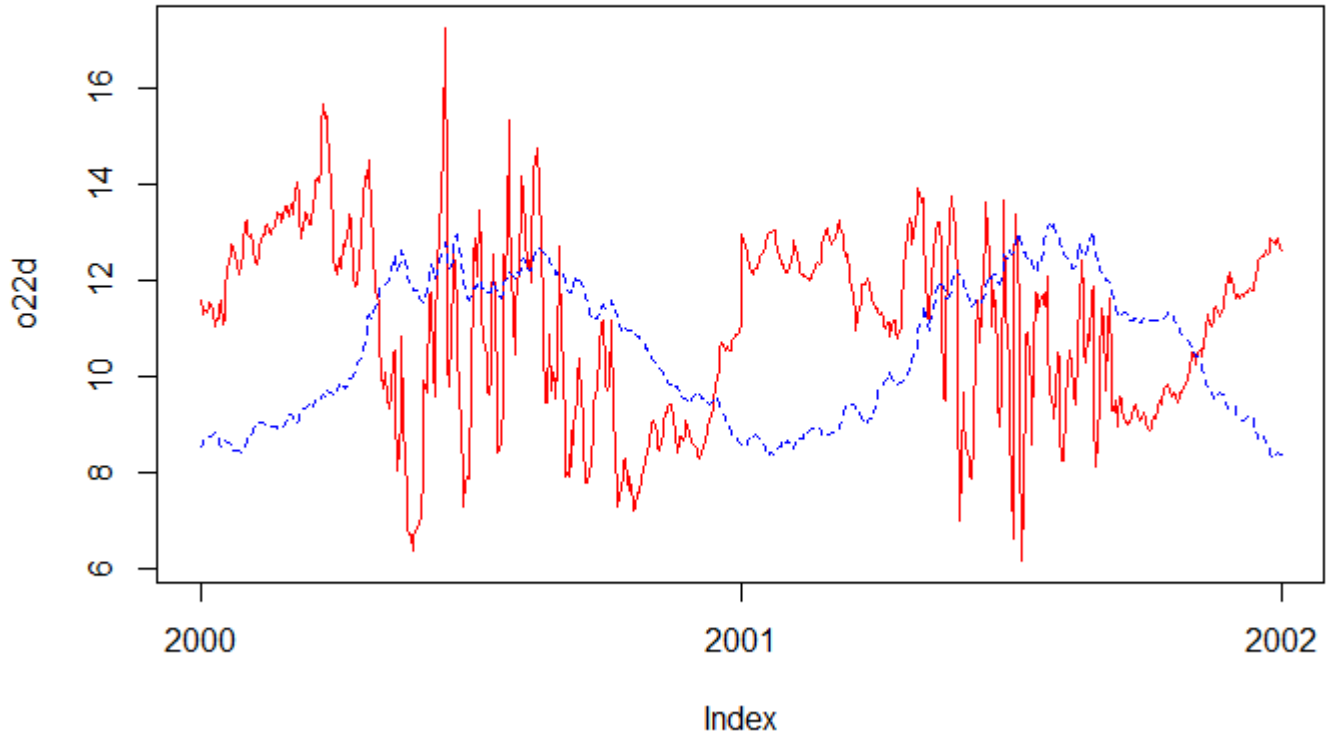
Modelling and Data Analysis with R

Lecture 2 Time Series - Analysis



Time Series Analysis

R packages

- R: programming language & software environment for statistical computing and graphics
- package: integrated collection of functions and datasets

- standard packages
- specialized (contributed) packages

Time Series Analysis

Overview Standard Packages in R

- **base** Base R functions (and datasets before R 2.0.0).
- compiler R byte code compiler (added in R 2.13.0).
- datasets Base R datasets (added in R 2.0.0).
- grDevices Graphics devices for base and grid graphics (added in R 2.0.0).
- **graphics** R functions for base graphics.
- grid rewrite of graphics layout capabilities + some support for interaction
- methods Formally defined methods and classes for R objects, plus other programming tools, as described in the Green Book.
- parallel Support for parallel computation, including by forking and by sockets, and random-number generation (added in R 2.14.0).
- splines Regression spline functions and classes.
- **stats** R statistical functions.
- stats4 Statistical functions using S4 classes.
- Tcltk Interface and language bindings to Tcl/Tk GUI elements.
- tools Tools for package development and administration.
- utils R utility functions.

Packages in R

Package Installation

- RStudio -> Tools -> Install Packages …

Basic Commands

- `library()` list installed packages
- `library(zoo)` load a package (e.g. zoo)
- `detach(package:zoo)` unload a package (e.g. zoo)
- `help(zoo)` help page for package zoo

Time Series Analysis

Specialized Packages in R (Contributed Packages)

- **https://cran.r-project.org**
  The **C**omprehensive **R** **A**rchive **N**etwork
- 7515 available packages (!)
  individual contributions
  - different background
  - different style
  - specific targets

-> heterogeneous environment

Time Series Analysis

Packages for Time Series Analysis (examples!)

- ArDec — Time series autoregressive-based decomposition
- astsa — Applied Statistical Time Series Analysis
- aTSA — Alternative Time Series Analysis
- cardidates — Identification of Cardinal Dates in Ecological Time Series
- ClamR — Time Series Modeling for Climate Change Proxies
- dyn — Time Series Regression
- fNonlinear — Nonlinear and Chaotic Time Series Modelling
- forecast — Forecasting Functions for Time Series and Linear Models
- freqdom — Frequency Domain Analysis for Multivariate Time Series
- funtimes — Functions for Time Series Analysis
- hydroTSM — Time series management, analysis and interpolation for hydrological modelling
- TSA — Time Series Analysis
- zoo — S3 Infrastructure for Regular and Irregular Time Series (Z's Ordered Observations)

Packages in R

Let's analyse the situation

- R Packages are not developed with one strategy
- collection of individual implementations
- overlapping functionality, no standards
- missing or inefficient/inappropriate functionality
- different data structure and approaches
- several different solutions possible
- not all problems are covered well by packages

Conclusion:

- you need your own strategy before to apply R packages

Packages in R

Problem Specific Strategy

- R provides methods/functions – no solution
- every problem has specific properties
- examples:

air temperature	partial periodic
precipitation	event oriented
reservoir outflow	human controlled

- combination of individual time series in a project leads to a specific, problem oriented time series analysis strategy
- partial problem classes / types specification

Time Series Example 2

Simple Water Quality Data Analysis

- measurements: six years 1996-2001 10 min interval one station

chlorophyll-a total	[mirco g/l]
chlorophyll-a alive	[%]
conductivity	[micro s]
oxygen content	[mg/l]
oxygen saturation	[%]
pH-value	
global radiation	[W/m]
air temperature	[°C]
turbidity	[%]
water temperature	[°C]
absorption of black light radiation 254nm	[%]

Time Series Example 2

Target

- to analyse the given data sets with R scripts
  to explore knowledge and understanding
  of the relationship between the different biophysical
  state variables for the given time window
- no “Scheme F” assignment !
- mixture of structured and intuitive strategy
- learning by doing: data mining / exploration

Time Series Example 2

Strategy Proposal

- writing a coarse work plan
- iterative adaptation and specialization
- general steps:
  1. data pre-processing
  2. data analysis univariate
  3. data analysis multivariate
  4. reporting, reporting, reporting, ….

Time Series Example 2

Proposals for Pre-Processing

- reading data files as time series
- value range identification and validation
- identifying and filling of gaps
- harmonization of time series (if necessary)
- basic statistics
- scaling/aggregation to hourly, daily, monthly, … values
- subdivision by years 1996, 1997,… , 2001
- …

Time Series Example 2

Proposals for Univariate Time Series Analysis

- standard statistics
- histogram
- clustering
- fast Fourier analysis
- seasonal decomposition
- regression
- autocorrelation
- …

Time Series Example 2

Proposals for Multivariate Time Series Analysis

- scatter plot
- correlation
- principal components analysis
- multivariate regression
- supervised and unsupervised clustering
- …

Time Series Example 2

Proposals for Reporting

- protocol oriented style
  input -> method -> output
- description of the initial data set
- description of the planned and processed strategy
- evaluation and discussion of the results
- summary of the main findings
- conclusion

Water Quality Data

Classes and Objects in R

- idea: encapsulation of semantic units
- 10 time series with same structure but diff. content
- 1 class -> 10 objects

```
setClass("Water_Ts",representation(name="character",     unit="character",
                                   rawdata="data.frame", timeseries="zoo"))

ce = new("Water_Ts"); ce@name="chlorophyll-a total"; ce@unit  = "[mirco g/l]"
lf = new("Water_Ts"); lf@name="conductivity";        lf@unit  = "[micro s]"
o2 = new("Water_Ts"); o2@name="oxygen content";      o2@unit  = "[mg/l]"
ot = new("Water_Ts"); ot@name="oxygen saturation";   ot@unit  = "[%]"
ph = new("Water_Ts"); ph@name="pH-value";            ph@unit  = ""
st = new("Water_Ts"); st@name="global radiation";    st@unit  = "[W/m]"
tl = new("Water_Ts"); tl@name="air temperature";     tl@unit  = "[C]"
tr = new("Water_Ts"); tr@name="turbidity";           tr@unit  = "[%]"
tw = new("Water_Ts"); tw@name="water temperature";   tw@unit  = "[°C"
uv = new("Water_Ts"); uv@name="light absorption";    uv@unit  = "[%]"
```

Water Quality Data

Reading Data Files

```
#
# reading csv files
ce@rawdata <- read.csv(file="ce.txt",sep="\t", header=F,dec=",")
lf@rawdata <- read.csv(file="lf.txt",sep="\t", header=F,dec=",")
o2@rawdata <- read.csv(file="o2.txt",sep="\t", header=F,dec=",")
ot@rawdata <- read.csv(file="ot.txt",sep="\t", header=F,dec=",")
ph@rawdata <- read.csv(file="ph.txt",sep="\t", header=F,dec=",")
st@rawdata <- read.csv(file="st.txt",sep="\t", header=F,dec=",")
tl@rawdata <- read.csv(file="tl.txt",sep="\t", header=F,dec=",")
tr@rawdata <- read.csv(file="tr.txt",sep="\t", header=F,dec=",")
tw@rawdata <- read.csv(file="tw.txt",sep="\t", header=F,dec=",")
uv@rawdata <- read.csv(file="uv.txt",sep="\t", header=F,dec=",")
#
# convert to zoo object incl. converting first column string to time values
ce@timeseries <- read.zoo(ce@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
lf@timeseries <- read.zoo(lf@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
o2@timeseries <- read.zoo(o2@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
ot@timeseries <- read.zoo(ot@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
ph@timeseries <- read.zoo(ph@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
st@timeseries <- read.zoo(st@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
tl@timeseries <- read.zoo(tl@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
tr@timeseries <- read.zoo(tr@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
tw@timeseries <- read.zoo(tw@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
uv@timeseries <- read.zoo(uv@rawdata[,1:2],format="%d.%m.%Y %H:%M",tz="UTC")
```

Water Quality Data

First Analysis: size of data

```
uv.rawdata          data.frame     3          27.7 MB     302927 obs. of 3 variables
tw.rawdata          data.frame     3          31.8 MB     347040 obs. of 3 variables
tr.rawdata          data.frame     3          27.9 MB     305117 obs. of 3 variables
tl.rawdata          data.frame     3          27.9 MB     304518 obs. of 3 variables
st.rawdata          data.frame     3          16.4 MB     179025 obs. of 3 variables
ph.rawdata          data.frame     3          27.9 MB     305071 obs. of 3 variables
ot.rawdata          data.frame     3          23.1 MB     252275 obs. of 3 variables
o2.rawdata          data.frame     3          27.9 MB     304676 obs. of 3 variables
lf.rawdata          data.frame     3          27.9 MB     304661 obs. of 3 variables
lc.rawdata          data.frame     3          25.9 MB     282931 obs. of 3 variables
uv.timeseries       zoo            302927     4.6 MB      Large zoo (302927 elements, 4.6 Mb)
tw.timeseries       zoo            347040     5.3 MB      Large zoo (347040 elements, 5.3 Mb)
tr.timeseries       zoo            305117     4.7 MB      Large zoo (305117 elements, 4.7 Mb)
tl.timeseries       zoo            304518     4.6 MB      Large zoo (304518 elements, 4.6 Mb)
st.timeseries       zoo            179025     2.7 MB      Large zoo (179025 elements, 2.7 Mb)
ph.timeseries       zoo            305071     4.7 MB      Large zoo (305071 elements, 4.7 Mb)
ot.timeseries       zoo            252275     3.9 MB      Large zoo (252275 elements, 3.9 Mb)
lf.timeseries       zoo            304661     4.6 MB      Large zoo (304661 elements, 4.6 Mb)
lc.timeseries       zoo            282931     4.3 MB      Large zoo (282931 elements, 4.3 Mb)
```

Water Quality Data

First Analysis: Metadata

```
printMetaData <- function(water_ts)
{
 print(paste(water_ts@name,":values",length(water_ts@timeseries),
                          "NaN:",sum(is.na(water_ts@timeseries)),
                          sprintf("%.1f%%",(sum(is.na(water_ts@timeseries)/
                          length(water_ts@timeseries)*100))))
 print(sprintf(" min  %6.2f %s",min(water_ts@timeseries,na.rm=TRUE),water_ts@unit))
…
# annual values
 title_text="                      year "
 na_text   ="                      NA   "
…
# loop on all years
 for(i in 1:6)
  {
   year_time_series = extract_year(water_ts@timeseries,i)
   title_text = paste(title_text,sprintf("  %4d",(1995+i)))
   na_text   = paste(na_text,sprintf("%6d",sum(is.na(year_time_series))))
…
   min_text  = paste(min_text,sprintf("%6.2f",min(year_time_series,na.rm=TRUE)))
…
  }
# print the results line by line
 print(title_text)
 print(na_text)
…
 print(min_text)
…
}
```

