Exploratory Data Analysis with SWMP

Marcus W. Beck[1] Todd D. O'Brien[2]

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

[2]NOAA/NMFS COPEPOD Project
Email: todd.obrien@noaa.gov

Objectives and agenda

- Objectives
  - What are some basic time series analysis techniques and when would you use them?
  - How are the data set up, what functions are used, and how are the results interpreted?

Objectives and agenda

- Objectives
  - What are some basic time series analysis techniques and when would you use them?
  - How are the data set up, what functions are used, and how are the results interpreted?
- Agenda
  - Analysis 1 - missing data and interpolation
  - Analysis 2 - smoothing and aggregation
  - Analysis 3 - basic trend analysis

Interactive portion

You can follow along in this module:

- dataset3
- script3

Interactive!

What is exploratory data analysis (EDA)?

A general term that describes preliminary evaluation of a variable or multiple variables in a dataset to assess quantitative properties for further analysis or hypothesis generation

EDA can inform you of the types of variables (categorical, continuous), distribution of variables (central tendency, spread), correlations between variables, and presence of outliers

R has many functions available for EDA - see the R reference card and the cookbook for some ideas

For now, we will focus on some tasks that have specific relevance to SWMP

Analysis 1 - Missing data and interpolation

Time series will usually include missing data - you will have to decide how to handle missing values

Let's import some wq data

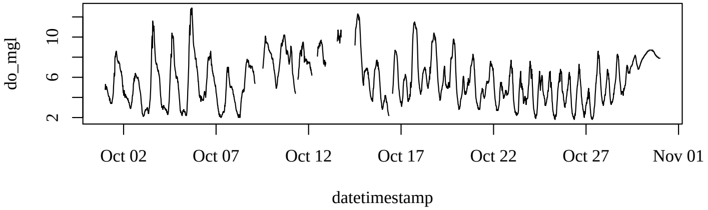
```
# import data, qaqc, and subset
# change this path for the flash drive
path <- 'C:/data/dataset3'
dat <- import_local(path, 'cbmmcwq2012')
```

```
# qaqc and subset do_mgl
dat <- qaqc(dat)
dat <- subset(dat, select = 'do_mgl')

# how many missing values?
sum(is.na(dat$do_mgl))


## [1] 419
```

Analysis 1 - Missing data and interpolation

Introducing the 'na.approx' function - this method can interpolate missing data

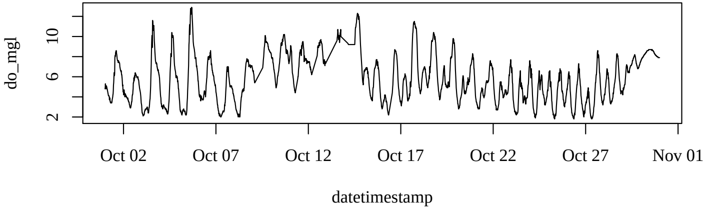
```
# subset the do time series for plotting
wq_dat <- subset(wq_dat, subset = c('2012-10-01 0:0', '2012-10-31 0:0'))
plot(do_mgl ~ datetimestamp, wq_dat, type = 'l')
```



Notice the missing values around October 12^{th}

Analysis 1 - Missing data and interpolation

Here's what the time series looks like after using 'na.approx'



The missing values have been linearly interpolated - a simple function that predicts missing values based on the starting and ending values in gaps

Analysis 1 - Missing data and interpolation

The ‘na.approx.swmpr’ function has only a few arguments

- object: input swmpr data
- params: which parameters to interpolate, default is all
- maxgap: what is the maximum gap size to interpolate (units are the timestep)?

See the help file for moreinfo

```
# see the help file
?na.approx.swmpr
```

Analysis 1 - Missing data and interpolation

Now you try an analysis! Open a new script and try the following:

- Import the file ‘cbmmcwq2012.csv’ in the dataset3 folder
- Handle QAQC flags and subset by October 1 to 31
- Plot the data - where are the missing values?
- Use ‘na.approx.swmpr’ to interpolate the missing values - what value to use for maxgap?
- Plot the data again - how does it look?

