

Procedures and R Scripts for QCing, Formatting and Deriving Summary Outputs for Continuous Temperature and Hydrologic Data (for Beginning R Users)



National Water Quality Monitoring Conference

Erik Leppo, Jen Stamp, Tetra Tech, Inc.

Britta Bierwagen, David Gibbs, EPA/ORD

The views expressed in this presentation are those of the author and they do not necessarily reflect the views or policies of the U.S. Environmental Protection Agency or other collaborating agencies

Agenda

- | | |
|-----------|--|
| 2:00-2:10 | Intro/background |
| 2:10-2:20 | Primer on R |
| 2:20-3:00 | Test dataset #1 - Non-vented pressure transducer
(ECO66G12_AW files) <ul style="list-style-type: none">• Modify (reformat, rename) 'raw' files (10 minutes)• QC script (15 minutes)• Aggregate (5 minutes)• Stats scripts (10 minutes) |
| 3:00-3:20 | Test dataset #2 – Standalone temperature sensors
(WSR air + water temperature files) <ul style="list-style-type: none">• Review 'raw' files (formatting, naming; show example of discrete data) (5 minutes)• QC script (5 minutes)• Aggregate (5 minutes)• Stats scripts (5 minutes) |
| 3:20-3:30 | Question and answer |
- Extras (if time permits)
- GetGage script (won't have internet; will use screenshots)
 - Adding gage data into Test dataset #2
 - Changing thresholds

Why did we develop these scripts?

Working with **EPA Regional offices, states, tribes** and other biomonitoring programs in the **Northeast, Mid-Atlantic, Southeast & Midwest** on **Regional Monitoring Networks (RMNs)**





Regional Monitoring Networks (RMNs)

Objective: detect potentially small, **climate-related trends** at a **regional scale**, in a **decision-relevant timeframe**, in the context of routine biomonitoring

Sampling goals:

- At least **30 sites with similar environmental and biological characteristics in each region**
 - Average 5 sites per state (range:1-15)
 - Design informed by power analyses on Northeast dataset (2012)
- Sample on an **annual basis for 10 or more years**, using as **consistent and comparable** methods as possible (QAPP)
- Initial focus is on **freshwater wadeable streams; could expand to inland lakes and wetlands**

Data Collection Targets



- **Biological:** macroinvertebrates.
 - Optional: fish and periphyton

- **Temperature:** year-round water and air temperature
- **Hydrologic:** year-round water level data (from USGS gages or pressure transducers), ideally converted to discharge.

- **Habitat**
- **Water chemistry**



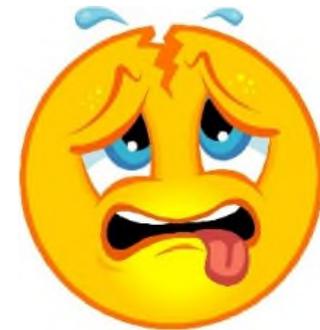
RMN partners have varying levels of experience with continuous sensors

Data Management



Ok great! now we are collecting the desired data....

But what are we going to do with all these data?





Challenges!

Lots of different data sources –

Data being collected with **different types of equipment**, sometimes at **different recording intervals**, sometimes in **different formats**

- **Standalone temperature** sensors (e.g. water and air temperature come in separate files)
- **Non-vented pressure transducers** (e.g., temperature and water level data together in one output; sometimes air temperature in a separate file)
- **Vented pressure transducers**
- **USGS discharge data** (needs to be pulled from USGS website and combined with temperature data)



Challenges!



Where to put the data?

There is not a single existing system that can accommodate all of the types of data being collecting at RMN sites.

