NERRS / SWMP
Training Workshop: *R, SWMPr, SWMPrats*
Williamsburg, VA, Nov 13, 2016

Time series topic 2: Decomposition

Marcus W. Beck[1]

[1]USEPA NHEERL Gulf Ecology Division
Email: beck.marcus@epa.gov

Objectives for the session (3:30-4:15)

- What is and why do we use time series decomposition
- Functions in SWMPr
- Application to NERRS data
    - Data prep
    - Execution
    - Interpretation

Interactive portion

Follow along as we go:

- flash drive
- online: swmprats.net 2016 workshop tab

Interactive portion

Follow along as we go:

- flash drive
- online: swmprats.net 2016 workshop tab

You will run examples whenever you see this guy:

Is everything installed?

We will use functions in the SWMPr package

Option 1, from the R Console prompt:

```r
install.packages('SWMPr')
library(SWMPr)
```

Is everything installed?

We will use functions in the SWMPr package

Option 1, from the R Console prompt:

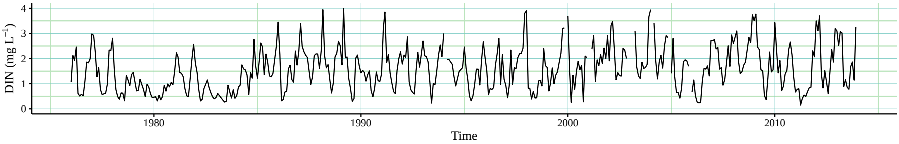
```
install.packages('SWMPr')
library(SWMPr)
```

Option 2, install the source file from the flash drive:

```
# change as needed
path_to_file <- 'C:/Users/mbeck/Desktop/SWMPr_2.2.0.tar.gz'

# install, load
install.packages(path_to_file, repos = NULL, type="source")
library(SWMPr)
```

Theory and background

Observed data represents effects of many processes



Climate

precipitation
temperature
wind events
ENSO effects

Local

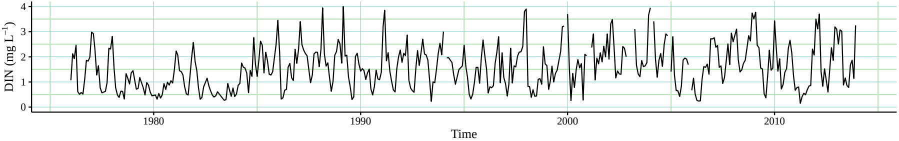
light/turbidity
residence time
invasive species
trophic effects

Regional/historical

watershed inputs
point sources
management actions
flow changes

Theory and background

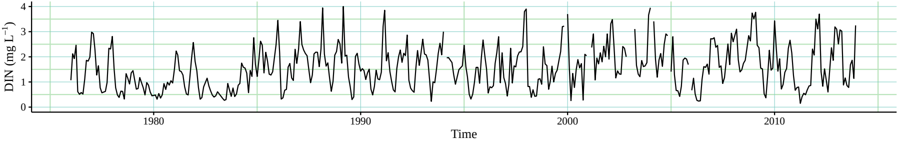
Observed data represents effects of many processes



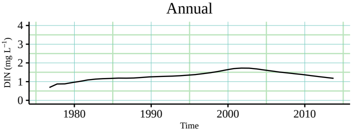
Models should describe components to evaluate effects

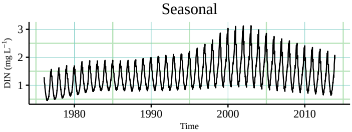
Theory and background

Observed data represents effects of many processes



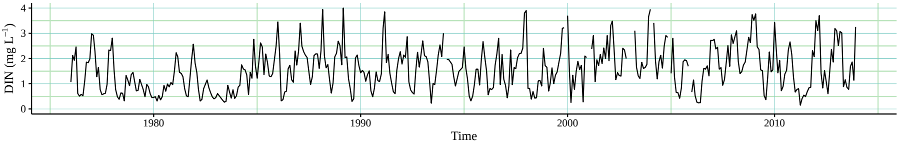
Models should describe components to evaluate effects



