SWMPr analyze

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 for the session

- What are some basic analyses that can be accomplished with SWMPr?
- What are some plotting functions provided by SWMPr?
- What are some resources for aditional learning?

Interactive portion

We will use the swmpr2.Rproj project for this session, double-click to open in RStudio

- location on flash drive
- location online

Interactive portion

We will use the swmpr2.Rproj project for this session, double-click to open in RStudio

- location on flash drive
- location online

You will run examples whenever you see this guy:

Don't forget to use your stickies: green for done/ok, red for problem

Basic analyses with SWMPr

How do we want to use the data?

Basic analyses with SWMPr

How do we want to use the data?

- What has happened at my site over time?
- Are there differences between sites?
- Can we remove seasonal trends?
- Are there differences between parameters?
- Others?

Basic analyses with SWMPr 

Take a few minutes to acquaint yourself with the ***analyze*** functions:

```
help.search('analyze', package = 'SWMPr')
```

Basic analyses with SWMPr 

Take a few minutes to acquaint yourself with the ***analyze*** functions:

```
help.search('analyze', package = 'SWMPr')
```

Which functions simplify the data?

Which functions could you use to explore or visualize the data?

Which functions are related to metabolism?

Basic analyses with SWMPr - missing data

Most datasets will have missing values - how do you deal with those?

Remove? Set as mean? Replace with similar?

SWMPr provides a function to interpolate missing data: `na.approx`

To start, let's import and plot some data...

Basic analyses with SWMPr - missing data

- Import the 2012 water quality data for cbmmc from the 'zip_ex' folder
- Deal with QAQC columns
- Select two columns of interest
- Plot the data subset (`?plot.swmpr`)

Basic analyses with SWMPr - missing data

- Import the 2012 water quality data for cbmmc from the 'zip_ex' folder

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmmcwq2012')
```

- Deal with QAQC columns

- Select two columns of interest

- Plot the data subset (`?plot.swmpr`)

Basic analyses with SWMPr - missing data

- Import the 2012 water quality data for cbmmc from the 'zip_ex' folder

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmmcwq2012')
```

- Deal with QAQC columns

```
tmp <- qaqc(dat)
```

- Select two columns of interest

- Plot the data subset (`?plot.swmpr`)

Basic analyses with SWMPr - missing data

- Import the 2012 water quality data for cbmmc from the 'zip_ex' folder

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmmcwq2012')
```

- Deal with QAQC columns

```
tmp <- qaqc(dat)
```

- Select two columns of interest

```
tmp <- subset(tmp, select = 'do_mgl', subset = c('2012-10-01 0:0',
  '2012-10-31 0:0'))
```

- Plot the data subset (`?plot.swmpr`)

Basic analyses with SWMPr - missing data

- Import the 2012 water quality data for cbmmc from the 'zip_ex' folder

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmmcwq2012')
```

- Deal with QAQC columns

```
tmp <- qaqc(dat)
```

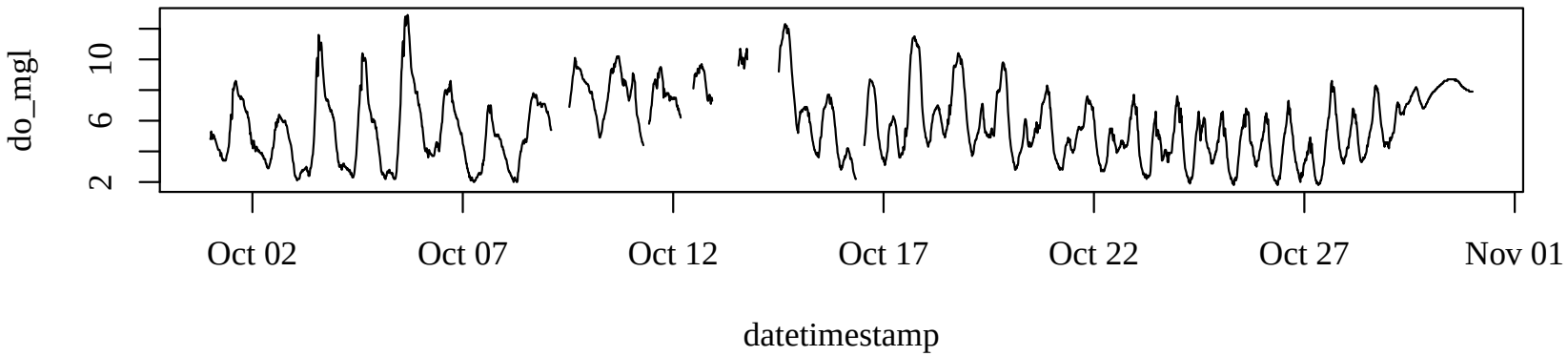
- Select two columns of interest

```
tmp <- subset(tmp, select = 'do_mgl', subset = c('2012-10-01 0:0',
  '2012-10-31 0:0'))
```

- Plot the data subset (`?plot.swmpr`)

```
plot(tmp)
```

Basic analyses with SWMPr - missing data



Notice the missing data around October 12th

The `na.approx` function (`?na.approx.swmpr`) has three arguments:

- `object`: swmpr data object to fill
- `params`: name(s) of parameter to fill
- `maxgap`: maximum gap size to interpolate

