Modeling Data with Dependencies

L. Torgo

ltorgo@fc.up.pt

Faculdade de Ciências / LIAAD-INESC TEC, LA
Universidade do Porto

Jan, 2017

Jožef Stefan Institute

Time Series Modeling

Time Series Data

A Definition

Definition

- A time series is a set of observations of a variable that are ordered by time.
- E.g.,
  $x_1, x_2, \cdots, x_{t-1}, x_t, x_{t+1}, \cdots, x_n$
  where x_t is the observation of variable X at time t.
- A multivariate time series is a set of observations of a set of variables over a certain period of time.

Jožef Stefan Institute

The Main Goals of Time Series Analysis

Explanation

Obtaining a Time Series Model help us to have
a Deeper Understanding of the Mechanism
that Generated the Observed Time Series Data.

Forecasting

- Given: $x_1, x_2, \cdots, x_{t-1}, x_t$ *The Past!*
- Obtain: a time series model
- Which is able to make predictions concerning:
  $x_{t+1}, \cdots, x_n$ *The Future!*

Other Goals

Time Series Data Mining

Main Time Series Data Mining Tasks

- *Indexing (Query by Content)*
  Given a query time series Q and a similarity measure $D(Q, X)$ find the most similar time series in a database **D**
- *Clustering*
  Find the natural goupings of a set of time series in a database **D** using some similarity measure $D(Q, X)$
- *Classification*
  Given an unlabelled time series Q, assign it a label C from a set of pre-defined labels (classes)

Jožef Stefan Institute

Summaries of Time Series Data

- Standard descriptive statistics (mean, standard deviation, etc.) do not allways work with time series (TS) data.
- TS may contain trends, seasonality and some other systematic components, making these stats misleading.
- So, for proving summaries of TS data we will be interested in concepts like trend, seasonality and correlation between sucessive observations of the TS.

Jožef Stefan Institute

Types of Variation

Seasonal Variation

Some time series exhibit a variation that is annual in period, e.g. demand for ice cream.

Other Cyclic Variation

Some time series have periodic variations that are not related to seasons but to other factors, e.g. some economic time series.

Trends

A trend is a long-term change in the mean level of the time series.

Time Series Decomposition

- It is frequent to decompose time series in three components:
  - Trend
  - Seasonal
  - Remainder component
- It is also frequent to classify the decomposition as been
  - Additive
    $Y_t = T_t + S_t + R_t$, where T_t is the value of the trend component at time t and S_t, R_t the values of the other components
  - Multiplicative, where the seasonal component is proportional to the level of the series
    $Y_t = T_t \times S_t \times R_t$

A Small Illustration in R

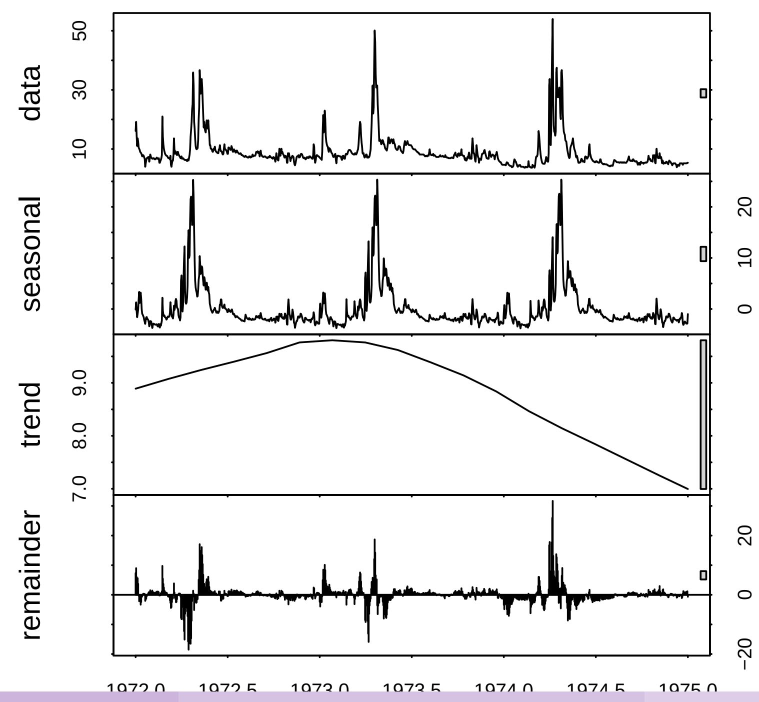
```r
data(ice.river,package='tseries')
head(ice.river)

## [1] 16.1 19.2 14.5 11.0 13.6 12.5

decomp <- stl(ice.river[,1], s.window=10)
```

STL (Seasonal and Trend decomposition with Loess) is robust decomposition based on the Loess non-linear regression method.

Jožef Stefan Institute

A Small Illustration in R - 2

```r
plot(decomp)
```



time

Jožef Stefan Institute

Stationarity

An Informal Definition

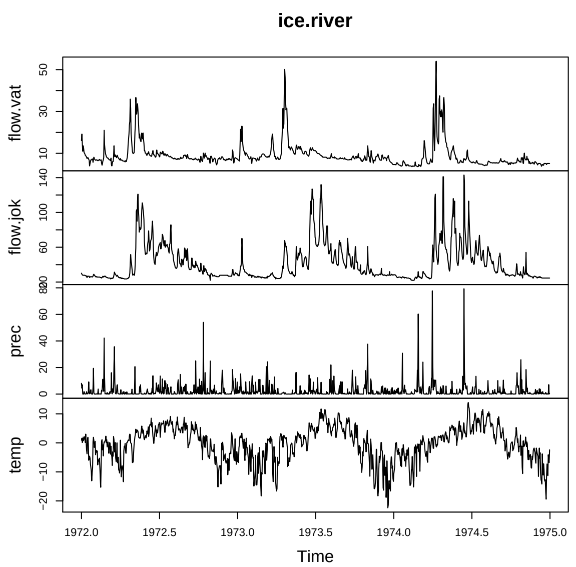
A time series is said to be stationary if

- there is no systematic change in mean (no trend),
- if there is no systematic change in variance and
- if strictly periodic variations have been removed.

Note that in these cases statistics like mean, standard deviation, variance, etc., bring relevant information!

