SWMPr overview, retrieve, and organize

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

- Why and what is SWMPr?
- How can data get from CDMO into R using SWMPr?
- What is the basic structure of a `swmpr` data object?
- What is data organization and how can SWMPr help?

Interactive portion

We will use the swmpr1.Rproj project for this session, double-click to open in RStudio

- location on flash drive
- location online

Interactive portion

We will use the swmpr1.Rproj project for this session, double-click to open in RStudio

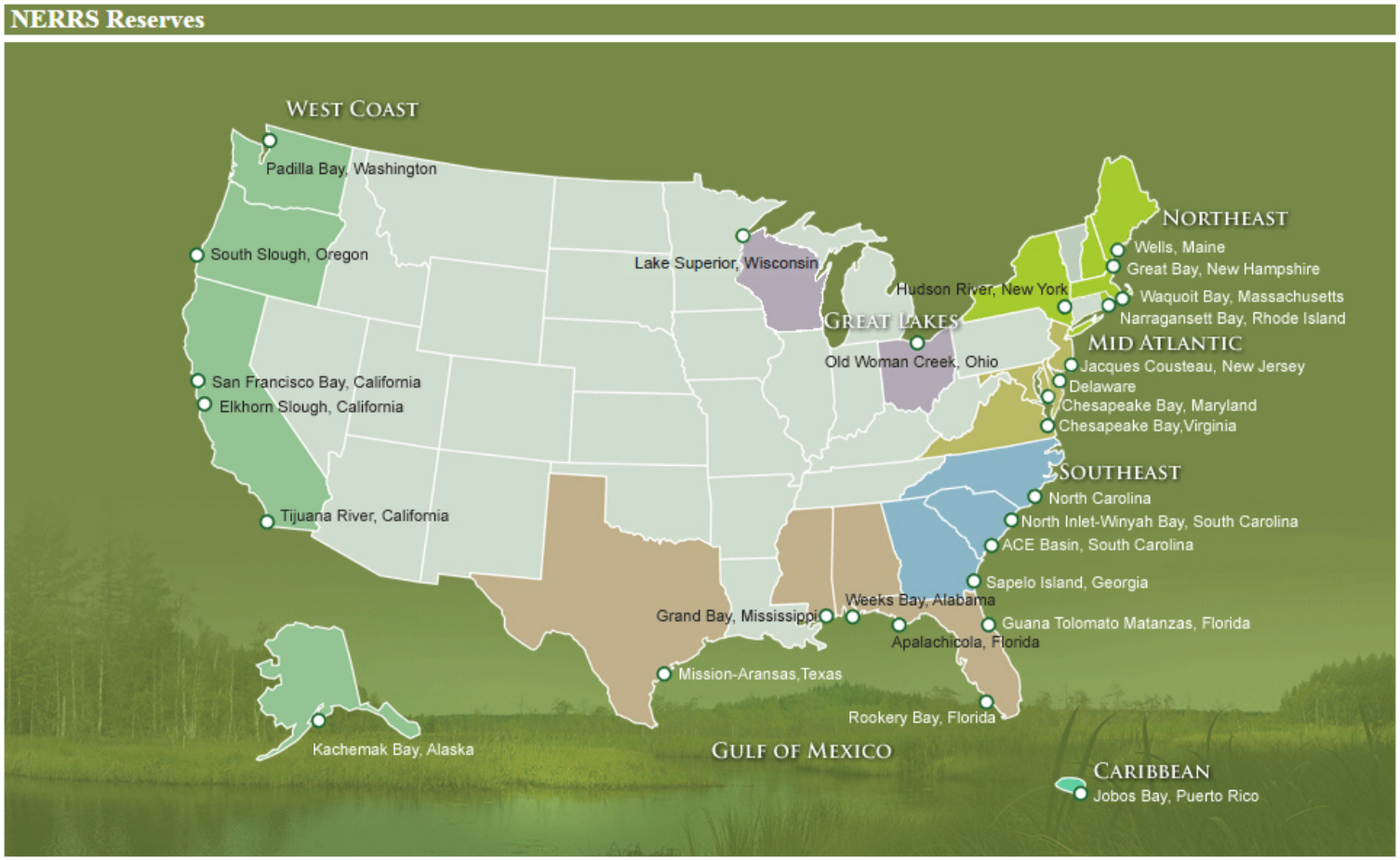
- location on flash drive
- location online

You will run examples whenever you see this guy:

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

Why and what is SWMPr?