Basic analyses with SWMPr - missing data

- Use `na.approx` to interpolate the missing data (`?na.approx.swmpr`)

- Plot the two to see the differences (`?plot.swmpr`)

Basic analyses with SWMPr - missing data

- Use `na.approx` to interpolate the missing data (`?na.approx.swmpr`)

```
tmp2 <- na.approx(tmp, params = 'do_mgl', maxgap = 100)
```

- Plot the two to see the differences (`?plot.swmpr`)

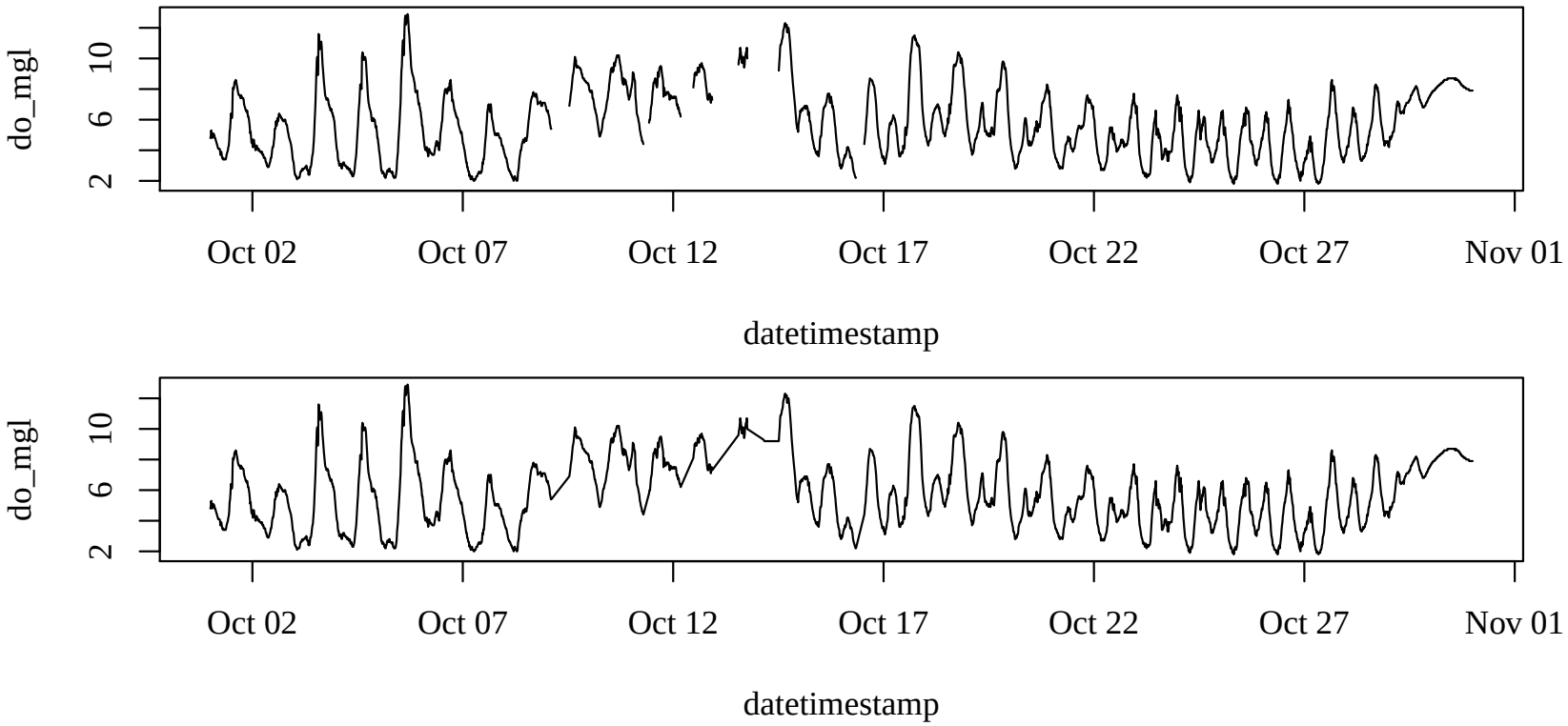
Basic analyses with SWMPr - missing data

- Use `na.approx` to interpolate the missing data (`?na.approx.swmpr`)

```
tmp2 <- na.approx(tmp, params = 'do_mgl', maxgap = 100)
```

- Plot the two to see the differences (`?plot.swmpr`)

```
plot(tmp)
plot(tmp2)
```

