sos4R: New Wrapper Functions for Easier SOS Access

Geospatial Sensing Conference 2019
2019-09-02

Eike H. Jürrens,
Benedikt Gräler,
Daniel Nüst

52°North GmbH
https://52north.org

Overview

Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

R is a statistical tool and programming language tailored for data analysis including spatial data.

It allows to **query data** from standard conform **SOS instances** using simple **R function calls** and does no require any knowledge about the Sensor Web. It is easily extendible for new data models and **opens** the huge amount of analysis and visualization **features of the R environment for the Sensor Web**.
https://52north.github.io/sos4R/

`sos4R` includes a collection of convenience functions which wrap the complex SOS interface with its specific terms (e.g. FOI, procedure).

The wrapper function use more generic terms easily accessible for all users, especially without a strong knowledge of the OGC standards of the Sensor Web Enablement (see "OGC SWE and SOS" vignette for details).

In a nutshell , sos4R...

- has had 589 commits made by 9 contributors representing 12,680 lines of code
- is mostly written in R with a well-commented source code
- has a well established, mature codebase maintained by a average size development team with increasing Y-O-Y commits
- took an estimated 13 years of effort (COCOMO model) starting with its first commit in May, 2013 ending with its most recent commit about three weeks ago
- CRAN: https://cran.r-project.org/package=sos4R

Sponsors

- NIWA: National Institute of Water and Atmospheric Research, New Zealand
- MuDak-WRM: Multidisciplinary data acquisition as the key for a globally applicable water resource management, funded by BMBF
- TaMIS: Talsperren-Mess-Informations-System, funded by BMBF
- ifgi: student contributions, initial version by Daniel Nüst

Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

52north

The key functions of the convenience API are:

- **SOS()** initializes the connection with an SOS endpoint
- **phenomena()** returns a list of phenomena that the SOS provides. It can also provide details on the recording period and the stations providing this phenomenon
- **sites()** retrieves details on the feature of interests/measurement stations. This can include their locations, observed phenomena and temporal extent of each time series.
- **getData()** fetches the desired time series data from the SOS.

The convenience API wraps and combines the default API reusing the available functions in sos4R.

Phenomena

```
library("sos4R")

niwaHydro <- SOS(url = "https://climate-sos.niwa.co.nz/",
                 binding = "KVP",
                 useDCPs = FALSE,
                 version = "2.0.0")
```


Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

52north

The function phenomena(..) provides information about observed phenomena and time periods of data.

```
phenomena <- phenomena(sos = niwaHydro)
str(phenomena)
```

```
'data.frame':   84 obs. of  1 variable:
 $ phenomenon: chr  "MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))" "MTH
```

```
head(phenomena)
```

```
                                                                                phenomenon
1 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
2                       MTHLY_STATS: DAYS OF OCCURRENCE (FOG) (MTHLY: FOG DAYS)
3                     MTHLY_STATS: DAYS OF OCCURRENCE (GALE) (MTHLY: GALE DAYS)
4      MTHLY_STATS: DAYS OF OCCURRENCE (GROUND FROST) (MTHLY: GROUND FROST DAYS)
5    MTHLY_STATS: DAYS OF OCCURRENCE (GUSTS over 23 knots) (MTHLY: GUST DAYS 24)
6    MTHLY_STATS: DAYS OF OCCURRENCE (GUSTS over 32 knots) (MTHLY: GUST DAYS 33)
```

The retrieved data can be extended by time intervals and site identifier for which data is available.

```
head(phenomena(sos = niwaHydro, includeTemporalBBox = TRUE))
```

```
                                                                  phenomenon
1                                MTHLY_STATS: TOTAL RAINFALL (MTHLY: TOTAL RAIN)
2            MTHLY_STATS: WET DAYS with rainfall 1 mm or more (MTHLY: WET DAYS)
3      MTHLY_STATS: MEAN AIR TEMPERATURE;   0.5* (MAX + MIN) (MTHLY: MEAN TEMP)
4  MTHLY_STATS: MEAN MAXIMUM TEMPERATURE from daily Maxs (MTHLY: MEAN MAX TEMP)
5  MTHLY_STATS: MEAN MINIMUM TEMPERATURE from daily Mins (MTHLY: MEAN MIN TEMP)
6 MTHLY_STATS: MEAN DAILY GRASS-MIN from dly Grass-mins (MTHLY: MEAN GRASS-MIN)
   timeBegin    timeEnd
1 1960-08-01 2019-08-01
2 1960-08-01 2019-08-01
3 1960-08-01 2019-08-01
4 1960-08-01 2019-08-01
5 1960-08-01 2019-08-01
6 1960-08-01 2019-08-01
```


Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

```
head(phenomena(sos = niwaHydro, includeSiteId = TRUE))
```

```
                                                                                phenomenon
61  MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
126 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
152 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
251 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
293 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
342 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
    siteID
61   17244
126  26958
152  25506
251  22719
293  37850
342  38224
```


Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

52north

One can also add both temporal extent and sites.

```
head(phenomena(sos = niwaHydro, includeTemporalBBox = TRUE, includeSiteId = TRUE))
```

```
                                                                                  phenomenon
61  MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
126 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
152 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
251 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
293 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
342 MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))
    siteID  timeBegin    timeEnd
61   17244 1999-08-01 2019-08-01
126  26958 2007-08-01 2019-08-01
152  25506 2004-12-01 2018-06-01
251  22719 2002-07-01 2018-11-01
293  37850 2011-11-01 2019-08-01
342  38224 2010-09-01 2019-08-01
```


Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

52north

Sites

The function `sites(..)` provides information about sites where observations are made, including metadata about the sites (e.g. location). The returned object is a `SpatialPointsDataFrame`.

```
sites <- sites(sos = niwaHydro)
head(sites)
```

```
            coordinates siteID
1    (173.926, -35.183)   1056
2  (172.851, -42.53433)  11234
3 (172.9716, -41.09798)  12429
4 (173.9628, -41.49891)  12430
5 (169.3148, -45.20724)  12431
6 (174.9844, -40.90392)  12442
```


Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

One can retrieve additional metadata about the phenomena and the time period for which data is available. Including temporal extent implies inclusion of phenomena. In the next chunks the object is coerced to a `date.frame` to get a tabular view.

```
sitesPhen <- sites(sos = niwaHydro, includePhenomena = TRUE)
head(colnames(sitesPhen@data))
```

```
[1] "siteID"
[2] "MTHLY_STATS: DAYS OF DEFICIT (WBal AWC=150mm) (MTHLY: DAYS OF DEFICIT (WBAL))"
[3] "MTHLY_STATS: DAYS OF OCCURRENCE (FOG) (MTHLY: FOG DAYS)"
[4] "MTHLY_STATS: DAYS OF OCCURRENCE (GALE) (MTHLY: GALE DAYS)"
[5] "MTHLY_STATS: DAYS OF OCCURRENCE (GROUND FROST) (MTHLY: GROUND FROST DAYS)"
[6] "MTHLY_STATS: DAYS OF OCCURRENCE (GUSTS over 23 knots) (MTHLY: GUST DAYS 24)"
```

```
colnames(sitesPhen@data)[c(1,3,80)] <- c("ID", "FOG", "RAIN")
```

List sites III

```
sitesPhen[,c(1,3,80)]
```

```
             coordinates    ID   FOG RAIN
1     (173.926, -35.183)  1056  TRUE TRUE
2   (172.851, -42.53433) 11234 FALSE TRUE
3  (172.9716, -41.09798) 12429 FALSE TRUE
4  (173.9628, -41.49891) 12430 FALSE TRUE
5  (169.3148, -45.20724) 12431 FALSE TRUE
6  (174.9844, -40.90392) 12442 FALSE TRUE
7  (168.3305, -46.41727) 12444 FALSE TRUE
8     (167.275, -45.525) 12482 FALSE TRUE
9  (166.7612, -77.84935) 12740 FALSE TRUE
10 (170.5147, -45.90129) 15752 FALSE TRUE
11    (172.324, -41.805) 16826 FALSE TRUE
12  (175.735, -37.87683) 17030 FALSE TRUE
13 (173.2629, -35.13352) 17067 FALSE TRUE
14 (172.6111, -43.32858) 17244 FALSE TRUE
15     (171.672, -43.47) 17610 FALSE TRUE
16 (-176.475, -43.81687) 17840 FALSE TRUE
17    (170.096, -43.736) 18125 FALSE TRUE
18    (174.867, -41.407) 18234 FALSE TRUE
19 (170.1356, -45.51814) 18437 FALSE TRUE
```