Jožef Stefan Institute

Time Plots

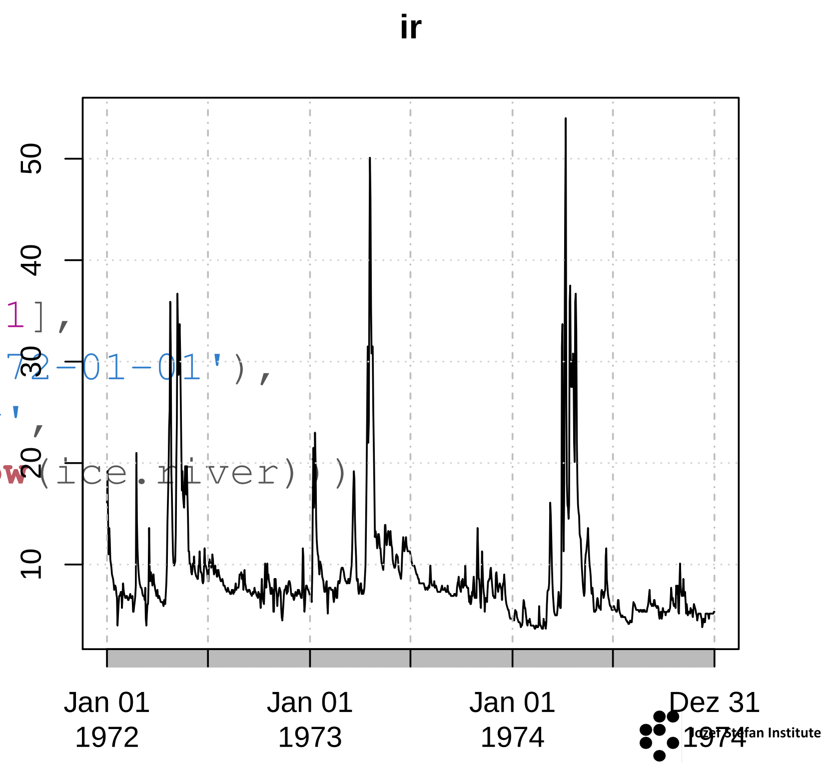


- Ploting the time series values against time is one of the most important tools for analysing its behaviour.
- Time plots show important features like trends, seasonality, outliers and discontinuities.

Jožef Stefan Institute

Time Plots in R

```
ir <- xts(ice.river[,1],
          seq(ymd('1972-01-01'),
              by='day',
              len=nrow(ice.river)))
plot(ir)
```



Transformations - I

Plotting the data may suggest transformations :

To stabilize the variance

Symptoms: trend with the variance increasing with the mean.
Solution: logarithmic transformation.

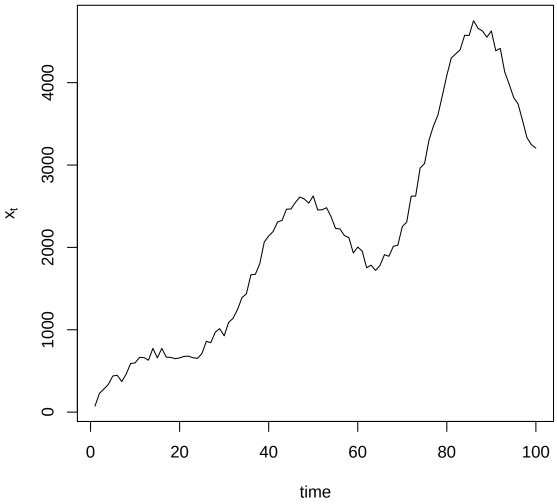
To make the seasonal effects additive

Symptoms: there is a trend and the size of the seasonal effect increases with the mean(multiplicative seasonality).
Solution: logarithmic transformation.

To remove trend

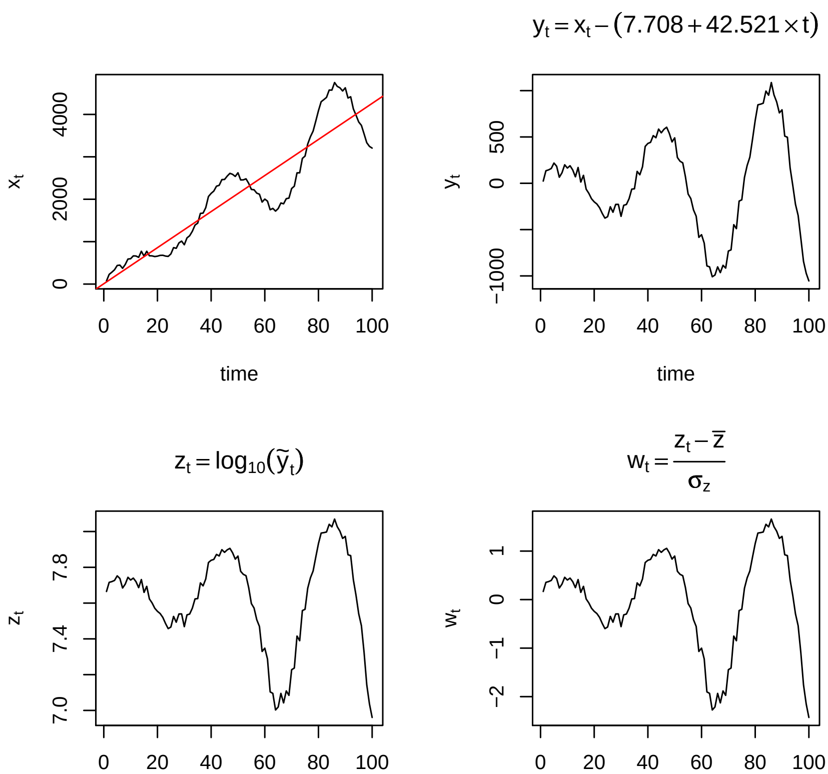
Symptoms: there is systematic change on the mean.
Solution 1: first order differentiation ($\nabla X_t = X_t - X_{t-1}$).
Solution 2: model the trend and subtractit from the original series ($Y_t = X_t - r_t$).

Transformations - a simple example (1)



An example time series with trend and a multiplicative seasonality effect.

