

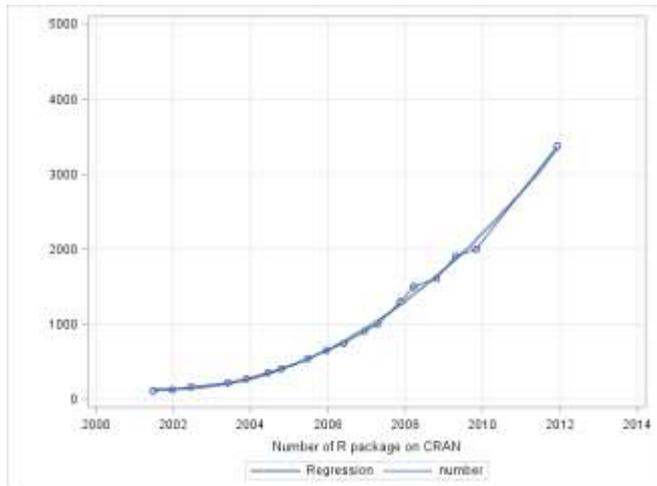
# R in the Sensor Web

Sensing a Changing World 2012

Daniel Nüst, 52°North Initiative for Geospatial Open Source Software GmbH  
d.nuest@52north.org

Edzer Pebesma, Institute for Geoinformatics, University of Muenster  
edzer.pebesma@wwu.de





R-sig-geo

# Sensor Web?

pachube

Feeds Apps Support Blog



What is it?

Use and build applications which update and respond in real-time

- Send your real-time
- Leverage
- Get applic
- Use and b
- communit

How it works

Hardware



# SAFECAST

ABOUT MAPS FAQ SUBMIT DOWNLOAD DATA

ENGLISH 日本語

Safecast is a global sensor network that empowers people with data



Disbursement

We publish our re

2 Responses

Measurements

OGC® SWE  
*LIGHT*

SOS

SOS-T

SIR/SOR

SES

WNS

# Sensor Web?

*Sensors* 2011, 11, 2652-2699; doi:10.3390/s110302652

OPEN ACCESS

**sensors**

ISSN 1424-8220

www.mdpi.com/journal/sensors

*Review*

## **New Generation Sensor Web Enablement**

**Arne Bröring**<sup>1,2,3,\*</sup>, **Johannes Echterhoff**<sup>4</sup>, **Simon Jirka**<sup>3</sup>, **Ingo Simonis**<sup>4</sup>, **Thomas Everding**<sup>1</sup>,  
**Christoph Stasch**<sup>1</sup>, **Steve Liang**<sup>5</sup> and **Rob Lemmens**<sup>2</sup>

<sup>1</sup> Institute for Geoinformatics, University of Muenster, Weseler Strasse 253, 48151 Muenster, Germany; E-Mails: thomas.everding@wwu.de (T.E.); christoph.stasch@wwu.de (C.S.)

<sup>2</sup> Faculty ITC, University of Twente, Hengelosestraat 99, 7514 AE Enschede, The Netherlands; E-Mail: lemmens@itc.nl (R.L.)

<sup>3</sup> 52° North Initiative for Geospatial Open Source Software, Martin-Luther-King-Weg 24, 48155 Muenster, Germany; E-Mail: jirka@52north.org (S.J.)

<sup>4</sup> International Geospatial Services Institute, Werner-Heisenberg-Str. 73, 26723 Emden, Germany; E-Mails: johannes.echterhoff@igsi.eu (J.E.); ingo.simonis@igsi.eu (I.S.)

<sup>5</sup> Geomatics Engineering, University of Calgary, Calgary, AB, T2N 1N4, Canada;



sos4R

spacetime

WPS4R

spacetime

WPS4R

sos4R

spacetime

sos4R

WPS4R

The logo consists of the text 'SOS4R'. The 'SOS' part is rendered in a bold, rounded, orange font. The '4' is a smaller, orange number with a plus sign on its top right. The 'R' is a large, bold, grey letter with a 3D metallic effect, including highlights and shadows.

<http://www.nordholmen.net/sos4r/>

# standard



client  
server

# implementation

# Open Geospatial Consortium Inc.

Date: 2007-10-26

Reference number of this document: **OGC 06-009r6**

Version: 1.0

Category: OpenGIS® Implementation Standard

Editors: Arthur Na (IRIS Corp.), Mark Priest (3eTI)



## Sensor Observation Service

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.  
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>

**OpenGIS® Filter Encoding Implementation Specification**

Open Geospatial Consortium Inc.

Date: 2007-10-26

**OGC® SWE Common Data Model  
Encoding Standard**

OGC 06-009r6

Version: 1.0

**Observations and Measurements –  
Part 2 - Sampling Features**

Category: OpenGIS® Implementation Standard

Editors: Arthur Na (IRIS Corp.), Mark Priest (3eTI)

**Observations and Measurements –  
Part 1 - Observation schema**

**OGC Web Services Common Specification**

Sensor Observation Service

**OpenGIS® Geography Markup Language (GML) Encoding  
Standard**

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.  
To obtain additional rights, please contact the Open Geospatial Consortium, Inc.

