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
  - Common functions for exploratory data analysis
  - 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!

Common functions for EDA

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

You may decide to omit variables or specific observations, transform, standardize, etc.

Many of the same principles that apply to standard data analysis apply to time series analysis

Common functions for EDA

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

We will cover a few basic techniques but keep in mind EDA is a general term and much of what we have already covered, and will cover, can be considered exploratory

Let's import some data:

```
# reload the SWMPr package if you started a new session
library(SWMPr)

# import data, qaqc, and subset
# change this path for the flash drive
path <- 'C:/data/dataset3'
nut_dat <- import_local(path, 'cbmmcnut')
nut_dat <- qaqc(nut_dat)
nut_dat <- subset(nut_dat, select = c('po4f', 'nh4f'))
```

Common functions for EDA

Perhaps the most useful function in R is 'summary'

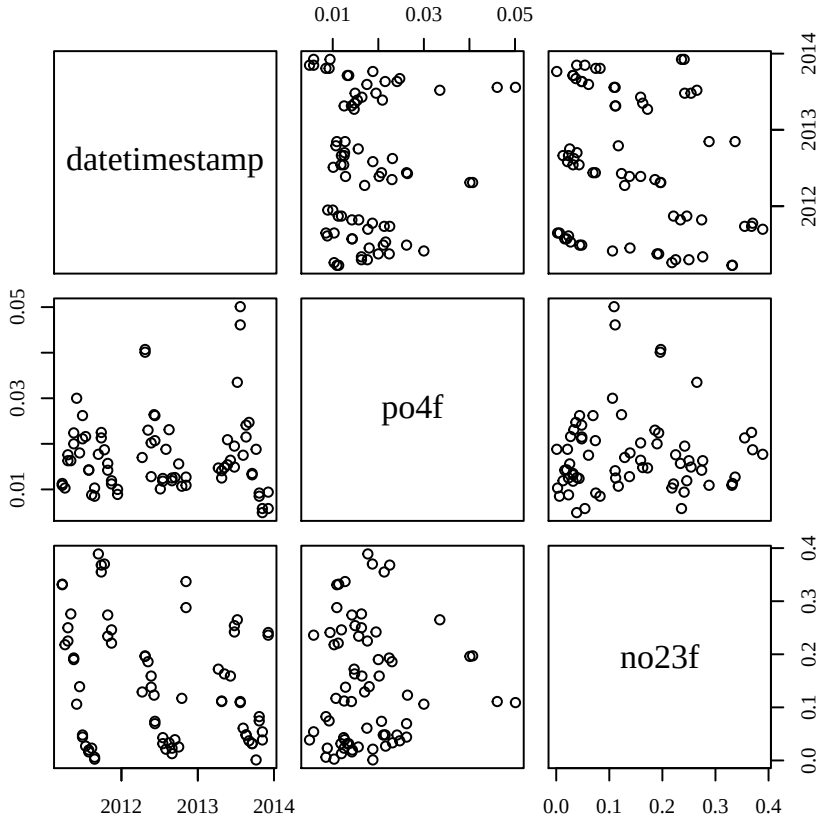
```
# get a summary of the data
summary(nut_dat)
```

```
##  datetimestamp                     po4f          no23f       chla_n
##  Min.   :2011-03-23 11:45:00   Min.   :0.00   Min.   :0   Min.   : 2
##  1st Qu.:2011-09-15 22:37:30   1st Qu.:0.01   1st Qu.:0   1st Qu.: 5
##  Median :2012-06-22 10:30:30   Median :0.02   Median :0   Median : 8
##  Mean   :2012-07-13 12:56:29   Mean   :0.02   Mean   :0   Mean   :17
##  3rd Qu.:2013-06-02 21:26:30   3rd Qu.:0.02   3rd Qu.:0   3rd Qu.:17
##  Max.   :2013-12-04 11:46:00   Max.   :0.05   Max.   :0   Max.   :98
##                                               NA's   :5   NA's   :4
```