Water Quality Data

R: Output Connections (stdout, stderr)

- connections in R

```
showConnections(all = TRUE)
  description class      mode text   isopen   can read can write
0 "stdin"     "terminal" "r"  "text" "opened" "yes"    "no"
1 "stdout"    "terminal" "w"  "text" "opened" "no"     "yes"
2 "stderr"    "terminal" "w"  "text" "opened" "no"     "yes"
```

- print(): text send to stdout
- message(): general message send to stderr
- warning(): warning text send to stderr
- stop(): error text send to stderr
- sink(): sink diverts R output to a connection

Water Quality Data

R: Output Connections (stdout, stderr)

Examples

- `message("start of time series analysis")`
- `warning("gap found in time series")`
- `stop("division by zero!")`

- `sink(file=NULL,append=FALSE,type=c("output","message"),split=FALSE)`

```
sink("MetaData.txt")              # open stdout connection to file
print("text for the output")      # write text in the file -> stdout
message("text for the message")   # write text to terminal -> stderr
sink()                            # close stdout connection to file
file.show("MetaData.txt")         # display file
```

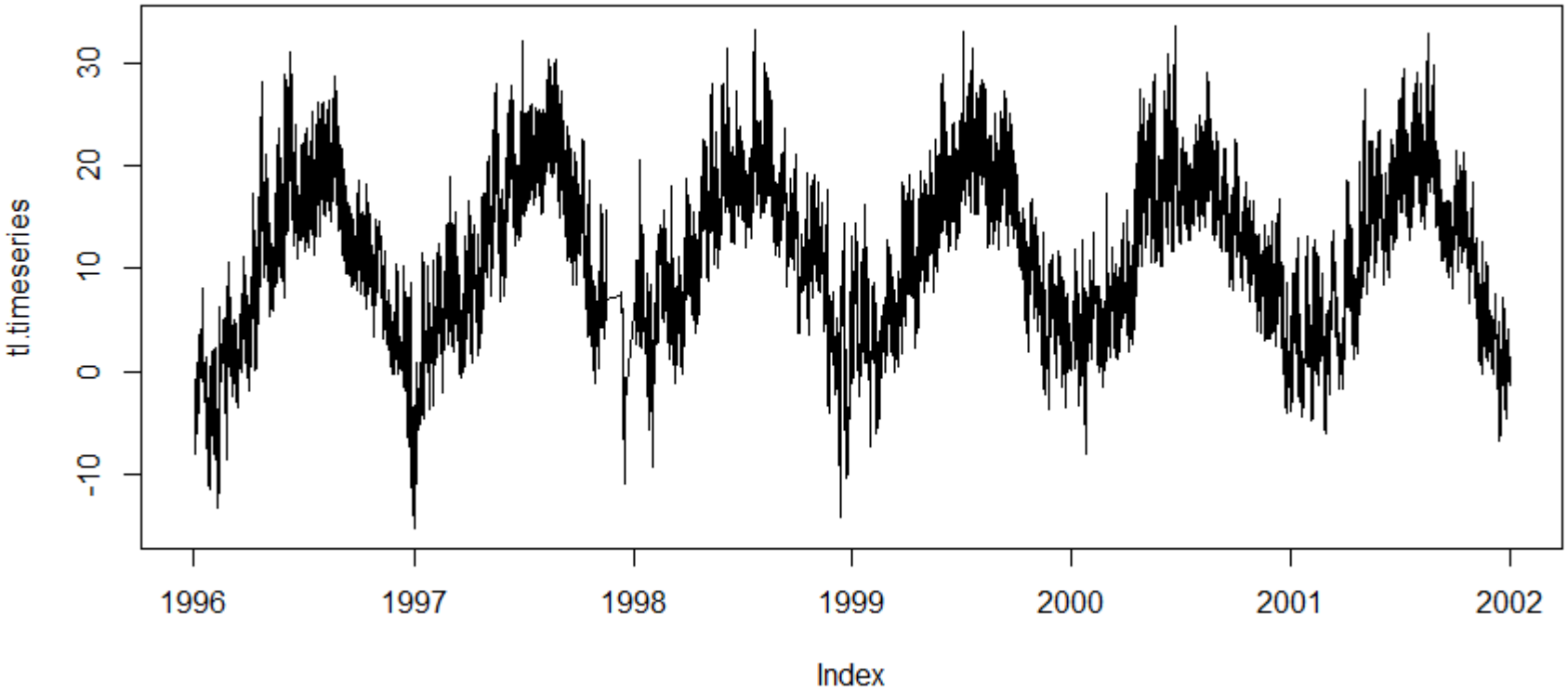
Water Quality Data

First Analysis: Metadata 1

```
air temperature : values 304518 NaN: 5629 1.8%
                        min  -15.30 [C]
                        max   33.70
                        mean  11.18
                        sd     7.87
                        year    1996   1997   1998   1999   2000   2001
                        NA      1011   1050   1664    681    745    478
                        NA %    2.74  15.42   6.73   1.58   2.53   2.34
                        min   -14.00 -15.30 -14.20  -7.30  -8.00  -6.80
                        max    31.10  32.20  33.40  33.10  33.70  33.00
                        mean    9.12  12.14  11.58  11.74  11.91  10.75
                        sd      8.71   8.27   7.50   7.60   6.95   7.72
water temperature : values 347040 NaN: 7344 2.1%
                        min    0.60 [°C
                        max   50.00
                        mean  13.21
                        sd     7.32
                        year    1996   1997   1998   1999   2000   2001
                        NA      1049   1642   1403   1275    790    560
                        NA %    2.82  16.55   5.96   2.71   2.61   2.50
                        min     0.60   1.80   1.40   2.20   1.96   1.60
                        max    25.30  27.00  25.10  26.10  26.40  50.00
                        mean   11.90  14.13  12.84  13.15  13.32  12.86
                        sd      7.32   7.43   7.07   7.41   6.90   7.50
```

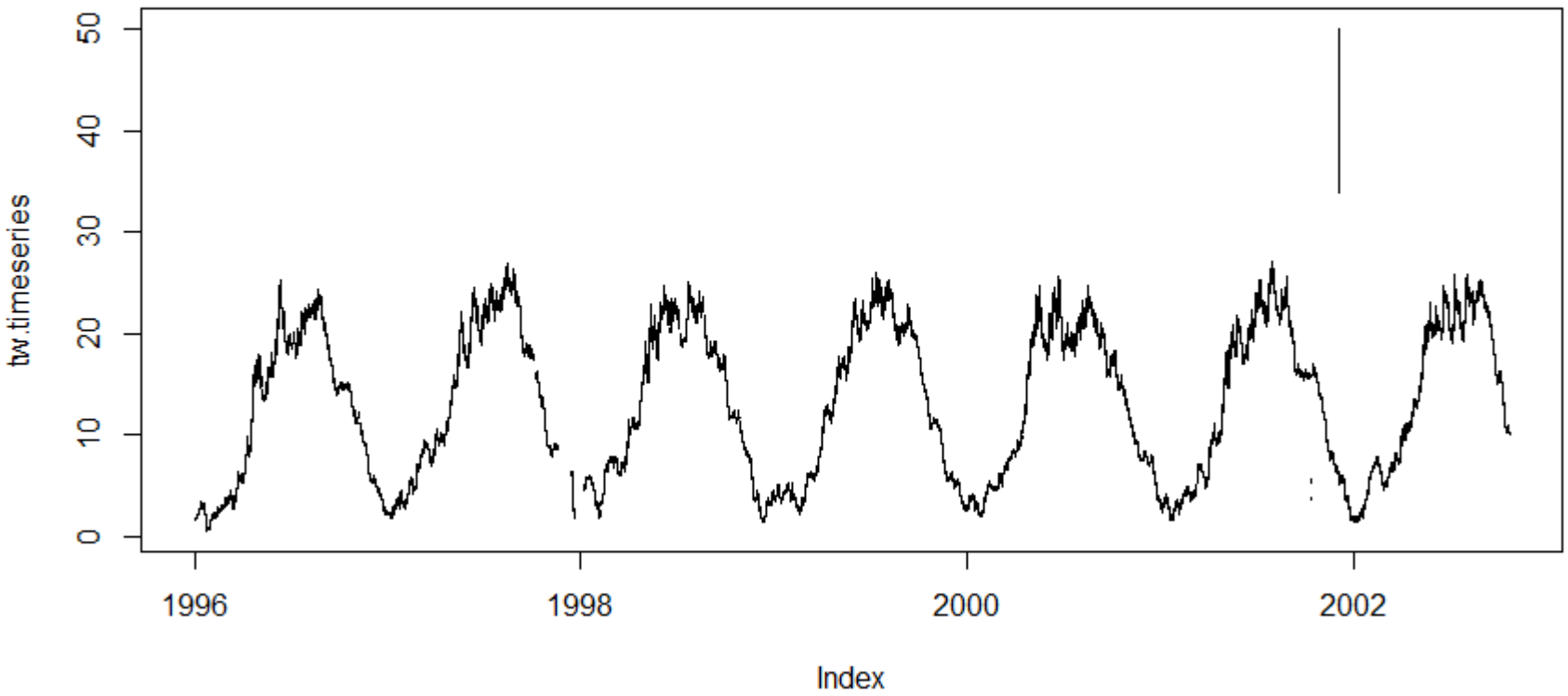
Water Quality Data

Raw Data Plot – Air Temperature [°C]



Water Quality Data

Raw Data Plot – Water Temperature [°C]



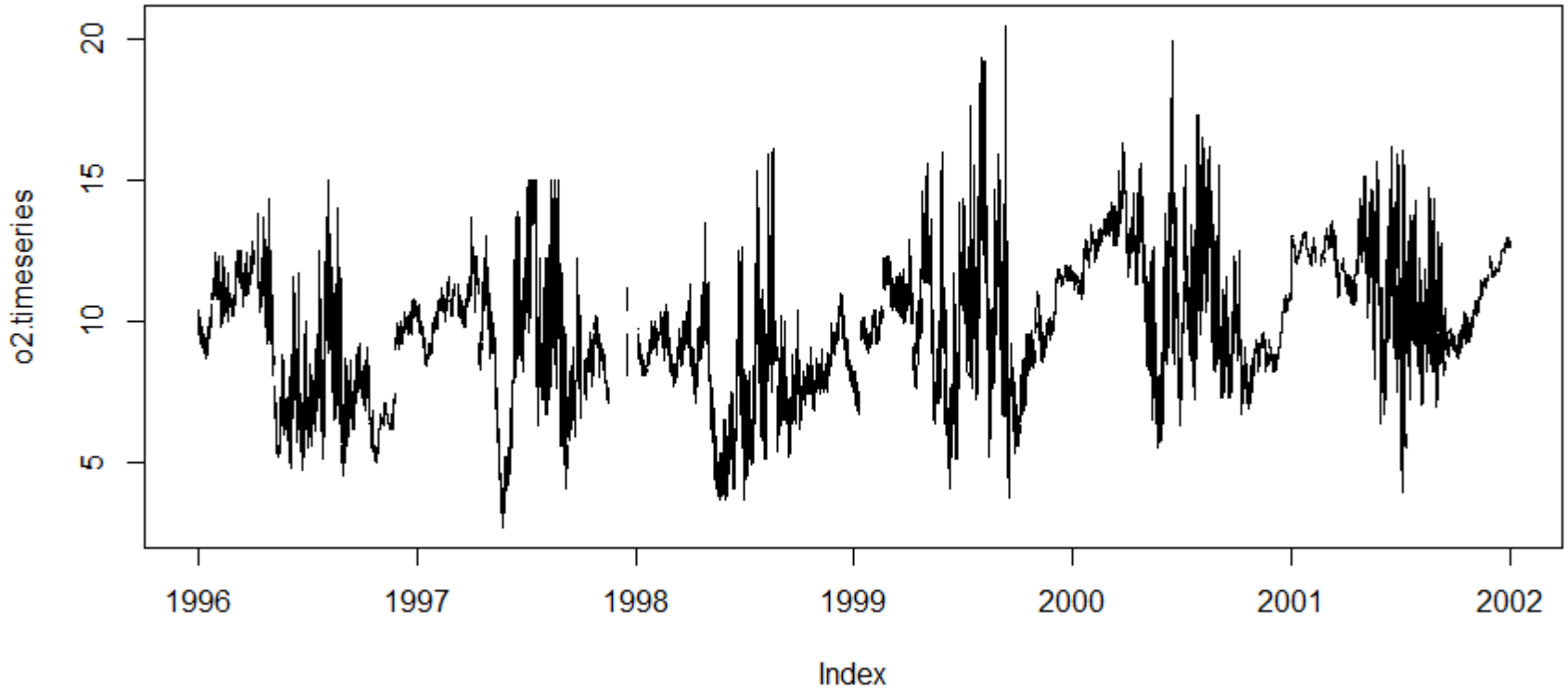
Water Quality Data

First Analysis: Metadata 2

```
oxygen content : values 304676 NaN: 12855 4.2%
                        min    2.70 [mg/l]
                        max   20.45
                        mean   9.73
                        sd     2.33
                        year    1996   1997   1998   1999   2000   2001
                        NA      2304   3804   1846   1489    913   2499
                        NA %    5.20  20.66   6.80   3.89   2.85   5.38
                        min     4.50   2.70   3.70   3.76   5.50   3.93
                        max    15.00  15.00  16.10  20.45  19.90  16.17
                        mean    8.76   9.30   8.16   9.96  10.88  11.21
                        sd      2.13   2.28   1.72   2.21   2.28   1.62
oxygen saturation : values 252275 NaN: 10854 4.3%
                        min   28.39 [%]
                        max  354.77
                        mean  88.37
                        sd    27.29
                        year    1996   1997   1998   1999   2000   2001
                        NA      2312   3853   1846   1991      0    852
                        NA %    5.22  20.76   6.80   4.85 100.00   3.05
                        min    47.10  28.39  39.48  40.96    Inf  47.62
                        max   171.19 185.88 187.16 233.56   -Inf 354.77
                        mean   79.11  90.10  75.86  94.16    NaN 102.34
                        sd     17.64  27.45  18.30  27.84     NA  32.57
```