SWMP - System Wide Monitoring Program, initiated in 1995 to provide continuous monitoring data at over 140 stations in 28 US estuaries



http://nerrs.noaa.gov/ReservesMap.aspx

Why and what is SWMPr?

CDMO is your one-stop shop for retrieving SWMP data

Why and what is SWMPr?

The raw data will look like this...

	A	B	C	D	E	F	G	H	I	J	K	L
1	StationCo	isSWMP	DateTimeStamp	Historical	Provisiona	CollMetho	REP	F_Record	PO4F	F_PO4F	NH4F	F_NH4F
2	apacpnut	P	1/10/2012 10:20	0	1	1	1		0.003	<-4> [SBL]	0.03	<0>
3	apacpnut	P	2/7/2012 11:41	0	1	1	1		0.005	<0>	0.019	<0>
4	apacpnut	P	3/5/2012 11:51	0	1	1	1		0.003	<-4> [SBL]	0.041	<0>
5	apacpnut	P	4/4/2012 10:30	0	1	1	1		0.003	<-4> [SBL]	0.043	<0>
6	apacpnut	P	5/9/2012 10:12	0	1	1	1		0.003	<0>	0.053	<0>
7	apacpnut	P	5/9/2012 10:15	0	1	1	2		0.003	<-4> [SBL]	0.022	<0>
8	apacpnut	P	5/9/2012 10:20	0	1	1	3		0.003	<0>	0.016	<0>
9	apacpnut	P	6/5/2012 8:30	0	1	1	1		0.003	<-4> [SBL]	0.04	<0>
10	apacpnut	P	7/3/2012 9:58	0	1	1	1	{CSM}	0.004	<0>	0.094	<0>
11	apacpnut	P	7/3/2012 9:59	0	1	1	2	{CSM}	0.004	<0>	0.066	<0>
12	apacpnut	P	7/3/2012 10:01	0	1	1	3	{CSM}	0.005	<0>	0.069	<0>
13	apacpnut	P	8/7/2012 9:53	0	1	1	1	{CSM}	0.003	<-4> [SBL]	0.05	<0>
14	apacpnut	P	9/5/2012 10:56	0	1	1	1		0.003	<-4> [SBL]	0.026	<0>
15	apacpnut	P	10/2/2012 9:22	0	1	1	1		0.003	<-4> [SBL]	0.042	<0>
16	apacpnut	P	10/2/2012 9:27	0	1	1	2		0.003	<-4> [SBL]	0.024	<0>
17	apacpnut	P	10/2/2012 9:32	0	1	1	3		0.003	<0>	0.042	<0>
18	apacpnut	P	11/6/2012 10:30	0	1	1	1		0.003	<-4> [SBL]	0.07	<0>
19	apacpnut	P	11/26/2012 11:39	0	1	1	1		0.003	<-4> [SBL]	0.041	<0>

Why and what is SWMPr?

What are the challenges for evaluating SWMP data?

Why and what is SWMPr?

What are the challenges for evaluating SWMP data?

- Knowing what we want
- Dealing with QAQC columns and removing 'bad' observations
- Data we don't want... extra columns or irrelevant parameters
- Combining data for comparison
- Issues inherent with time series, e.g., missing data
- Others?

Why and what is SWMPr?

What: An R package to ***augment*** existing CDMO services and to provide a ***bridge*** to analysis

Why and what is SWMPr?

What: An R package to ***augment*** existing CDMO services and to provide a ***bridge*** to analysis

Why: There are many challenges working with SWMP data... a toolkit for addressing these challenges will be useful

Why and what is SWMPr?

