant, i.e.

$$E[Z(s)] = \mu$$

    .
  - Ordinary kriging: The large scale component is unkown but constant
  - Universal kriging: The large scale component is unknown but a linear combination of known functions of locations

$$\mu(s) = \sum_{i=0}^{p} \beta_i f_i(s).$$

    where the parameters $\beta_1, \ldots, \beta_p$ have to be estimated.

What is kriging ?

- Predicts at any location using a weighted average of measured values

$$\hat{Z}(s) = \lambda_1 z(s_1) + \cdots + \lambda_n z(s_n)$$

- How to choose the weights ?
  1. We impose the prediction to be unbiased
  2. The weights are chosen in order to minimize the MSE at each location (in this sense it is a optimal predictor)

$$MSE(s) = E[Z(s) - \hat{Z}(s)]^2$$

- Kriging weights $(\lambda_1, \ldots, \lambda_n)$ are derived as solution of the kriging linear system of equations
- As part of the solution of the kriging system we get the MSE of each prediction

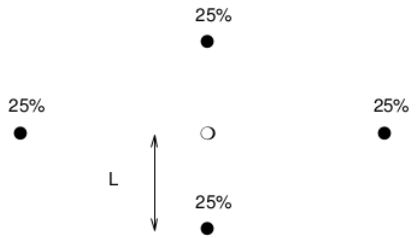
Kriging weights

They depend on:

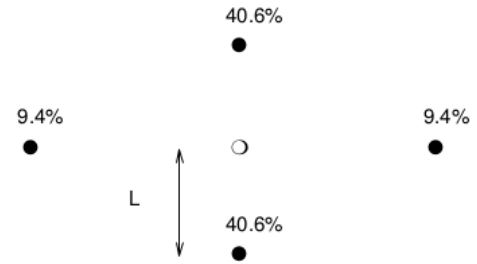
- Variogram model and its parameters
- The spatial pattern of samples points
- The location of the prediction point w.r.t. sample points
- They do not depend on the values $z(s_i)$

Toy example (1)

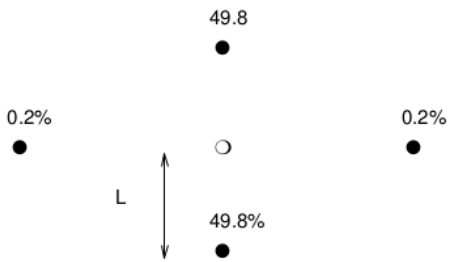
Nugget-effect model $\boxed{\sigma^2_{\text{OK}} = 1.25}$



Spherical model with range a/L = 2 $\boxed{\sigma^2_{\text{OK}} = .84}$



Gaussian model with range a/L = 1.5 $\boxed{\sigma^2_{\text{OK}} = .30}$



Top: No spatial effect. Middle: Spherical($a = 2L$). Bottom: Gaussien ($a = 1.5L$