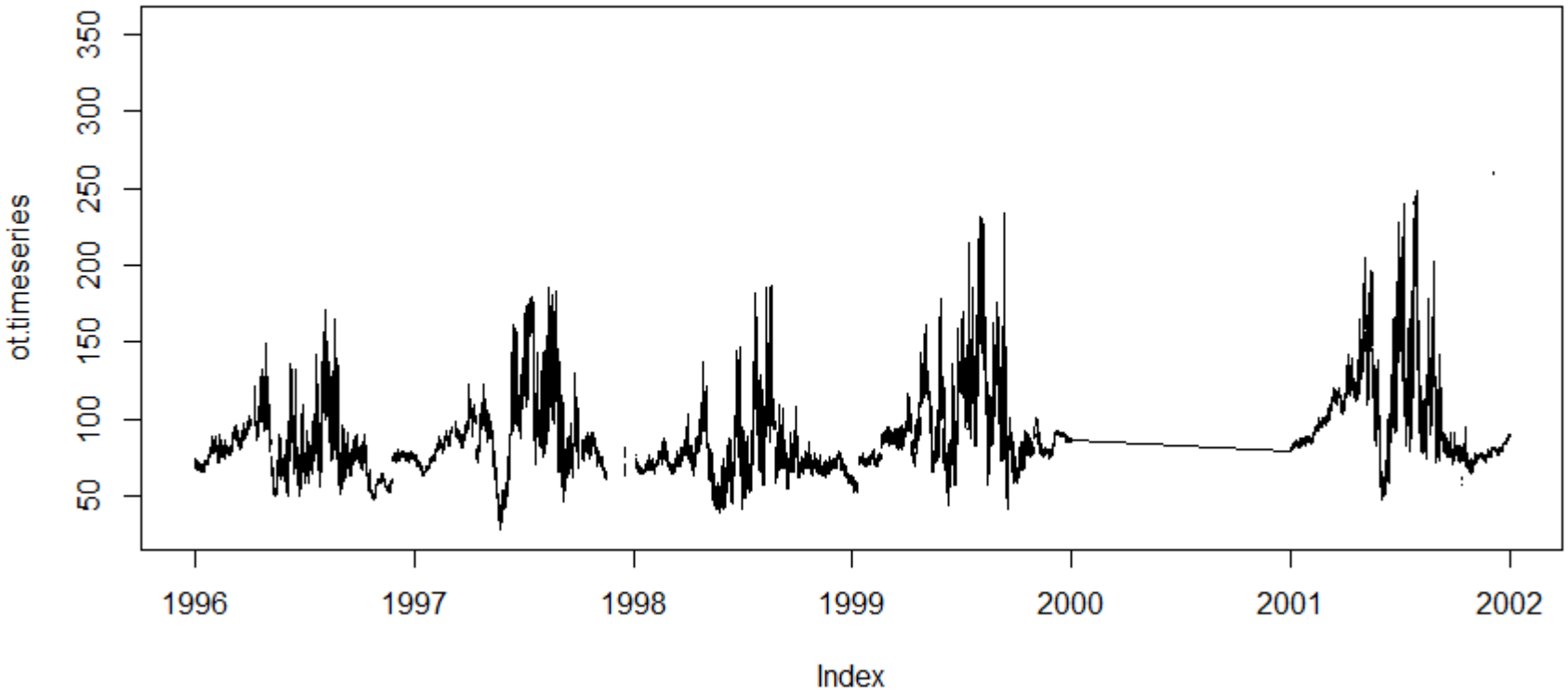
Water Quality Data

Raw Data Plot – Oxygen Content [mg/l]



Water Quality Data

Raw Data Plot – Oxygen Saturation [%]



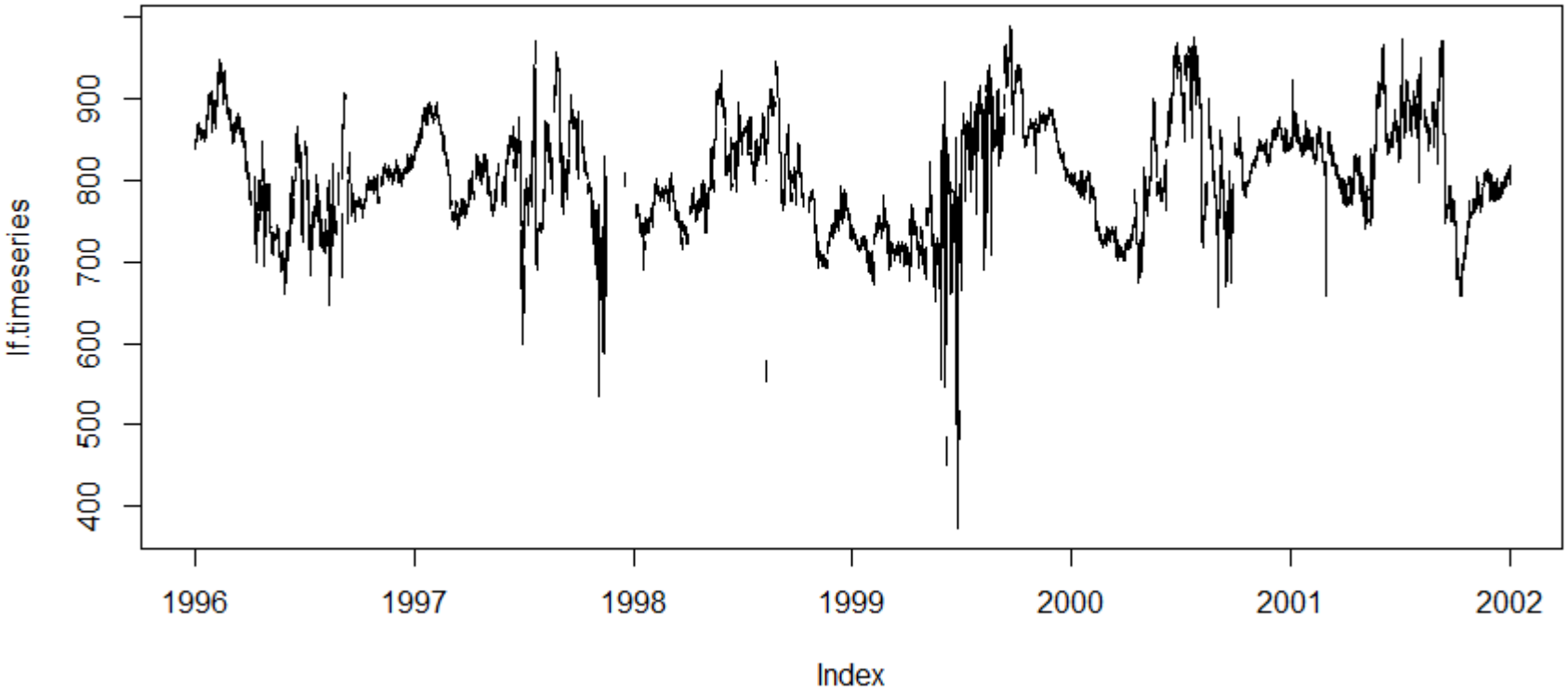
Water Quality Data

First Analysis: Metadata 3

```
conductivity : values 304661 NaN: 25177 8.3%
                      min  373.50 [micro s]
                      max  989.50
                      mean 807.68
                      sd    63.75
                      year    1996   1997   1998   1999   2000   2001
                      NA      4870   3799   4303   4247   2192   5766
                      NA %   10.09  20.65  11.47   8.36   5.28  12.40
                      min   640.00 536.00 553.00 373.50 644.60 657.30
                      max   949.00 972.00 945.00 989.50 975.00 974.30
                      mean  805.13 810.99 795.22 804.72 809.60 820.94
                      sd     56.52  55.80  55.00  80.27  68.70  57.48
pH-value : values 305071 NaN: 9119 3.0%
                      min    7.25
                      max    9.60
                      mean   8.15
                      sd     0.41
                      year    1996   1997   1998   1999   2000   2001
                      NA      1064   1181   1607   1229   1501   2537
                      NA %    2.84  15.67   6.34   2.62   4.00   5.45
                      min     7.40   7.40   7.50   7.60   7.25   7.69
                      max     9.00   9.30   9.60   9.40   8.95   9.53
                      mean    7.88   8.02   8.28   8.30   8.10   8.31
                      sd      0.29   0.37   0.38   0.36   0.39   0.44
```

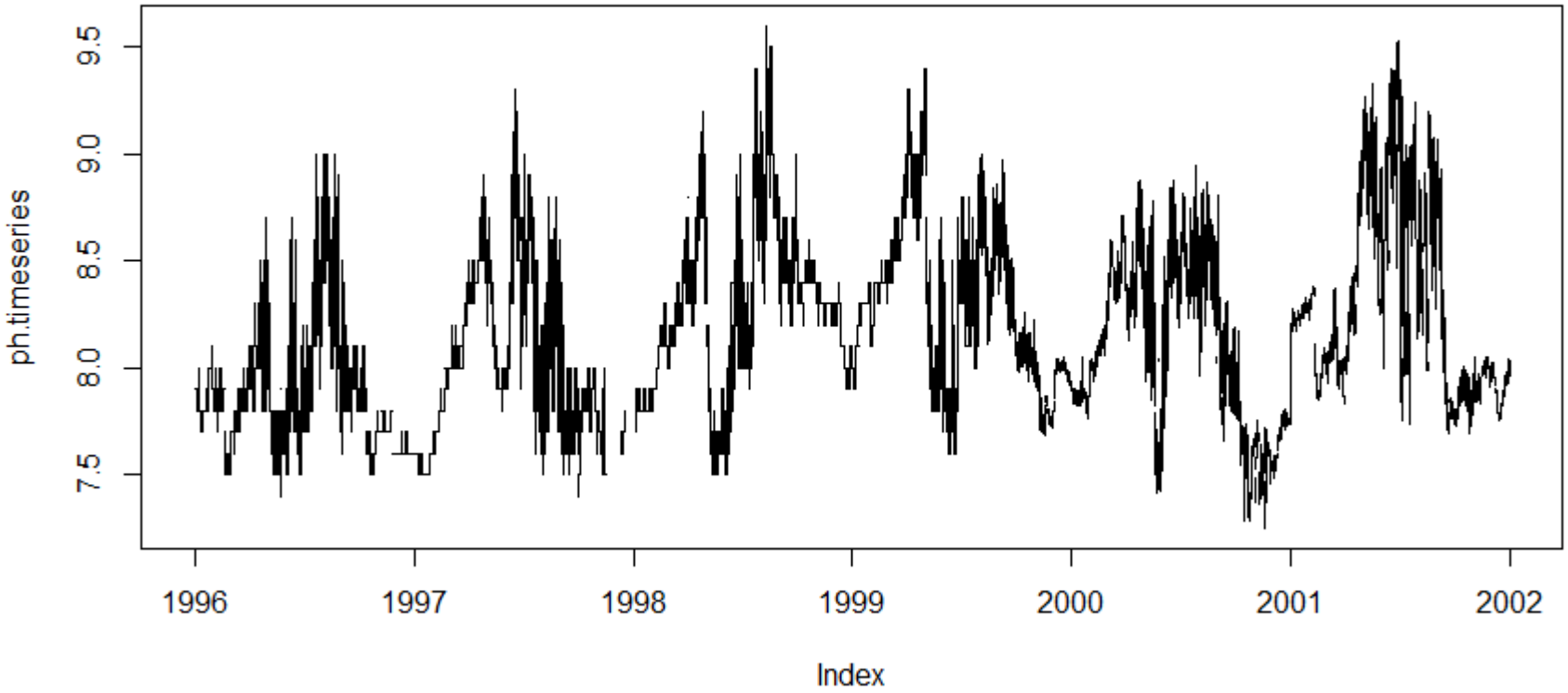
Water Quality Data

Raw Data Plot – Conductivity [µS/cm]