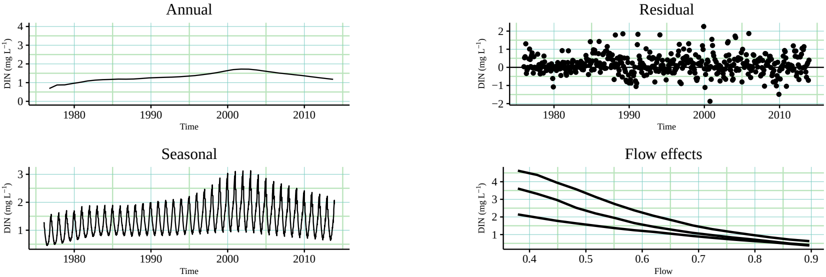


Theory and background

Observed data represents effects of many processes



Models should describe components to evaluate effects



Theory and background

- Time series decomposition is a very general topic - WRTDS is a form of decomposition

Theory and background

- Time series decomposition is a very general topic - WRTDS is a form of decomposition

- There are more generic and simpler approaches

Theory and background

- Time series decomposition is a very general topic - WRTDS is a form of decomposition
- There are more generic and simpler approaches
- Objective is to decompose a time series into individual components, isolate or otherwise remove components of interests

Theory and background

- Time series decomposition is a very general topic - WRTDS is a form of decomposition
- There are more generic and simpler approaches
- Objective is to decompose a time series into individual components, isolate or otherwise remove components of interests
- The individual components are sometimes additive or multiplicative components of the complete time series

Theory and background

- Time series decomposition is a very general topic - WRTDS is a form of decomposition
- There are more generic and simpler approaches
- Objective is to decompose a time series into individual components, isolate or otherwise remove components of interests
- The individual components are sometimes additive or multiplicative components of the complete time series
- But be warned... just because you can doesn't mean you should

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp()`

- Almost identical to the `decompose` function

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

decomp()

- Almost identical to the `decompose` function
- Works well for time series with daily or seasonal cycles

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp()`

- Almost identical to the `decompose` function
- Works well for time series with daily or seasonal cycles
- Separates components as trend, cyclical variation (e.g., annaul, daily), and random

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp()`

- Almost identical to the `decompose` function
- Works well for time series with daily or seasonal cycles
- Separates components as trend, cyclical variation (e.g., annaul, daily), and random
- Additive or multiplicative decomposition

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp()`

1. Gets trend by moving average, removes it from the time series.

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp()`

1. Gets trend by moving average, removes it from the time series.
2. Gets seasonal variation by averaging across time periods

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp()`

1. Gets trend by moving average, removes it from the time series.
2. Gets seasonal variation by averaging across time periods
3. Gets error as the remainder from the trend and seasonal components

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp_cj()`

- Based on a deprecated method in the wq package, described in [Cloern and Jassby, 2010]

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp_cj()`

- Based on a deprecated method in the wq package, described in [Cloern and Jassby, 2010]
- Works only for time series with seasonal cycles

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp_cj()`

- Based on a deprecated method in the wq package, described in [Cloern and Jassby, 2010]
- Works only for time series with seasonal cycles
- Separates components as grandmean, annual, seasonal, and events

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp_cj()`

- Based on a deprecated method in the wq package, described in [Cloern and Jassby, 2010]
- Works only for time series with seasonal cycles
- Separates components as grandmean, annual, seasonal, and events
- Additive or multiplicative decomposition

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp_cj()`

1. Takes grandmean, removes it from time series

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp_cj()`

1. Takes grandmean, removes it from time series
2. Annual trends as averages within years, removes from time series

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp_cj()`

1. Takes grandmean, removes it from time series
2. Annual trends as averages within years, removes from time series
3. Seasonal trend as averages between periods, removes from time series

Theory and background

Two very general decomposition methods are provided in SWMPr: `decomp()` and `decomp_cj`