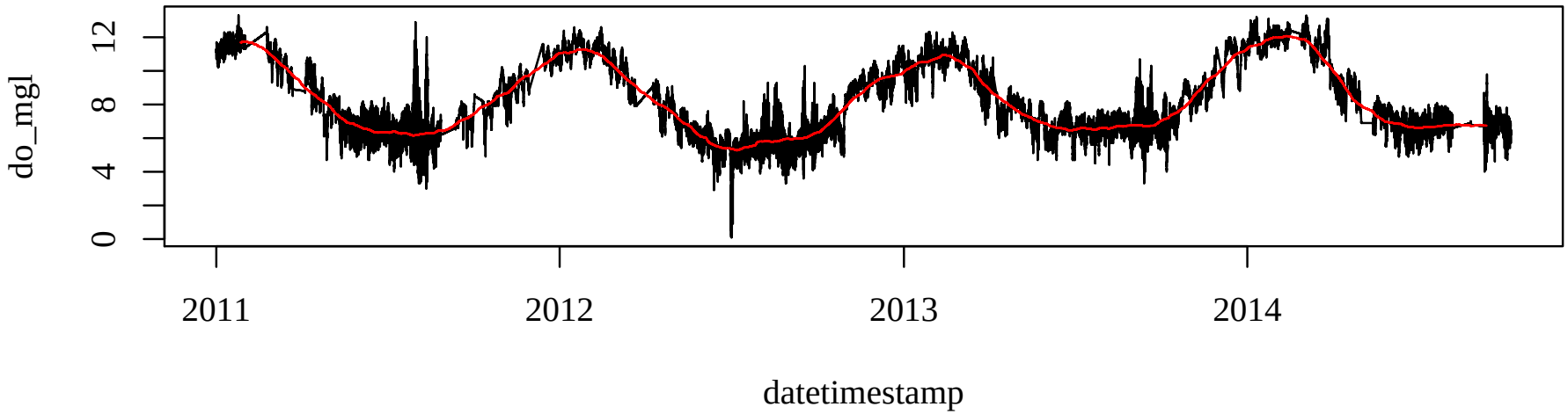


Basic analyses with SWMPr - smoothing data

Now we know how to fill missing data, let's see how it can help...

Trend evaluation often considers variation at different spatial scales

As an example, we want to evaluate variation in dissolved oxygen at seasonal or annual scales



Analysis 2 - Smoothing and aggregation

The `smoother` function (`?smoother`) calculates a moving window average of a time series

- `x`: 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)
- `params`: which parameters to smooth, default all

What would be a good window to look at seasonal or annual variation?

Basic analyses with SWMPr - smoothing data

- Import all years of water quality data for cbmip from the 'zip_ex' folder
- Deal with QAQC columns and subset DO
- Use `smoother` to remove daily and short-term variation, which window to use?
- Plot results (see `?plot.swmpr` and `?lines.swmpr`)

Basic analyses with SWMPr - smoothing data

- Import all years of water quality data for cbmip from the 'zip_ex' folder

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
```

- Deal with QAQC columns and subset DO

- Use `smoother` to remove daily and short-term variation, which window to use?

- Plot results (see `?plot.swmpr` and `?lines.swmpr`)

Basic analyses with SWMPr - smoothing data

- Import all years of water quality data for cbmip from the 'zip_ex' folder

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
```

- Deal with QAQC columns and subset DO

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

- Use `smoother` to remove daily and short-term variation, which window to use?

- Plot results (see `?plot.swmpr` and `?lines.swmpr`)

Basic analyses with SWMPr - smoothing data

- Import all years of water quality data for cbmip from the 'zip_ex' folder

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
```

- Deal with QAQC columns and subset DO

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

- Use `smoother` to remove daily and short-term variation, which window to use?

```
do_smooth <- smoother(tmp2, window = 5000)
```

- Plot results (see `?plot.swmpr` and `?lines.swmpr`)

Basic analyses with SWMPr - smoothing data

- Import all years of water quality data for cbmip from the 'zip_ex' folder

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
```

- Deal with QAQC columns and subset DO

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

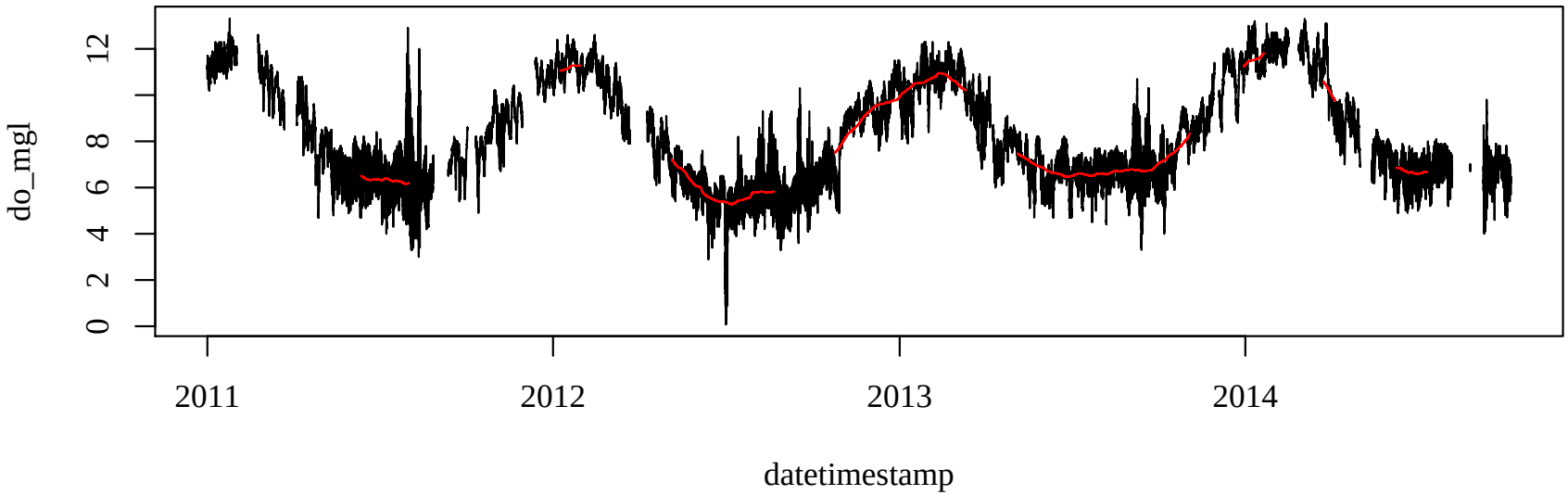
- Use `smoother` to remove daily and short-term variation, which window to use?

```
do_smooth <- smoother(tmp2, window = 5000)
```

- Plot results (see `?plot.swmpr` and `?lines.swmpr`)

```
plot(tmp)
lines(do_smooth)
```

Basic analyses with SWMPr - smoothing data



What happened?