List sites IV

```
20  (175.545, -39.19589) 18464 FALSE TRUE
21 (174.0727, -41.64724) 18468 FALSE TRUE
22 (170.1004, -45.12427) 18593 FALSE TRUE
23 (177.5294, -38.28566)  1905 FALSE TRUE
24 (174.8638, -37.20637)  2006 FALSE TRUE
25 (173.0948, -41.31727) 21937 FALSE TRUE
26 (175.3898, -41.25231) 21938 FALSE TRUE
27 (175.6092, -40.38195) 21963 FALSE TRUE
28 (174.7764, -36.96177) 22719 FALSE TRUE
29  (174.305, -39.33546) 23872 FALSE TRUE
30 (171.1916, -42.46022) 23934 FALSE TRUE
31 (172.6077, -43.53074) 24120 FALSE TRUE
32 (170.1343, -43.36548) 24926 FALSE TRUE
33 (177.9218, -38.62747) 24976 FALSE TRUE
34 (173.8532, -35.93145) 25119 FALSE TRUE
35  (169.267, -43.71882) 25506 FALSE TRUE
36 (174.7202, -38.62033) 25726 FALSE TRUE
37 (172.1557, -41.27058) 25777 FALSE TRUE
38 (171.5628, -42.94152) 25821 FALSE TRUE
39 (175.3052, -37.77389) 26117 FALSE TRUE
40 (169.7315, -46.29282) 26163 FALSE TRUE
41 (176.1103, -40.20812) 26958 FALSE TRUE
42 (175.4128, -39.41755) 31621 FALSE TRUE
```

```
43 (173.2241, -42.82722) 31832 FALSE TRUE
44 (175.8575, -40.68048) 31851 FALSE TRUE
45 (171.3108, -44.12476) 35704 FALSE TRUE
46 (169.3921, -45.25366) 36592 FALSE TRUE
47 (172.9657, -43.80938) 36593 FALSE TRUE
48     (175.58, -38.517) 37016 FALSE TRUE
49 (167.6825, -45.29644) 37047 FALSE TRUE
50 (177.1629, -38.33334) 37239 FALSE TRUE
51  (168.018, -44.97781) 37382 FALSE TRUE
52 (175.1967, -39.07486) 37850 FALSE TRUE
53 (174.7138, -36.74827) 37852 FALSE TRUE
54 (176.4489, -40.57728) 38057 FALSE TRUE
55 (175.9159, -40.50652) 38224 FALSE TRUE
56 (172.0943, -43.83207) 38866 FALSE TRUE
57 (172.0842, -43.47962) 39063 FALSE TRUE
58 (171.8601, -42.11578)  3925  TRUE TRUE
59 (169.3182, -45.20342) 39564 FALSE TRUE
60  (169.6842, -45.0401)  5535 FALSE TRUE
```

List sites VI

```
sitesTempBB <- sites(sos = niwaHydro,
                     includeTemporalBBox = TRUE, includePhenomena = TRUE)
str(sitesTempBB[1:3,c(1,3,80)]@data)
```

```
'data.frame':   3 obs. of  3 variables:
 $ siteID                                                  : chr  "1056" "11234" "12429"
 $ MTHLY_STATS: DAYS OF OCCURRENCE (FOG) (MTHLY: FOG DAYS):List of 3
  ..$ MTHLY_STATS: DAYS OF OCCURRENCE (FOG) (MTHLY: FOG DAYS):'data.frame': 1 obs. of  2 variables:
  .. ..$ timeBegin: POSIXct, format: "1981-10-01"
  .. ..$ timeEnd  : POSIXct, format: "1995-11-01"
  ..$ MTHLY_STATS: DAYS OF OCCURRENCE (FOG) (MTHLY: FOG DAYS): logi NA
  ..$ MTHLY_STATS: DAYS OF OCCURRENCE (FOG) (MTHLY: FOG DAYS): logi NA
 $ MTHLY_STATS: TOTAL RAINFALL (MTHLY: TOTAL RAIN)        :List of 3
  ..$ MTHLY_STATS: TOTAL RAINFALL (MTHLY: TOTAL RAIN):'data.frame': 1 obs. of  2 variables:
  .. ..$ timeBegin: POSIXct, format: "1981-10-01"
  .. ..$ timeEnd  : POSIXct, format: "2019-08-01"
  ..$ MTHLY_STATS: TOTAL RAINFALL (MTHLY: TOTAL RAIN):'data.frame': 1 obs. of  2 variables:
  .. ..$ timeBegin: POSIXct, format: "1995-06-01"
  .. ..$ timeEnd  : POSIXct, format: "2019-08-01"
  ..$ MTHLY_STATS: TOTAL RAINFALL (MTHLY: TOTAL RAIN):'data.frame': 1 obs. of  2 variables:
  .. ..$ timeBegin: POSIXct, format: "1995-04-01"
  .. ..$ timeEnd  : POSIXct, format: "2018-02-01"
```