These are not new methods, they just make it easy to decompose NERRS data - work with swmpr objects

`decomp_cj()`

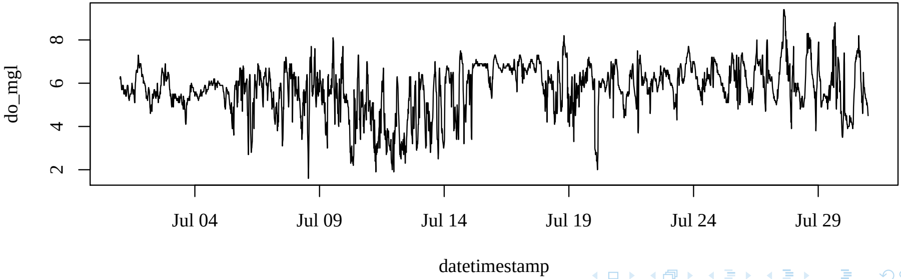
1. Takes grandmean, removes it from time series
2. Annual trends as averages within years, removes from time series
3. Seasonal trend as averages between periods, removes from time series
4. Events as remainder

Using decomp with NERRS data

Load some water quality data from Apalachicola Bay, Dry Bar station

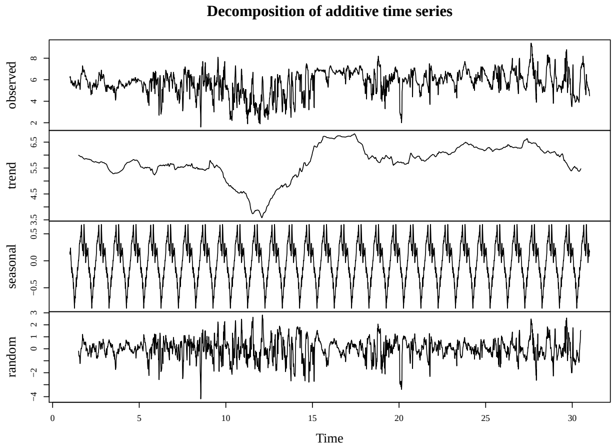
Let's look at DO variation over one month

```
# load SWMPr
library(SWMPr)

# subset for daily decomposition
dat <- subset(apadbwq, subset = c('2013-07-01 00:00', '2013-07-31 00:00'),
  select = 'do_mgl')
plot(dat)
```



Using decomp with NERRS data

```
dat_add <- decomp(dat, param = 'do_mgl', frequency = 'daily', type = 'additive')
plot(dat_add)
```



Using decomp with NERRS data

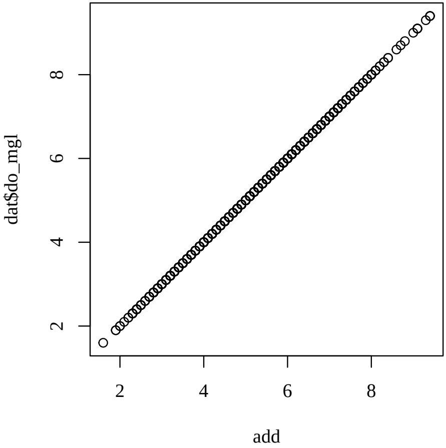
What's in the decomposed object?

```
str(dat_add)

## List of 6
##  $ x       : Time-Series [1:2881] from 1 to 31: 6.2 6.3 6.3 6.2 6 5.9 5.7 5.8 5.
##  $ seasonal: Time-Series [1:2881] from 1 to 31: 0.165 0.12 0.178 0.239 0.163 ...
##  $ trend   : Time-Series [1:2881] from 1 to 31: NA NA NA NA NA NA NA NA NA NA ..
##  $ random  : Time-Series [1:2881] from 1 to 31: NA NA NA NA NA NA NA NA NA NA ..
##  $ figure  : num [1:96] 0.165 0.12 0.178 0.239 0.163 ...
##  $ type    : chr "additive"
##  - attr(*, "class")= chr "decomposed.ts"

str(dat_add$trend)

##  Time-Series [1:2881] from 1 to 31: NA NA NA NA NA NA NA NA NA NA ...
```

Using decomp with NERRS data

What does additive mean?