How can we fix the problem?

Basic analyses with SWMPr - smoothing data

Repeat the analysis but use `na.approx` to fill missing data

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(tmp, select = 'do_mgl')
do_smooth <- smoother(tmp, window = 5000)
plot(tmp)
lines(do_smooth, col = 'red')
```

Where would you use `na.approx`?

Basic analyses with SWMPr - smoothing data

Repeat the analysis but use `na.approx` to fill missing data

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(tmp, select = 'do_mgl')
do_smooth <- smoother(tmp, window = 5000)
plot(tmp)
lines(do_smooth, col = 'red')
```

Where would you use `na.approx`?

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(tmp, select = 'do_mgl')
tmp <- na.approx(tmp, maxgap = 5000)
do_smooth <- smoother(tmp, window = 5000)
plot(tmp)
lines(do_smooth, col = 'red')
```

Basic analyses with SWMPr - smoothing data

Repeat the analysis but use `na.approx` to fill missing data

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(tmp, select = 'do_mgl')
do_smooth <- smoother(tmp, window = 5000)
plot(tmp)
lines(do_smooth, col = 'red')
```

Where would you use `na.approx`?

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(tmp, select = 'do_mgl')
tmp <- na.approx(tmp, maxgap = 5000)
do_smooth <- smoother(tmp, window = 5000)
plot(tmp)
lines(do_smooth, col = 'red')
```

Bonus: Try changing `maxgap` or `window`

Basic analyses with SWMPr - aggregating data

Finally, we can use `aggreswmp` to summarize and plot for an alternative interpretation

`aggreswmp` has five main arguments:

- `swmpr_in`: input data object
- `by`: aggregation period ('years', 'quarters', etc.)
- `FUN`: aggregation function, defaults to mean
- `params`: which parameters to aggregate, defaults all
- `aggs_out`: get the raw data, use this to make plots

Basic analyses with SWMPr - aggregating data

- Import all years of water quality data for cbmip from the 'zip_ex' folder, QAQC cleanup, and subset DO

- Use `aggreswmp` (`?aggreswmp`) to get quarterly summaries of the data

Basic analyses with SWMPr - aggregating data

- Import all years of water quality data for cbmip from the 'zip_ex' folder, QAQC cleanup, and subset DO

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(dat, select = 'do_mgl')
```

- Use `aggreswmp` (`?aggreswmp`) to get quarterly summaries of the data

Basic analyses with SWMPr - aggregating data

- Import all years of water quality data for cbmip from the 'zip_ex' folder, QAQC cleanup, and subset DO

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(dat, select = 'do_mgl')
```

- Use `aggreswmp` (`?aggreswmp`) to get quarterly summaries of the data

```
aggtmp <- aggreswmp(tmp, by = 'quarters')
```

Basic analyses with SWMPr - aggregating data

- Import all years of water quality data for cbmip from the 'zip_ex' folder, QAQC cleanup, and subset DO

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(dat, select = 'do_mgl')
```

- Use `aggreswmp` (`?aggreswmp`) to get quarterly summaries of the data

```
aggtmp <- aggreswmp(tmp, by = 'quarters')
```

- Bonus: Try different aggregation periods

- Bonus: Try different aggregation functions

Basic analyses with SWMPr - aggregating data

- Import all years of water quality data for cbmip from the 'zip_ex' folder, QAQC cleanup, and subset DO

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(dat, select = 'do_mgl')
```

- Use `aggreswmp` (`?aggreswmp`) to get quarterly summaries of the data

```
aggtmp <- aggreswmp(tmp, by = 'quarters')
```

- Bonus: Try different aggregation periods

```
aggtmp <- aggreswmp(tmp, by = 'years')
aggtmp <- aggreswmp(tmp, by = 'weeks')
```

- Bonus: Try different aggregation functions

Basic analyses with SWMPr - aggregating data

- Import all years of water quality data for cbmip from the 'zip_ex' folder, QAQC cleanup, and subset DO

```
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')
tmp <- qaqc(dat)
tmp <- subset(dat, select = 'do_mgl')
```

- Use `aggreswmp` (`?aggreswmp`) to get quarterly summaries of the data

```
aggtmp <- aggreswmp(tmp, by = 'quarters')
```

- Bonus: Try different aggregation periods

```
aggtmp <- aggreswmp(tmp, by = 'years')
aggtmp <- aggreswmp(tmp, by = 'weeks')
```

- Bonus: Try different aggregation functions