Analysis 2 - Smoothing and aggregation

Problem: trend evaluation is difficult if the data are noisy

Noise can be addressed byaggregating or smoothing data, both are similar

The 'aggregate.swmpr' function aggregates a time series by set periods of observation and calculates summary data for a parameter(s)

The 'smoother.swmpr' function calculates a moving window average of a time series

Analysis 2 - Smoothing and aggregation

The (relevant) arguments for 'aggregate.swmpr':

- x: Input data object
- by: How are the data aggregated - 'years', 'quarters', 'months', 'weeks', 'days', 'hours'
- FUN: What function is used to aggregate the data? Defaults to mean.
- aggs_out: T or F, to return the data at an intermediate step for plotting

```
# see help for all arguments
?aggregate.swmpr
```

Analysis 2 - Smoothing and aggregation

The (relevant) arguments for 'smoother.swmpr':

- swmpr_in: Input data object
- window: the size of the smoothing window, defaults to five observations at the current time step
- sides: what defines the window, centered on an observation (2, default) or use only the preceding observations (1)

```
# see help for all arguments
?smoother.swmpr
```

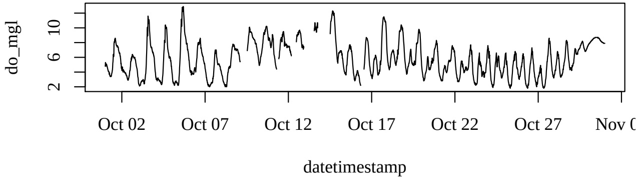
Analysis 2 - Smoothing and aggregation

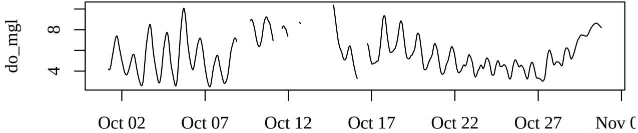
Now you try an analysis! Open a new script and try the following:

- Import the the same data as before - 'cbmmcwq2012.csv' in the dataset3 folder
- Handle QAQC flags and subset by October 1 to 31
- Plot the raw data
- Use 'smoother.swmpr', save to new object, plot again. How does it look? Try different window sizes.
- Aggregate the data by weeks and view the raw data (do not plot). Now try aggregation by months, what's the difference?

Analysis 2 - Smoothing and aggregation

```
# use the same data as in analysis 1 but...
# subset by these date ranges
dat <- subset(dat, subset = c('2012-10-01 0:0', '2012-10-31 0:0'))

# smooth
new_dat <- smoother.swmpr(dat, window = 40)

# plot original, then new
plot(do_mgl ~ datetimestamp, data = dat, type = 'l')
plot(do_mgl ~ datetimestamp, data = new_dat, type = 'l')
```





Analysis 2 - Smoothing and aggregation

```
# try an aggregation by 'weeks'
aggregate(dat, by = 'weeks')

##   datetimestamp do_mgl
## 1    2012-09-30    5.4
## 2    2012-10-07    6.4
## 3    2012-10-14    6.6
## 4    2012-10-21    4.5
## 5    2012-10-28    6.2

# try an aggregation by 'months'
aggregate(dat, by = 'months')

##   datetimestamp do_mgl
## 1    2012-10-01    5.8
```

Analysis 3 - Basic trend analysis

More often, we are concerned with long-term trends over time – a missing data point here or there or noisy data on short time periods may not be very important

We need plots to characterize long-term trends over time – both raw and summarized data

This analysis will show you two ways to evaluate trends by plotting

Analysis 3 - Basic trend analysis

Start by importing all the water quality data for the 'Iron Pot Landing' station at the Chesapeake Bay Maryland reserve

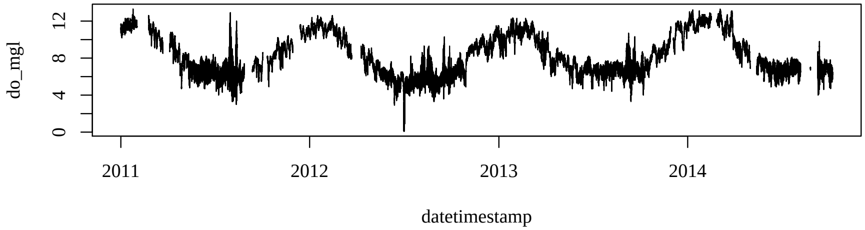
```
# import all wq data for cbmip
# change path as needed
path <- 'C:/data/dataset3/'
dat <- import_local(path, 'cbmipwq')

# qaqc checks
dat <- qaqc(dat)
```

Our questions: What are the dissolved oxygen dynamics over the last four years? Can we characterize trends, both seasonal and annual?