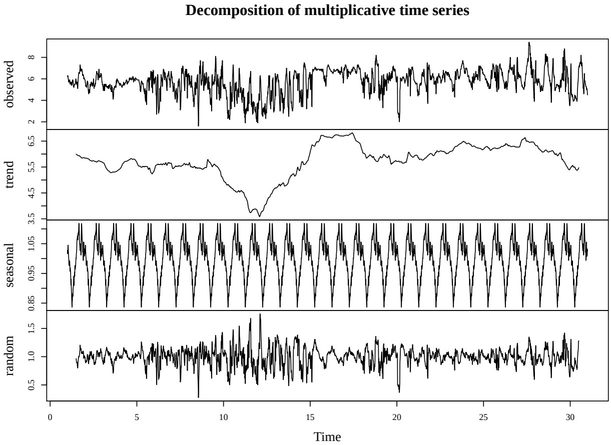
```
add <- with(dat_add, seasonal + trend + random)
plot(add, dat$do_mgl)
```



Using decomp with NERRS data

Let's try a multiplicative decomposition

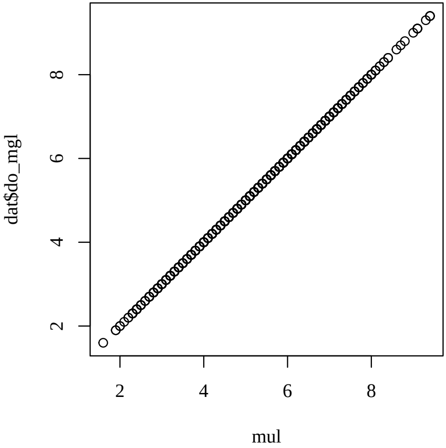
```
dat_mul <- decomp(dat, param = 'do_mgl', frequency = 'daily',
  type = 'multiplicative')
plot(dat_mul)
```



Using decomp with NERRS data

What does multiplicative mean?

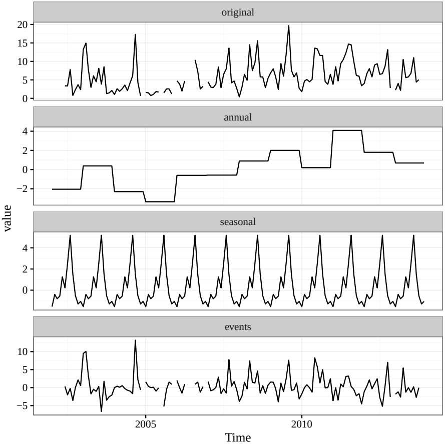
```r
mul <- with(dat_mul, seasonal * trend * random)
plot(mul, dat$do_mgl)
```



Using decomp_cj with NERRS data

Use discrete, monthly data with `decomp_cj`: Apalachicola Bay, Cat Point nutrient station

```
apacpnut <- qaqc(apacpnut, qaqc_keep = c(0, 4))
decomp_cj(apacpnut, param = 'chla_n', type = 'add')
```



Using decomp_cj with NERRS data

Note that the default behavior for `decomp_cj` is a plot, use `vals_out = TRUE` for values

```
add <- decomp_cj(apacpnut, param = 'chla_n', type = 'add', vals_out = TRUE)
head(add)


##         Time original annual  seasonal  events
## 1 2002-01-01       NA  -2.05 -1.550000      NA
## 2 2002-02-01       NA  -2.05 -0.400000      NA
## 3 2002-03-01       NA  -2.05 -0.807500      NA
## 4 2002-04-01      1.6  -2.05 -0.562500 -0.6875
## 5 2002-05-01       NA  -2.05  1.258333      NA
## 6 2002-06-01      3.4  -2.05  0.220000  0.3300
```