Water Quality Data

Raw Data Plot – pH Value



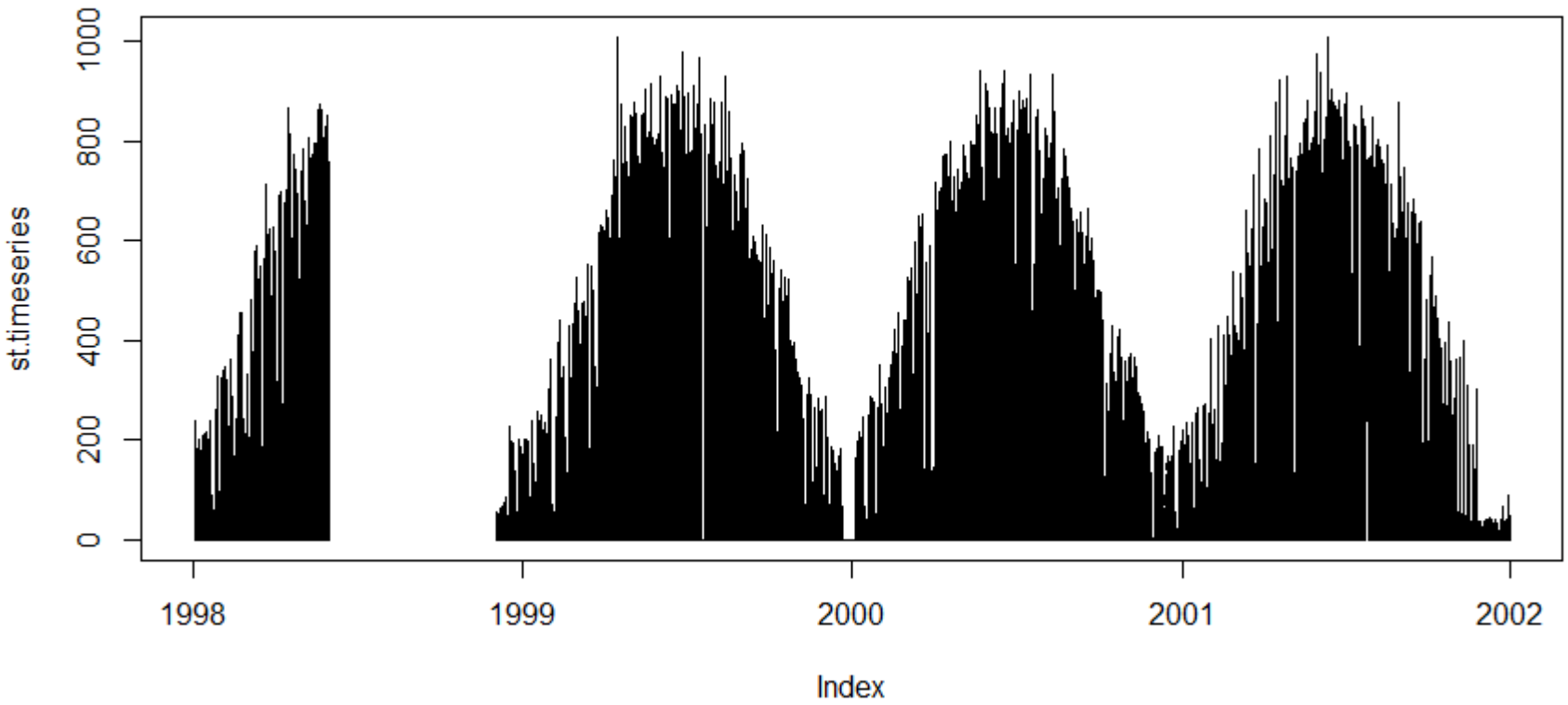
Water Quality Data

First Analysis: Metadata 4

```
global radiation : values 179025 NaN: 3399 1.9%
                       min    0.00 [W/m]
                       max  1009.10
                       mean  93.90
                       sd   174.66
                       year    1996   1997   1998   1999   2000   2001
                       NA         0      0    655   1754    275    715
                       NA %  100.00 100.00  53.85   7.17   2.28   2.56
                       min      Inf    Inf   0.00   0.00   0.00   0.00
                       max     -Inf   -Inf 875.00 1009.00 941.20 1009.10
                       mean     NaN    NaN  74.51 101.99  99.39  89.86
                       sd        NA     NA 153.25 182.39 178.60 171.81
light absorption : values 302927 NaN: 31284 10.3%
                       min    4.90 [%]
                       max   65.70
                       mean  21.93
                       sd     8.49
                       year    1996   1997   1998   1999   2000   2001
                       NA     10200   3189   7216   2599   2468   5612
                       NA %   20.23  19.49  18.66   6.88   5.81  12.11
                       min     5.10   5.20   5.30   4.90   8.10  12.30
                       max    30.20  42.50  58.40  65.70  29.90  30.40
```

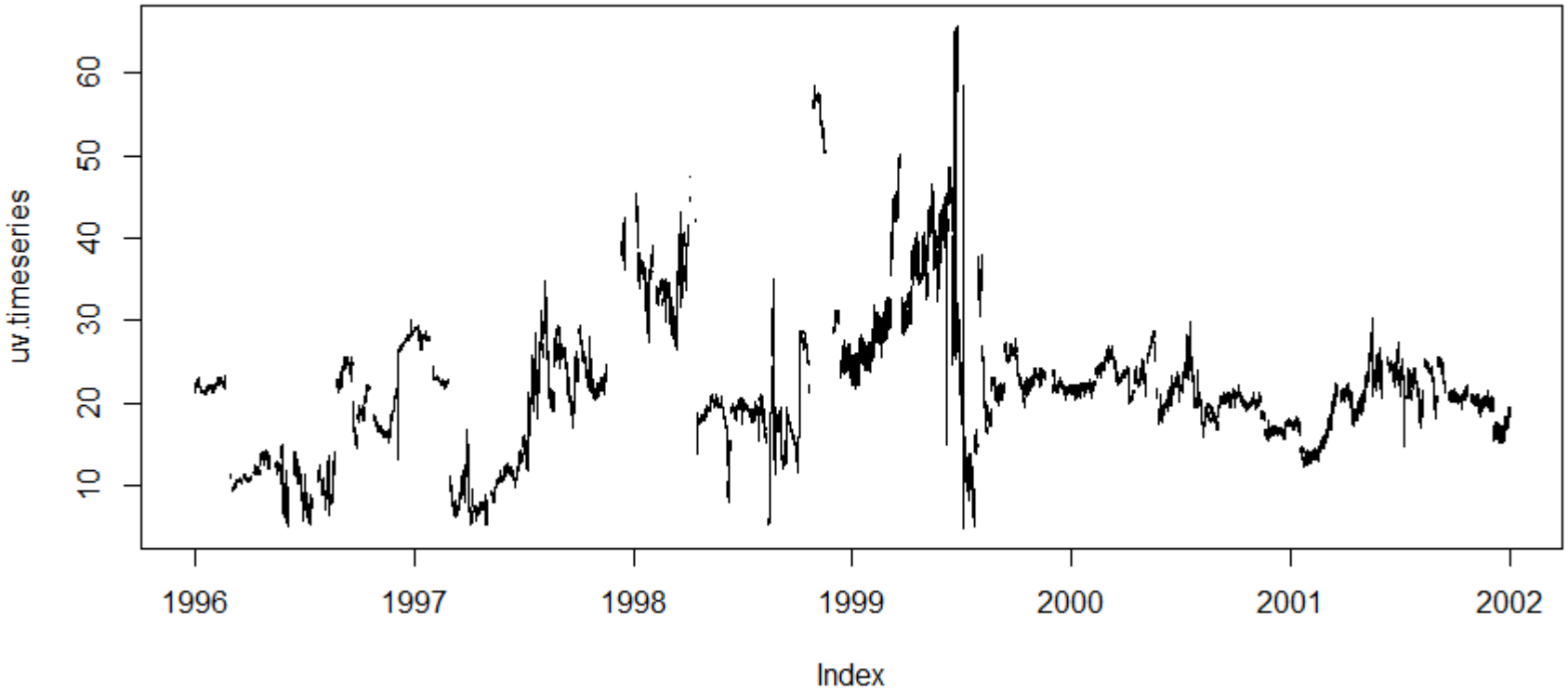
Water Quality Data

Raw Data Plot – Global Radiation [W/m]



Water Quality Data

Raw Data Plot – absorption UV light 245nm [%]



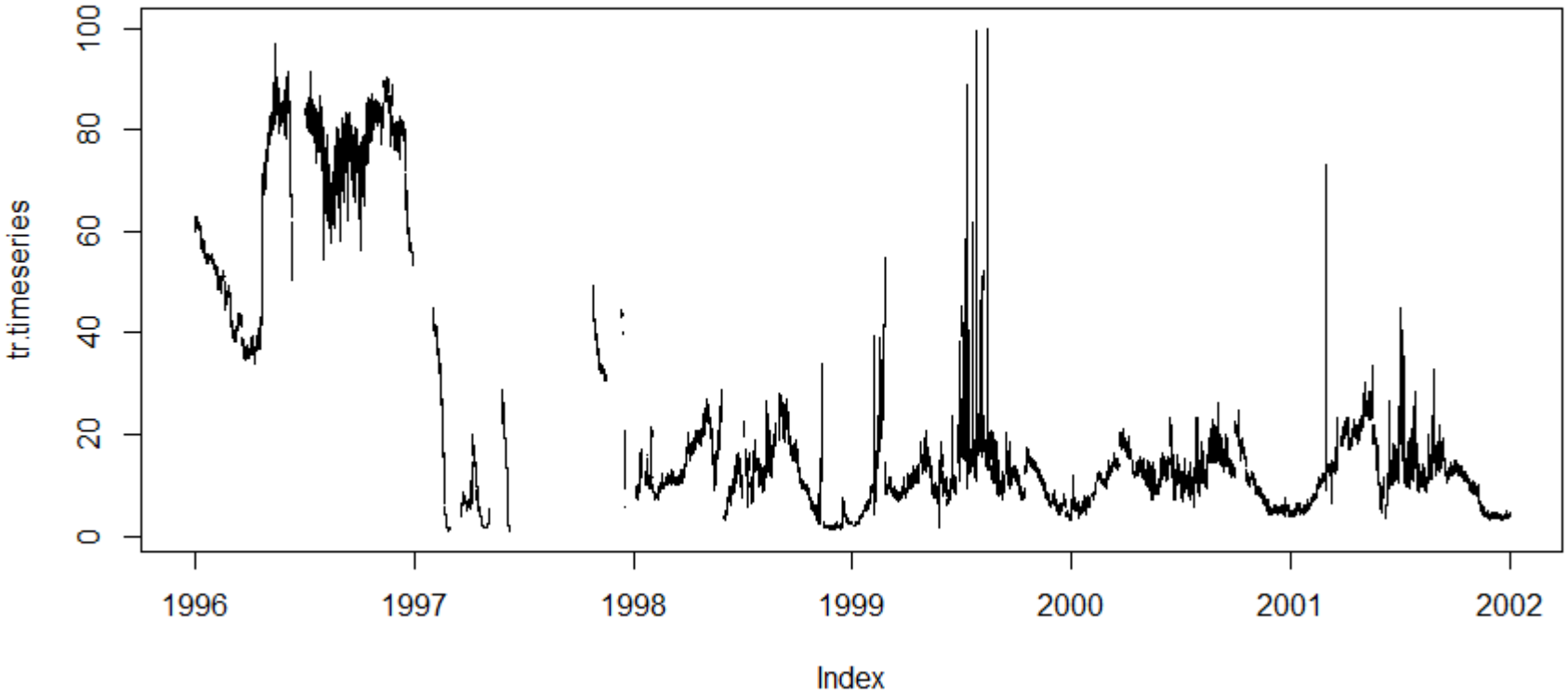
Water Quality Data

First Analysis: Metadata 5

```
turbidity : values 305117 NaN: 50321 16.5%
                      min    1.00 [%]
                      max   99.97
                      mean  22.61
                      sd    23.65
                      year    1996   1997   1998   1999   2000   2001
                      NA      5115  29966   5411   3777   3430   2622
                      NA %   10.55  70.44  13.58   7.47   7.46   5.74
                      min     6.80   1.00   1.30   1.70   3.31   3.11
                      max    97.10  49.40  33.80  99.97  26.30  99.47
                      mean   67.80  17.97  11.86  12.22  11.47  12.17
                      sd     16.66  14.83   6.42   7.73   4.42   6.46
chlorophyll-a total : values 284439 NaN: 35832 12.6%
                      min    0.00 [mirco g/l]
                      max  212.80
                      mean  44.76
                      sd    33.36
                      year    1996   1997   1998   1999   2000   2001
                      NA      7961   9790   4655    863   6783   5780
                      NA %   30.21  39.72  14.88   5.19  19.21  17.78
                      min     0.20   0.00   0.50   0.30   0.84   0.82
                      max   199.90  98.70 212.80 150.40 169.04 199.94
                      mean   40.63  28.30  46.18  43.10  53.75  51.95
                      sd     27.95  18.25  42.01  26.10  34.34  37.08
```