What: An R package to ***augment*** existing CDMO services and to provide a ***bridge*** to analysis

Why: There are many challenges working with SWMP data... a toolkit for addressing these challenges will be useful

How: Use the SWMPr functions to ***retrieve***, ***organize***, and ***analyze*** SWMP data

Why and what is SWMPr?

Some housekeeping...

```
# install from CRAN (only do once)
install.packages('SWMPr')

# load for your current session
library(SWMPr)
```

https://cran.r-project.org/web/packages/SWMPr/index.html

Why and what is SWMPr?

Uses an ***object-oriented*** structure... data are imported into R as a `swmpr` data object, with functions built to use this object

What are the ***retrieve***, ***organize***, and ***analyze*** functions?

Run this code one line at a time... What comes up?

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

Why and what is SWMPr?

Uses an ***object-oriented*** structure... data are imported into R as a `swmpr` data object, with functions built to use this object

What are the ***retrieve***, ***organize***, and ***analyze*** functions?

Run this code one line at a time... What comes up?

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

What about this?

```
?import_local
```

What pieces of information are in the help file?

Getting SWMP data into R

Let's get some data into R!

The ***retrieval*** functions do two things:

Import data directly from the CDMO:

```
all_params
all_params_dtrng
single_param
site_codes
site_codes_ind
```

These functions require registering your IP address with CDMO

Import data from a local path:

```
import_local
```

Imports data obtained from (and only from) the zip downloads feature

Getting SWMP data into R 

The ‘zip_ex‘ folder in the project is a sample dataset that looks exactly like a folder you get from CDMO

Let’s import some data from that folder, try to import `‘apacpwq’`...

Getting SWMP data into R 

The ‘zip_ex‘ folder in the project is a sample dataset that looks exactly like a folder you get from CDMO

Let’s import some data from that folder, try to import ‘apacpwq’...

```
# get data for apacpwq, all years

# location of data
mypath <- 'zip_ex'

# import and assign to 'dat'
dat <- import_local(mypath, 'apacpwq', trace = T)
```

Getting SWMP data into R 

The ‘zip_ex‘ folder in the project is a sample dataset that looks exactly like a folder you get from CDMO

Let’s import some data from that folder, try to import ‘apacpwq’...

```
# get data for apacpwq, all years

# location of data
mypath <- 'zip_ex'

# import and assign to 'dat'
dat <- import_local(mypath, 'apacpwq', trace = T)
```

What about this?

```
dat2 <- import_local(mypath, 'apacp2012', trace = T)
dat3 <- import_local(mypath, 'apadbnut', trace = F)
```

Structure of the swmpr data object

Now we have data in our 'workspace' that we can organize/analyze

Try running the following...

```
head(dat)
tail(dat)
View(dat)
str(dat)
attributes(dat)
```

Structure of the swmpr data object 

Now we have data in our 'workspace' that we can organize/analyze

Try running the following...

```
head(dat)
tail(dat)
View(dat)
str(dat)
attributes(dat)
```

How are the data organized?

What are the column names?

What are the attributes?

Structure of the swmpr data object

The swmpr object is a data.frame and a list of attributes

```
head(dat, 3)

##          datetimestamp temp f_temp spcond f_spcond sal f_sal do_pct f_do_pct
## 1 2011-01-01 00:00:00   11   <0>      44     <0>   28  <0>      68     <0>
## 2 2011-01-01 00:15:00   11   <0>      44     <0>   28  <0>      68     <0>
## 3 2011-01-01 00:30:00   11   <0>      44     <0>   28  <0>      68     <0>
##   do_mgl f_do_mgl depth f_depth cdepth f_cdepth level f_level clevel f_clevel
## 1      6     <0>      2    <0>       2     <3>     NA   <-1>      NA        NA
## 2      6     <0>      2    <0>       2     <3>     NA   <-1>      NA        NA
## 3      6     <0>      2    <0>       2     <3>     NA   <-1>      NA        NA
##   ph f_ph turb f_turb chlfluor f_chlfluor
## 1  8 <0>     3   <0>        NA      <-1>
## 2  8 <0>     3   <0>        NA      <-1>
## 3  8 <0>     2   <0>        NA      <-1>

names(attributes(dat))

## [1] "names"       "row.names"   "class"       "station"     "parameters"
## [6] "qaqc_cols"   "date_rng"    "timezone"    "stamp_class"

attr(dat, 'parameters')

##  [1] "temp"     "spcond"   "sal"      "do_pct"   "do_mgl"   "depth"
##  [7] "cdepth"   "level"    "clevel"   "ph"       "turb"     "chlfluor"
```