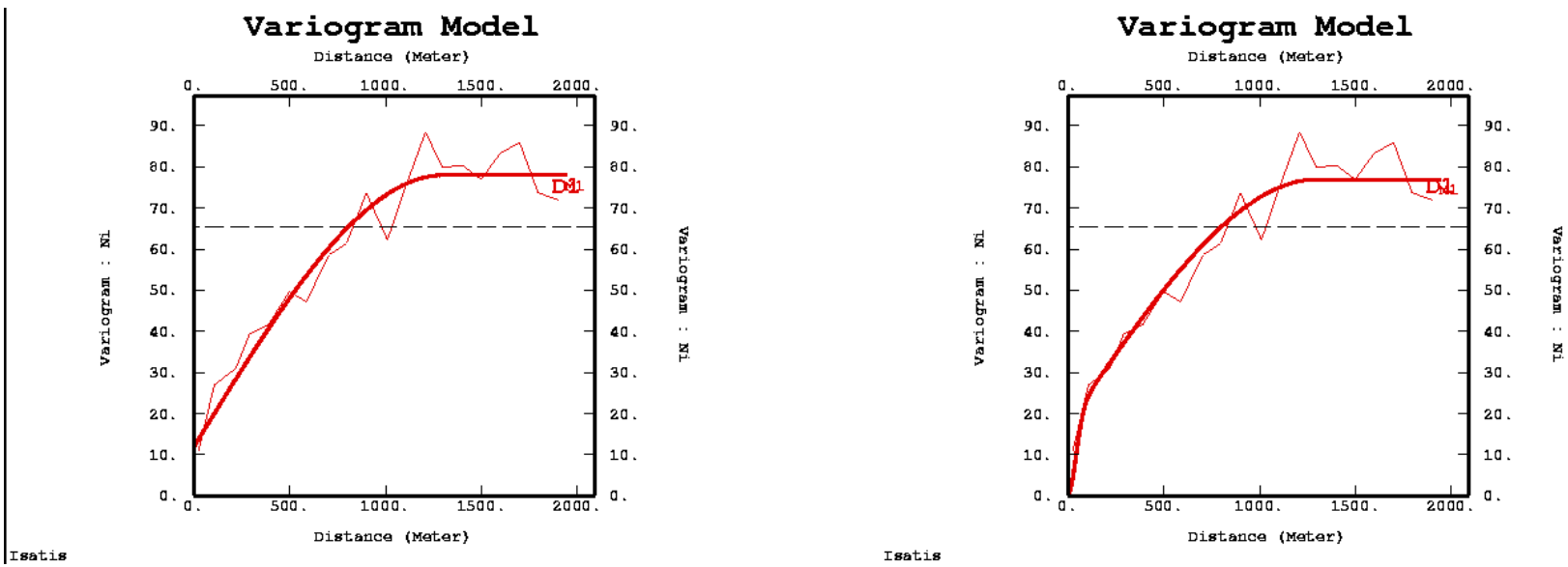
Toy example (2)

$\sigma^2_{\text{OK}} = 1.14$

$\sigma^2_{\text{OK}} = 0.87$

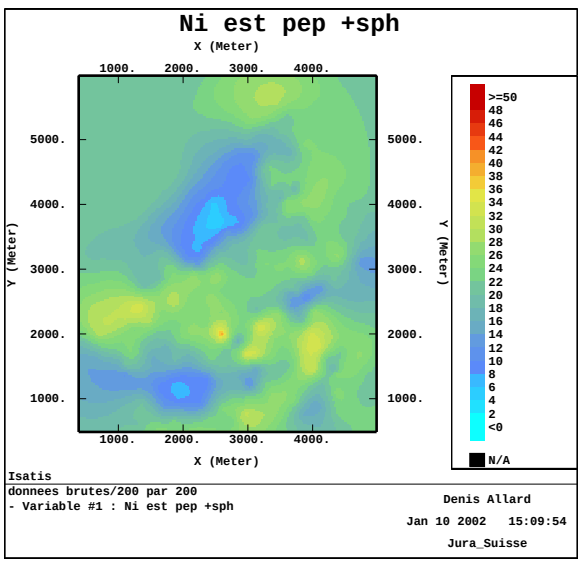
Spherical($a = 2L$)