Transformations - a simple example (2)



Some useful functions in R

```
(s <- ir[1:10])

##               [,1]
## 1972-01-01 16.10
## 1972-01-02 19.20
## 1972-01-03 14.50
## 1972-01-04 11.00
## 1972-01-05 13.60
## 1972-01-06 12.50
## 1972-01-07 10.50
## 1972-01-08 10.10
## 1972-01-09  9.68
## 1972-01-10  9.02

diff(s)

##               [,1]
## 1972-01-01    NA
## 1972-01-02  3.10
## 1972-01-03 -4.70
## 1972-01-04 -3.50
## 1972-01-05  2.60
## 1972-01-06 -1.10
## 1972-01-07 -2.00
## 1972-01-08 -0.40
## 1972-01-09 -0.42
## 1972-01-10 -0.66
```

```
diff(s,diff=2)

##               [,1]
## 1972-01-01    NA
## 1972-01-02    NA
## 1972-01-03 -7.80
## 1972-01-04  1.20
## 1972-01-05  6.10
## 1972-01-06 -3.70
## 1972-01-07 -0.90
## 1972-01-08  1.60
## 1972-01-09 -0.02
## 1972-01-10 -0.24

log10(s)

##                    [,1]
## 1972-01-01 1.2068259
## 1972-01-02 1.2833012
## 1972-01-03 1.1613680
## 1972-01-04 1.0413927
## 1972-01-05 1.1335389
## 1972-01-06 1.0969100
## 1972-01-07 1.0211893
## 1972-01-08 1.0043214
## 1972-01-09 0.9858754
## 1972-01-10 0.9552065
```


Autocorrelation

Sample Autocorrelation Coefficients

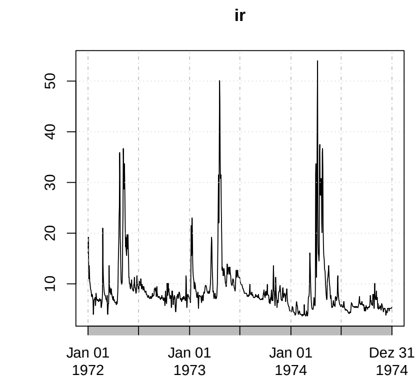
They measure the correlation between observations different distances apart.

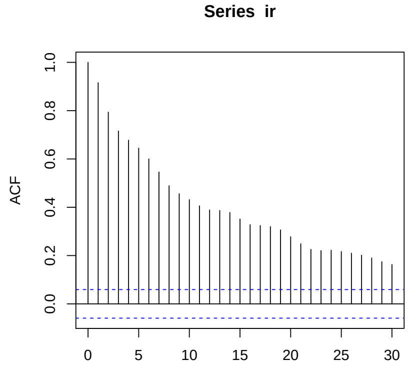
$$r_k = \frac{\sum_{t=1}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{N}(x_t - \bar{x})^2}$$

Jožef Stefan Institute

Correlograms in R

```r
plot(ir)
acf(ir)
```





Jožef Stefan Institute

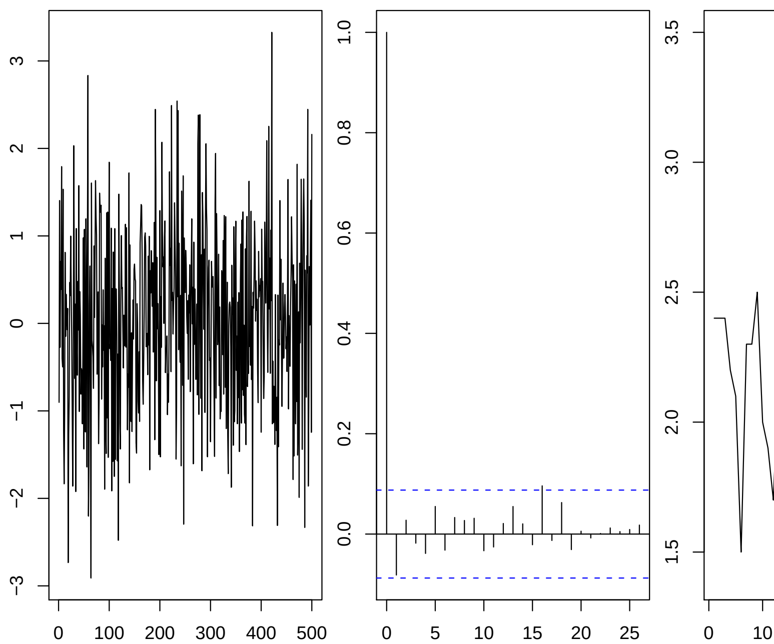
Interpreting the Correlogram

Random Series

Most r_k's near 0. Still, it is possible that 1 on 20 is significant...

Short-Term Correlation

Fairly large value of r_1 with sucessive values rapidly tending to non-significant.



Jožef Stefan Institute

Interpreting the Correlogram (cont.)

Alternating Series

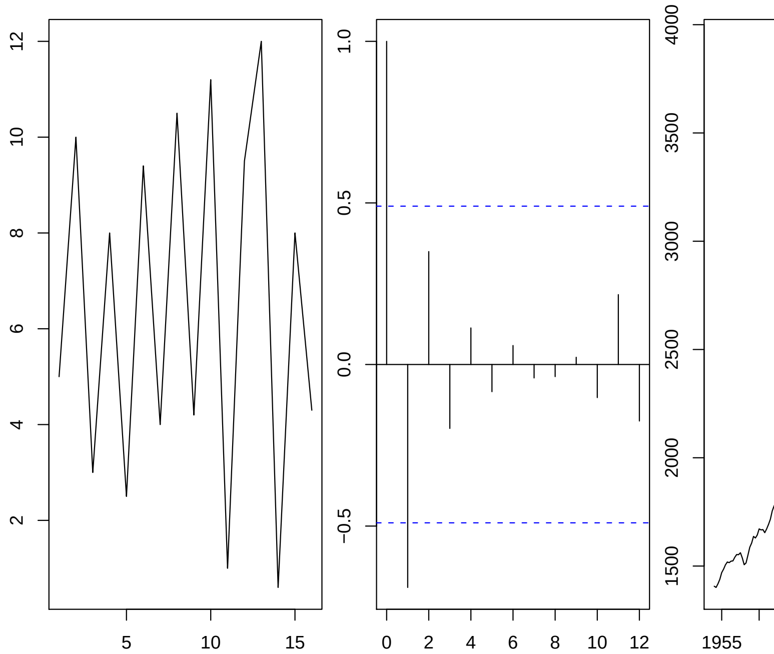
Similar pattern on the values of r_k.