Common functions for EDA

The pairs function is useful for evaluating simple bivariate correlations

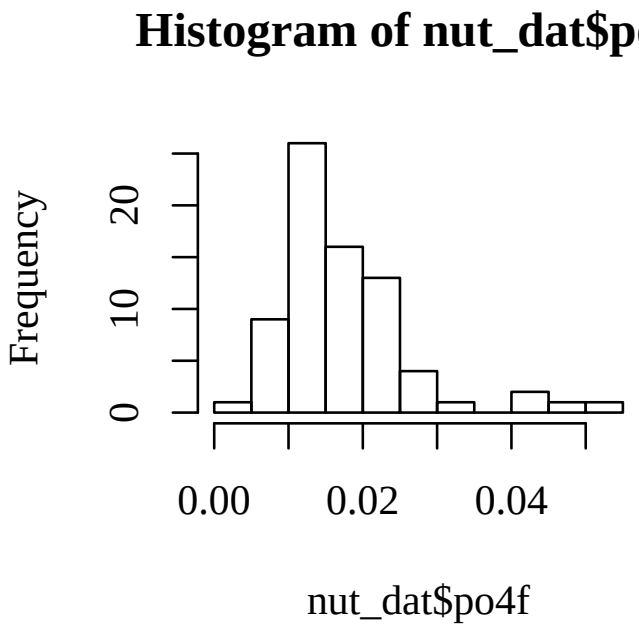
```
# bivariate scatterplots
pairs(nut_dat)
```

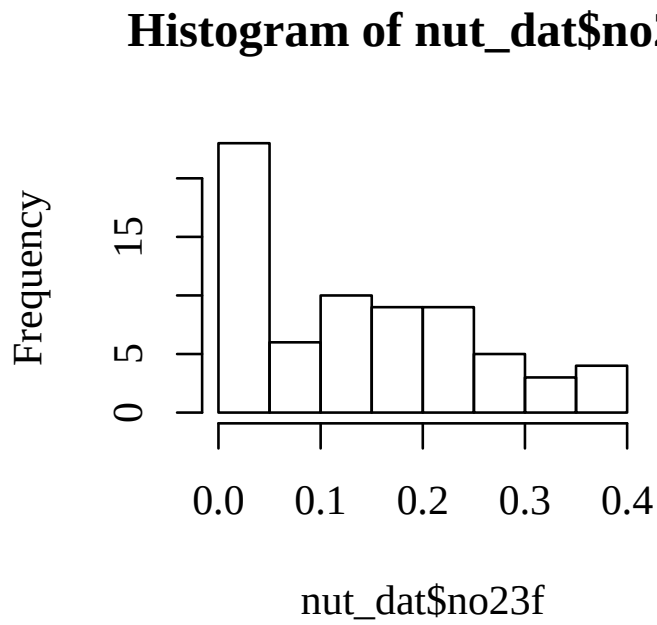


Common functions for EDA

Histograms are useful...

```
# some histograms
hist(nut_dat$po4f)
hist(nut_dat$no23f)
```

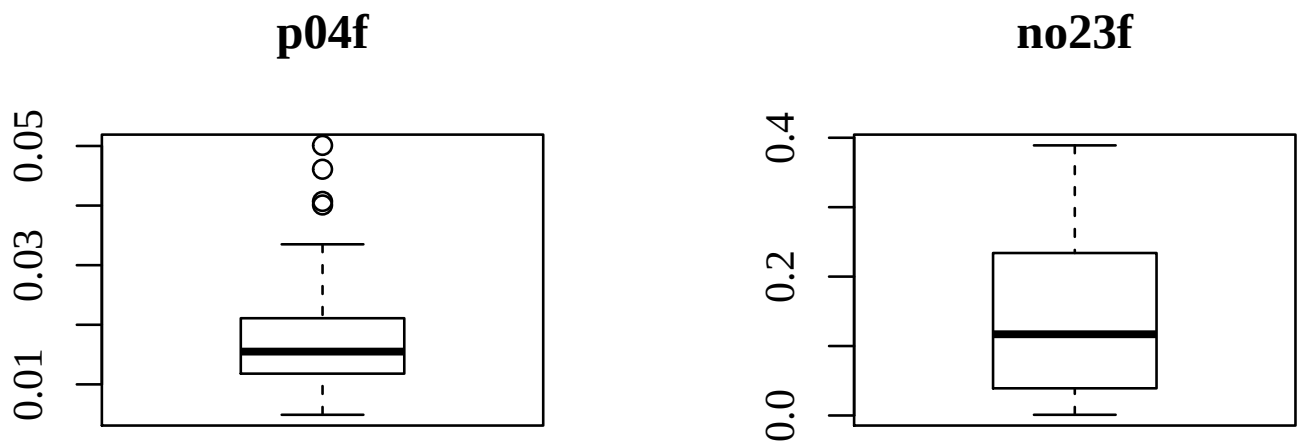




Common functions for EDA

Boxplots are useful...

```
# some boxplots
boxplot(nut_dat$po4f, main = 'p04f')
boxplot(nut_dat$no23f, main = 'no23f')
```