**OpenGIS® Sensor Model Language (SensorML)  
Implementation Specification**

## Sensor Data Analysis Workflow

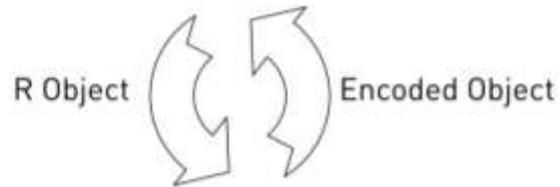
Build Request



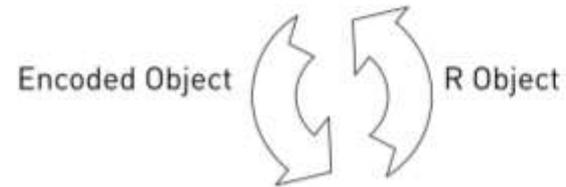
Send and Receive Request



Data Analysis



**Encoding Functions**



**Parsing Functions**

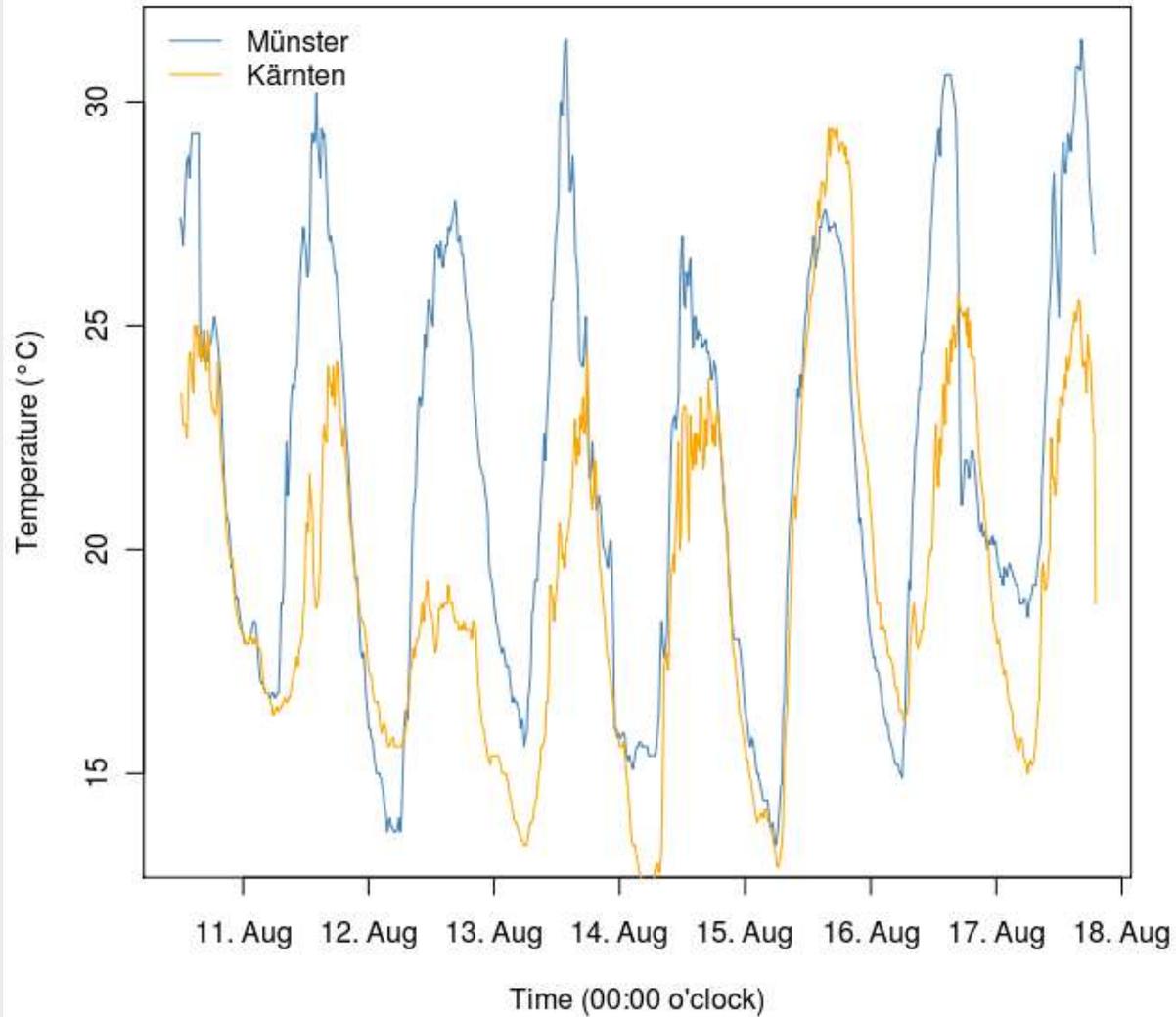


**Conversion Functions**

## Mapped Classes (excerpt)

R Classes	XML Schema Elements	
<b>SOS</b>		
SOS		
SOS_1.0.0		
SosCapabilities_1.0.0	← <i>sos:GetCapabilities</i>	} Response Parsing
SosFilter_Capabilities	<i>sos:Filter_Capabilities</i>	
SosContents	<i>sos:Contents</i> ←	} Request Encoding
SosObservationOffering	<i>sos:ObservationOffering</i>	
SosDescribeSensor	<i>sos:DescribeSensor</i>	
SosGetObservation	<i>sos:GetObservation</i>	
<b>O&amp;M</b>		
OmObservationCollection	<i>om:ObservationCollection</i>	
OmObservation	<i>om:Observation</i>	
OmObservationProperty	← <i>om:ObservationProperty</i>	} Response Parsing
OmMeasurement	<i>om:Measurement</i>	
SaSamplingPoint	<i>sa:SamplingPoint</i>	
<b>OWS Common</b>		
OwsGetCapabilities,	<i>ows:GetCapabilities</i> -----	} Request Encoding
OwsGetCapabilities_1.1.0		
OwsExceptionReport	← <i>ows:ExceptionReport</i>	} Response Parsing
OwsCapabilities	<i>ows:Capabilities</i>	
OwsServiceIdentification	<i>ows:ServiceIdentification</i>	
OwsContents	<i>ows:Contents</i> -----	