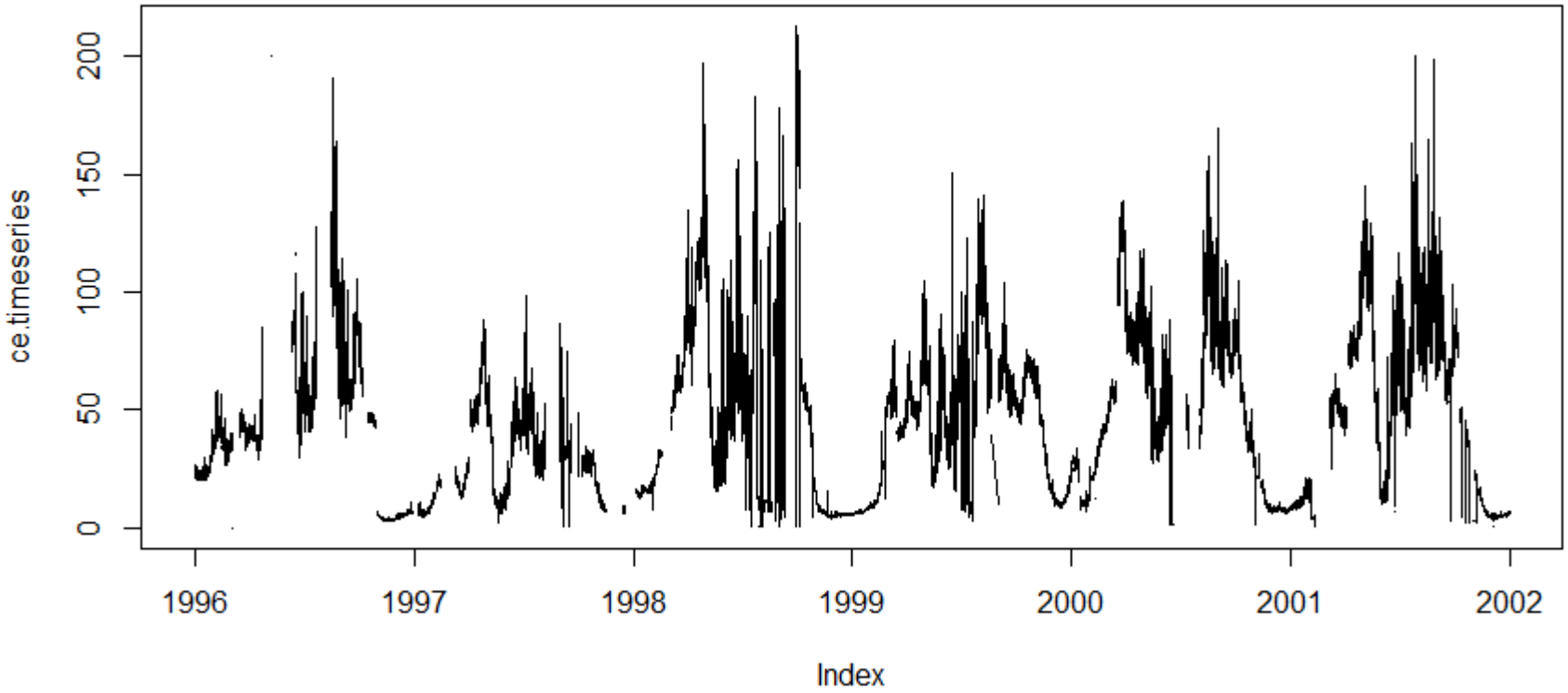
Water Quality Data

Raw Data Plot – Turbidity [%]



Water Quality Data

Raw Data Plot – Chlorophyll-a [µg/l]



Water Quality Data

Metadata and Raw Data Plot

- “some” irregularities concerning values
- a lot of value gaps -> NA values
- time gaps: parts of the time series are missing

-> regular time series

-> filtering of outliers

-> filling value gaps

-> identifying relevant time windows (e.g. 1m, 1y, season)

-> identifying relevant time scales (e.g. 10 min, 1h, 1d, 1m)

Packages in R

Zoo Package

- S3 Infrastructure for Regular and Irregular Time Series (**Z**'s **O**rdered **O**bservations)

-> window method to extract subsets by time

-> convert irregular to regular time series

-> na.**** methods to fill NA value gaps

-> aggregate method

Extracting Subsets by time

Zoo: window() function

- structure

```
window(x,start=NULL,end=NULL,frequency=NULL,deltat=NULL,extend=FALSE, ...)
```

- application examples:

```
# time series water temperature year 2001
wt_2001=window(tw@timeseries,
               start=as.POSIXct("2001-01-01 00:00:00","UTC"),
               end=as.POSIXct("2001-12-31 23:59:59","UTC"))

# time series water temperature month October 2001
wt_oct2001=window(tw@timeseries,start=as.POSIXct("2001-10-01
00:00:00","UTC"),end=as.POSIXct("2001-10-31 23:59:59","UTC"))
```

Irregular to Regular Time Series

hydroTSM: izoo2rzoo() function

- structure

```
izoo2rzoo(x,from=start(x),to=end(x),date.fmt= "%Y-%m-%d",tstep="days", ...)
```

- application examples:

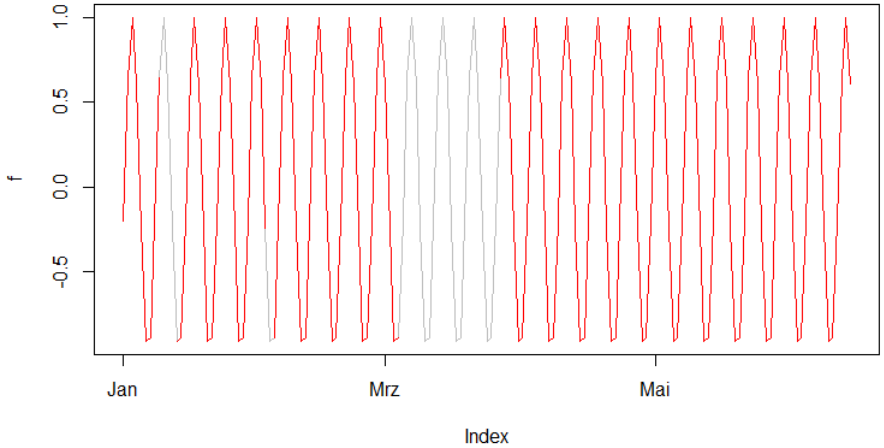
```
# convert time series with gaps to regular time series
# example water temperature year 2001
# keep date format and 10 min time step
# time gaps will be filled with NA values
wt_2001_600 = izoo2rzoo(wt_2001,date.fmt="%Y-%m-%d %H:%M:%S",
                                tstep="10 min")
```

Filling gaps of NA Values

Example 1

```
t = as.Date("2015-01-01") + c(1:165) - 1 # 165 days of a year
f = zoo(sin(as.numeric(t)*6.2831/7.),t)  # sin funtion one week
f_org=f                  # keep the original
plot(f_org,type="l",col="grey")
f[34]=NA                 # one value
for(i in 10:12)f[i]=NA # few values
for(i in 64:85)f[i]=NA # several values
lines(f,col="red")

error <- function(f_fill)
{ error = 0.
  for(i in 1: length(f_org))
  { error = error +
    abs(as.numeric(f_org[i])-
        as.numeric(f_fill[i]))}
 print(paste("Error:",error))
}
```

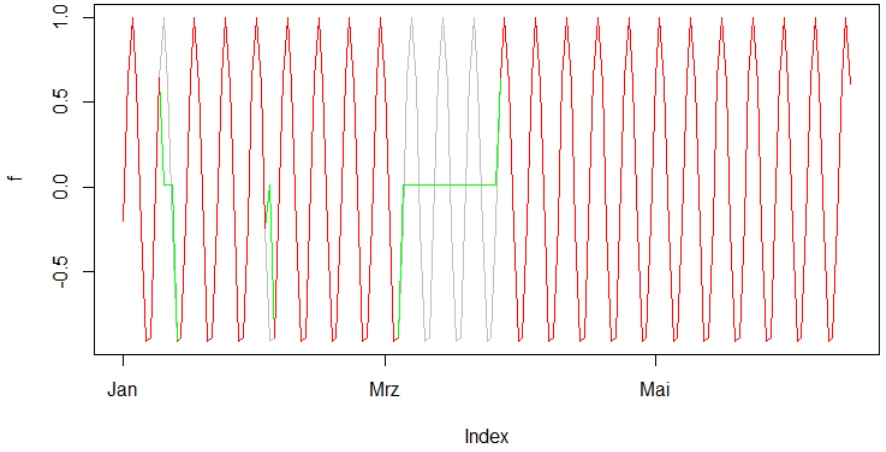


Filling gaps of NA Values

Example na.aggregate

```
f2 = na.aggregate(f)
plot(f_org,type="l",col="grey")
lines(f2,col="green")
lines(f,col="red")
error(f2)
```

Error: 16.4910173114068

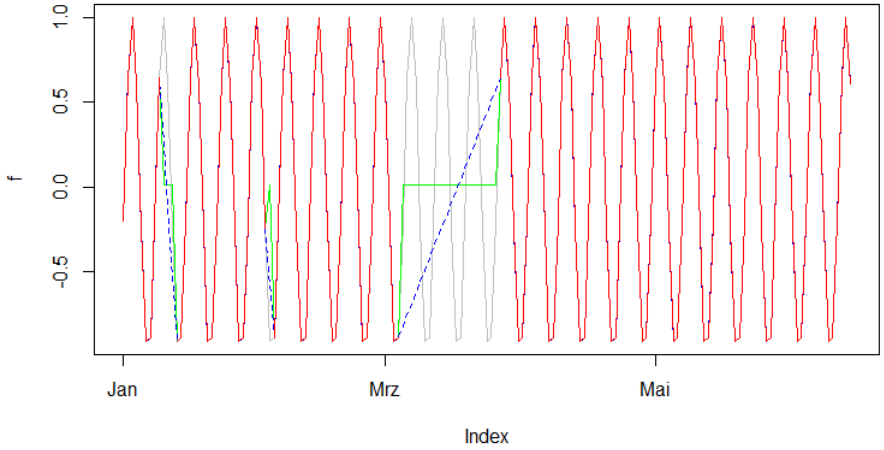


Filling gaps of NA Values

Example na.fill

```
f3 = na.fill(f,"extend")
lines(f3,col="blue")
lines(f,col="red")
error(f3)
```

Error: 19.2722790176197

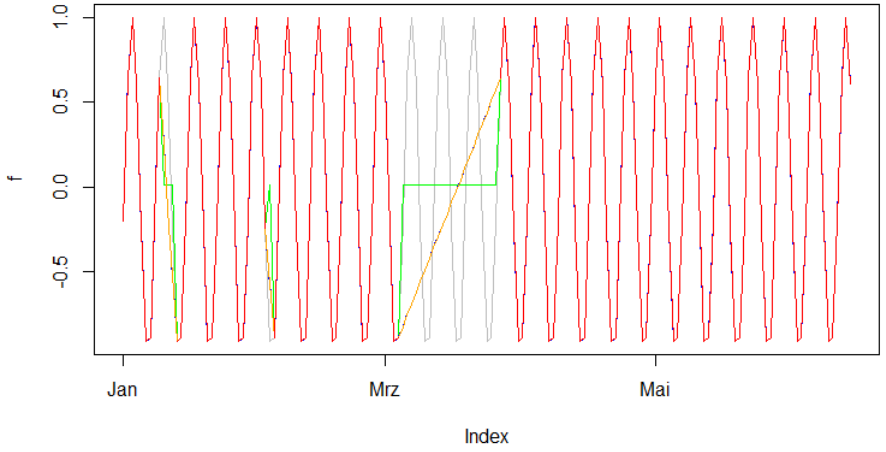


Filling gaps of NA Values

Example na.approx

```
f4 = na.approx(f)
lines(f4,col="orange")
lines(f,col="red")
error(f4)
```

Error: 19.2722790176197

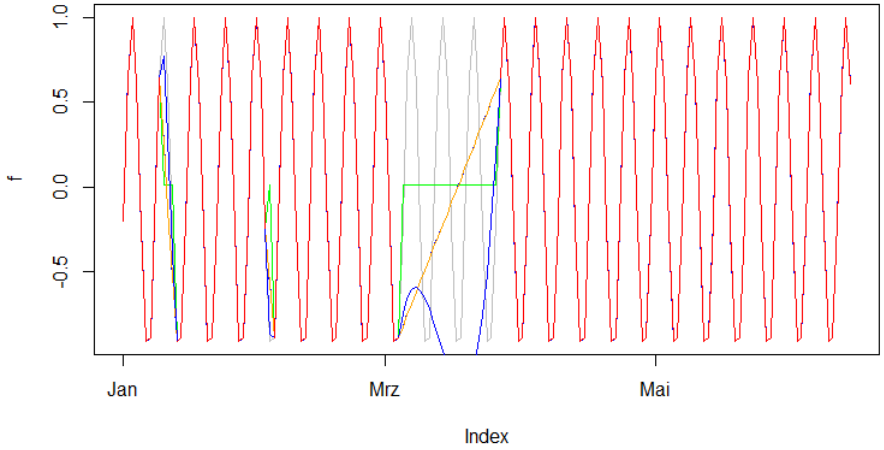


Filling gaps of NA Values

Example na.spline

```
f5 = na.spline(f)
lines(f5,col="blue")
lines(f,col="red")
error(f5)
```

Error: 21.6947642720613

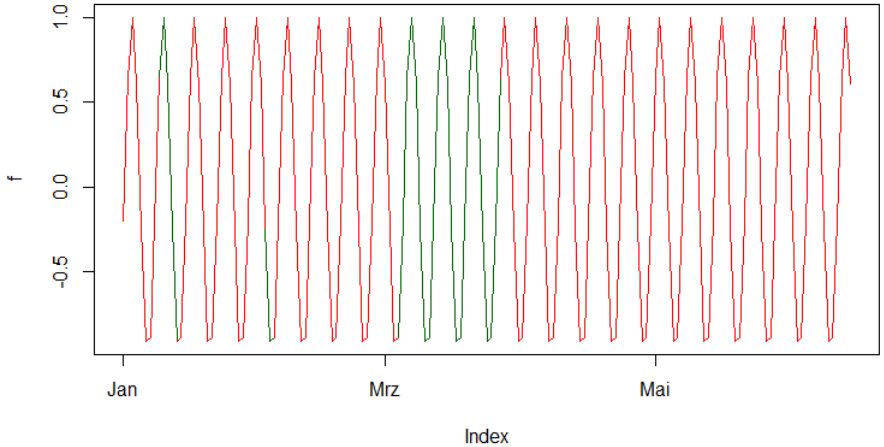


Filling gaps of NA Values

Example na.aggregate by weekday

```
as.index <- function(t) as.numeric(format(t,"%u")) # aggregate by weekday 1-7
aggregate(f_org,as.index,FUN=mean)
 1          2          3          4          5          6          7
-0.2449911 -0.9107497 -0.8907126 -0.1999259  0.6413859  0.9997332  0.6052801
f8 = na.aggregate(f,as.index,FUN=mean)
plot(f_org,type="l",col="grey")
lines(f8,col="green")
lines(f,col="red")
error(f8)
```