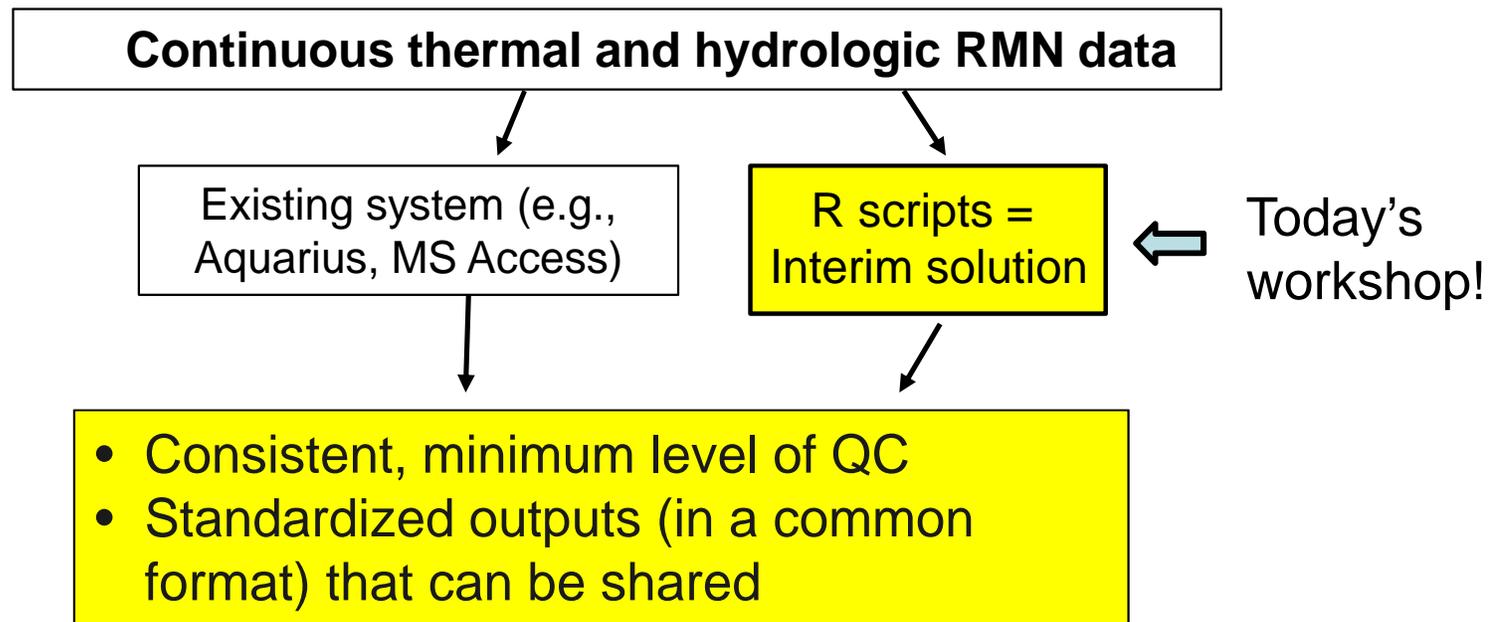
For continuous data, there is an ongoing multi-agency effort (led by Dwane Young, EPA) to address this but full implementation is a few years off...

Until a permanent data management system is in place, the individual organizations will be the custodians and owners of the continuous thermal and hydrologic monitoring data.

- Some people already have a system in place (e.g., Aquarius)
- Some do not

Overall Goals of the R scripts

- Ensure that a certain **(minimum) level of QC** is performed on the continuous thermal and hydrologic RMN data
- Produce **standardized sets of outputs** (summary statistics, plots) that can be shared



Why R?



Our preliminary exploration of performing QC checks and summary statistic calculations in Excel and Access proved too burdensome. Some of the **biggest hurdles** were –

- Excel and Access getting **bogged down by the large data files**, or maxing out capacity
- **Formatting issues** (e.g., inconsistencies across files)
- **Difficulties with combining the different data files**

We did our best **to keep procedures and scripts as simple and generic as possible**, and set them up in a way that **people with limited R experience can understand and run.**

We also tried to utilize existing scripts and resources whenever possible...



Disclaimers

- These R scripts should be regarded as a **proof of concept**. We're learning as we go and won't 'wow' you with these scripts.
- This is not necessarily the 'best' solution; rather this is the **best we could come up with at this time**, with the information we have.
- This was done with **limited time and resources**
 - We won't be able to do everything we'd like to do in this phase of work. Please give us feedback on what additional features you'd like us to add in. We're keeping a 'wish list' and hope to accomplish more through additional phases of work.
- These are **minimum QC standards**. Some organizations may choose to do more.

Primer on R



Show of hands...



- How many of you have worked with continuous sensor data before?
- How many of you have used R before (in any capacity)?
- How many of you have R and R Studio installed on your laptops?
 - If so, do you have R version 3.2?



What is R?

- R is available as Free Software
- It runs on a wide variety of platforms and systems (e.g., Windows and MacOS)
- The R language is widely used among statisticians and data miners for developing statistical software, data analysis and graphics
- R's popularity has increased substantially in recent years





Don't be intimidated!

There is a learning curve and you may run into some stumbling blocks initially, but you can do this!