Common functions for EDA

Plotting individual variables or simple scatterplots between two variables will get you familiar with a dataset

Again, R has many functions for EDA and we don't want to focus on generic approaches that can be learned at home

A quick google search of 'exploratory data analysis in r' will point you in the right direction

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'
wq_dat <- import_local(path, 'cbmmcwq2012')
```

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

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

## [1] 419
```

Analysis 1 - Missing data and interpolation

Mising data can be removed with the subset function or replaced with the mean

```
# a temporary object so we don't overwrite wq_dat
wq_tmp <- wq_dat

# remove missing values with subset function
wq_tmp <- subset(wq_tmp, rem_row = T)

# or replace missing values with the mean
wq_tmp <- wq_dat
wq_tmp[is.na(wq_tmp$do_mgl), 'do_mgl'] <- mean(wq_tmp$do_mgl, na.rm = T)
```

What are some issues with these approaches?

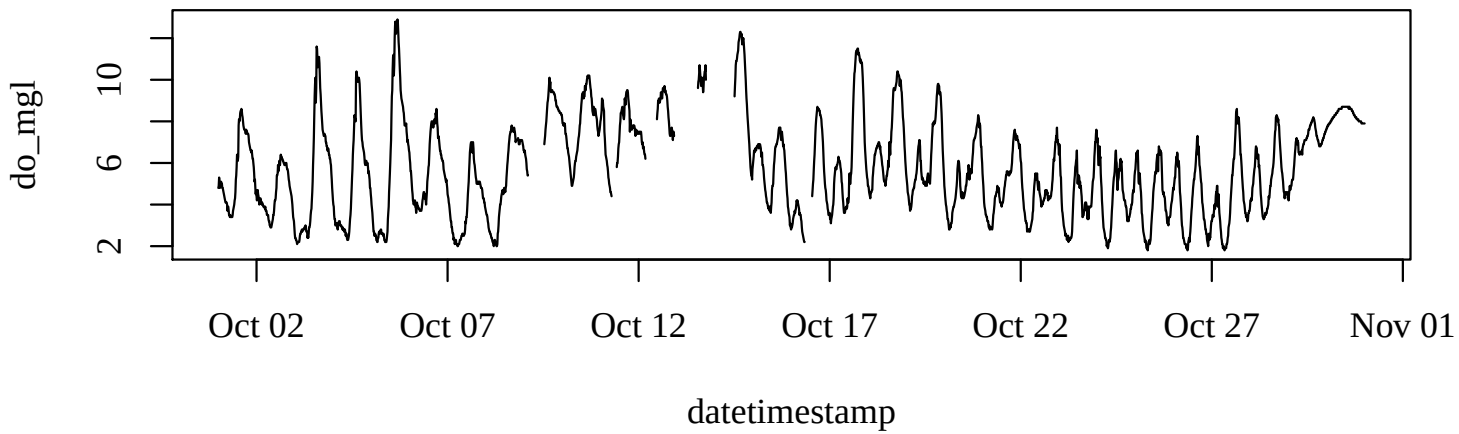
'subset' will change the time step

Neither approach is very true to the data...