One can filter sites using phenomena and temporal extent.

```
head(sites(sos = niwaHydro, phenomena = phenomena[3,]))
```

```
           coordinates siteID
1    (173.926, -35.183)   1056
2  (172.851, -42.53433)  11234
3 (172.9716, -41.09798)  12429
4 (173.9628, -41.49891)  12430
5 (169.3148, -45.20724)  12431
6 (174.9844, -40.90392)  12442
```

Filter Sites II

```
head(sites(sos = niwaHydro,
           begin = as.POSIXct("1904-01-01"),
           end = as.POSIXct("1905-12-31")))
```

```
            coordinates siteID
1    (173.926, -35.183)   1056
2  (172.851, -42.53433)  11234
3 (172.9716, -41.09798)  12429
4 (173.9628, -41.49891)  12430
5 (169.3148, -45.20724)  12431
6 (174.9844, -40.90392)  12442
```

The SpatialPointsDataFrame allows access to coordinates with coordinate reference system (CRS).

```
library(sp)
coordinates(sites)[1:3,]
```

```
          lon       lat
[1,] 173.9260 -35.18300
[2,] 172.8510 -42.53433
[3,] 172.9717 -41.09798
```

```
sites@proj4string
```

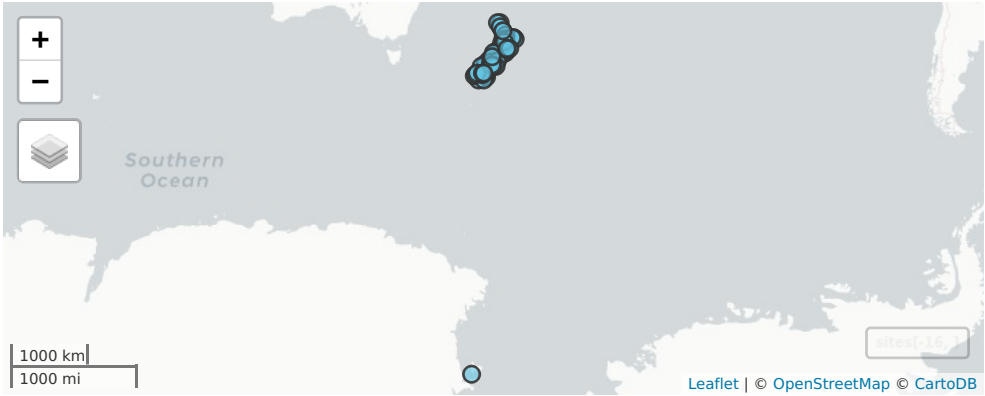
```
CRS arguments:
 +init=epsg:4326 +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0
```

```
bbox(sites)
```

```
           min       max
lon -176.47500 177.92180
lat  -77.84935 -35.13352
```

One way of plotting and exploring the sites is:

```
library("mapview")
mapview(sites[-16,], legend=FALSE, col.regions="#65c6e4")
```



Eike H. Jürrens,
Benedikt Gräler,
Daniel Nüst

The function `siteList(..)` is an analogue, but the result is a list rather than a spatial object.

```
siteList <- siteList(sos = niwaHydro)
str(siteList)
```

```
'data.frame':   60 obs. of  1 variable:
 $ siteID: chr  "1056" "11234" "12429" "12430" ...
```

```
head(siteList)
```

```
  siteID
1   1056
2  11234
3  12429
4  12430
5  12431
6  12442
```

getData

The function `getData(..)` retrieves data and returns them in a long form 'data.frame' that is ready-to-use for the `xts` or `spacetime` package.