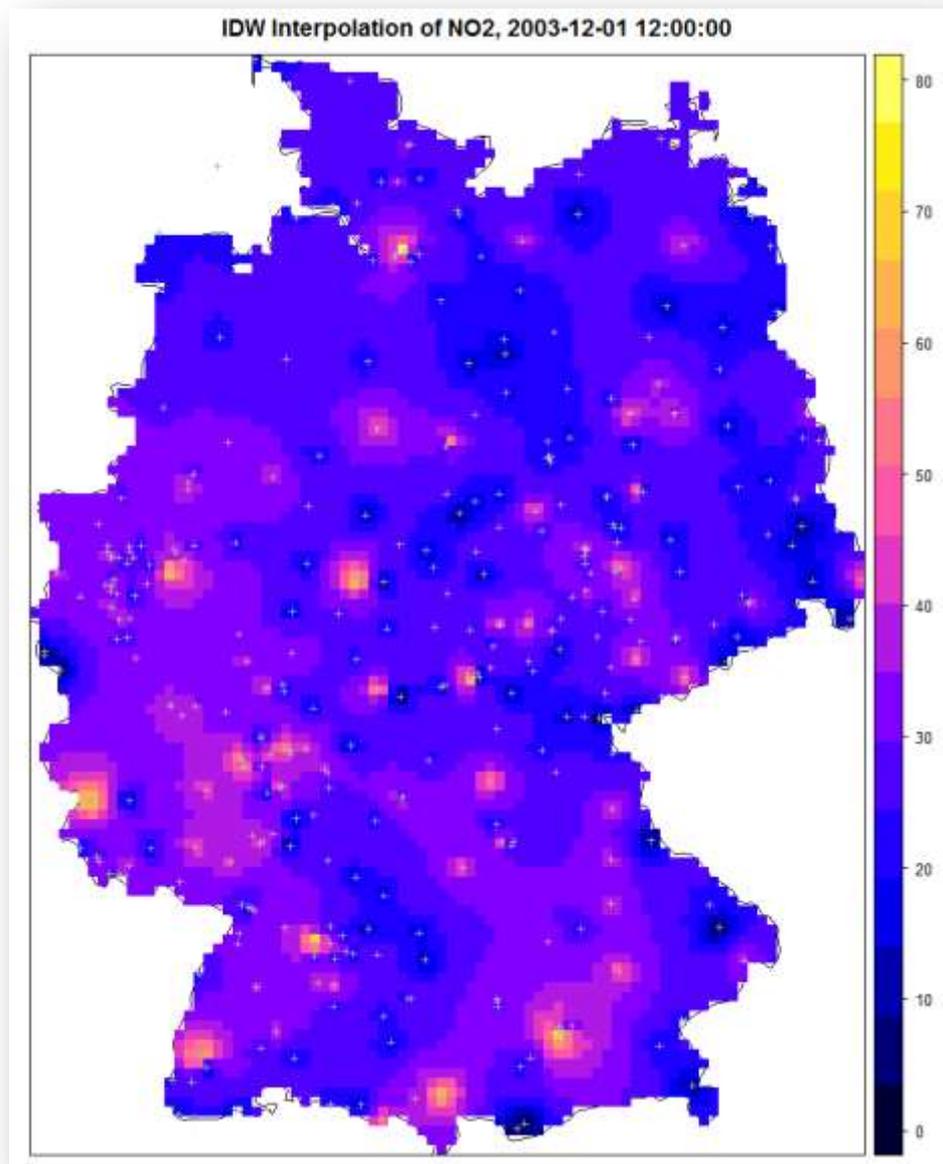
## Temperature in Münster and Kärnten, 2009



```

6 # establish a connection to a SOS instance with default settings
7 weathersos <- SOS(url = "http://v-swe.uni-muenster.de:8080/WeatherSOS/sos")
8
9 #####
10 # Request two procedures
11 obs <- getObservation(sos = weathersos, offering = off,
12                     procedure = sosProcedures(off),
13                     eventTime = sosCreateEventTimeList(sosCreateTimePeriod(sos = weathersos,
14                                     begin = as.POSIXct("2009-08-10 12:00"),
15                                     end = as.POSIXct("2009-08-20 12:00"))))
16 obs
17
18 # Create plot
19 # Attention: plots ignore the fact that the times do NOT perfectly match!
20 x <- 800
21 plot(x = obs[[1]]@result[[1]][1:x], y = obs[[1]]@result[[3]][1:x], type = "l",
22      col = "steelblue", main = "Temperature in Muenster and Kaernten, 2009",
23      xlab = "Time (00:00 o'clock)",
24      ylab = "Temperature (°C)",
25      xaxt="n") # do not plot x-axis
26 r <- as.POSIXct(round(range(obs[[1]]@result[[1]]), "days"))
27 axis.POSIXct(side = 1, x = obs[[1]]@result[[1]][1:x], format = "%d. %h",
28             at = seq(r[1], r[2], by="day"))
29 lines(x = obs[[2]]@result[[1]][1:x], y = obs[[2]]@result[[3]][1:x],
30      col = "orange")
31 legend("topleft", legend = c("Muenster", "Kaernten"),
32      col = c("steelblue", "orange"), lty = 1, bty="n")
33

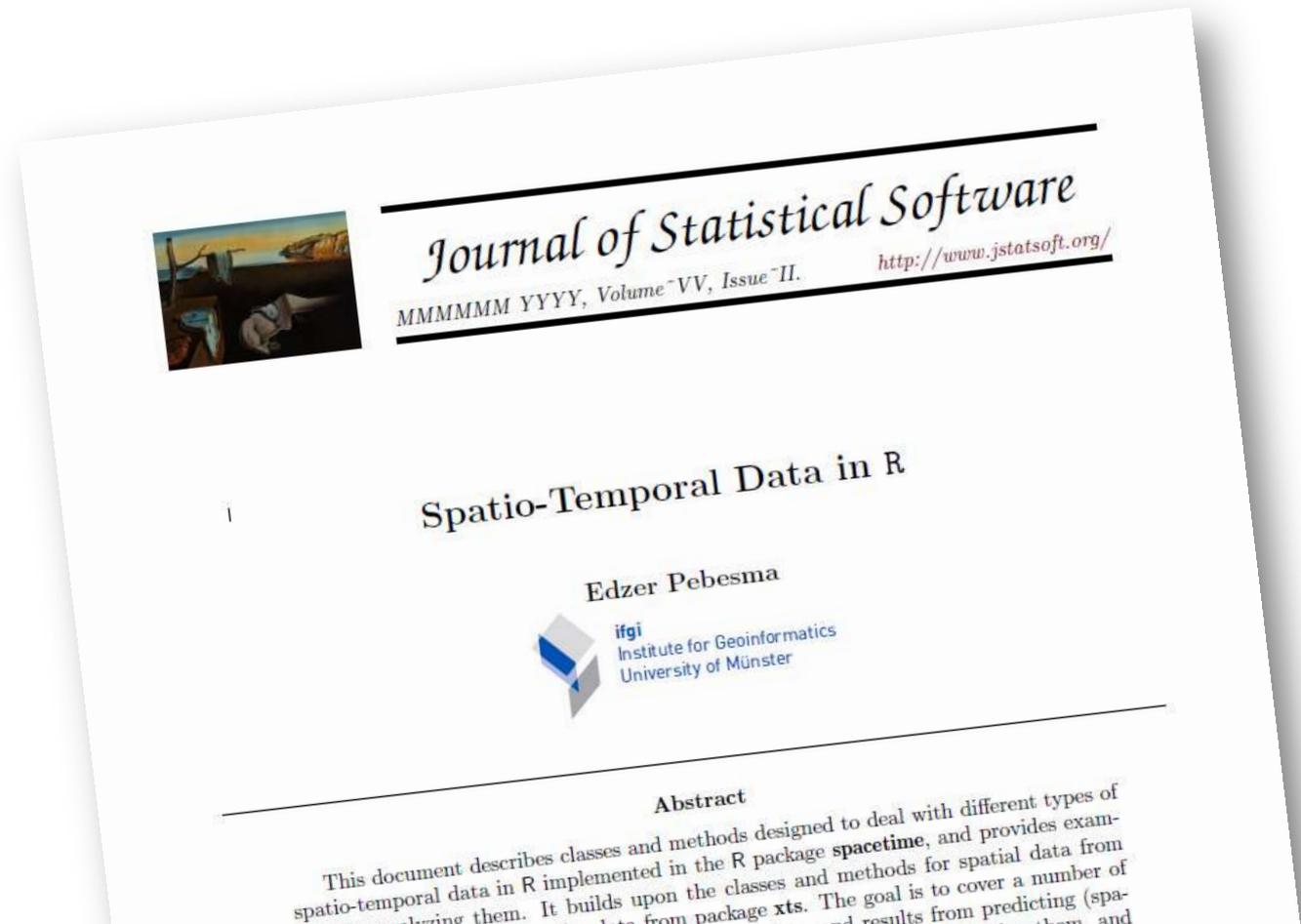
```