Analysis 1 - Missing data and interpolation

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

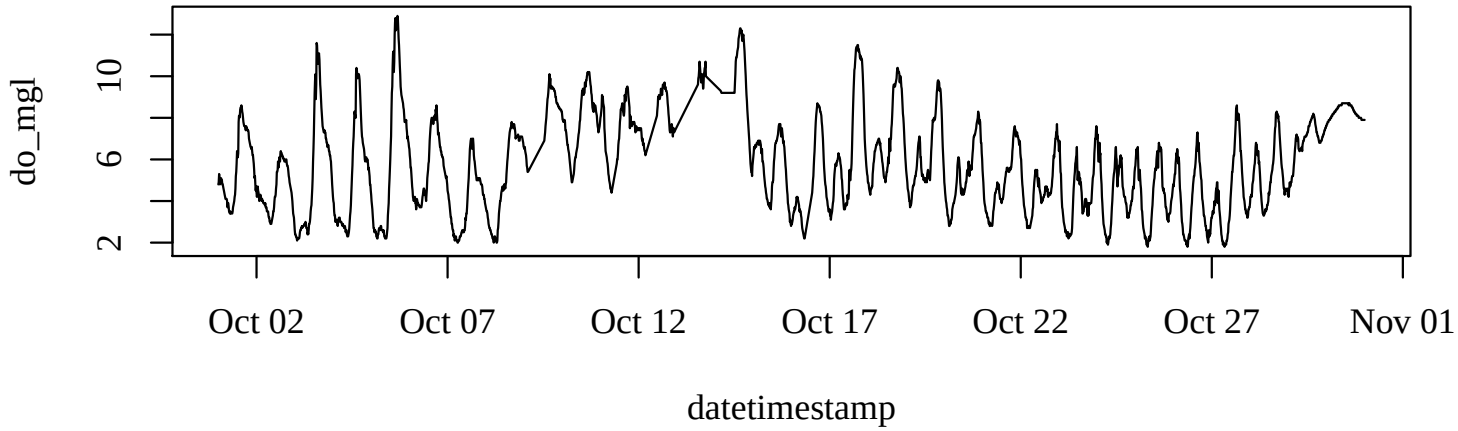
```
# subset the do time series for plotting
wq_dat <- subset(wq_dat, subset = c('2012-10-01 0:0', '2012-10-31 0:0'))
plot(do_mgl ~ datetimestamp, wq_dat, type = 'l')
```



Notice the missing values around October 12th

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

May not be a true representation but better than some other approaches

Analysis 1 - Missing data and interpolation

The 'na.approx' function has only a few arguments

```
# arguments for na.approx.swmpr, a method for na.approx
formals(na.approx.swmpr)

## $swmpr_in
##
##
## $params
## NULL
##
## $maxgap
```

- swmpr_in: input swmpr data
- params: which parameters to interplate, default is all
- maxgap: what is the maximum gap size to interpolate?

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?
- Does interpolation affect the mean? variance?

Analysis 1 - Missing data and interpolation

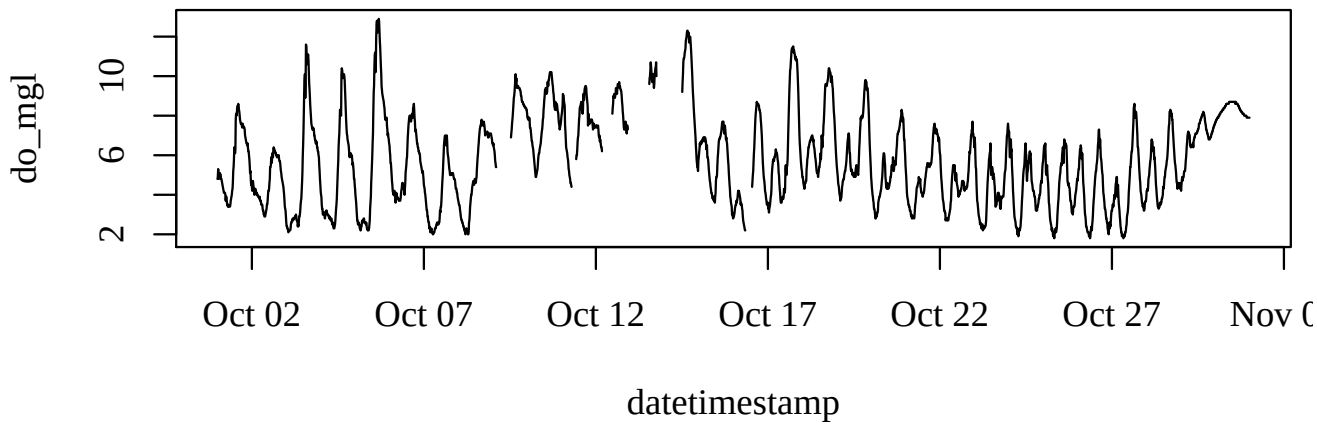
Now you try an analysis! Open a new script and try the following:

- 'import_local', make sure path is correct
- 'qaqc' then 'subset.swmpr' with appropriate 'subset' argument
- 'plot' y $\sim$ x, data, type = 'l'
- 'na.approx.swmpr', maxgap = some number, assign results to new objectt
- 'plot' y $\sim$ x, data, type = 'l'
- 'mean' or 'var' of do˙mgl, must include na.rm = T

Consult the cookbook or help files for how to use a function ('?na.approx.swmpr')