The returned data can be limited by thematic, spatial, and temporal filters. Thematic filtering (phenomena) support the values of the previous functions as inputs. Spatial filters are either sites, or a bounding box. Temporal filter is a time period during which observations are made.

Without a temporal extent, the used SOS only returns the last measurement.

```
obsData <- getData(sos = niwaHydro,
                   phenomena = phenomena[18,1],
                   sites = siteList[1:2,1]
)
str(obsData,1)
```

```
'data.frame':   2 obs. of  3 variables:
 $ siteID                                                          : Factor w/ 2 levels "1056","11234": 1 2
 $ timestamp                                                       : POSIXct, format: "2019-08-01" "2019-08-01"
 $ MTHLY_STATS: EXTREME MAXIMUM TEMPERATURE (MTHLY: EXTR MAX TEMP): num  20.3 18.8
  ..- attr(*, "metadata")=Formal class 'WmlMeasurementTimeseriesMetadata' [package "sos4R"] with 1 slot
  ..- attr(*, "defaultPointMetadata")=Formal class 'WmlDefaultTVPMeasurementMetadata' [package "sos4R"] with 2 slots
```

The result data.frame includes additional metadata.

```
attributes(obsData[[3]])
```

```
$metadata
An object of class "WmlMeasurementTimeseriesMetadata"
Slot "temporalExtent":
GmlTimePosition [ time: 2019-08-01 ]

$defaultPointMetadata
An object of class "WmlDefaultTVPMeasurementMetadata"
Slot "uom":
[1] "DEGREE_CELSIUS"

Slot "interpolationType":
An object of class "WmlInterpolationType"
Slot "href":
[1] "http://www.opengis.net/def/timeseriesType/WaterML/2.0/continuous"

Slot "title":
[1] "Instantaneous"
```

Request more data with a temporal extent for all sites.

```
obsData <- getData(sos = niwaHydro,
                   phenomena = phenomena[18,1],
                   sites = siteList[1,1],
                   begin = parsedate::parse_iso_8601("1970-01-01T12:00:00+12:00"),
                   end = parsedate::parse_iso_8601(Sys.time()))
```

```
str(obsData, 2)
```

```
'data.frame':   421 obs. of  3 variables:
 $ siteID                                                    : Factor w/ 1 level "1056": 1 1 1 1 1 1 1 1 1 1 ...
 $ timestamp                                                 : POSIXct, format: "1981-10-01" "1981-11-01" ...
 $ MTHLY_STATS: EXTREME MAXIMUM TEMPERATURE (MTHLY: EXTR MAX TEMP): num  20.6 21.9 25 27.8 29.2 24.5 22.9 20.7 18.3 18 ...
  ..- attr(*, "metadata")=Formal class 'WmlMeasurementTimeseriesMetadata' [package "sos4R"] with 1 slot
  ..- attr(*, "defaultPointMetadata")=Formal class 'WmlDefaultTVPMeasurementMetadata' [package "sos4R"] with 2 slots
```


Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

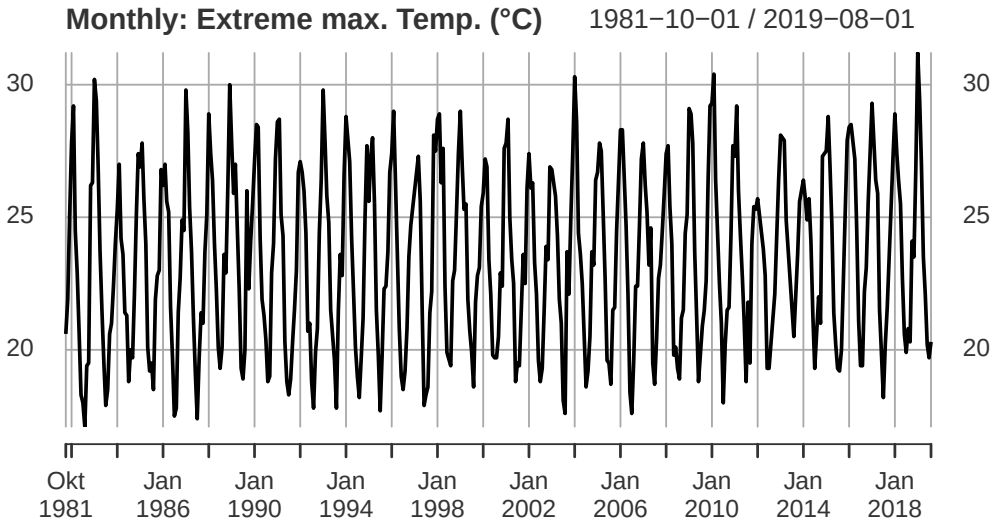
Plotting and Analytics

Plot received data as time series:

```
library(xts)
```

```
Loading required package: zoo


Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Registered S3 method overwritten by 'xts':
  method     from
  as.zoo.xts zoo
```

```
ts1056 <- xts(obsData[obsData$siteID == '1056',3],
              obsData[obsData$siteID == '1056', "timestamp"])
plot(x = ts1056,
     main = "Monthly: Extreme max. Temp. (°C)")
```



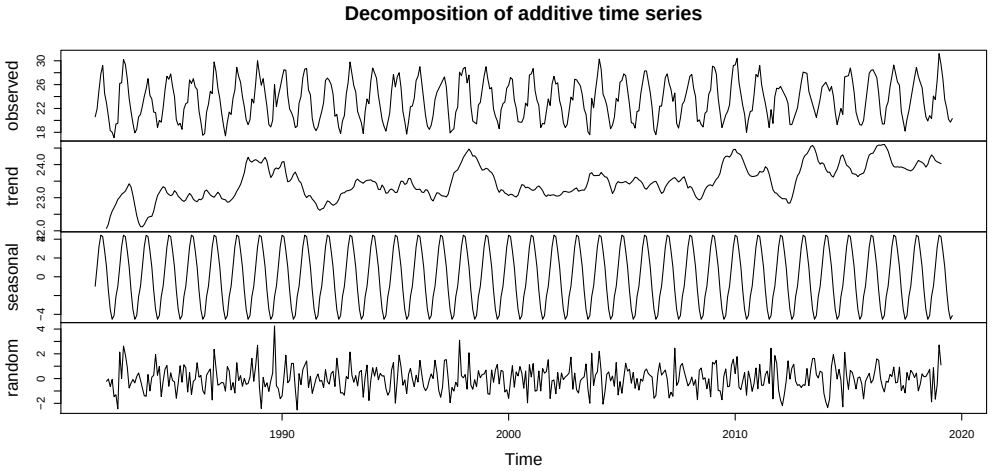
Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

52north

Conversion of the time series:

```
extTs <- merge(ts1056, as.xts(ts(start = c(1981,10), end = c(2018, 8), frequency = 12)))
smplTs <- ts(as.numeric(extTs[,1]), c(1981, 10), frequency = 12)
smplTsFill <- imputeTS::na.interpolation(smplTs)
decTs <- decompose(smplTsFill)
```

```
plot(decTs)
```



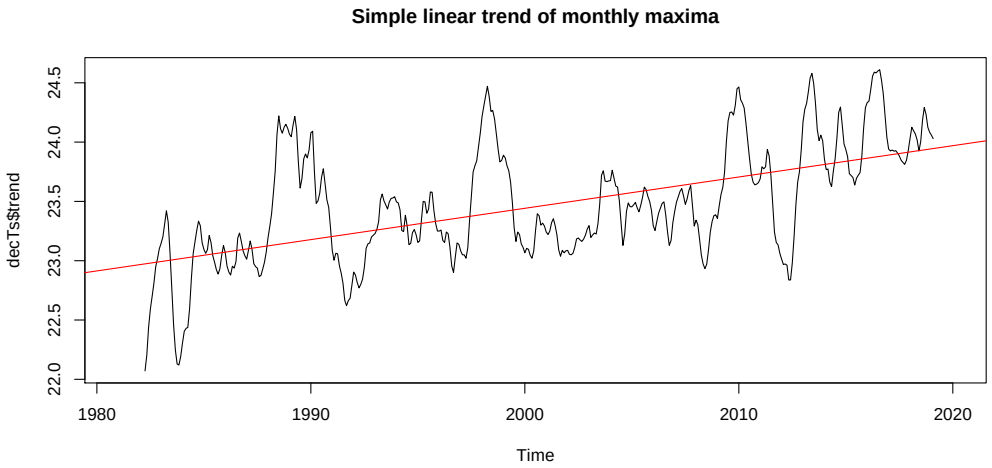
Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