Using decomp_cj with NERRS data

A word of caution, `decomp_cj` uses `setstep` before decomposing

```
head(apacpnut)

##          datetimestamp  po4f  nh4f  no2f  no3f no23f chla_n
## 1 2002-04-02 11:55:00 0.004 0.027 0.002 0.048 0.050    1.8
## 2 2002-04-02 11:56:00 0.004 0.029 0.002 0.046 0.048    1.8
## 3 2002-04-30 11:15:00 0.014 0.138 0.005 0.115 0.120    1.2
## 4 2002-06-04 11:03:00 0.006 0.049 0.002 0.024 0.026    3.4
## 5 2002-07-02 09:53:00 0.014 0.083 0.002    NA 0.039    3.7
## 6 2002-07-02 09:55:00 0.017 0.093 0.002    NA 0.040    3.0

head(add)

##          Time original annual  seasonal  events
## 1 2002-01-01       NA  -2.05 -1.550000      NA
## 2 2002-02-01       NA  -2.05 -0.400000      NA
## 3 2002-03-01       NA  -2.05 -0.807500      NA
## 4 2002-04-01      1.6  -2.05 -0.562500 -0.6875
## 5 2002-05-01       NA  -2.05  1.258333      NA
## 6 2002-06-01      3.4  -2.05  0.220000  0.3300
```

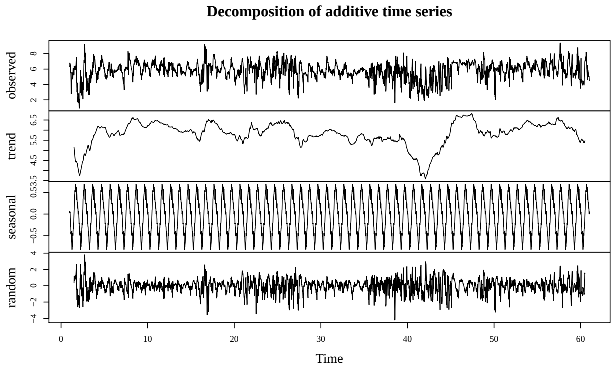
Summary

A word of caution, `decomp` does not work with missing data

```
dat <- subset(apadbwq, subset = c('2013-06-01 00:00', '2013-07-31 00:00'))

# this returns an error
test <- decomp(dat, param = 'do_mgl', frequency = 'daily')

## Error in na.omit.ts(x):  time series contains internal NAs
```

Summary

```
# use na.approx to interpolate missing data
dat <- subset(apadbwq, subset = c('2013-06-01 00:00', '2013-07-31 00:00'))
dat <- na.approx(dat, params = 'do_mgl', maxgap = 10)

# decomposition and plot
dat_fl <- decomp(dat, param = 'do_mgl', frequency = 'daily')
plot(dat_fl)
```


Decomposition of additive time series

Summary

Things to ask before decomposition:

- What is the time step? Is it regular? Does it need be standardized?

Summary

Things to ask before decomposition:

- What is the time step? Is it regular? Does it need be standardized?
- How do I deal with missing data?

Summary

Things to ask before decomposition:

- What is the time step? Is it regular? Does it need be standardized?
- How do I deal with missing data?
- Is there any expected cyclical variation? If so, what is the period (e.g., seasonal, daily)?

Summary

Things to ask before decomposition:

- What is the time step? Is it regular? Does it need be standardized?
- How do I deal with missing data?
- Is there any expected cyclical variation? If so, what is the period (e.g., seasonal, daily)?
- Is stationarity a reasonable expectation of the cyclical variation (yes = additive, no = multiplicative)?

NERRS / SWMP
Training Workshop: *R, SWMPr, SWMPrats*
Williamsburg, VA, Nov 13, 2016

Up next... Time Series Topic 3: Seasonal Kendall

Questions??

References

Cloern JE, Jassby AD. 2010.
Patterns and scales of phytoplankton variability in estuarine-coastal ecosystems.
Estuaries and Coasts, 33(2):230–241.