```
fun_in <- function(x)  var(x, na.rm = TRUE)
aggtmp <- aggreswmp(swmpr_in, FUN = fun_in, 'years')
```

Basic analyses with SWMPr - aggregating data

Plot the aggregated data by quarters - use `aggs_out = TRUE`

Basic analyses with SWMPr - aggregating data

Plot the aggregated data by quarters - use `aggs_out = TRUE`

```
# use aggs_out to get all
aggtmp <- aggreswmp(tmp, by = 'quarters', aggs_out = TRUE)
```

Basic analyses with SWMPr - aggregating data

Plot the aggregated data by quarters - use `aggs_out = TRUE`

```
# use aggs_out to get all
aggtmp <- aggreswmp(tmp, by = 'quarters', aggs_out = TRUE)
```

Then use `boxplot` (`?boxplot`) from the R stats package

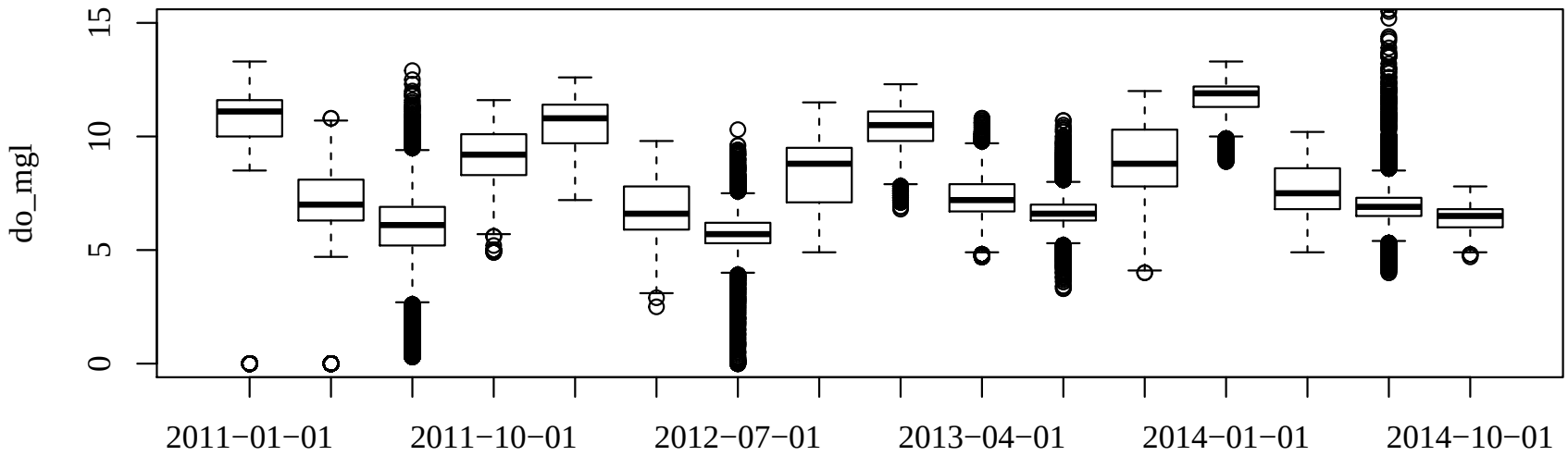
Basic analyses with SWMPr - aggregating data

Plot the aggregated data by quarters - use `aggs_out = TRUE`

```
# use aggs_out to get all
aggtmp <- aggreswmp(tmp, by = 'quarters', aggs_out = TRUE)
```

Then use `boxplot` (`?boxplot`) from the R stats package

```
# use boxplot
boxplot(do_mgl ~ datetimestamp, data = aggtmp, ylab = 'do_mgl', ylim = c(0, 15))
```



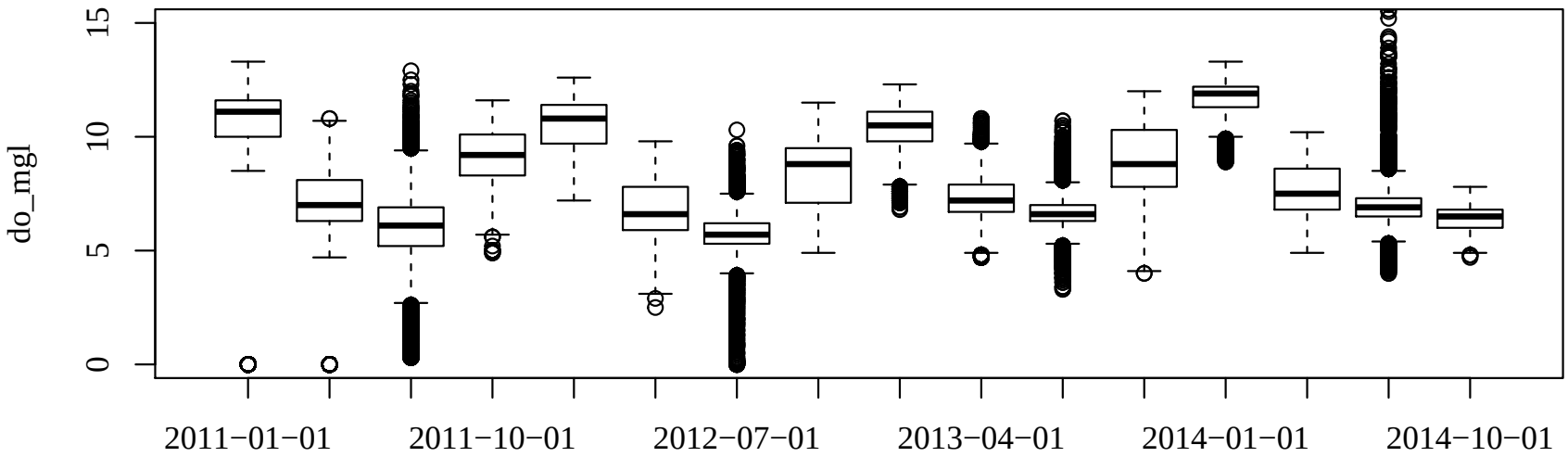
Basic analyses with SWMPr - aggregating data

Plot the aggregated data by quarters - use `aggs_out = TRUE`