demo("airquality")

# spacetime

- <http://cran.r-project.org/package=spacetime>

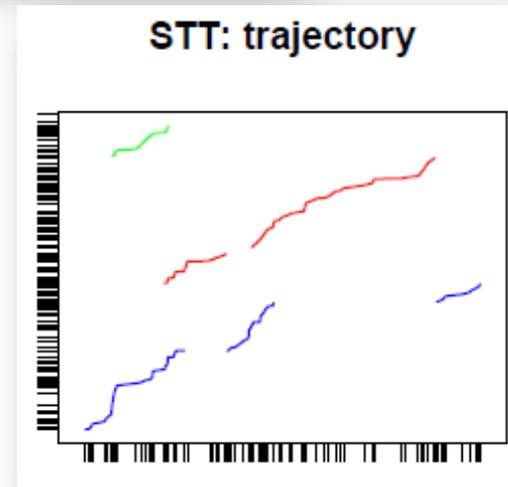
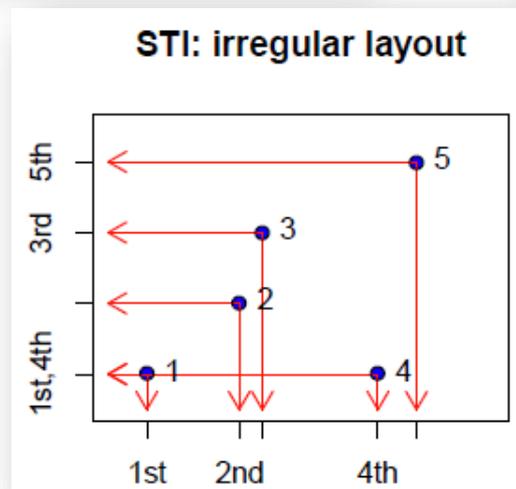
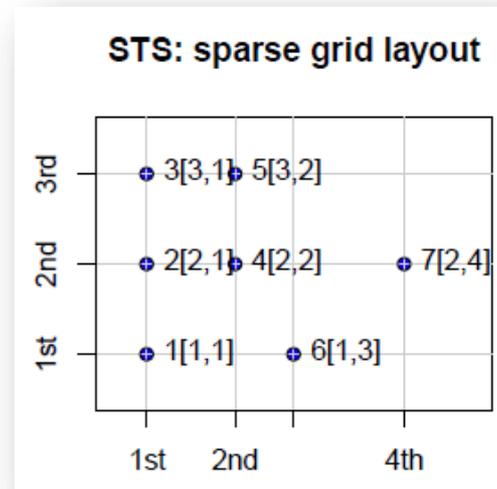
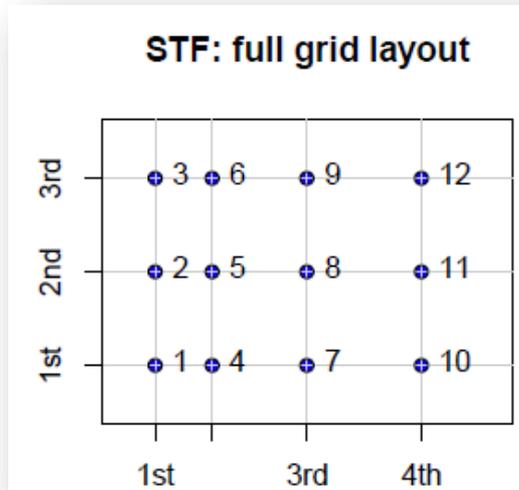


# spatio-temporal data in tables

- **time-wide** where different columns reflect different moments in time,
- **space-wide** where different columns reflect different measurement locations or areas, or
- **long formats** where each record reflects a single time and space combination.