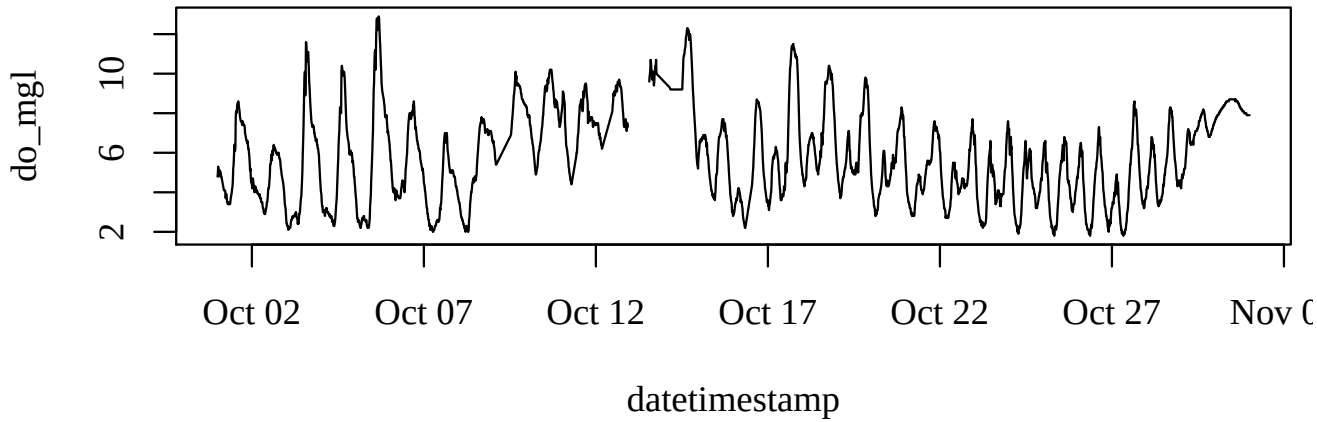
Analysis 1 - Missing data and interpolation

```
# the analysis should look something like this

# change path as needed
path <- 'C:/data/dataset3'
dat <- import_local(path, 'cbmmcwq2012')

# qaqc and subset imported data
dat <- qaqc(dat)
dat <- subset(dat, subset = c('2012-10-01 00:00', '2012-10-31 00:00'))

# plot
plot(do_mgl ~ datetimestamp, data = dat, type = 'l')
```



Analysis 1 - Missing data and interpolation

```r
# interpolate missing
new_dat <- na.approx(dat, maxgap = 50) # try different maxgaps

# plot
plot(do_mgl ~ datetimestamp, data = new_dat, type = 'l')
```




```r
mean(dat$do_mgl, na.rm = T)

## [1] 5.8

mean(new_dat$do_mgl, na.rm = T)

## [1] 5.9
```

Analysis 2 - Smoothing and aggregation

Problem: trend evaluation is difficult if the data are noisy

Noise can be caused by many factors (e.g., measurement error, process uncertainty), we are usually more concerned with the true signal in a time series

Noise can be addressed by aggregating 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 variable

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

Analysis 2 - Smoothing and aggregation

The arguments for 'aggregate.swmpr':

- swmpr_in: Input data object
- by: How are the data aggregated - 'year', 'quarters', 'months', 'weeks', 'days', 'hours'
- FUN: What function is used to aggregate the data? Defaults to mean.
- params: Which parameters do you aggregate? Defaults to all.
- na.action: how are missing data treated, default is to retain missing data in results

Analysis 2 - Smoothing and aggregation

The 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) or use only the preceding observations (1)
- params: Which parameters do you aggregate? Defaults to all.

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', 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