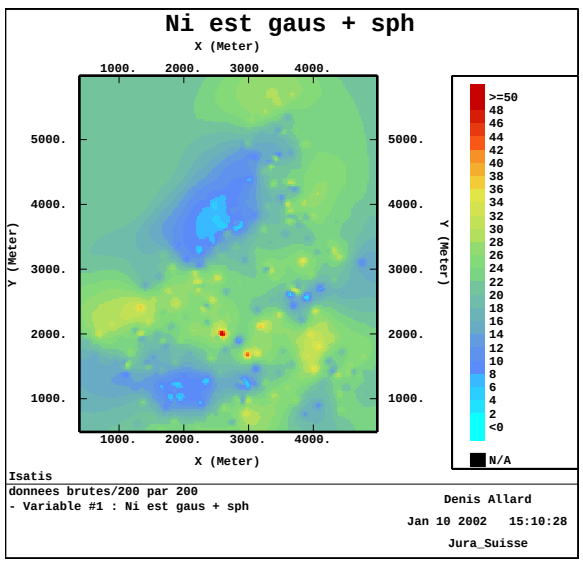
Illustration: variograms



Left: Nugget + Spherical; Right: Gaussian very short range + Spherical

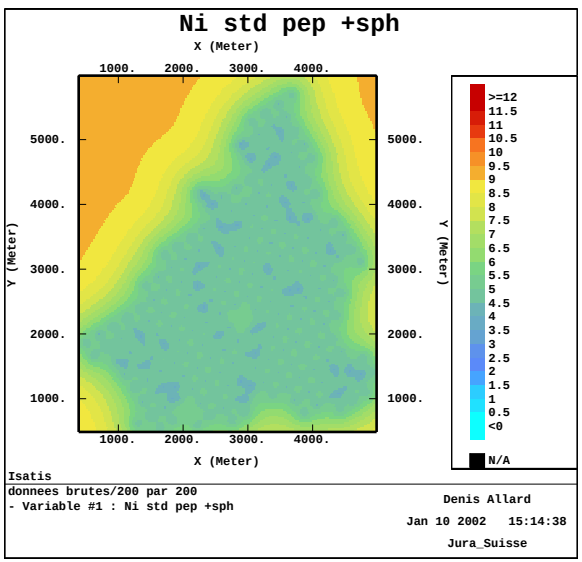
Illustration: Kriging

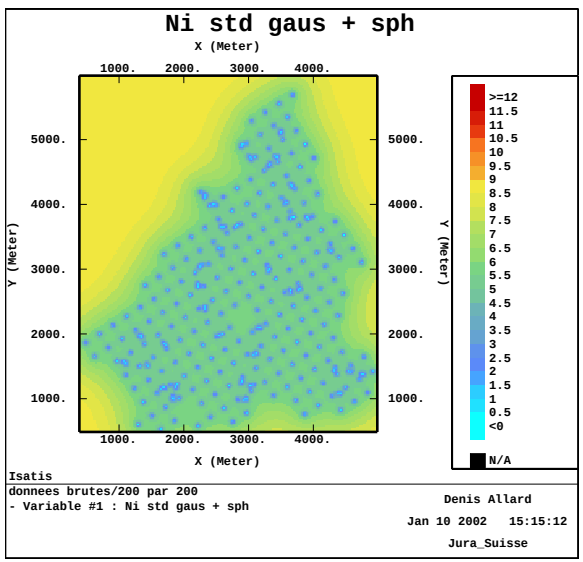




Left: Nugget + Spherical; Right: Gaussian very short range + Spherical

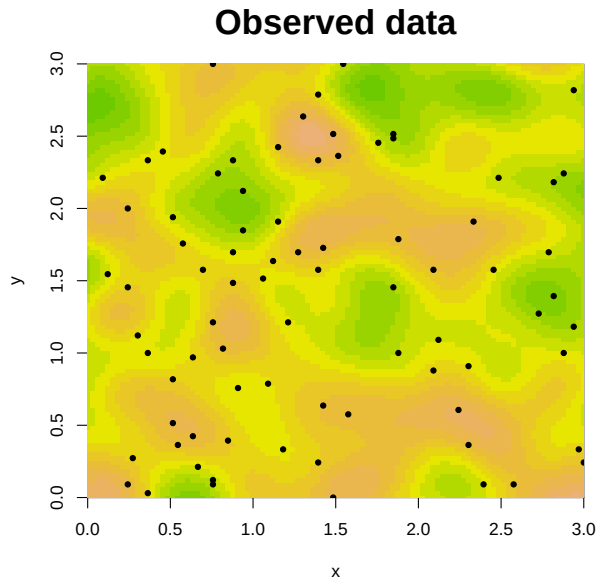
Illustrations: Kriging variance

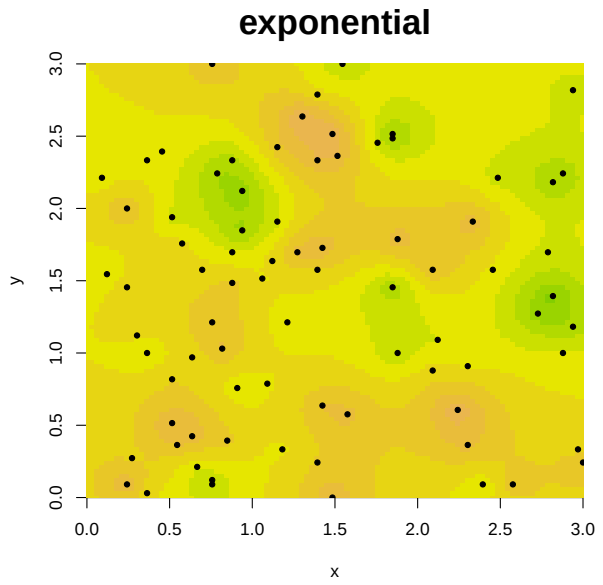


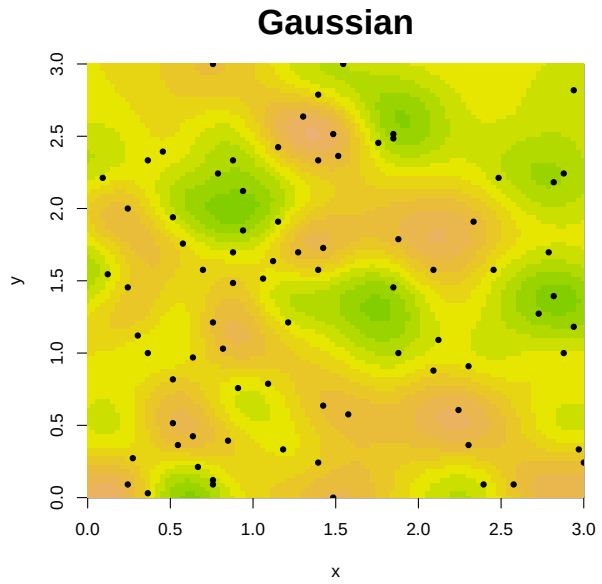


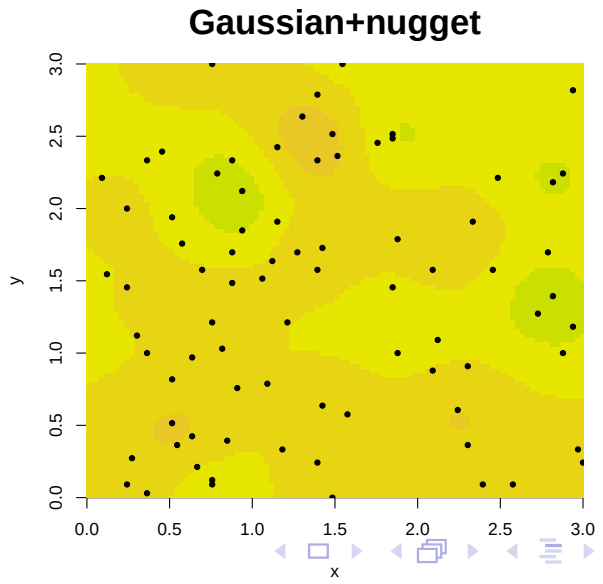
Left: Nugget + Spherical; Right: Gaussian very short range + Spherical

Illustration of Ordinary Kriging









How do we use Kriging in practice ?

1. Sample, preferably at different resolutions: stratified random sampling
2. Calculate the experimental variogram
3. Model the variogram with one or more variogram models
4. Apply the kriging system of equations, with the variogram model of spatial dependence, at each point to be predicted
5. Predictions are often at each point on a regular grid (e.g. a raster map)
6. As part of the solution of the kriging system, calculate the variance of each prediction; this is based only on the sample point locations, not their data values.
7. Display maps of the predictions and their mean squared errors.

Ready-to-use functions from packages `gstat`, `geoR` and `RandomFields`

Example: daily rainfall in Switzerland

1. Sample, preferably at different resolutions

```
> data(SIC)
```

2. Calculate the experimental variogram

```
> data(SIC)
> max.dist<-220
> sic.bin<-variog(sic.100,max.dist=max.dist)
```

3. Model the variogram with one or more variogram models

```
> cov.model<-"exp"
> ini<-c(15000,50)
> wls <- variofit(sic.bin, ini = ini,cov.model=cov.model,
                  weights="cressie",fix.nugget=TRUE)
```

4. Apply the kriging system of equations, with the variogram model of spatial dependence, at each point to be predicted. Predictions are often at each point on a regular grid (e.g. a raster map).