```
# use aggs_out to get all
aggtmp <- aggreswmp(tmp, by = 'quarters', aggs_out = TRUE)
```

Then use `boxplot` (`?boxplot`) from the R stats package

```
# use boxplot
boxplot(do_mgl ~ datetimestamp, data = aggtmp, ylab = 'do_mgl', ylim = c(0, 15))
```



Have a look at the data from `aggs_out = TRUE` and `aggs_out = FALSE`, how do they differ and why?

Plotting functions in SWMPr

R provides near limitless options to visualize data - a full coverage of these tools would take days

We will briefly go over some key plotting functions in SWMPr, each is designed for simplicity and efficiency to summarize lots of data

Plotting functions in SWMPr:

- `decomp`: time series decomposition
- `decomp_cj`: time series decomposition, alternative
- `map_reserve`: plot a basic map of a reserve
- `overplot`: plot multiple parameters on the same plot
- `plot_metab`: plot metabolism estimates
- `plot_summary`: plot multiple summaries for a parameter

Plotting functions in SWMPr 

The `map_reserve` function can be used to map sites:

- `nerr_site_id`: site(s) to map, usually first three letters
- `zoom`: zoom factor for the map (usually between 5–15)
- `map_type`: `'terrain'`, `'satellite'`, `'roadmap'`, or `'hybrid'`

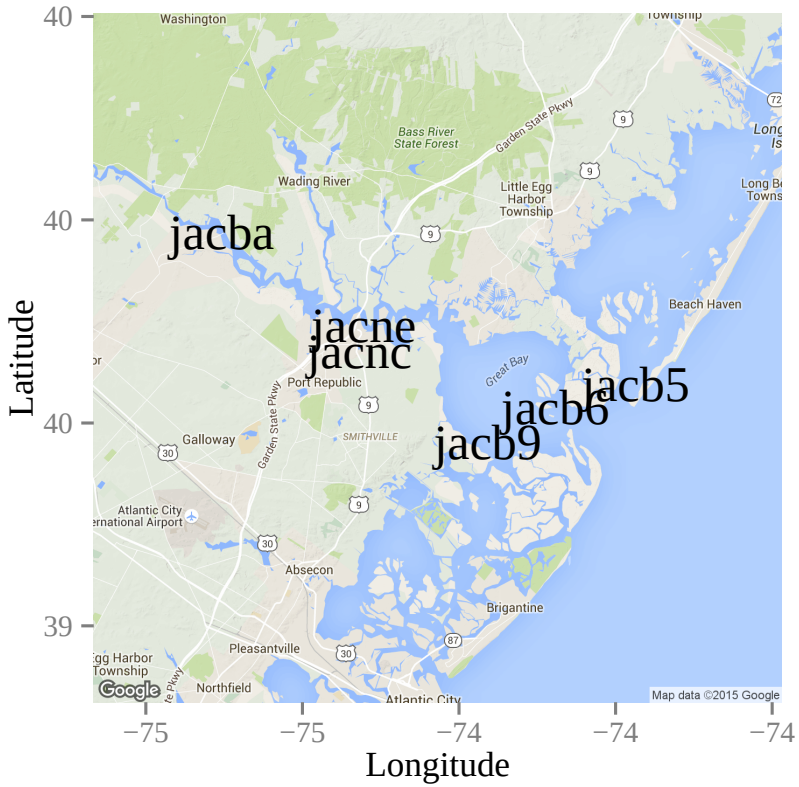
```
# try any of these examples
map_reserve('jac')

map_reserve('elk', zoom = 13,
  map_type = 'hybrid')

map_reserve('gtmss', zoom = 15,
  map_type = 'satellite',
  text_col = 'lightblue')
```

Plotting functions in SWMPr

The `map_reserve` function can be used to map sites:

- `nerr_site_id`: site(s) to map, usually first three letters
- `zoom`: zoom factor for the map (usually between 5–15)
- `map_type`: `'terrain'`, `'satellite'`, `'roadmap'`, or `'hybrid'`