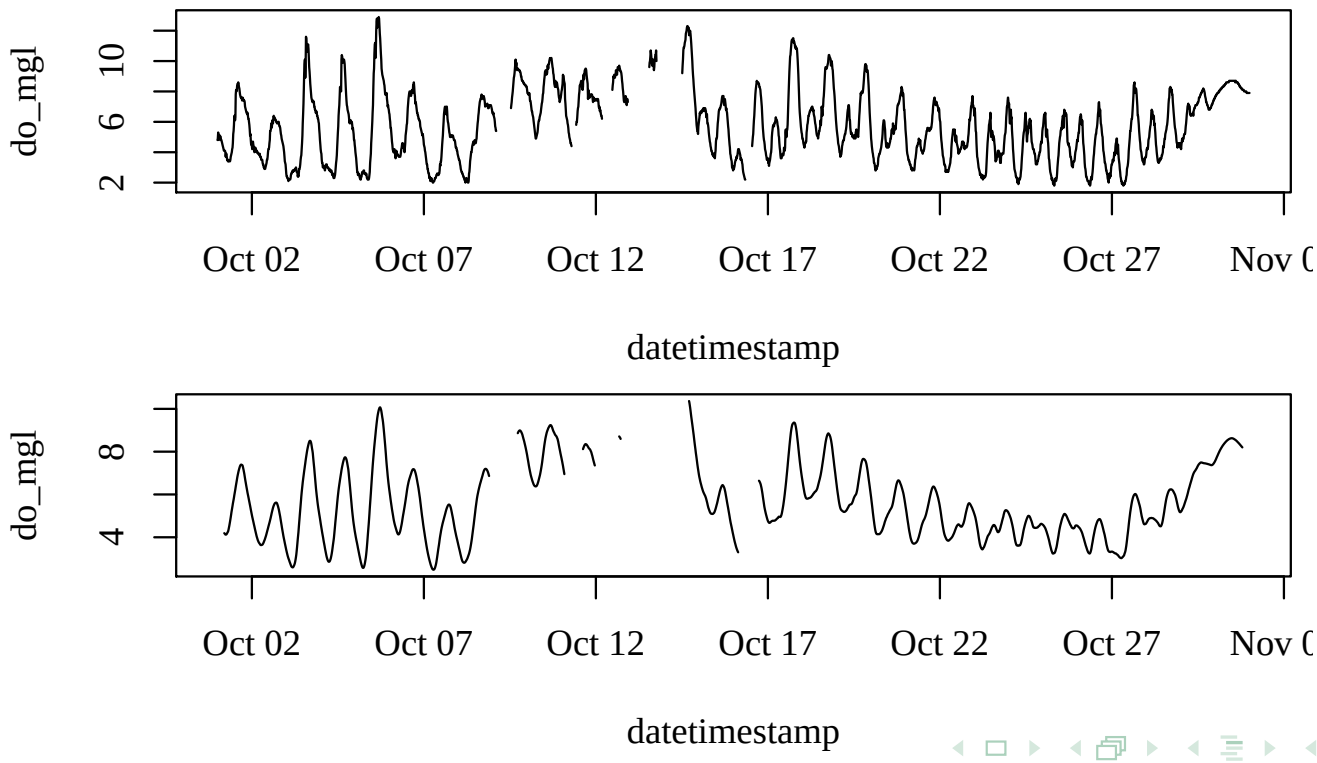
Now you try an analysis! Open a new script and try the following:

- 'import_local', make sure path is correct
- 'qaqc' then 'subset.swmpr' with appropriate 'subset' argument
- 'plot' y $\sim$ x, data, type = 'l'
- 'smoother.swmpr', try a large number (e.g., 30), assign results to new object
- 'plot' y $\sim$ x, data, type = 'l'
- 'aggregate.swmpr', by = 'weeks', by = 'days'

Consult the cookbook or help files for how to use a function ('?smoother.swmpr')

Analysis 2 - Smoothing and aggregation

```
# use the same data as in analysis 1

# 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 temp spcond  sal do_pct do_mgl depth cdepth level clevel  ph
## 1    2012-09-30   21   0.75 0.36     61    5.4  0.47    NaN   NaN    NaN 6.5
## 2    2012-10-07   16   0.67 0.33     65    6.4  0.51    NaN   NaN    NaN 6.6
## 3    2012-10-14   15   0.70 0.34     67    6.6  0.51    NaN   NaN    NaN 6.8
## 4    2012-10-21   16   0.71 0.35     46    4.5  0.52    NaN   NaN    NaN 6.8
## 5    2012-10-28   14   0.57 0.29     59    6.2  0.64    NaN   NaN    NaN 6.7
##   turb chlfluor
## 1   22       23
## 2   34       23
## 3   35       20
## 4   31       14
## 5   88       14
```

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

```
##   datetimestamp temp spcond  sal do_pct do_mgl depth cdepth level clevel  ph
## 1    2012-10-01   16   0.69 0.33     59    5.8  0.52    NaN   NaN    NaN 6.7
##   turb chlfluor
## 1   39       19
```

Analysis 3 - Basic trend analysis


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

Questions??