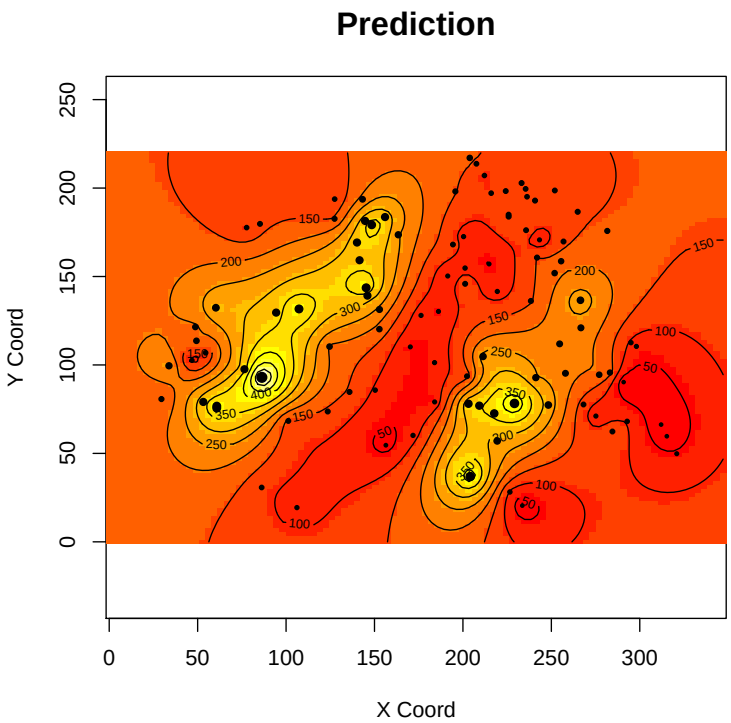
```
> ngridx<-100
> ngridy<-100
> xgrid<-seq(min(sic.borders[,1]),max(sic.borders[,1]),l=ngridx)
> ygrid<-seq(min(sic.borders[,2]),max(sic.borders[,2]),l=ngridy)
> pred.grid <- expand.grid(xgrid,ygrid)
> krige.par<-krige.control(type.krige='ok',cov.pars=wls$cov.pars,
                           cov.model=wls$cov.model)
> ksic<-krige.conv(sic.100,locations=pred.grid,krige=krige.par)
```

Example: daily rainfall in Switzerland

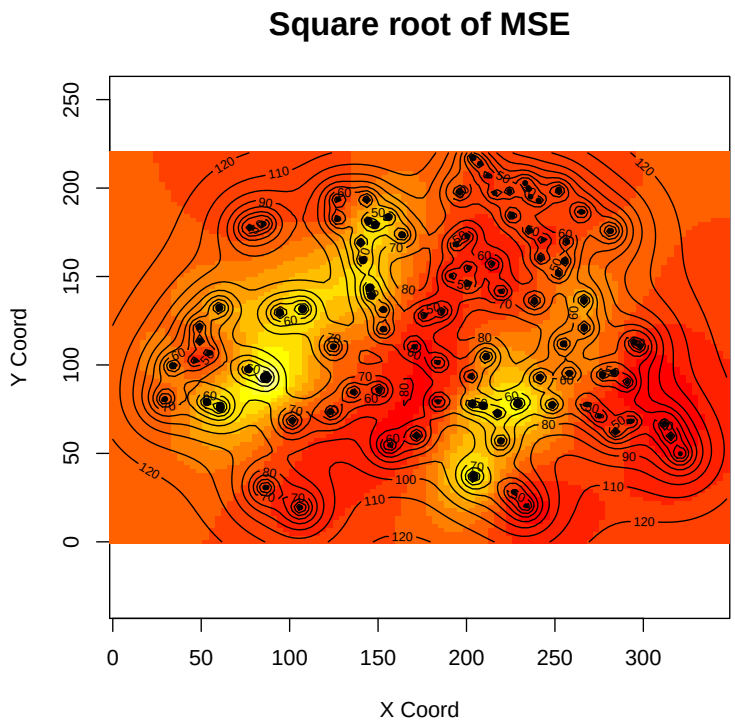
5. As part of the solution of the kriging system, calculate the variance of each prediction; this is based only on the sample point locations, not their data values.
6. Display maps of the predictions and their (square root) mean squared errors.

```
> image(ksic,main ="Prediction")
> contour(ksic,add=TRUE)
> points(sic.100,pch=20,add=TRUE)
> se<-sqrt(ksic$krige.var)
> image(ksic, main ="Square root of MSE")
> contour(ksic, val=se,add=TRUE)
> points(sic.100,pch=20,add=TRUE)
```