# space-time layouts

Spatial features

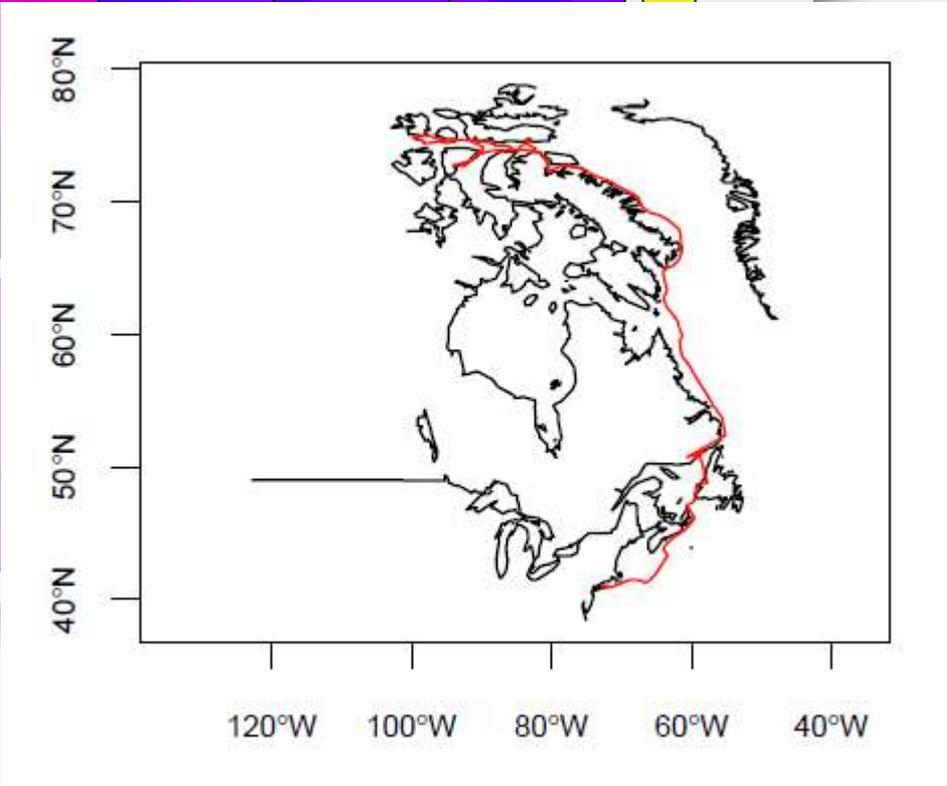
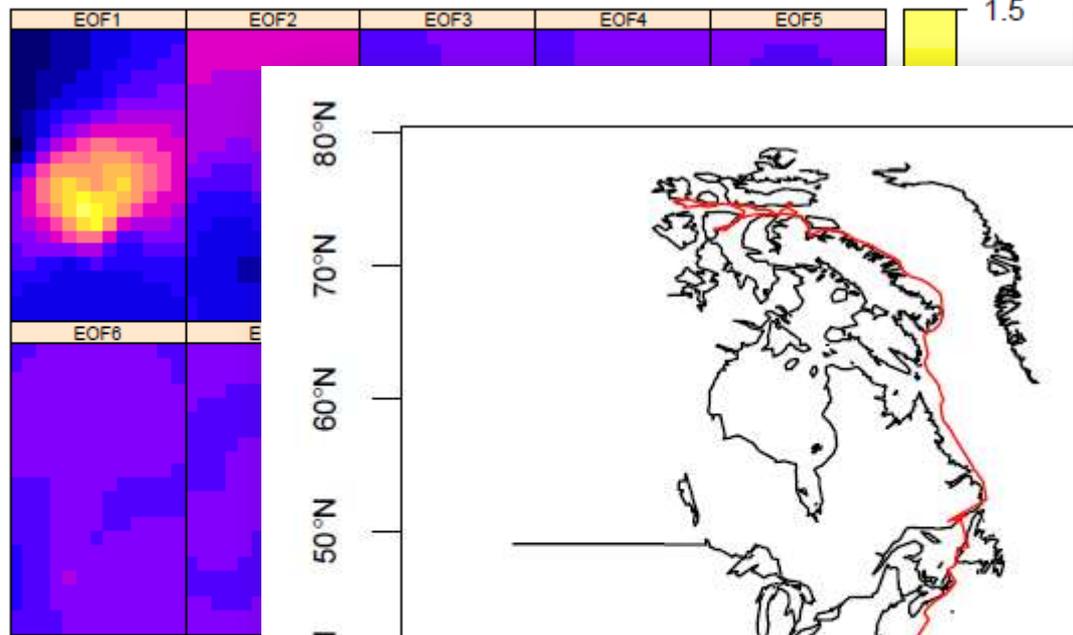
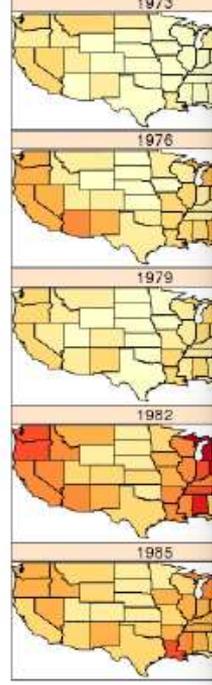
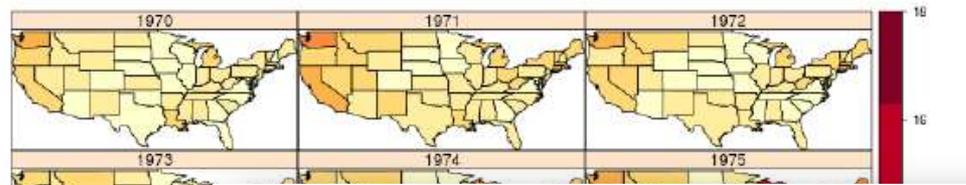
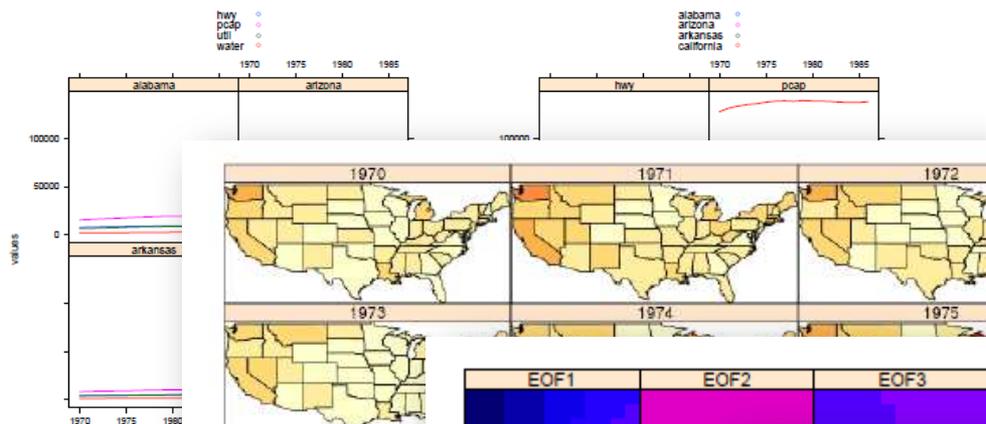


Time points

```
R> air_quality[Germany, "2008::2009", "PM10"]