```
# try any of these examples
map_reserve('jac')

map_reserve('elk', zoom = 13,
  map_type = 'hybrid')

map_reserve('gtmss', zoom = 15,
  map_type = 'satellite',
  text_col = 'lightblue')
```



Plotting functions in SWMPr

The `overplot` function can be used to plot one to many time series:

- `dat_in`: input `swmpr` object
- `select`: parameter(s) to plot (passed to `subset`)
- `subset`: date ranges to plot (passed to `subset`)

Plotting functions in SWMPr 

The `overplot` function can be used to plot one to many time series:

- `dat_in`: input `swmpr` object
- `select`: parameter(s) to plot (passed to `subset`)
- `subset`: date ranges to plot (passed to `subset`)

Import the 2011 data for cbmip

Plot DO, temperature, and salinity for August (`?overplot`)

Plotting functions in SWMPr 

The `overplot` function can be used to plot one to many time series:

- `dat_in`: input `swmpr` object
- `select`: parameter(s) to plot (passed to `subset`)
- `subset`: date ranges to plot (passed to `subset`)

Import the 2011 data for cbmip

```
# import
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq2011')
```

Plot DO, temperature, and salinity for August (`?overplot`)

Plotting functions in SWMPr 

The `overplot` function can be used to plot one to many time series:

- `dat_in`: input `swmpr` object
- `select`: parameter(s) to plot (passed to `subset`)
- `subset`: date ranges to plot (passed to `subset`)

Import the 2011 data for cbmip

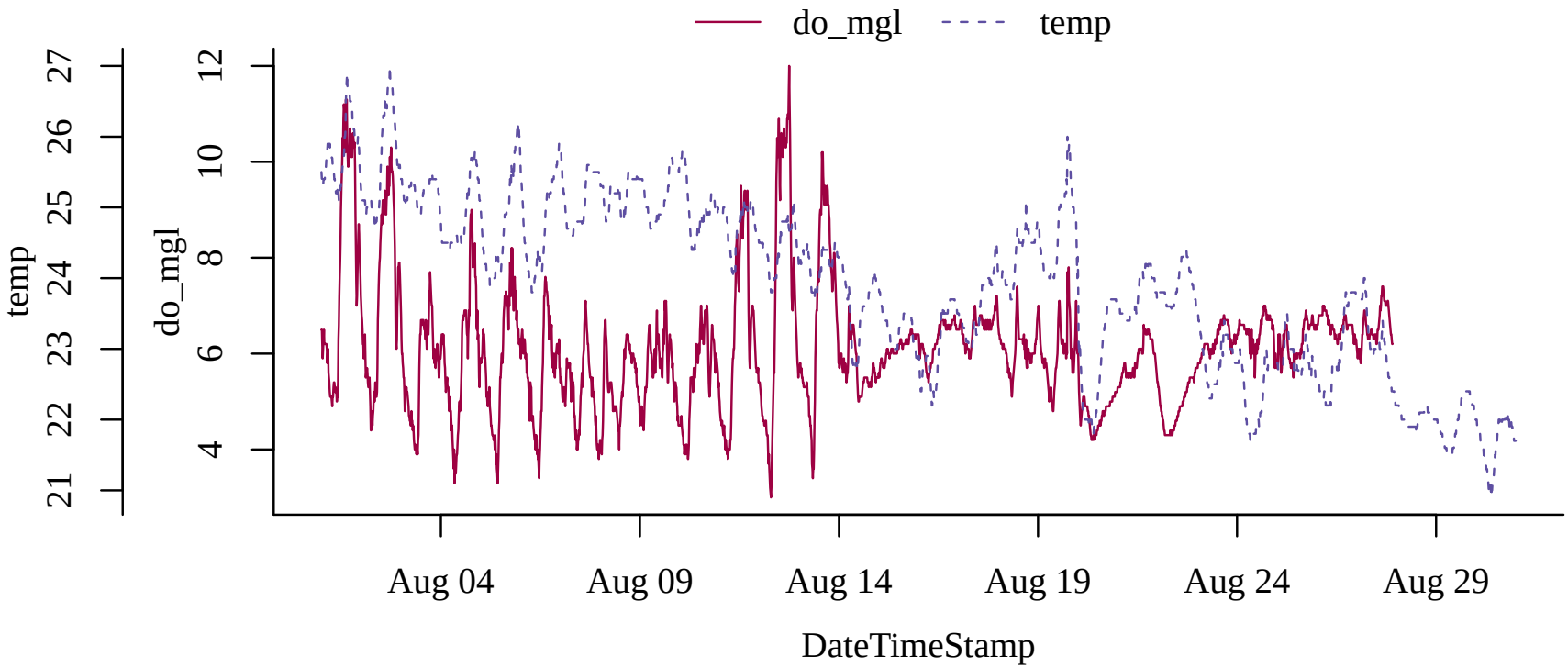
```
# import
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq2011')
```

Plot DO, temperature, and salinity for August (`?overplot`)

```
# plot
overplot(dat, select = c('do_mgl', 'temp'),
  subset = c('2011-08-01 0:0', '2011-08-31 0:0'))
```

Plotting functions in SWMPr