Non-Stationary Series

For series with a trend the values of r_k will not go down till very large values of the lag.

Seasonal Series

The correlagram tends to exhibit the same periodicity as the original series.



Jožef Stefan Institute

Time Series Forecasting

- Given:
  $x_1, x_2, \cdots, x_{t-1}, x_t$ *The Past!*
- Obtain:
  a time series model
- Which is able to make predictions concerning:
  $x_{t+1}, \cdots, x_n$ *The Future!*

Jožef Stefan Institute

Moving Average Models

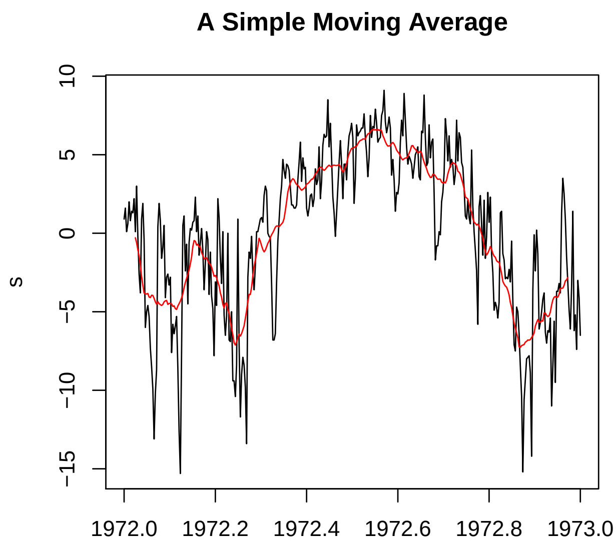
Definition

A moving average of order q, MA(q),is a time series given by

$$Y_t = \sum_{i=0}^{q} \beta_i X_{t-i}$$

Moving Average Models - example in R

```
library(forecast)
s <- window(ice.river[,"temp"],start=1972,end=1973)
plot(s, main="A Simple Moving Average")
lines(ma(s,order=20,centre=FALSE),col="red")
```



Exponential Moving Average Models

Definition

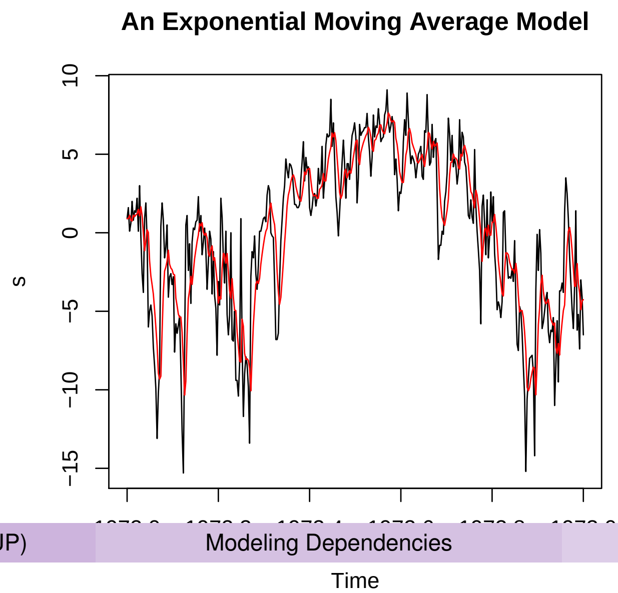
An exponential moving average is a series given by

$$\begin{aligned} Y_t &= \alpha \times X_t + (1 - \alpha) \times \mathrm{EMA}_\alpha(X_{t-1}) \\ Y_1 &= X_1 \end{aligned}$$

where $\alpha \in]0..1[$ is a smoothing parameter.

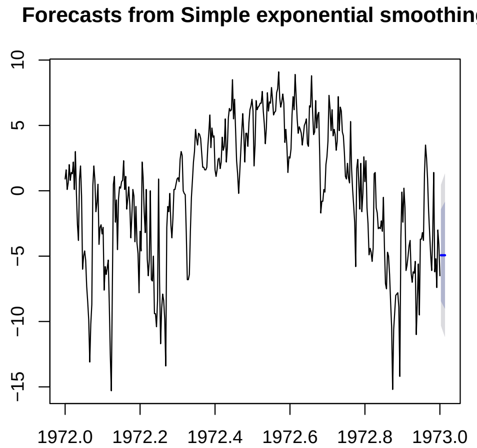
Exponential Moving Average Models - an example in R

```
library(forecast)
s <- window(ice.river[,"temp"],start=1972,end=1973)
model <- ses(s, alpha=0.3, initial="simple")
plot(s, main="An Exponential Moving Average Model")
lines(fitted(model),col="red")
```



Exponential Moving Average Models - forecasting

```
forecast(model,h=5)

##           Point Forecast       Lo 80      Hi 80     Lo 95     Hi 95
## 1973.003       -4.931377 -8.446668 -1.4160849 -10.30755 0.4447981
## 1973.005       -4.931377 -8.601449 -1.2613043 -10.54427 0.6815146
## 1973.008       -4.931377 -8.749961 -1.1127924 -10.77140 0.9086440
## 1973.011       -4.931377 -8.892909 -0.9698440 -10.99002 1.1272646
## 1973.014       -4.931377 -9.030876 -0.8318772 -11.20102 1.3382667

plot(forecast(model,h=5))
```



Holt-Winters Models

The model equations depend on the type of seasonality:

- *Multiplicative* - effects increase with the increase on the mean.
- *Additive* - effects are constant (just adding up to the series).