Error: 0.00390736391651286



Aggregate Values of Time Series

Zoo: aggregate() function

- structure

```
aggregate(x, by, FUN = sum, ..., regular = NULL, frequency = NULL)
```

- application examples:

```
as.yearhour<-function(t){return(as.numeric(format(t,"%j"))*24+
                                as.numeric(format(t,"%H")))}
as.hour <- function(t) as.numeric(format(t,"%H"))
as.yearweek <- function(t) as.numeric(format(t,"%V"))
as.week <- function(t) as.numeric(format(t,"%u"))
tyh = aggregate(wt_2001,as.yearhour,FUN=mean)
tyd = aggregate(wt_2001,as.Date(time(wt_2001_final)),FUN=mean)
tyw = aggregate(wt_2001,as.yearweek,FUN=mean)
tym = aggregate(wt_2001,as.yearmon(time(wt_2001_final)),FUN=mean)
tw = aggregate(wt_2001,as.week,FUN=mean)
tq = aggregate(wt_2001,as.yearqtr,FUN=mean)
plot(tyh) # aggregate by hour  -> hourly values
plot(tyd) # aggregate by day   -> daily values
plot(tyw) # aggregate by week  -> weekly values
plot(tym) # aggregate by month -> monthly values
plot(tw)  # aggregate by weekday -> average of the weekdays
plot(tq)  # aggregate by quarters -> average of the four quarters
```

Data Analysis Strategy

Part 1: Example Water Temperature Year 2001

- extract for 2001 -> window()
- transform to regular time series -> izoo2rzoo()
- filter outliers -> specific range filter function
  -> October and December
- fill NA values (if possible) -> na.approx()
- aggregate to other time steps (1h, 1d, 1m, …)

…

Data Analysis Strategy

Part 2: comparison with other state variables

- comparison of data in 2000 and 2001
- homogenization of the time series
  - eliminate duplicated values
  - filter outliers
  - transform to regular time series
  - fill NA values

  time gaps: parts of the time series are missing
- daily mean values -> eliminate night/day effects
- mean values per hour -> night/day effects
- correlation and scatter plots
- seasonal analysis (spring, summer, autumn, winter)

Repetition Harmonization

Water Temperature 2001

```
wt_pre_processing <- function(year)
 {
# extract year
  wt = window(tw@timeseries,
         start = as.POSIXct(sprintf("%4d-01-01 00:00:00",year),"UTC"),
         end   = as.POSIXct(sprintf("%4d-12-31 23:59:59",year),"UTC"))
# transform to regular time series zoo object
  wt_600 = izoo2rzoo(wt,date.fmt="%Y-%m-%d %H:%M:%S",tstep="10 min")
# set NA values for outliers in October and December in 2001
 if(year==2001)
  { wt=filterTimeWindow(wt,"2001-10-12 00:00:00 UTC",
                           "2001-10-13 00:00:00 UTC",10.,50.)
    wt=filterTimeWindow(wt,"2001-11-01 00:00:00 UTC",
                           "2001-12-31 00:00:00 UTC",0.,32.)
  }
# filling NA values
  wt = na.approx(wt_600)
  return(wt)
 }
wt_2000=wt_pre_processing(2000)
```

Repetition Harmonization

Water Temperature 2001

```
# filter values in a time window for min max
filterTimeWindow <- function(time_series,start,end,minimum,maximum)
{
# start and end index
 is = which(time(time_series)==start)
 ie = which(time(time_series)==end)
# loop on values
 for(it in is:ie)
  {
   value = coredata(time_series)[it]
# check values
   if(is.na(value))next;
   if(value >= maximum) {coredata(time_series)[it] <- NA}
   if(value <= minimum) {coredata(time_series)[it] <- NA}
  }
  return(time_series)
}
```

Harmonization of Time Series

2000 and 2001

- preprocessing: harmonization for 2000 and 2001

```
wt_2000=wt_pre_processing(2000)
wt_2001=wt_pre_processing(2001)
at_2000=at_pre_processing(2000)
at_2001=at_pre_processing(2001)
o2_2000=o2_pre_processing(2000)
o2_2001=o2_pre_processing(2001)
…
```

- rbind method for zoo objects

```
   merge two zoo objects by common indexes (times
### two years 2000 and 2001 in one time series
wt2 = rbind(wt_2000,wt_2001)
at2 = rbind(at_2000,at_2001)
o22 = rbind(o2_2000,o2_2001)
```

Time Series Analysis

Seasonal Extraction

- extract method from the hydroTSM package

```
extract(x, trgt, ...)
```

parameter trgt (string)

```
DJF : December, January, February
MAM : March, April, May
JJA : June, July, August
SON : September, October, November
DJFM: December, January, February, March
AM : April, May
JJAS: June, July, August, September
ON : October, November
```

example summer

```
wt_summer = extract(wt_2001,trgt="JJA")
o2_summer = extract(o2_2001,trgt="JJA")
```

Time Series Analysis

Correlation State Variables

- correlation function (Pearson correlation coefficient -> linear relationship)

```
cor(x, y = NULL, use = "everything",…)
```

- application

```
# correlation table
correlation_table <- function(ts_array,ts_name)
{ number = length(ts_array)
  row="    "
  for(i in 1:number) row = paste(row,sprintf("%6s",ts_name[i]))
  print(row)
  for(i in 1:number)
  {row=""
   row = paste(row,sprintf("%6s",ts_name[i]))
   for(j in 1:number)
    { row = paste(row,sprintf("%6.3f",
      cor(ts_array[[i]],ts_array[[j]],use="na.or.complete"))) }
   print(row)
} }
ts_n = list("at","wt","gr","o2","ph","ch","tu","co","uv")
ts   = list(at2,wt2,gr2,o22,ph2,ch2,tu2,co2,uv2)
correlation_table(ts,ts_n)
```

Time Series Analysis

Correlation State Variables

- correlation table: 2 years time window 10 min values

```
[1] "          at     wt     gr     o2     ph     ch     tu     co     uv"
[1] "     at  1.000  0.913  0.407 -0.192  0.472  0.546  0.452  0.278  0.436"
[1] "     wt  0.913  1.000  0.318 -0.336  0.452  0.551  0.451  0.338  0.434"
[1] "     gr  0.407  0.318  1.000  0.003  0.257  0.228  0.214  0.107  0.193"
[1] "     o2 -0.192 -0.336  0.003  1.000  0.481  0.134  0.125 -0.222  0.037"
[1] "     ph  0.472  0.452  0.257  0.481  1.000  0.534  0.463  0.212  0.422"
[1] "     ch  0.546  0.551  0.228  0.134  0.534  1.000  0.789 -0.206  0.402"
[1] "     tu  0.452  0.451  0.214  0.125  0.463  0.789  1.000 -0.122  0.359"
[1] "     co  0.278  0.338  0.107 -0.222  0.212 -0.206 -0.122  1.000 -0.015"
[1] "     uv  0.436  0.434  0.193  0.037  0.422  0.402  0.359 -0.015  1.000"
```

Time Series Analysis

Scatter Plots

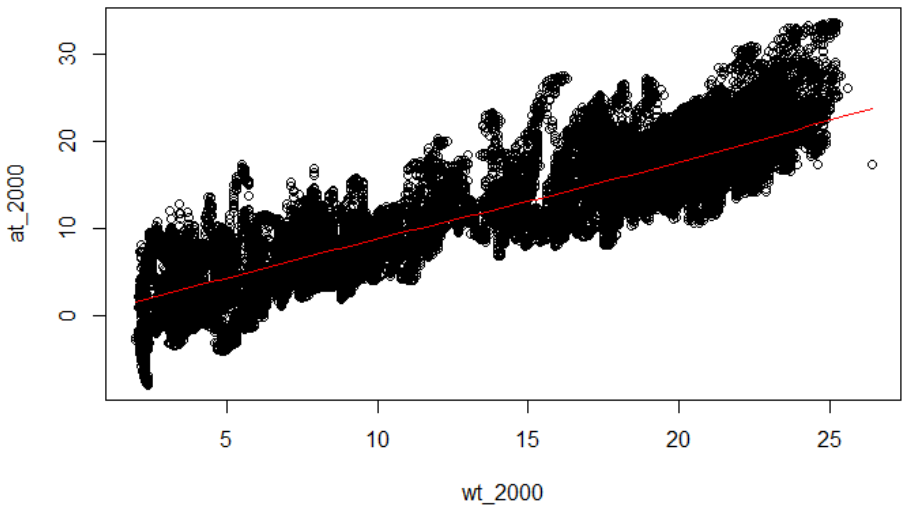
- plotting two data sets in one diagram
- one data set T1 for the vertical axis
- one data set T2 for the horizontal axis
- data points T1/T2 for the same time
- in R:

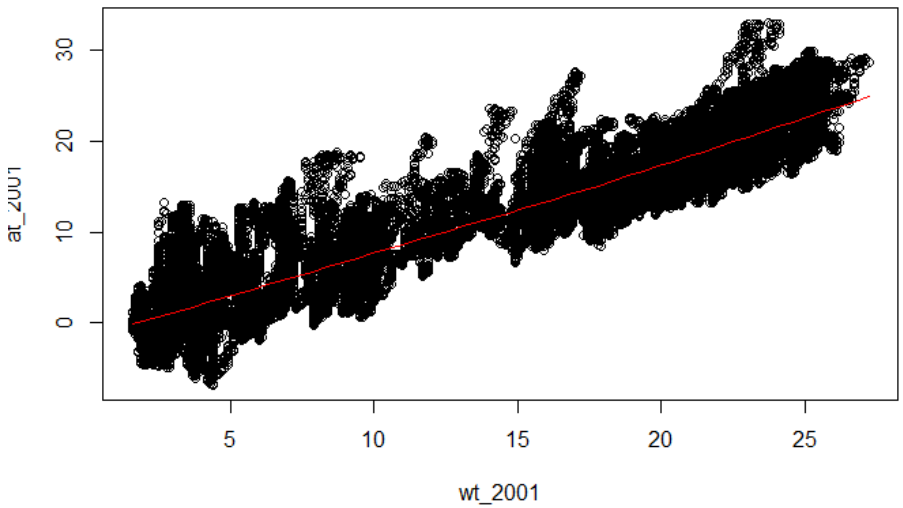
```
scatter.smooth(x, y = NULL, span = 2/3, degree = 1,
family = c("symmetric", "gaussian"),  xlab = NULL, ylab = NULL,
ylim=range(y,pred$y,na.rm=TRUE),evaluation=50,...,lpars=list())
```

Time Series Analysis

Scatter Plot Example

- water temperature vs. air temperature





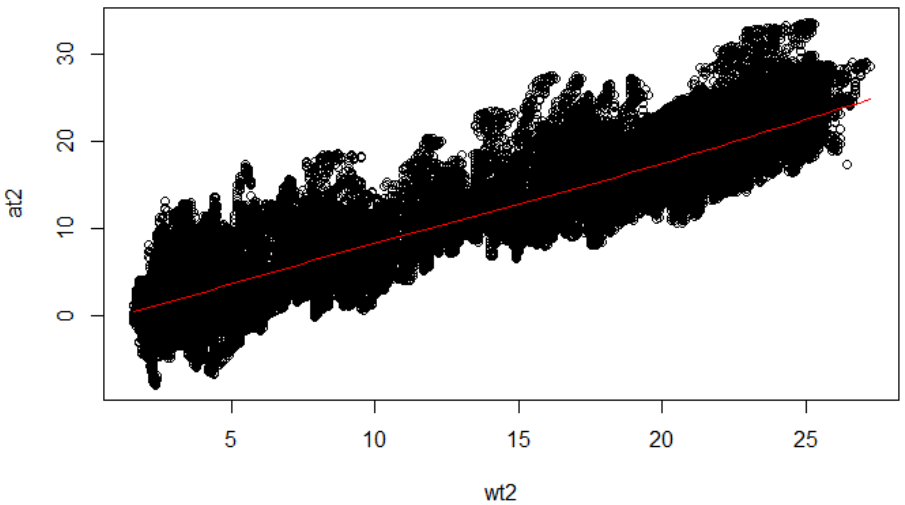
- red line:

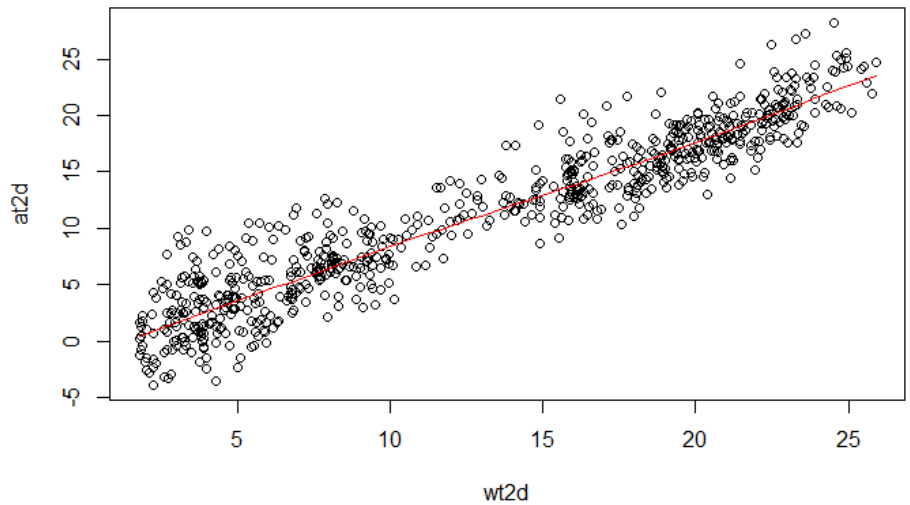
```
lines(loess.smooth(at****,wt****,span=2/3),col="red")
```

Time Series Analysis

Scatter Plot Example

- water temperature vs. air temperature





- 2000-2001: 10 min and 1 day time step

Time Series Analysis

LOESS: Local Polynomial Regression Fitting

- fits a polynomial surface determined by one or more numerical predictors, using local
- fitting based on least squares:

  minimum of the sum of squared residuals
- important parameter:

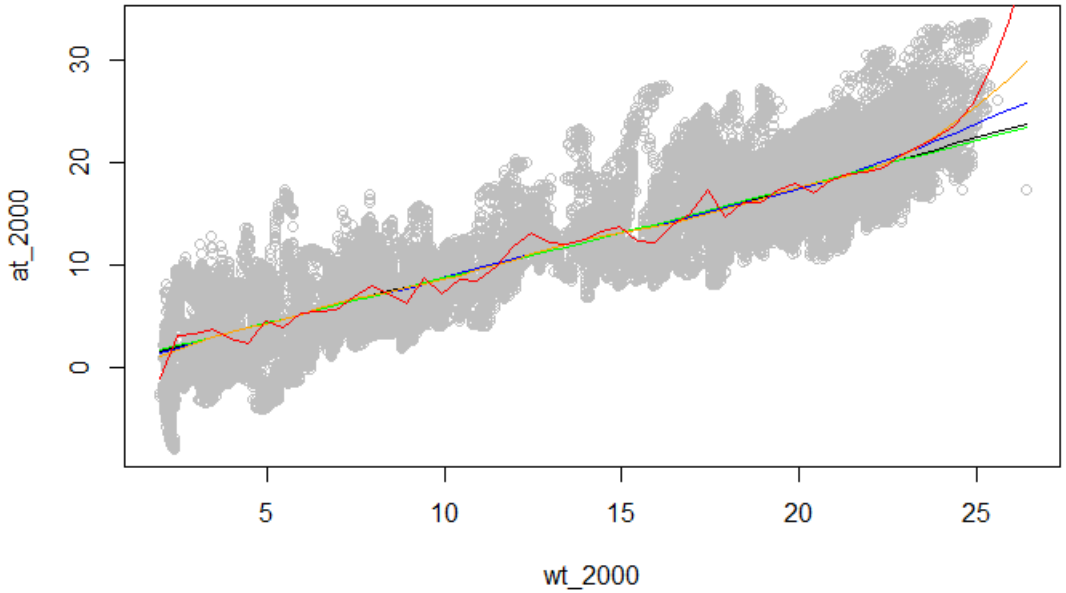
```
degree      the degree of the polynomials to be used (1 or 2)
span        proportions of points to be considered
            (> 1 all points are used)
```

Time Series Analysis

LOESS Example

- water temperature / air temperature 2000

```
scatter.smooth(wt_2000,at_2000,col="grey")
lines(loess.smooth(wt_2000,at_2000,span=1/3),col="blue")
lines(loess.smooth(wt_2000,at_2000,span=1),col="green")
lines(loess.smooth(wt_2000,at_2000,span=1/3,degree=2),col="orange")
lines(loess.smooth(wt_2000,at_2000,span=1/30,degree=2),col="red")
```



Time Series Analysis

Water Temperature vs. Air Temperature

- scatter plot -> linear relationship (all times)
- correlation test

```
cor(wt_2000,at_2001)      -> 0.9046793
cor(wt_2001,at_2001)      -> 0.9219079
cor(wt,at)                -> 0.9132889
cor(wt2d,at2d)            -> 0.9425983
```

- linear regression

```
lm(wt_2000 ~ at_2000)     -> 2.5890        0.9011
lm(wt_2001 ~ at_2001)     -> 3.2666        0.8913
lm(wt2 ~ at2)             -> 2.9709        0.8926
lm(wt2d ~ at2d)           -> 2.2861        0.9533
                             intersect     slope
```

-> linear relationship
water temperature vs. air temperature

Time Series Analysis

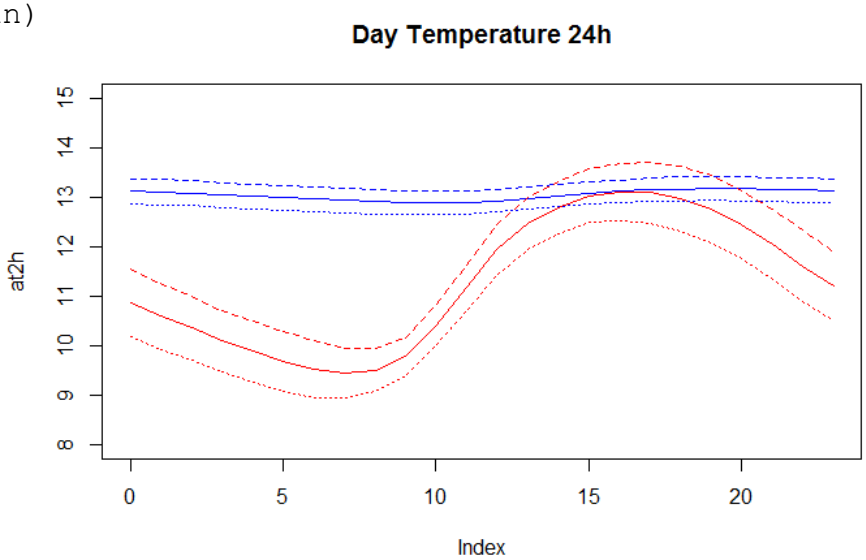
Water Temperature vs. Air Temperature

- one day (24 hours) analysis:

```
as.hour <- function(t) as.numeric(format(t,"%H"))
wt2h = aggregate(wt2,as.hour,FUN=mean)
at2h = aggregate(at2,as.hour,FUN=mean)
wt_2000h = aggregate(wt_2000,as.hour,FUN=mean)
at_2000h = aggregate(at_2000,as.hour,FUN=mean)
wt_2001h = aggregate(wt_2001,as.hour,FUN=mean)
at_2001h = aggregate(at_2001,as.hour,FUN=mean)

plot(at2h,main="Day Temperature 24h",
          col="red",ylim=c(8,15))
lines(wt2h,col="blue")
lines(at_2000h,col="red",lty="dashed")
lines(wt_2000h,col="blue",lty="dashed")
lines(at_2001h,col="red",lty="dotted")
lines(wt_2001h,col="blue",lty="dotted")
```

-> discussion, interpretation

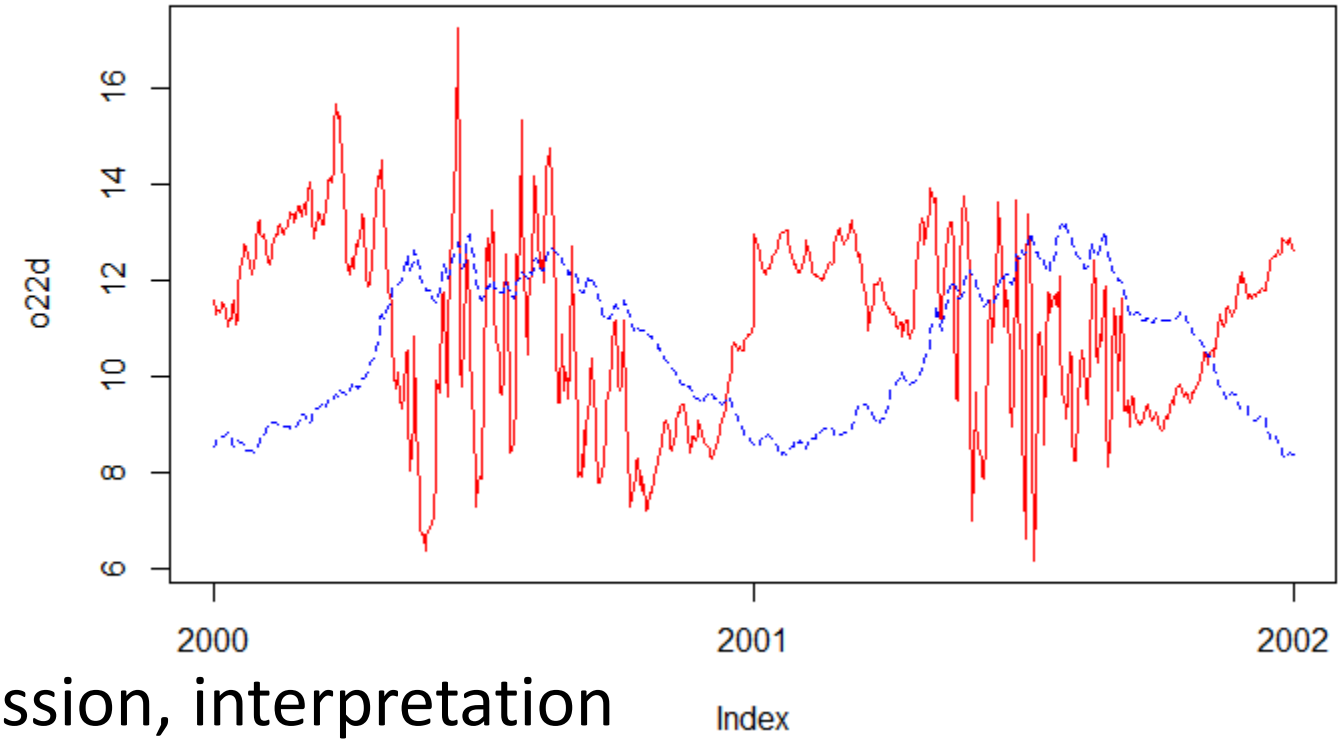


Time Series Analysis

Water Temperature vs. Oxygen

```
plot(o22d,col= "red")

lines(wt2d*0.2+8,col="blue",lty="dashed")
```