```

```
R> aggregate(x, by, FUN, ..., simplify = TRUE)
```

```
R> over(x, y)
```



# WPS4R

<http://52north.org/wps4r>

Expose arbitrary R scripts via WPS

Variety of input and output formats

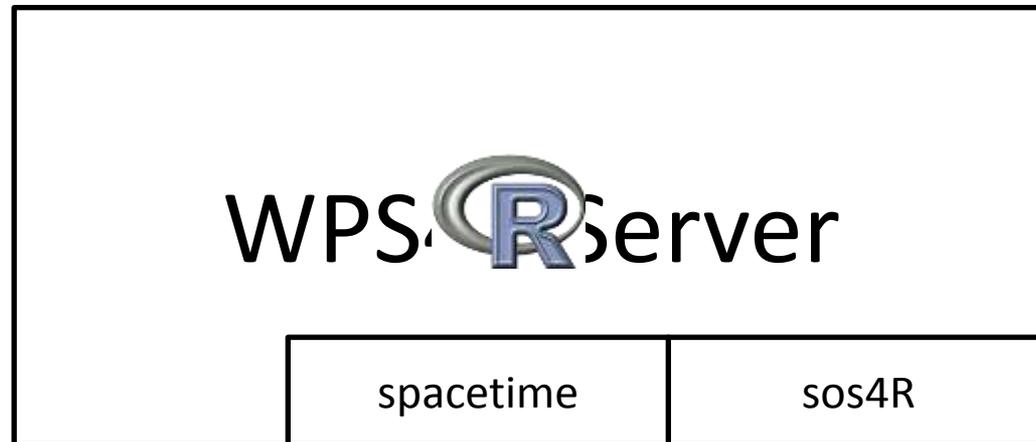
- Under development!
- Vector, Raster, ...