Analysis 3 - Basic trend analysis

First a simple plot...

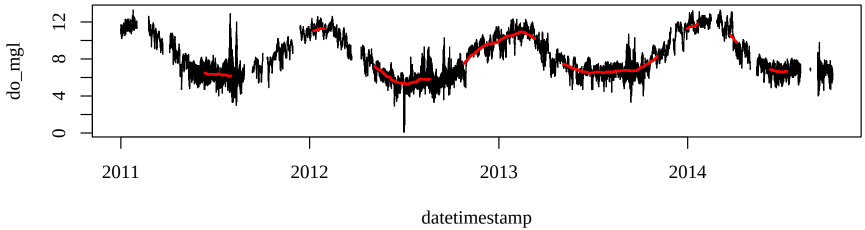
```r
# plot DO for the time series
plot(do_mgl ~ datetimestamp, data = dat, type = 'l')
```



Analysis 3 - Basic trend analysis

If we are concerned with long-term trends, we want to reduce the noise related to annual variability... we can use the smoother function

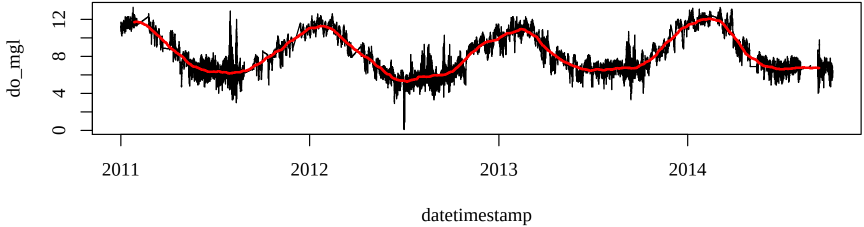
```
# smoother using a large window (5000 steps ~ 52 days)
do_smooth <- smoother(dat, params = 'do_mgl', window = 5000)
plot(do_mgl ~ datetimestamp, data = dat, type = 'l')
lines(do_smooth$datetimestamp, do_smooth$do_mgl, col = 'red', lwd = 2)
```



Analysis 3 - Basic trend analysis

Try it again but use ‘na.approx’ first to fill gaps

```
# use na.approx, then smooth
new_dat <- na.approx(dat, param = 'do_mgl', maxgap = 3000)
do_smooth <- smoother(new_dat, params = 'do_mgl', window = 5000)
plot(do_mgl ~ datetimestamp, data = new_dat, type = 'l')
lines(do_smooth$datetimestamp, do_smooth$do_mgl, col = 'red', lwd = 2)
```



Now we have a time series that primarily shows annual variation, independent of short-term variation

Analysis 3 - Basic trend analysis

Finally, we can use the 'aggregate.swmpr' function with boxplots for an alternative interpretation

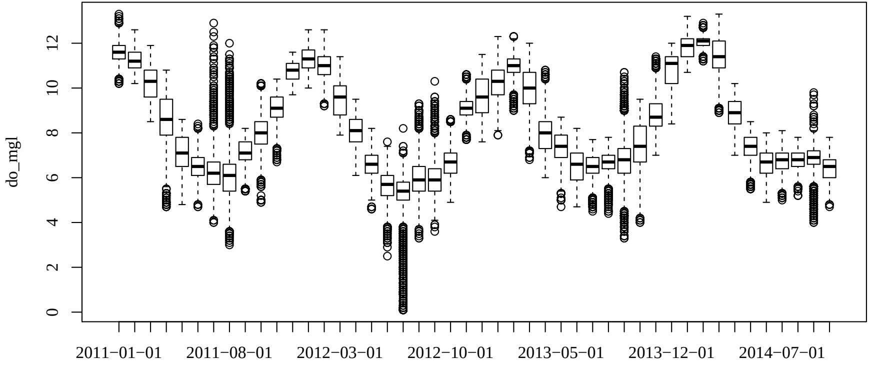
The 'aggs_out' argument can be used...

```
# get reformatted data from aggregate for plotting
agg_dat <- aggregate(dat, by = 'months', params = 'do_mgl', aggs_out = T)
head(agg_dat)

##   datetimestamp do_mgl
## 1    2011-01-01     11
## 2    2011-01-01     11
## 3    2011-01-01     11
## 4    2011-01-01     11
## 5    2011-01-01     11
## 6    2011-01-01     11

# note same row number in aggregated data
dim(agg_dat)

## [1] 132132      2

dim(dat)
```