Example: daily rainfall in Switzerland



Kriging



$\sqrt{\text{MSE}}$

Use of the prediction and the MSE

- One of the major advantages of kriging is that it produces both a prediction $\hat{Z}(s)$ and its mean squared error $MSE(s)$.
- This can be used to construct prediction intervals around the predicted value
- The two-sided interval which has (approximately) probability $(1-\alpha)$ of containing the unboserved value $Z(s)$ is:

$$[\hat{Z}(s) - q_{1-\alpha/2}\sqrt{MSE(s)}, \hat{Z}(s) + q_{1-\alpha/2}\sqrt{MSE(s)}$$

  $q_{1-\alpha/2}$ quantile of the standard Gaussian distribution.
- For instance if you want a prediction interval which has (approximately) probability 0.95 choose

$$[\hat{Z}(s) - 1.96\sqrt{MSE(s)}, \hat{Z}(s) + 1.96\sqrt{MSE(s)}$$

Cross-validation

The underlying idea in cross-validation is to put aside each observation in turn and use kriging to predict its value using the other observations without again estimating the variogram model.

- For each site we then have an observed value $z(s_i)$ and a predicted value $\widehat{Z}(s_i)$.
- We then compute

$$\begin{aligned} BIAS &= 1/n \sum_{i=1}^{n} (z(s_i - \widehat{Z}(s_i))), \\ RMSE &= 1/n \sum_{i=1}^{n} (z(s_i - \widehat{Z}(s_i))^2, \end{aligned}$$