Current Use Cases: Reproducibility,  
Clients/Usability, Publishing

# Demo

Client/UI

KML, Images,  
Rasters, Tables,  
Shapefiles



```
--
14 # wps.in: sos_url, string, title = SOS service URL,
15 # abstract = SOS URL endpoint,
16 # value = http://v-swe.uni-muenster.de:8080/WeatherSOS/sos,
17 # minOccurs = 0, maxOccurs = 1;
18 #sos_url <- "http://v-swe.uni-muenster.de:8080/WeatherSOS/sos"
19
20 # wps.in: offering_id, type = string, title = identifier for the
21 # value = ATMOSPHERIC_TEMPERATURE;
22 #offering_id <- "ATMOSPHERIC_TEMPERATURE"
23
```

```
22 #offering_id <- "ATMOSPHERIC_TEMPERATURE"
23
24 # wps.in: offering_days, integer, temporal extent,
25 # the number of days the plot spans to the past,
26 # value = 7,
27 # minOccurs = 0, maxOccurs = 1;
28 #offering_days <- 7
29
30 # wps.in: offering_station, type = integer, title = identifier for the used offering,
31 # value = 1,
32 # minOccurs = 0, maxOccurs = 1;
33 #offering_station <- 1
34
35 # wps.in: image_width, type = integer, title = width of the generated image in pixels,
36 # value = 800, minOccurs = 0, maxOccurs = 1;
37 # wps.in: image_height, type = integer, title = height of the generated image in pixels,
38 # value = 500, minOccurs = 0, maxOccurs = 1;
39 #image_width = 800;
40 #image_height = 500;
41
42 #####
43 # SOS and time series analysis
44
45 # establish a connection to a SOS instance with default settings
46 sos <- SOS(url = sos_url)
47
```

# 52north Web Admin Console

WPS Config Configuration

WPS Test Client

Save and Activate Configuration

Load Active Configuration

Upload Configuration File

Reset

Upload Process

Update Remote Repositories

Upload R Script

Server Settings

Algorithm Repositories

Name LocalAlgorithmRepository

Class org.n52.wps.server.t.LocalAlgori

Active

Please enter the process name:

(only if process name should be unlike filename)