Analysis 3 - Basic trend analysis

Plot the aggregated data

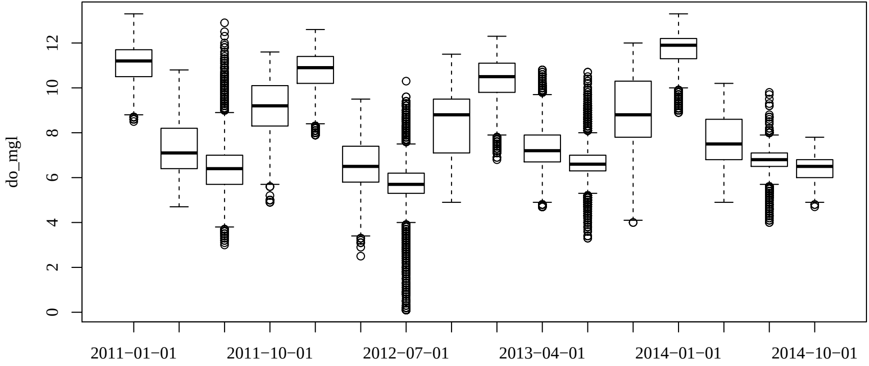
```r
# use boxplots
boxplot(do_mgl ~ datetimestamp, data = agg_dat, ylab = 'do_mgl')
```



Analysis 3 - Basic trend analysis

This can be repeated for different time steps...

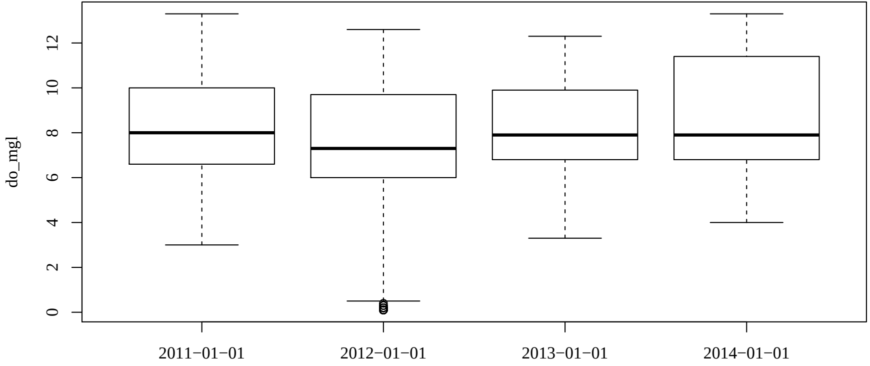
```
# by season
agg_dat <- aggregate(dat, by = 'quarters', params = 'do_mgl', aggs_out = T)
boxplot(do_mgl ~ datetimestamp, data = agg_dat, ylab = 'do_mgl')
```



Analysis 3 - Basic trend analysis

This can be repeated for different time steps...

```
# by year
agg_dat <- aggregate(dat, by = 'years', params = 'do_mgl', aggs_out = T)
boxplot(do_mgl ~ datetimestamp, data = agg_dat, ylab = 'do_mgl')
```



Analysis 3 - Basic trend analysis

A final note about trend analysis – this can be as simple or as complex as you like

The key question - has my variable of interest significantly changed and when did it occur?

You must define what change means and how you will assess

E.g., Has it increased/decreased? How has the central tendency changed? Has the variance changed? What factors could have influenced this change?

As a first step, always plot the raw or summarized data!

More detailed approaches are beyond the scope of this workshop - but check out the CRAN task view on time series for more you can do in R!


NERRS / SWMP
Data Analysis Workshop: *Time Series*
November 17, 2014

Questions??