Definition

A prediction for the horizon h can be obtained with,

$\hat{X}_{t+h} = \hat{a}_t + h \times \hat{b}_t + \hat{s}_t$

where,

$\hat{a}_t = \alpha(X_t - \hat{s}_{t-p}) + (1-\alpha)(\hat{a}_{t-1} + \hat{b}_{t-1})$

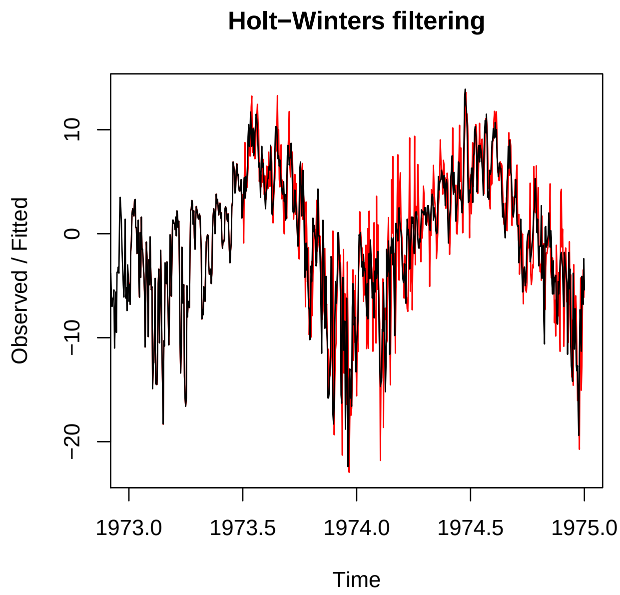
$\hat{b}_t = \beta(\hat{a}_t - \hat{a}_{t-1}) + (1-\beta)\hat{b}_{t-1}$

and $\hat{s}_t = \gamma(X_t - \hat{a}_t) + (1-\gamma)\hat{s}_{t-p}$ for additive seasonality of period p

$\hat{s}_t = \gamma\frac{X_t}{\hat{a}_t} + (1-\gamma)\hat{s}_{t-p}$ for multiplicative seasonality of period p

Holt-Winters Models - an example in R

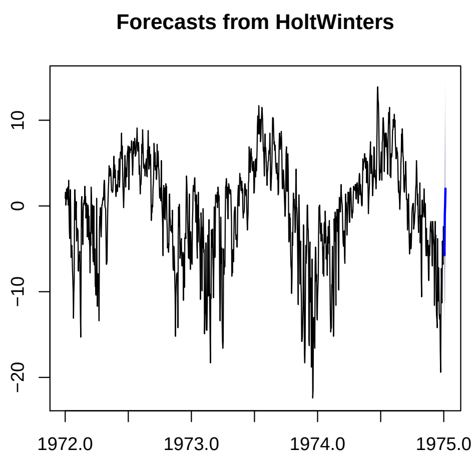
```
hw <-  HoltWinters(ice.river[,"temp"])
plot(hw)
```



Jožef Stefan Institute

Holt-Winters Models - forecasting

```
forecast(hw,h=5)

##          Point Forecast        Lo 80       Hi 80     Lo 95     Hi 95
## 1975.003     -4.2927349  -8.622913  0.03744281 -10.91517  2.329700
## 1975.005     -5.7463757 -11.371934 -0.12081711 -14.34993  2.857174
## 1975.008     -1.0681061  -7.742189  5.60597660 -11.27523  9.139023
## 1975.011      0.2402665  -7.338642  7.81917477 -11.35067 11.831206
## 1975.014      2.0234497  -6.363225 10.41012486 -10.80286 14.849763

plot(forecast(hw,h=5))
```



Jožef Stefan Institute

Autoregressive (AR) Models

Definition

An autoregressive model of order p is a series given by

$$Y_t = \sum_{i=0}^{p} \alpha_i Y_{t-i}$$

Mixed Autoregressive and Moving Average Models

Definition

A mixed ARMA model of order p, q is a series given by

$$Y_t = \sum_{i=0}^{p} \alpha_i Y_{t-i} + \sum_{i=0}^{q} \beta_i X_{t-i}$$

Integrated ARMA (or ARIMA) Models

Definition

An integrated ARMA (or ARIMA) model of order p, d, q is a series given by

$W_t = \sum_{i=0}^{p} \alpha_i W_{t-i} + \sum_{i=0}^{q} \beta_i X_{t-i}$

where $W_t = \nabla^d X_t$ is a d order difference.

Jožef Stefan Institute

ARIMA Models - an example in R

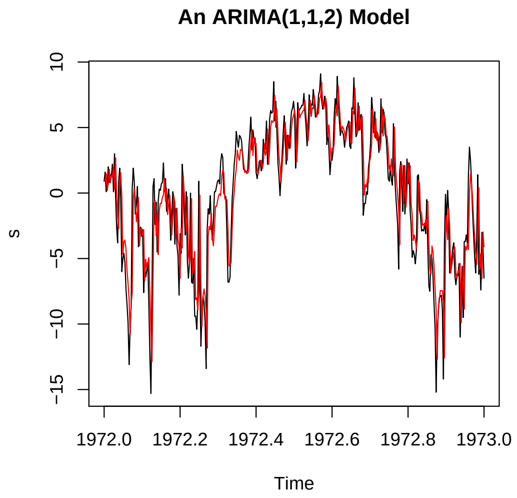
Defining your own model,

```
a1 <- Arima(s,order=c(1,1,2))
a1

## Series: s
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.6408  -0.8398  -0.0678
## s.e.  0.0888   0.1037   0.0782
##
## sigma^2 estimated as 5.732:  log likelihood=-835.33
## AIC=1678.65   AICc=1678.76   BIC=1694.25
```

ARIMA Models - an example in R

Defining your own model,

```
plot(s, main="An ARIMA(1,1,2) Model")
lines(fitted(a1),col="red")
```



Jožef Stefan Institute

ARIMA Models - an example in R

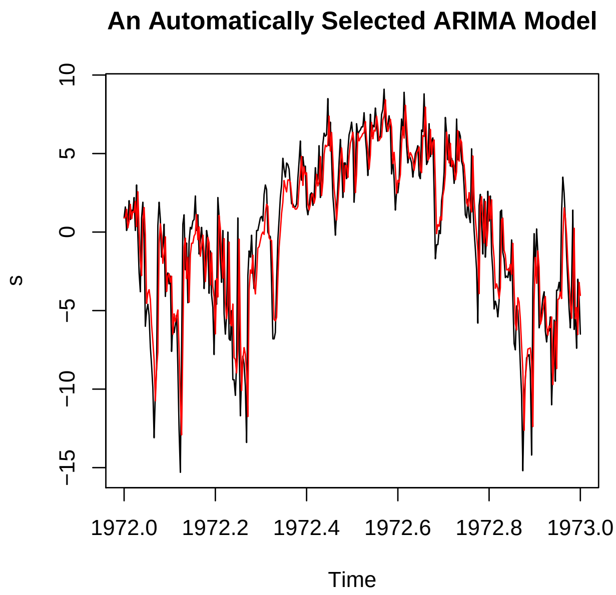
Automatically tuning the model,

```
a2 <- auto.arima(s)
a2

## Series: s
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.6936  -0.9235
## s.e.  0.0549   0.0281
##
## sigma^2 estimated as 5.728:  log likelihood=-835.7
## AIC=1677.39   AICc=1677.46   BIC=1689.09
```