Data organization with SWMPr

First problem is solved... we know how to get SWMP data from CDMO into R:

- Download a dataset from zip downloads
- Find where the data have downloaded
- Import using `import_local`
- Have a look at the data (`head`, `View`, `attributes`)
- Lost? Check the help files: `?import_local`

Now we can think about preprocessing or organizing prior to analysis

Data organization with SWMPr

What are the challenges for evaluating SWMP data?

Data organization with SWMPr

What are the challenges for evaluating SWMP data?

- Knowing what we want
- Dealing with QAQC columns and removing 'bad' observations
- Data we don't want... extra columns or irrelevant parameters
- Combining data for comparison
- Issues inherent with time series, e.g., missing data
- Others?

Data organization with SWMPr 

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

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

Data organization with SWMPr 

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

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

Which function would you use first?

Which would you use to reduce data volume or select certain variables?

Can any be used to combine `swmpr` data objects?

Data organization with SWMPr 

Perhaps you want to deal with QAQC columns first...

From the zips folder, import all of the weather data for apaebmet (`?import_local`)

Data organization with SWMPr 

Perhaps you want to deal with QAQC columns first...

From the zips folder, import all of the weather data for apaebmet (`?import_local`)

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

Data organization with SWMPr 

Perhaps you want to deal with QAQC columns first...

From the zips folder, import all of the weather data for apaebmet (`?import_local`)

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

View the data, what are the columns?

Try running `qaqc` (`?qaqc`) and view again, what happened?

Data organization with SWMPr 

Perhaps you want to deal with QAQC columns first...

From the zips folder, import all of the weather data for apaebmet (`?import_local`)

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

View the data, what are the columns?

Try running `qaqc` (`?qaqc`) and view again, what happened?

```
View(dat)
dat2 <- qaqc(dat)
View(dat2)
```

Data organization with SWMPr 

Try playing with the `qaqc_keep` argument (`?qaqc`)...

How are these different?

```
# different options for qaqc
dat2 <- qaqc(dat)
dat3 <- qaqc(dat, qaqc_keep = c('0', '-1'))
dat4 <- qaqc(dat, qaqc_keep = NULL)
dat5 <- qaqc(dat, qaqc_keep = 'CSM')
```

Data organization with SWMPr

Try playing with the `qaqc_keep` argument (`?qaqc`)...

How are these different?

```
# different options for qaqc
dat2 <- qaqc(dat)
dat3 <- qaqc(dat, qaqc_keep = c('0', '-1'))
dat4 <- qaqc(dat, qaqc_keep = NULL)
dat5 <- qaqc(dat, qaqc_keep = 'CSM')
```

Changes are hard to visualize for lots of data - as a proof of concept, try running `qaqcchk` on any of the datasets

```
qaqcchk(dat)
qaqcchk(dat2)
qaqcchk(dat3)
qaqcchk(dat4)
qaqcchk(dat5)
```

Data organization with SWMPr 

We'll continue with the water quality data for apadb - import again and run the `qaqc` function

Data organization with SWMPr

We'll continue with the water quality data for apadb - import again and run the `qaqc` function

```
# import apawq
mypath <- 'zip_ex'
dat <- import_local(mypath, 'apadbwq')
dat <- qaqc(dat)
```

Data organization with SWMPr 

We'll continue with the water quality data for apadb - import again and run the `qaqc` function

```
# import apawq
mypath <- 'zip_ex'
dat <- import_local(mypath, 'apadbwq')
dat <- qaqc(dat)
```

What is the next logical step after dealing with QAQC values?