-> discussion, interpretation

Time Series Analysis

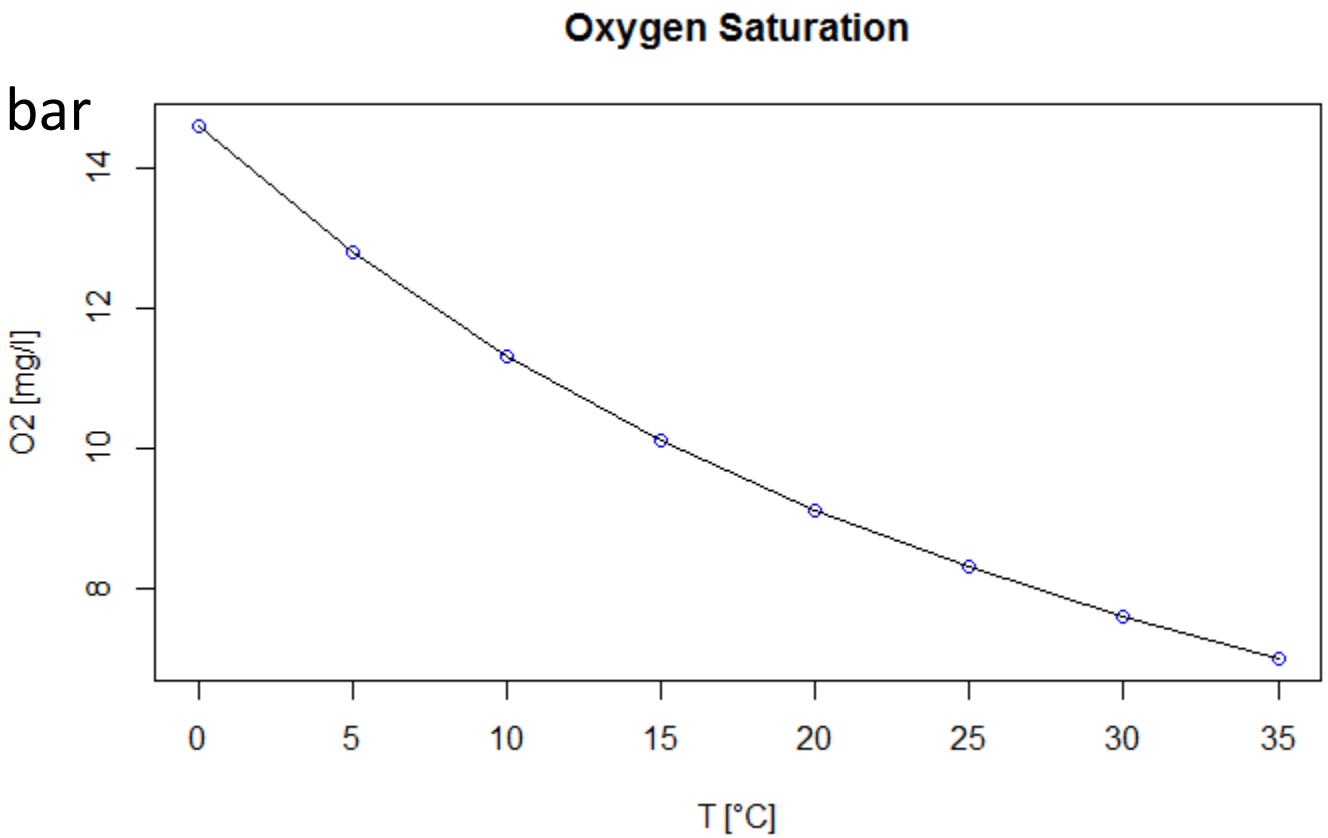
Water Temperature vs. Oxygen

- Oxygen Solubility/Saturation in Fresh Water

  no salinity

  air pressure: 1 bar

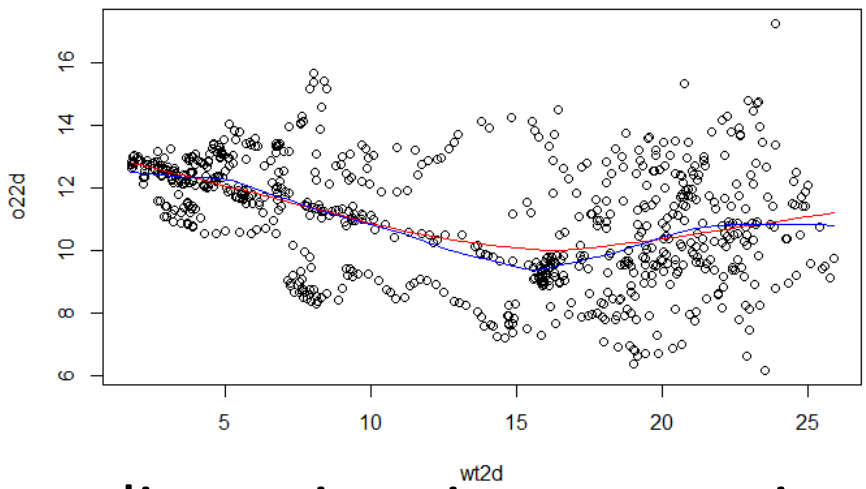
T[°C]	S[mg/l]
0	14.6
5	12.8
10	11.3
15	10.1
20	9.1
25	8.3
30	7.6
35	7.0

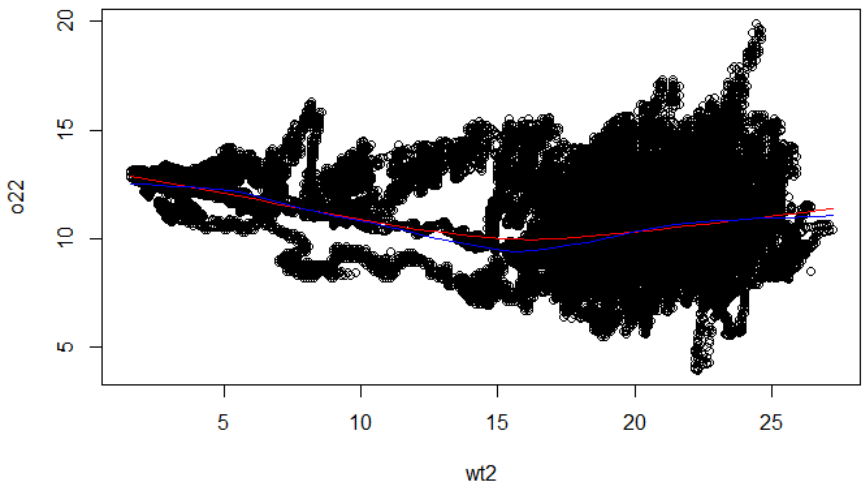


Time Series Analysis

Water Temperature vs. Oxygen

- ```
  scatter.smooth(wt2d,o22d)
  lines(loess.smooth(wt2d,o22d,span=2/3),col="red")
  lines(loess.smooth(wt2d,o22d,span=1/3),col="blue")
  scatter.smooth(wt2,o22)
  lines(loess.smooth(wt2,o22,span=2/3),col="red")
  lines(loess.smooth(wt2,o22,span=1/3),col="blue")
  ```





-> discussion, interpretation
  ```

Time Series Analysis

Water Temperature vs. Oxygen

- correlation test

```
cor(wt_2000,o2_2001)      -> -0.224
cor(wt_2001,o2_2001)      -> -0.492
cor(wt,o2)                -> -0.336
cor(wt2d,o22d)            -> -0.363
```

-> seasonal analysis

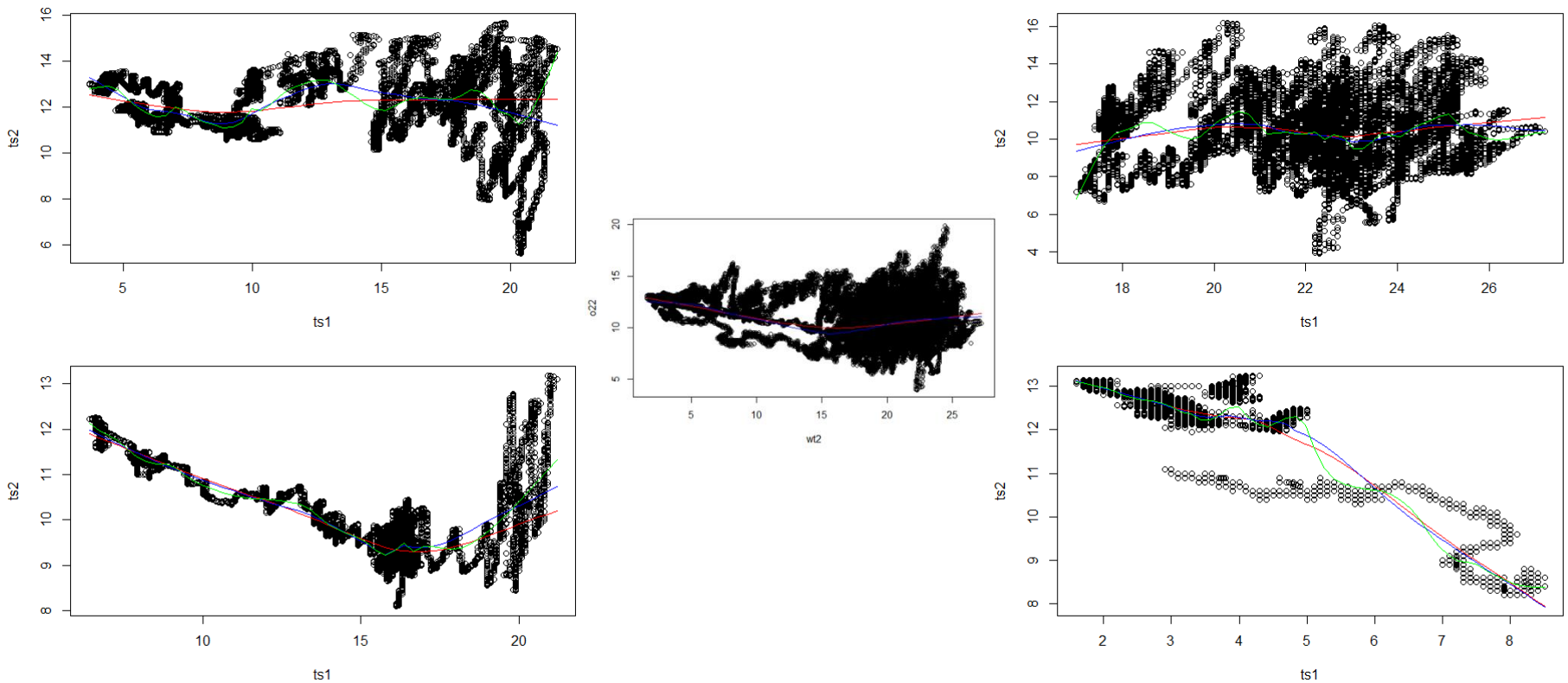
```
cor(wt_spring,o2_spring) -> -0.095
cor(wt_summer,o2_summer) ->  0.059
cor(wt_autumn,o2_autumn) -> -0.695
cor(wt_winter,o2_winter) -> -0.904
```

-> discussion, interpretation

Time Series Analysis

Water Temperature vs. Oxygen

- seasonal scatter plots



Time Series Analysis

Correlation State Variables

- correlation table: 2 years time window 10 min values

```
[1] "          at     wt     gr     o2     ph     ch     tu     co     uv"
[1] "     at  1.000  0.913  0.407 -0.192  0.472  0.546  0.452  0.278  0.436"
[1] "     wt  0.913  1.000  0.318 -0.336  0.452  0.551  0.451  0.338  0.434"
[1] "     gr  0.407  0.318  1.000  0.003  0.257  0.228  0.214  0.107  0.193"
[1] "     o2 -0.192 -0.336  0.003  1.000  0.481  0.134  0.125 -0.222  0.037"
[1] "     ph  0.472  0.452  0.257  0.481  1.000  0.534  0.463  0.212  0.422"
[1] "     ch  0.546  0.551  0.228  0.134  0.534  1.000  0.789 -0.206  0.402"
[1] "     tu  0.452  0.451  0.214  0.125  0.463  0.789  1.000 -0.122  0.359"
[1] "     co  0.278  0.338  0.107 -0.222  0.212 -0.206 -0.122  1.000 -0.015"
[1] "     uv  0.436  0.434  0.193  0.037  0.422  0.402  0.359 -0.015  1.000"
```

Time Series Analysis

Correlation State Variables

- correlation table: 2 years time window, daily values

```
[1] "          at     wt     gr     o2     ph     ch     tu     co     uv"
[1] "     at  1.000  0.943  0.723 -0.255  0.478  0.568  0.471  0.289  0.458"
[1] "     wt  0.943  1.000  0.698 -0.363  0.458  0.557  0.460  0.339  0.439"
[1] "     gr  0.723  0.698  1.000  0.040  0.594  0.524  0.494  0.220  0.442"
[1] "     o2 -0.255 -0.363  0.040  1.000  0.466  0.121  0.103 -0.228  0.036"
[1] "     ph  0.478  0.458  0.594  0.466  1.000  0.540  0.468  0.219  0.432"
[1] "     ch  0.568  0.557  0.524  0.121  0.540  1.000  0.799 -0.208  0.410"
[1] "     tu  0.471  0.460  0.494  0.103  0.468  0.799  1.000 -0.123  0.369"
[1] "     co  0.289  0.339  0.220 -0.228  0.219 -0.208 -0.123  1.000 -0.017"
[1] "     uv  0.458  0.439  0.442  0.036  0.432  0.410  0.369 -0.017  1.000"
```

Time Series Analysis

Correlation State Variables

- correlation table: summer 2001, 10 min values

```
[1] "          at     wt     gr     o2     ph     ch     tu     co     uv"
[1] "     at  1.000  0.771  0.344  0.394  0.050  0.689  0.490 -0.000 -0.074"
[1] "     wt  0.771  1.000  0.062  0.059 -0.162  0.803  0.517  0.171 -0.368"
[1] "     gr  0.344  0.062  1.000  0.107  0.057  0.080  0.087  0.037  0.049"
[1] "     o2  0.394  0.059  0.107  1.000  0.707  0.251  0.242 -0.366  0.452"
[1] "     ph  0.050 -0.162  0.057  0.707  1.000  0.091  0.093 -0.457  0.381"
[1] "     ch  0.689  0.803  0.080  0.251  0.091  1.000  0.516 -0.099 -0.065"
[1] "     tu  0.490  0.517  0.087  0.242  0.093  0.516  1.000 -0.111 -0.011"
[1] "     co -0.000  0.171  0.037 -0.366 -0.457 -0.099 -0.111  1.000 -0.376"
[1] "     uv -0.074 -0.368  0.049  0.452  0.381 -0.065 -0.011 -0.376  1.000"
```

Time Series Analysis

Correlation State Variables

- correlation table: winter 2001, 10 min values

```
[1] "          at     wt     gr     o2     ph     ch     tu     co     uv"
[1] "      at  1.000  0.543  0.161 -0.458 -0.332 -0.238 -0.010  0.351  0.067"
[1] "      wt  0.543  1.000 -0.019 -0.904 -0.854 -0.175 -0.049  0.238  0.399"
[1] "      gr  0.161 -0.019  1.000  0.079  0.005  0.210  0.221 -0.180 -0.100"
[1] "      o2 -0.458 -0.904  0.079  1.000  0.881  0.417  0.331 -0.360 -0.448"
[1] "      ph -0.332 -0.854  0.005  0.881  1.000  0.159  0.035 -0.140 -0.442"
[1] "      ch -0.238 -0.175  0.210  0.417  0.159  1.000  0.737 -0.614 -0.242"
[1] "      tu -0.010 -0.049  0.221  0.331  0.035  0.737  1.000 -0.433 -0.381"
[1] "      co  0.351  0.238 -0.180 -0.360 -0.140 -0.614 -0.433  1.000  0.175"
[1] "      uv  0.067  0.399 -0.100 -0.448 -0.442 -0.242 -0.381  0.175  1.000"
```

Data Analysis Strategy

Conclusions

- investigations should be continued

- other effects/correlations:
  measurement set-up changes ?
  rainfall/runoff events with storm water inflow ?
  “upstream” events/impact ?
  human impact ?