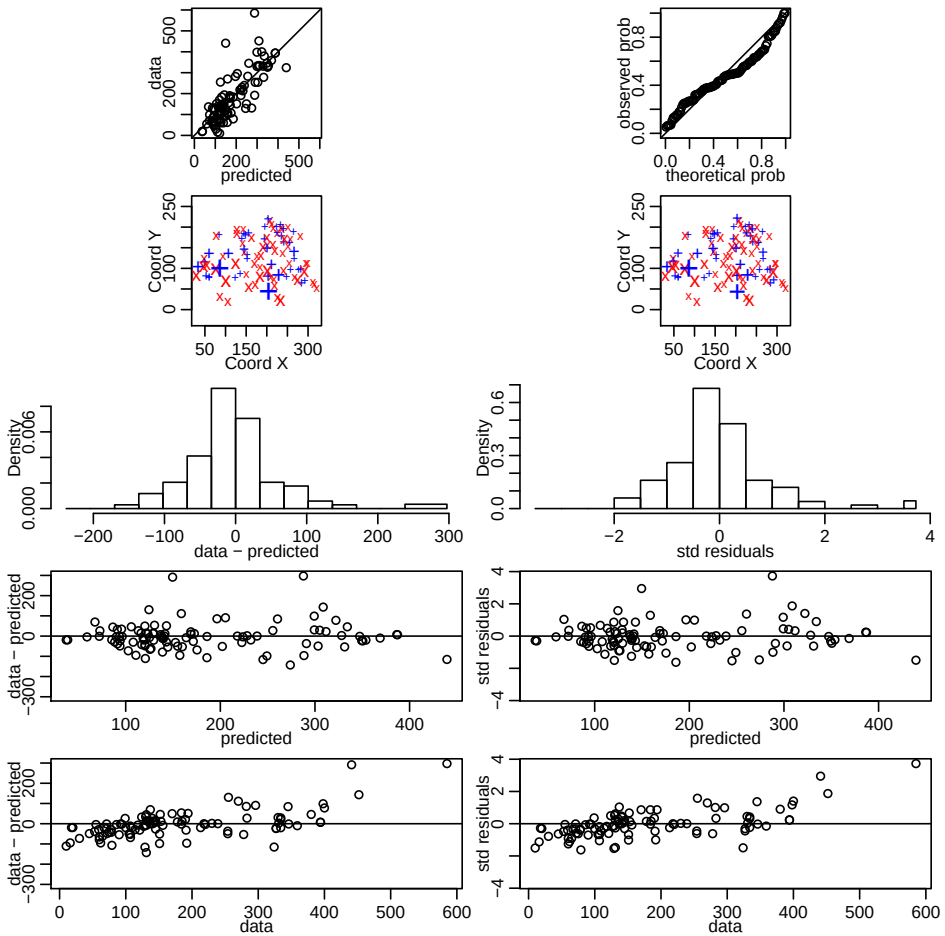
and

$$CV = \frac{1}{n} \sum_{i=1}^{n} \frac{(z(s_i) - \widehat{Z}(s_i))^2}{MSE(s_i)},$$

- If the variogram model is correctly identified and well-estimated, then the BIAS should be close to 0 and CV should be close to 1.
- Residuals $z(s_i) - \widehat{Z}(s_i)$ can be also analyzed !

Cross-validation in R

```
> xv <- xvalid(sic.100, model = wls)
> cv<-mean(xv$std.error^2)
> plot(xv)
> print(cv)
[1] 0.7233951
```



Comparing two variogram models

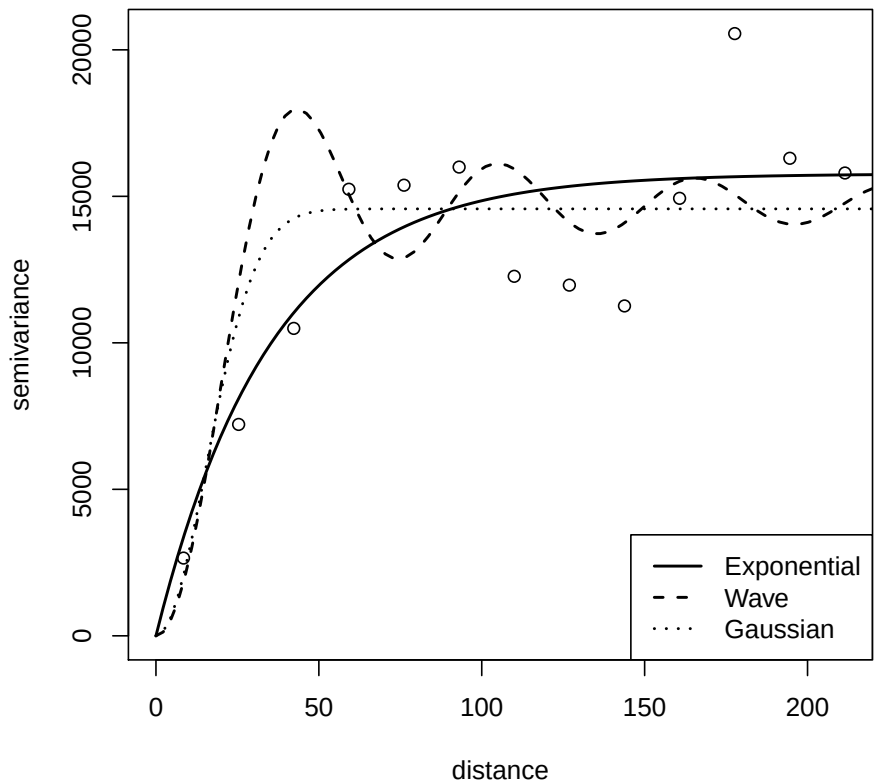
- For the sake of comparison, consider and fit a Gaussian model

```
> cov.model<-"gauss"
> ini<-c(15000,30)
> wls.gauss <- variofit(sic.bin, ini = ini,cov.model=cov.model,
                        weights="cressie",fix.nugget=TRUE)
> xv <- xvalid(sic.100, model = wls.gauss)
> cv<-mean(xv$std.error^2)
> print(cv)
[1] 3.730225
```