How would we further want to organize the data?

Data organization with SWMPr

We'll continue with the water quality data for apadb - import again and run the `qaqc` function

```
# import apawq
mypath <- 'zip_ex'
dat <- import_local(mypath, 'apadbwq')
dat <- qaqc(dat)
```

What is the next logical step after dealing with QAQC values?

How would we further want to organize the data?

Maybe we want to subset the data...

```
# view help file
?subset.swmpr
```

Data organization with SWMPr

The `subset` function has several arguments (help file `?subset.swmpr`)

Not all are necessary for every task

- `swmpr_in`: input data (`swmpr` object)
- `subset`: dates to keep
- `select`: parameters to keep
- `operator`: less than, greater than, etc. if only one date in subset
- `rem_rows`: remove empty rows
- `rem_cols`: remove empty columns

Data organization with SWMPr

The `select` argument of `subset` is used to select parameters of interest - one to many

```
# select the DO column
tmp <- subset(dat, select = 'do_mgl')
head(tmp)
```

```
##          datetimestamp do_mgl
## 1 2011-01-01 00:00:00     NA
## 2 2011-01-01 00:15:00     NA
## 3 2011-01-01 00:30:00     NA
## 4 2011-01-01 00:45:00     NA
## 5 2011-01-01 01:00:00     NA
## 6 2011-01-01 01:15:00     NA
```

Selecting more than one column...

Data organization with SWMPr

The `select` argument of `subset` is used to select parameters of interest - one to many

```
# select the DO column
tmp <- subset(dat, select = 'do_mgl')
head(tmp)
```

```
##          datetimestamp do_mgl
## 1 2011-01-01 00:00:00     NA
## 2 2011-01-01 00:15:00     NA
## 3 2011-01-01 00:30:00     NA
## 4 2011-01-01 00:45:00     NA
## 5 2011-01-01 01:00:00     NA
## 6 2011-01-01 01:15:00     NA
```

Selecting more than one column...

```
# select DO and salinity
tmp <- subset(dat, select = c('do_mgl', 'sal'))
head(tmp)
```

Data organization with SWMPr

The `subset` argument of `subset.swmpr` selects a date range

The dates must have a specific format: 'YYYY-mm-dd HH:MM'

```
# select a date range, July 2012
dates <- c('2012-07-01 12:00', '2012-07-31 6:30')
tmp <- subset(dat, subset = dates)
head(tmp) # view first six rows
```

```
##          datetimestamp temp spcond sal do_pct do_mgl depth cdepth level clevel
## 1 2012-07-01 12:00:00   NA     50  33    104      7     2     NA    NA     NA
## 2 2012-07-01 12:15:00   NA     50  33    101      7     2     NA    NA     NA
## 3 2012-07-01 12:30:00   NA     50  33    104      7     2     NA    NA     NA
## 4 2012-07-01 12:45:00   NA     50  33    104      7     2     NA    NA     NA
## 5 2012-07-01 13:00:00   NA     50  33    104      7     2     NA    NA     NA
## 6 2012-07-01 13:15:00   NA     52  34    104      7     2     NA    NA     NA
##   ph turb chlfluor
## 1  8    3       NA
## 2  8   11       NA
## 3  8    8       NA
## 4  8   10       NA
## 5  8   15       NA
## 6  8   12       NA
```

Data organization with SWMPr

- Import the weather data at apaeb
- Deal with QAQC columns
- Select two columns of interest
- Subset a date range

Data organization with SWMPr

- Import the weather data at apaeb

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

- Deal with QAQC columns

- Select two columns of interest

- Subset a date range

Data organization with SWMPr

- Import the weather data at apaeb

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

- Deal with QAQC columns

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

- Select two columns of interest

- Subset a date range

Data organization with SWMPr

- Import the weather data at apaeb

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

- Deal with QAQC columns

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

- Select two columns of interest

```
tmp <- subset(tmp, select = c('temp', 'wind'))
```

- Subset a date range

Data organization with SWMPr

- Import the weather data at apaeb

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

- Deal with QAQC columns

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

- Select two columns of interest

```
tmp <- subset(tmp, select = c('temp', 'wind'))
```

- Subset a date range

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

Data organization with SWMPr

- Import the weather data at apaeb

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

- Deal with QAQC columns

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

- Select two columns of interest

```
tmp <- subset(tmp, select = c('temp', 'wind'))
```

- Subset a date range

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

- Bonus: can you select all observations before or after a date?