52north

```
lm(y~x, data.frame(y=as.numeric(decTs$trend), x=1:length(decTs$trend)))
```

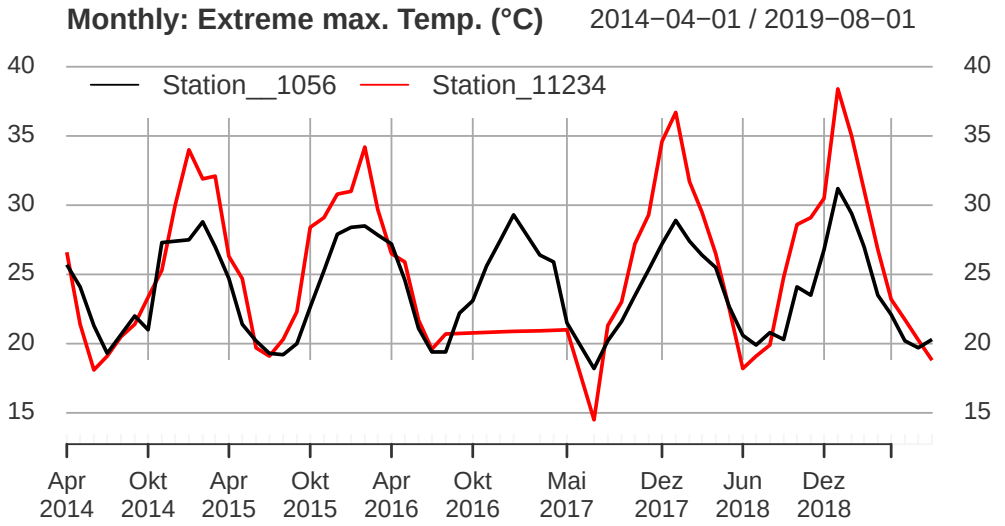
```
Call:
lm(formula = y ~ x, data = data.frame(y = as.numeric(decTs$trend),
    x = 1:length(decTs$trend)))

Coefficients:
(Intercept)            x
  22.963303     0.002235
```

```
plot(decTs$trend,xlim=c(1981, 2020), main="Simple linear trend of monthly maxima")
abline(22.96-(1981+9/12)*0.0022*12, 0.0022*12, col="red")
```


Simple linear trend of monthly maxima
decTs$trend
22.0
22.5
23.0
23.5
24.0
24.5
1980
1990
2000
2010
2020
Time

You can also retrieve data for phenomena from multiple sites.

```
multipleSites <- siteList$siteID[35:38]
obsData <- getData(sos = niwaHydro,
                   phenomena = phenomena[18,1],
                   sites = sites[1:2,]$siteID, # siteList[1:2,1]
                   begin = parsedate::parse_iso_8601(Sys.time()-5.5*365*24*60*60),
                   end = parsedate::parse_iso_8601(Sys.time())
)


ts1056 <- xts(obsData[obsData$siteID == '1056',3],
              obsData[obsData$siteID == '1056',"timestamp"])
names(ts1056) <- "Station__1056"
ts11234 <- xts(obsData[obsData$siteID == '11234',3],
               obsData[obsData$siteID == '11234',"timestamp"])
names(ts11234) <- "Station_11234"
p <- plot(x = na.fill(merge(ts1056, ts11234), list(NA, "extend", NA)),
     main = "Monthly: Extreme max. Temp. (°C)",
     ylim=c(14,41))
addLegend("topleft", ncol=2, col = c("black", "red"), lty=1)
plot(p)
```



Eike H. Jürrens, Benedikt Gräler, Daniel Nüst

52north

```
library(sp)
mapview(sites[1:2,], map.types="OpenTopoMap")
```