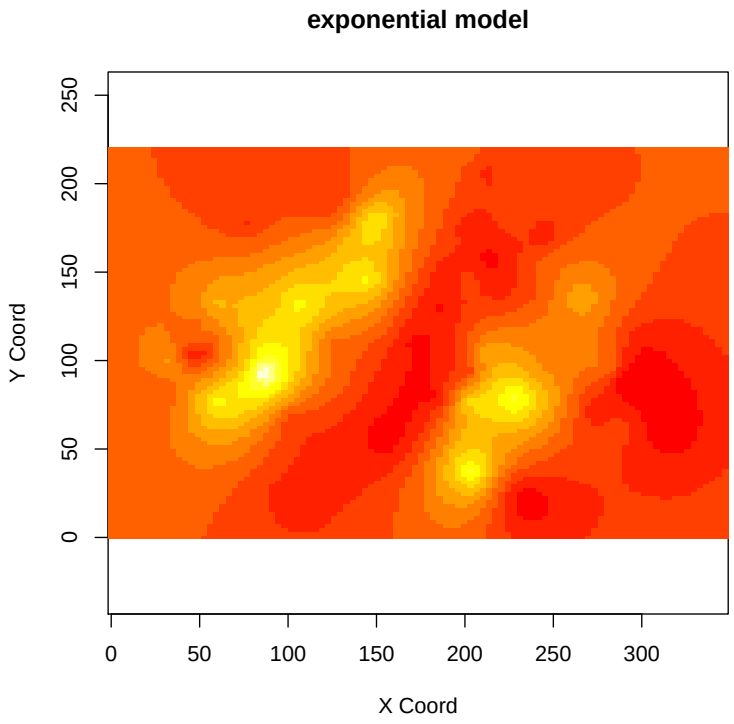
Comparing two variogram models

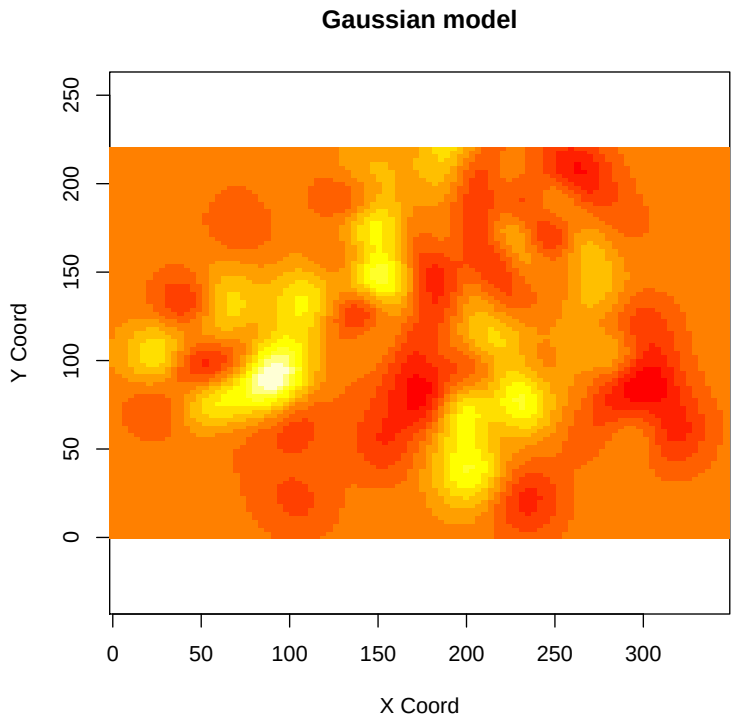
- We compare the fitting

```
> plot(sic.bin)
> lines(wls.fit,lwd=1.8)
> lines(wls.gauss,lty=3,lwd=1.8s)
> legend("bottomright",legend=c("Exponential","Gaussian"),
        lty=c(1,3),lwd=1.8)
```



Comparing two variogram models





Thank you very much for your attention and your willingness