Data organization with SWMPr

- Import the weather data at apaeb

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

- Deal with QAQC columns

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

- Select two columns of interest

```
tmp <- subset(tmp, select = c('temp', 'wind'))
```

- Subset a date range

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

- Bonus: can you select all observations before or after a date?

```
# get observations after Jan 1, 2013
dates <- '2013-01-01 00:00'
tmp <- subset(dat, subset = dates, operator = '>=')
```

Data organization with SWMPr

A final nod to the `comb` and `setstep` functions

Run the following, view the results, discuss with your neighbors:

```
mypath <- 'zip_ex'
dat_met <- import_local(mypath, 'apaebmet')
dat_met <- qaqc(dat_met)
dat_wq <- import_local(mypath, 'apadbwq')
dat_wq <- qaqc(dat_wq)

# what does this do (hint: use View to see the data)?
tmp1 <- comb(dat_wq, dat_met, timestep = 120)
```

Data organization with SWMPr

A final nod to the `comb` and `setstep` functions

Run the following, view the results, discuss with your neighbors:

```
mypath <- 'zip_ex'
dat_met <- import_local(mypath, 'apaebmet')
dat_met <- qaqc(dat_met)
dat_wq <- import_local(mypath, 'apadbwq')
dat_wq <- qaqc(dat_wq)

# what does this do (hint: use View to see the data)?
tmp1 <- comb(dat_wq, dat_met, timestep = 120)
```

Now try this...

```
tmp2 <- setstep(dat_wq, timestep = 60)
```

What happened?

Organize SWMP data

The `setstep` function is used to standardize the time step of a `swmpr` object

The `comb` function is used to combine `swmpr` objects

`setstep` is used within `comb` so you should not have to use it directly

Arguments for `comb`:

- ... : input swmpr data, separated by comma
- timestep: minutes defining the standardized time step
- differ: maximum difference in minutes for matching observations with original time steps to standardized time steps
- method: how the data are combined using the time stamps - union, intersect, or using a station

Organize SWMP data

A final note about combining... what about combining data with different ***time ranges***?

Consider combining two datasets

The `method` argument of allows flexibility under different scenarios - time range `'intersect'`, `'union'`, or range of one station

Organize SWMP data

A final note about combining... what about combining data with different ***time ranges***?

Consider combining two datasets

- ***Scenario 1***: Time ranges are the same

The `method` argument of allows flexibility under different scenarios - time range `'intersect'`, `'union'`, or range of one station

Organize SWMP data

A final note about combining... what about combining data with different ***time ranges***?

Consider combining two datasets

- ***Scenario 1***: Time ranges are the same
  - all methods can be used, same results for each

The `method` argument of allows flexibility under different scenarios - time range `'intersect'`, `'union'`, or range of one station

Organize SWMP data

A final note about combining... what about combining data with different ***time ranges***?

Consider combining two datasets

- ***Scenario 1***: Time ranges are the same
  - all methods can be used, same results for each
- ***Scenario 2***: Time ranges are not the same, but there is overlap

The `method` argument of allows flexibility under different scenarios - time range `'intersect'`, `'union'`, or range of one station

Organize SWMP data

A final note about combining... what about combining data with different ***time ranges***?

Consider combining two datasets

- ***Scenario 1***: Time ranges are the same
  - all methods can be used, same results for each
- ***Scenario 2***: Time ranges are not the same, but there is overlap
  - all methods can be used, different results for each

The `method` argument of allows flexibility under different scenarios - time range `'intersect'`, `'union'`, or range of one station

Organize SWMP data

A final note about combining... what about combining data with different ***time ranges***?