ARIMA Models - an example in R

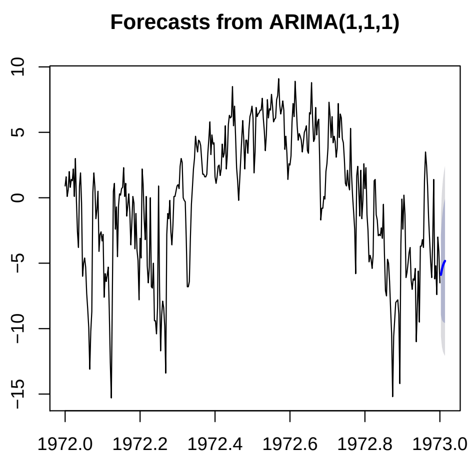
Automatically tuning the model,

```
plot(s, main="An Automatically Selected ARIMA Model")
lines(fitted(a2),col="red")
```



ARIMA Models - forecasting

```
forecast(a2,h=5)

##          Point Forecast     Lo 80      Hi 80     Lo 95      Hi 95
## 1973.003      -5.889288 -8.956410 -2.8221659 -10.58005 -1.1985295
## 1973.005      -5.465671 -9.337006 -1.5943351 -11.38637  0.4550261
## 1973.008      -5.171831 -9.472559 -0.8711026 -11.74923  1.4055654
## 1973.011      -4.968010 -9.534151 -0.4018698 -11.95132  2.0152991
## 1973.014      -4.826631 -9.573899 -0.0793639 -12.08695  2.4336877

plot(forecast(a2,h=5))
```



Delay-Coordinate Embedding

Theorem (Takens, 1981)

Informally, it states that given a correct embed size, e, and the right delay, τ, delay-coordinate embedding is sufficient to uncover the dynamics of any time series.

Definition

The embedded vector r_t, of dimension e, and delay τ, of a time series X, is obtained as,

$$r_t = < x_t, x_{t-\tau}, x_{t-2\tau}, \cdots, x_{t-(e-1)\tau} >$$

An Example of Delay-Coordinate Embedding

Example

Given the time series, $y_1, y_2, y_3, \cdots, y_{100}$, a embed dimension of 3 and a delay of 2, the embed vectors are,

$$\begin{aligned} r_5 &= < y_5, y_3, y_1 > \\ r_6 &= < y_6, y_4, y_2 > \\ r_7 &= < y_7, y_5, y_3 > \\ r_8 &= < y_8, y_6, y_4 > \\ &\ldots \end{aligned}$$

Consequences of Delay-Coordinate Embedding

If the system dynamics can be captured by a certain embed, then we may try to model the relationship between the state of the system and the future values of the series.

That is, we can try to obtain a model of the form,
$$Y_{t+h} = f(r_t)$$

This modelling task can be handled by any multiple regression tool!

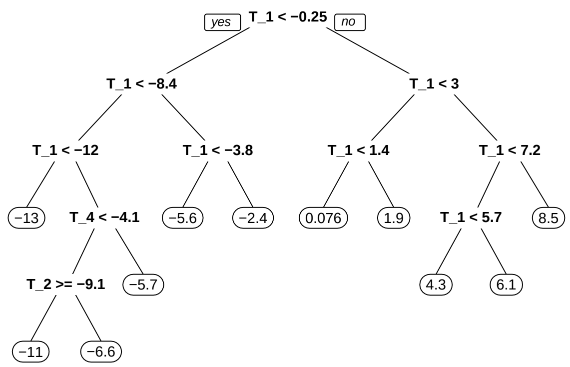
Jožef Stefan Institute

Creating an embed data set in R

```
library(DMwR2)
dat <- createEmbedDS(ice.river[,"temp"], emb = 6)
head(dat)

##         T T_1 T_2 T_3 T_4 T_5
## [1,] 0.8 2.0 0.6 0.1 1.6 0.9
## [2,] 1.4 0.8 2.0 0.6 0.1 1.6
## [3,] 1.3 1.4 0.8 2.0 0.6 0.1
## [4,] 2.2 1.3 1.4 0.8 2.0 0.6
## [5,] 0.1 2.2 1.3 1.4 0.8 2.0
## [6,] 3.0 0.1 2.2 1.3 1.4 0.8

head(ice.river[,"temp"])

## [1] 0.9 1.6 0.1 0.6 2.0 0.8
```

Using a regression tree to model

```
library(DMwR2)
library(rpart.plot)
tr <- rpartXse(T ~ ., as.data.frame(dat))
prp(tr)
```



Hands On Time Series

Package **quantmod** (an extra package that you need to install) contains several facilities to handle financial time series. Among them, the function `getSymbols` allows you to download the prices of financial assets from *yahoo finance*. Explore the help page of the function to try to understand how it works, and the answer the following:

1. Obtain the prices of Apple during the last year
2. Using these prices create a time series of the percentage variation of the Closing prices (tip: check function `Cl()` and `Delt` from package **quantmod**)
3. Create and embed data set of the previous series using function `createEmbedDS()` of package **DMwR2**
4. Split the data set in two consecutive periods. Train a random forest with the first and apply it to the second.
5. Analyse the results

Modeling and Forecasting Spatial Data

Spatial Interpolation/Imputation

Problem Definition/Motivation

- Filling in unknown values in geo-referenced data sets
- Data collection is not fully controllable and it is prone to failures
- Data incompleteness may be caused by poor data collection, measurement errors, costs management and many other factors

Illustrative Application Areas

Wind speed forecasting, oil resources analysis, water quality assessment, satellite images, pictures and/or paintings repair, surveillance, security, etc.

Jožef Stefan Institute

State of the art : Spatial Interpolators

1^{st} Law of Geography

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

Inverse Distance Weighing - IDW

Approximates values with the weighted average of the known neighbourhood values - weights inversely proportional to the distance from the target location.