```
# plot
overplot(dat, select = c('do_mgl', 'temp'),
  subset = c('2011-08-01 0:0', '2011-08-31 0:0'))
```



Plotting functions in SWMPr 

The `plot_summary` function can be used to summarize a parameter across the time series

- `swmpr_in`: input swmpr object
- `param`: parameter to summarize
- `years`: years to plot

Import all years of data for cbmip, plot a summary of temperature (`?plot_summary`)

Plotting functions in SWMPr 

The `plot_summary` function can be used to summarize a parameter across the time series

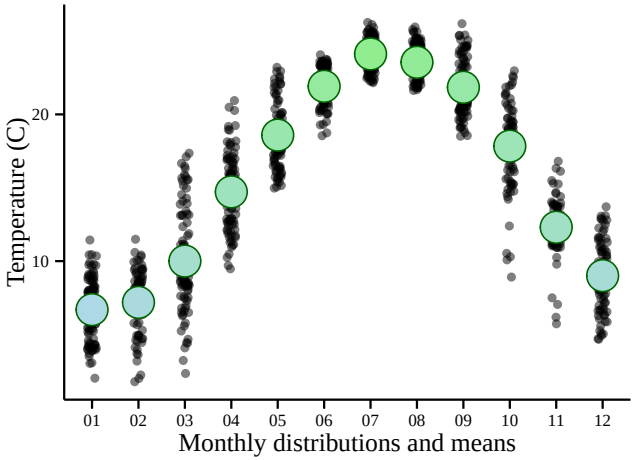
- `swmpr_in`: input swmpr object
- `param`: parameter to summarize
- `years`: years to plot

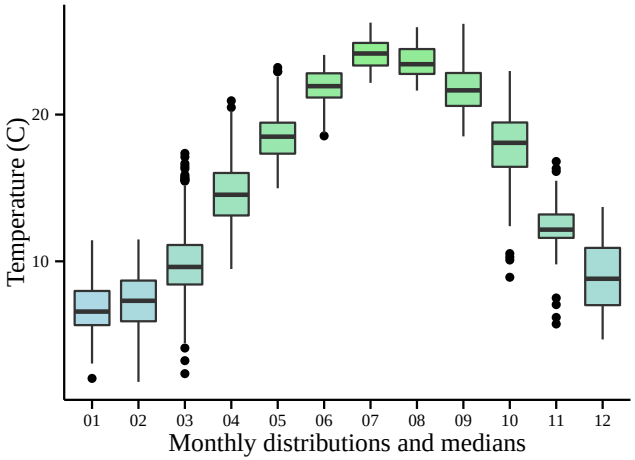
Import all years of data for cbmip, plot a summary of temperature (`?plot_summary`)

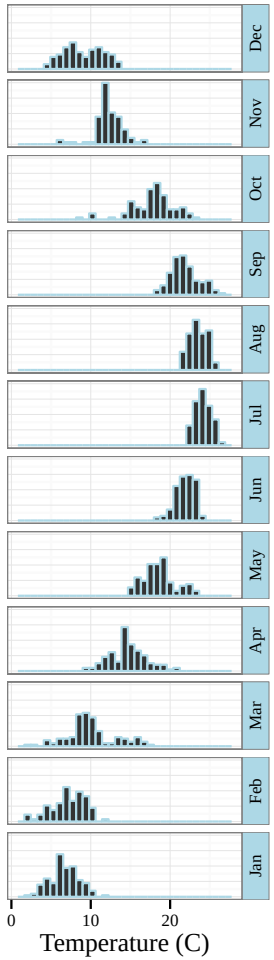
```
# import
mypath <- 'zip_ex'
dat <- import_local(mypath, 'cbmipwq')

# plot
plot_summary(dat, 'temp')
```

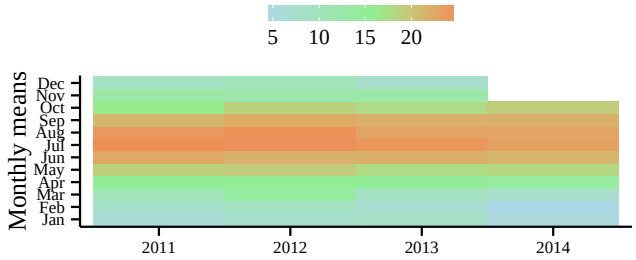
Plotting functions in SWMPr

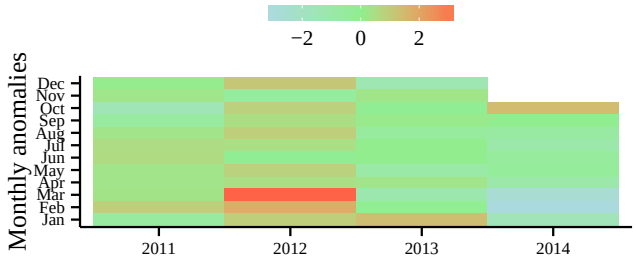
```
# plot
plot_summary(dat, 'temp')
```

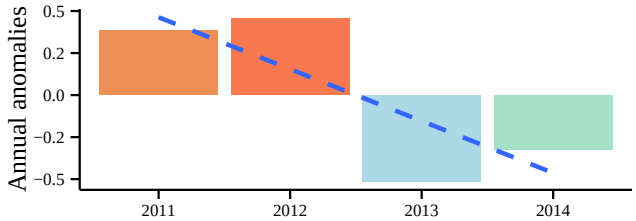












Additional resources

You've been exposed to some basic tools to ***retrieve***, ***organize***, and ***analyze*** SWMP data with SWMPr

There are multiple resources available for continued learning:

- swmprats.net
  - User forum to post questions, plot of the month
  - Widgets for data viz
  - Access to workshop materials
  - Access to 2014 workshop materials
- SWMPr online reference manual: list of all functions
- SWMPr cookbook: step-by-step scripts as stand-alone analyses
- Email instructors: beck.marcus@epa.gov, todd.obrien@noaa.gov

Questions??