Consider combining two datasets

- ***Scenario 1***: Time ranges are the same
  - all methods can be used, same results for each
- ***Scenario 2***: Time ranges are not the same, but there is overlap
  - all methods can be used, different results for each
- ***Scenario 3***: Time ranges are not the same, there is no overlap

The `method` argument of allows flexibility under different scenarios - time range `'intersect'`, `'union'`, or range of one station

Organize SWMP data

A final note about combining... what about combining data with different ***time ranges***?

Consider combining two datasets

- ***Scenario 1***: Time ranges are the same
  - all methods can be used, same results for each
- ***Scenario 2***: Time ranges are not the same, but there is overlap
  - all methods can be used, different results for each
- ***Scenario 3***: Time ranges are not the same, there is no overlap
  - only `'union'` will work

The `method` argument of allows flexibility under different scenarios - time range `'intersect'`, `'union'`, or range of one station

Summary

Now you have an idea of how to organize SWMP data for analysis!

Here's what we did:

- ***Import*** SWMP data into R
- Evaluate and ***handle QAQC*** flags in the data
- ***Subset*** to select variables or time ranges of interest
- ***Combine*** data for comparison or data simplification

Consult the SWMP cookbook for an example workflow

Check the help files for usage (reference manual on CRAN)

A final exercise 

Import 2012 apadbwq and apaebmet data, deal with ***QAQC***, ***subset*** one month of data and one variable of interest, ***combine*** at hourly time step and intersect method, and ***plot*** two variables against each other...

A final exercise 

Import 2012 apadbwq and apaebmet data, deal with ***QAQC***, ***subset*** one month of data and one variable of interest, ***combine*** at hourly time step and intersect method, and ***plot*** two variables against each other...

`?import_local`, `?qaqc`, `?subset.swmpr`, `?comb`, `?plot`

A final exercise 

Import 2012 apadbwq and apaebmet data, deal with ***QAQC***, ***subset*** one month of data and one variable of interest, ***combine*** at hourly time step and intersect method, and ***plot*** two variables against each other...

`?import_local`, `?qaqc`, `?subset.swmpr`, `?comb`, `?plot`

```
wqdat <- import_local('zip_ex', 'apadbwq2012')
wqdat <- qaqc(wqdat)
wqdat <- subset(wqdat, select = 'turb',
  subset = c('2012-08-01 0:0', '2012-08-31 0:0'))
metdat <- import_local('zip_ex', 'apaebmet2012')
metdat <- qaqc(metdat)
metdat <- subset(metdat, select = 'wspd',
 subset = c('2012-08-01 0:0', '2012-08-31 0:0'))
dat <- comb(wqdat, metdat, method = 'intersect', timestep = 60)
plot(turb ~ wspd, data = dat)
```

A final exercise 

Import 2012 apadbwq and apaebmet data, deal with ***QAQC***, ***subset*** one month of data and one variable of interest, ***combine*** at hourly time step and intersect method, and ***plot*** two variables against each other...

`?import_local`, `?qaqc`, `?subset.swmpr`, `?comb`, `?plot`

```
wqdat <- import_local('zip_ex', 'apadbwq2012')
wqdat <- qaqc(wqdat)
wqdat <- subset(wqdat, select = 'turb',
  subset = c('2012-08-01 0:0', '2012-08-31 0:0'))
metdat <- import_local('zip_ex', 'apaebmet2012')
metdat <- qaqc(metdat)
metdat <- subset(metdat, select = 'wspd',
 subset = c('2012-08-01 0:0', '2012-08-31 0:0'))
dat <- comb(wqdat, metdat, method = 'intersect', timestep = 60)
plot(turb ~ wspd, data = dat)
```

Bonus: can you plot a regression line on this plot? Hint: `?lm`, `?abline`

After break... what are some ways we can analyze or visualize the data?

Questions??