Kriging

Kriging uses the same basic principle as IDW - weights are calculated using the covariation between known data at various spatial locations.

Potential Difficulties

Data Shortage

- On some applications data collection on different locations is hard
  - Costs
  - Access difficulties
  - Etc.
- Few data for interpolation within the neighborhood
- Increased risk of wrong interpolations

Jožef Stefan Institute

Calculating spatial distances in R

```
library(sp)
data(meuse)
coordinates(meuse) <- ~x+y
proj4string(meuse) <- CRS("+init=epsg:28992")
meuse[1:2,]
```

```
##        coordinates cadmium copper lead zinc  elev       dist   om ffreq
## 1 (181072, 333611)    11.7     85  299 1022 7.909 0.00135803 13.6     1
## 2 (181025, 333558)     8.6     81  277 1141 6.983 0.01222430 14.0     1
##   soil lime landuse dist.m
## 1    1    1      Ah     50
## 2    1    1      Ah     30
```


Calculating spatial distances in R

Euclidean or Great Circle distances

```
distsTo1 <- spDistsN1(meuse, meuse[1,]) # Euclidean
nn3 <- meuse[order(distsTo1)[2:4],]
meuse[1,]
```

```
##        coordinates cadmium copper lead zinc  elev       dist   om ffreq
## 1 (181072, 333611)    11.7     85  299 1022 7.909 0.00135803 13.6     1
##   soil lime landuse dist.m
## 1    1    1      Ah     50
```

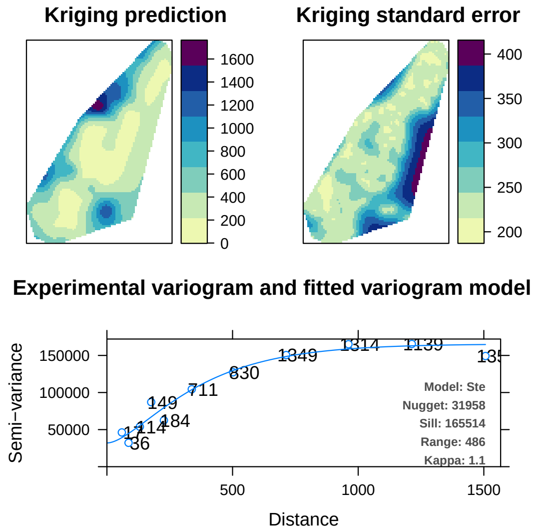
```
nn3
```

```
##        coordinates cadmium copper lead zinc  elev      dist   om ffreq
## 2 (181025, 333558)     8.6     81  277 1141 6.983 0.0122243 14.0     1
## 3 (181165, 333537)     6.5     68  199  640 7.800 0.1030290 13.0     1
## 8 (181027, 333363)     2.8     29  150  406 8.490 0.0921516  9.5     1
##   soil lime landuse dist.m
## 2    1    1      Ah     30
## 3    1    1      Ah    150
## 8    1    0      Ab    120
```

```
distsTo1GC <- spDistsN1(meuse, meuse[1,], longlat=TRUE) # Great Circle
```

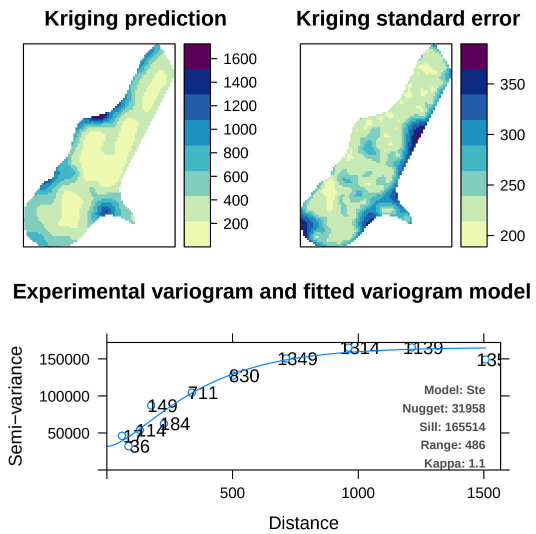
Kriging in R

```
library(automap) # extra package you need to install
kr <- autoKrige(zinc ~ 1, meuse)

## [using ordinary kriging]

plot(kr)
```



Forecasting with Kriging in R

```
data(meuse.grid)
gridded(meuse.grid) <- ~x+y
kr2 <- autoKrige(zinc ~ 1, meuse, meuse.grid)

## [using ordinary kriging]

plot(kr2)
```



Spatial Interpolation using Regression

Ohashi & Torgo: Spatial Interpolation using Multiple Regression. IEEE ICDM'2012

Key Idea

Allow the use of data from faraway regions provided these neighbourhoods have similar spatial dynamics to the target location

Why?

- Being faraway $\neq$ being different
- Using data from other (faraway) region $\rightarrow$ more data

How to achieve this?

- Map the spatial interpolation problem into a regression task
- Propose a series of spatial indicators to better describe the spatial dynamics of a region

Spatial Dynamics of a Variable

Spatial Indicators

Goal: Describe the Spatial Dynamics around a Location

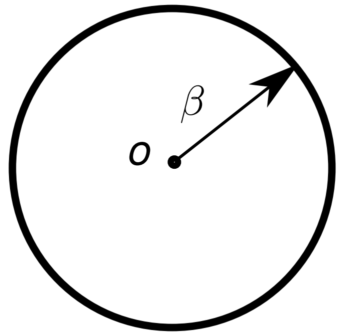
- Inspired by financial technical indicators
  - Try to capture key time-related properties of the time series
    - e.g.: tendency, acceleration, momentum, volatility, etc.
  - Describe what happened to the time series in previous time steps (time neighborhood)
- Try to do the same but for a spatial neighborhood

Jožef Stefan Institute

Spatial Indicators and Spatial Neighborhoods

Spatial Neighborhood

Given a location o, its spatial neighborhood, $\mathcal{N}_o^\beta$, is formed by the set of measurements within a radius β from o



Proposed Spatial Indicators

For a given variable of interest Z, a location o and its spatial neighbourhood $\mathcal{N}_o^\beta$:

- Centrality (averages and weighted averages)

$$\overline{Z}(\mathcal{N}_o^\beta) \qquad \widetilde{Z}(\mathcal{N}_o^\beta)$$

- Variability/Spread

$$\sigma_Z(\mathcal{N}_o^\beta)$$

- **Spatial Tendency**

$$\overline{Z}_o^{\beta_1,\beta_2} = \frac{\overline{Z}(\mathcal{N}_o^{\beta_1})}{\overline{Z}(\mathcal{N}_o^{\beta_2})} \qquad \widetilde{Z}_o^{\beta_1,\beta_2} = \frac{\widetilde{Z}(\mathcal{N}_o^{\beta_1})}{\widetilde{Z}(\mathcal{N}_o^{\beta_2})}$$

Spatial Interpolation as a Multiple Regression Task

- Target: the variable of interest value at location o
- Predictors: a description of the spatial dynamics of the variable in the neighbourhood of o
- An illustrative formalization of the prediction task:

$$Z_o = f(\overline{Z}(\mathcal{N}_o^{k_1}), \overline{Z}(\mathcal{N}_o^{k_2}), \overline{Z}(\mathcal{N}_o^{k_3}), \overline{Z}_o^{k_1,k_2}, \overline{Z}_o^{k_2,k_3}, \\ \widetilde{Z}(\mathcal{N}_o^{k_1}), \widetilde{Z}(\mathcal{N}_o^{k_2}), \widetilde{Z}(\mathcal{N}_o^{k_3}), \widetilde{Z}_o^{k_1,k_2}, \widetilde{Z}_o^{k_2,k_3}, \\ \sigma_Z(\mathcal{N}_o^{k_1}), \sigma_Z(\mathcal{N}_o^{k_2}), \sigma_Z(\mathcal{N}_o^{k_3}))$$

- Procedure: (i) build a data set with the available data; (ii) use a regression method to approximate the unknown function $f()$; (iii) use the obtained model to carry out spatial interpolation

Jožef Stefan Institute

An Illustrative Example

location	$\mathcal{Z}_o$
(44, 39)	390.89
(45, 39)	410.15
(45, 38)	400.07
(55, 42)	780.45
(25, 32)	800.34
⋮	⋮

location	$\mathcal{Z}_o$	$\overline{z}(\mathcal{N}_o^{k_1})$	$\overline{z}(\mathcal{N}_o^{k_2})$	⋯
(44, 39)	390.89	405.11	597.75	⋯
⋮	⋮	⋮	⋮	⋱

Jožef Stefan Institute

Forecasting with Spatial Indicators in R

A simple function to create the spatial indicators

```
getVars <- function(location, data, var, nns=c(3,5,10), funcs=c("mean","var")) {
    dists <- spDistsN1(data,location)
    o <- order(dists)
    res <- lapply(nns, function(nn) {
        ns <- data[o[1:nn],]
        vals <- ns[[var]]
        nms <- paste(var,funcs,nn,sep=".")
        vs <- sapply(funcs,function(f) do.call(f,list(vals)))
        names(vs) <- nms
        vs
    })
    unlist(res)
}
getVars(meuse.grid[1,],meuse,"zinc")

##   zinc.mean.3   zinc.var.3  zinc.mean.5   zinc.var.5 zinc.mean.10
##      934.3333   68514.3333     681.2000  155390.7000     580.5000
##   zinc.var.10
##   134412.2778
```


Forecasting with Spatial Indicators in R - 2

Getting a training set

```
set.seed(1234)
traindat <- NULL
for(r in 1:nrow(meuse)) traindat <- rbind(traindat,getVars(meuse[r,],meuse[-r,],"zinc"))
traindat <- data.frame(traindat,tgtZinc=meuse[["zinc"]])
head(traindat)

##   zinc.mean.3 zinc.var.3 zinc.mean.5 zinc.var.5 zinc.mean.10 zinc.var.10
## 1    729.0000 140997.000       558.0   126315.5        528.5   110533.61
## 2    689.3333  96689.333       702.0   118958.0        516.6    95752.49
## 3    806.6667 230140.333       634.4   171162.3        566.7   134591.12
## 4    418.3333  38334.333       511.6   104112.3        491.4   112093.16
## 5    294.6667   2120.333       269.2     3428.2        325.4    17116.27
## 6    236.3333   2169.333       248.8     4457.2        309.6    19322.27
##   tgtZinc
## 1    1022
## 2    1141
## 3     640
## 4     257
## 5     269
## 6     281
```

Forecasting with Spatial Indicators in R - 3

Now a test set

```
set.seed(1234)
szTest <- 100
testPoints <- meuse.grid[sample(1:nrow(meuse.grid),szTest),]
dat <- NULL
for(r in 1:szTest) dat <- rbind(dat,getVars(testPoints[r,],meuse,"zinc"))
head(dat)
```

```
##      zinc.mean.3  zinc.var.3 zinc.mean.5 zinc.var.5 zinc.mean.10
## [1,]    820.0000  41329.0000       667.2    65089.7        636.9
## [2,]    203.3333    421.3333       293.2    46105.2        473.4
## [3,]    203.3333    421.3333       397.8    71209.2        473.4
## [4,]    177.0000   2923.0000       181.8     1545.2        416.2
## [5,]    850.3333 586158.3333       581.0   429334.5        437.1
## [6,]    407.0000 103573.0000       312.2    69937.7        246.4
##      zinc.var.10
## [1,]    69468.54
## [2,]   109516.04
## [3,]   109516.04
## [4,]   101609.73
## [5,]   246621.43
## [6,]    36532.93
```


Forecasting with Spatial Indicators in R - 4

Trying an SVM on the data

```
library(e1071)
s <- svm(tgtZinc ~ ., traindat, cost=10, epsilon=0.1)
ps <- predict(s,dat)
head(ps)
```

```
##         1         2         3         4         5         6
## 1004.0961  200.2492  147.7198  267.7419  481.0211  258.1782
```

Hands On Spatial Forecasting

Using the `meuse` data set from package **sp**:

1. Build a multiple regression data set to predict the variable *cadmium* through the function `getVars()` shown during the classes. Explore other statistics apart from the defaults of the function.
2. Split the obtained data set randomly in two 70-30% partitions
3. Obtain an SVM with the larger partition
4. Apply the model to obtain predictions for the smaller partition and analyse the results

Jožef Stefan Institute