# 52north Web Admin Console

WPS Config Configuration

WPS Test Client

Save and Activate Configuration

Load Active Configuration

Upload Configuration File

Reset

Upload Process

Update Remote Repositories

Upload R Script

Server Settings

Algorithm Repositories

✖

Name LocalAlgorithmRepository

Class org.n52.wps.server.t.LocalAlgorithmRep

Active

Properties

Name

Value

Algorithm

org.n52.wps.server.t.intersection

✔ ✖ 🗑

Algorithm

org.n52.wps.server.t.RandomNumbers

✔ ✖ 🗑

Algorithm

org.n52.wps.server.t.Sum

✔ ✖ 🗑

Rserve\_Host

localhost

✔ ✖ 🗑

Rserve\_Port

6311

✔ ✖ 🗑

Parsers

Generators

Remote Repositories

```

- <ns:ProcessDescriptions xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd"
  version="1.0.0">
- <ProcessDescription statusSupported="true" storeSupported="true" ns:processVersion="1.0.0">
  <ns:Identifier>org.n52.wps.server.r.SosPlot</ns:Identifier>
  <ns:Title>Plot SOS Time Series</ns:Title>
- <ns:Abstract>
  Accesses a SOS with sos4R and create
  <ns:Abstract>
  <ns:Metadata xlin:title="R Script" about="The R script which is used for this process" xlin:href="http://localhost:8080/wps/scripts/SosPlot.R"/>
  <ns:Metadata xlin:title="R Session Info" about="R Console output of sessionInfo() method in R, content is generated dynamically for the current state"
    /sessioninfo.jsp"/>
- <DataInputs>
  - <Input minOccurs="0" maxOccurs="1">
  <Input minOccurs="0" maxOccurs="1">
    <ns:Identifier>sos_url</ns:Identifier>
    <ns:Title>null</ns:Title>
    <ns:Abstract>SOS URL endpoint</ns:Abstract>
  - <LiteralData>
    <ns:DataType ns:reference="xs:string"/>
    <ns:AnyValue/>
    <DefaultValue>http://v-swe.uni-muenster.de:8080/WeatherSOS/sos</DefaultValue>
  </LiteralData>
  </Input>
  </LiteralData>
  </Input>
- <Input minOccurs="0" maxOccurs="1">
  <ns:Identifier>offering_days</ns:Identifier>
  <ns:Title>temporal extent</ns:Title>
  <ns:Abstract>the number of days the plot spans to the past</ns:Abstract>
- <LiteralData>
  <ns:DataType ns:reference="xs:integer"/>

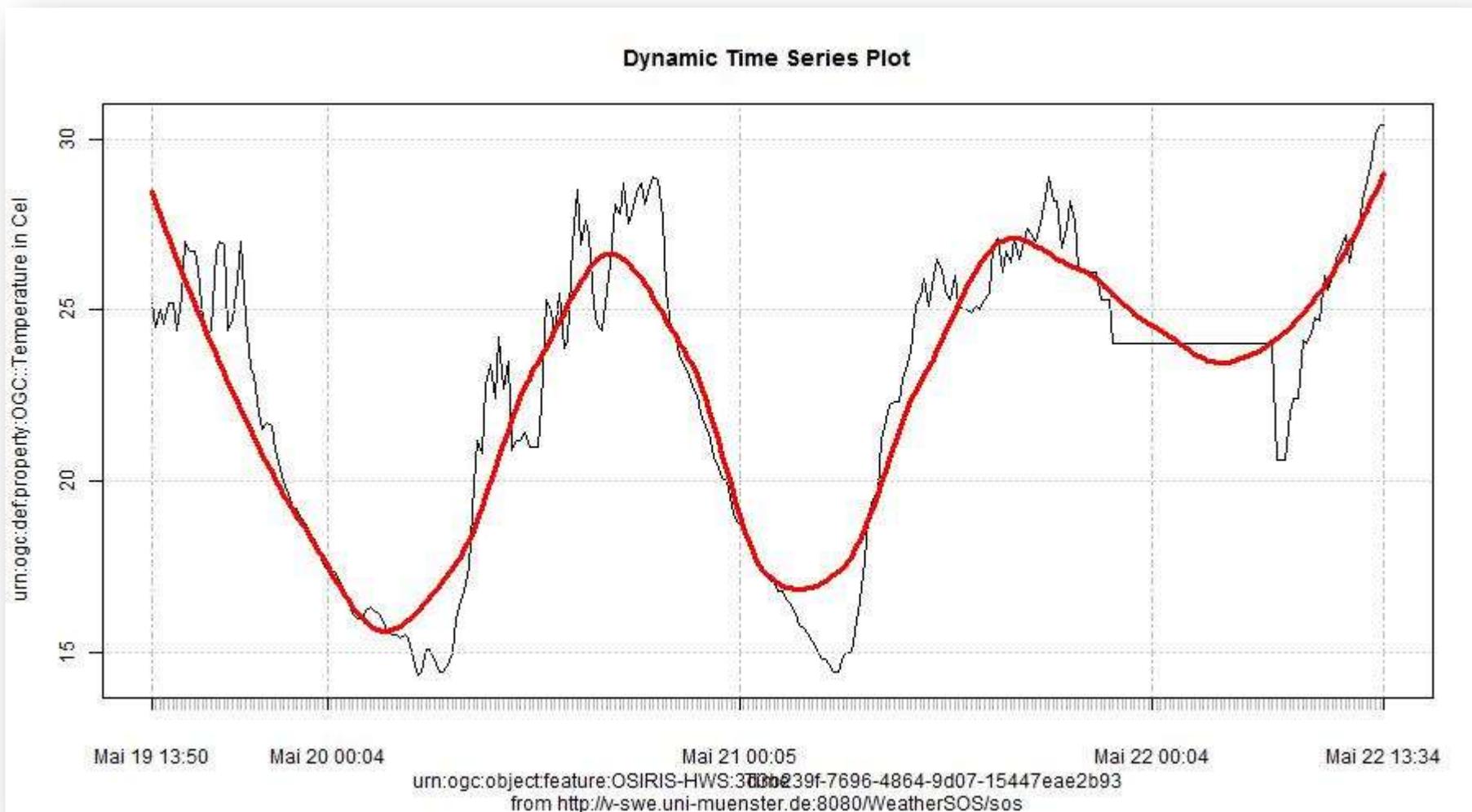
```

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute service="WPS" version="1.0.0"
  xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>org.n52.wps.server.r.SosPlot</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>offering_days</ows:Identifier>
      <ows:Title></ows:Title>
      <wps>Data>
        <wps:LiteralData>3</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <!-- <wps:Input -->
    <!-- <ows:Identifier>offering_id</ows:Identifier -->
    <!-- <ows:Title></ows:Title -->
    <!-- <wps>Data -->
    <!-- <wps:LiteralData>ATMOSPHERIC_TEMPERATURE</wps:LiteralData -->
    <!-- </wps>Data -->
    <!-- </wps:Input -->
    <wps:Input>
      <ows:Identifier>image_width</ows:Identifier>
      <ows:Title></ows:Title>
      <wps>Data>
        <wps:LiteralData>500</wps:LiteralData>
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>image_height</ows:Identifier>
      <ows:Title></ows:Title>
      <wps>Data>
        <wps:LiteralData>500</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps>DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="image/jpeg">
      <ows:Identifier>output_image</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

# Send Request

- <http://localhost:8080/52n-wps-webapp/test.html>
- Select R\_sosPlot.xml
- DescribeProcess:  
<http://localhost:8080/52n-wps-webapp/WebProcessingService?Request=DescribeProcess&Service=WPS&identifier=org.n52.wps.server.r.SosPlot>

# Generated Plot



# Goal

Make Sensor Web work, make it accessible

Scalable and sustainable information systems

**... across domains**

# Outlook

## O&M

- GDAL and rgdal?
- om package?



to WPS4R

Promoting/Advancing **spacetime**

**WPS4R Use Cases**



# Talk to me...

Quality (Modelling, Visualisation)

Open Data

# Questions?

Do you use R and the Sensor Web?





- INTeroperability and Automated MAPping
- “backend” R package
  - <http://www.intamap.org/intamap-package.php>
- WPS Server
- O&M Input
- Clients
  - JAVA API
  - Stand-alone: <http://www.intamap.org/tryGenericClient.php>