We set these scripts up so that there are only a few places where you need to make entries (e.g., siteID and time period). Then it's just a matter of hitting the Run button, knowing where to find the outputs, and learning how to interpret the outputs.



Setting up R and R studio

IMPORTANT: make sure you are using R version 3.2 or later, otherwise you may not be able to download all the files necessary to run the script.

"R" and "RStudio" (free downloads)

R is the base program (we typically only install the 32-bit version) -

<https://cran.r-project.org/bin/windows/base/>

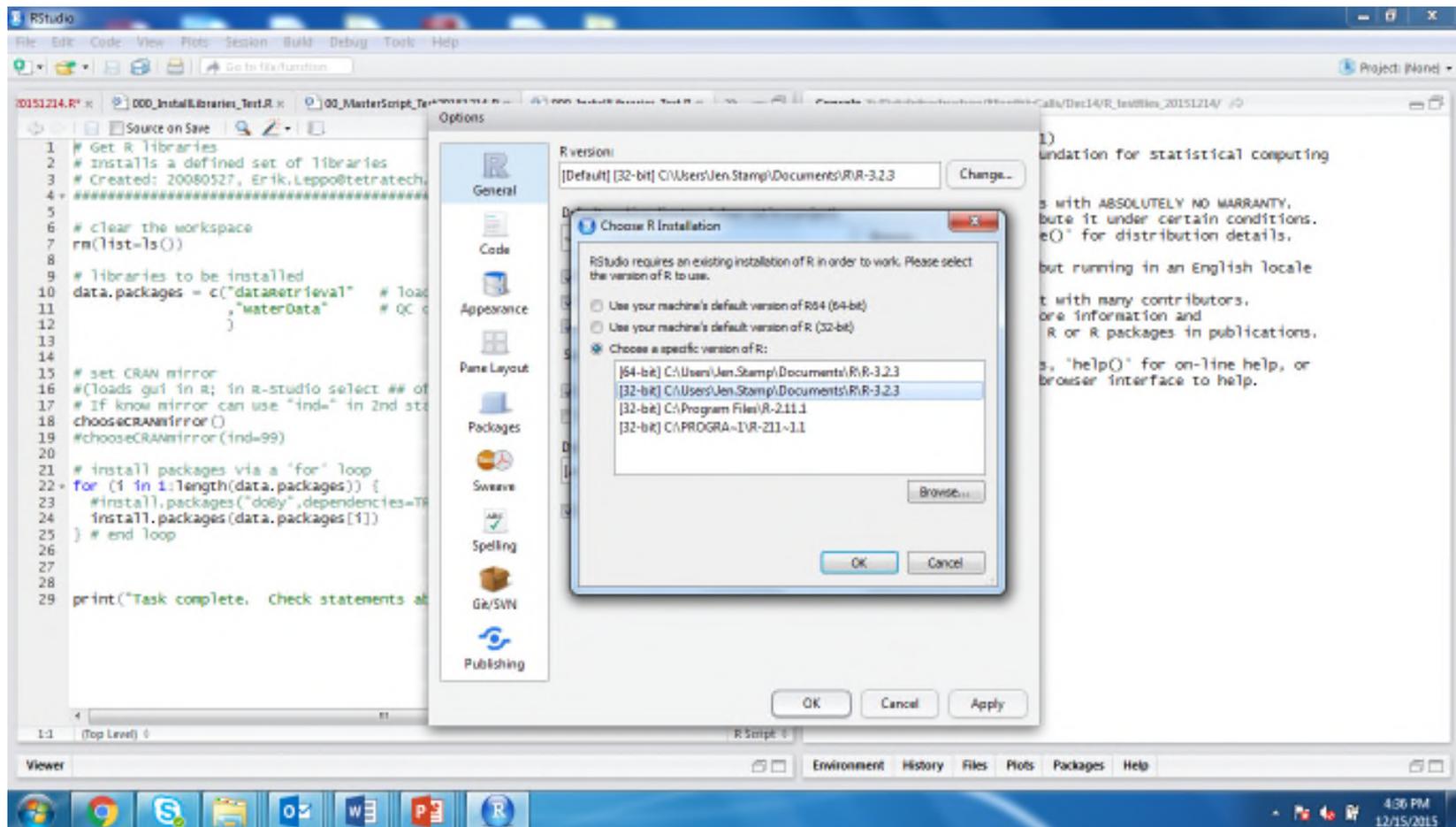
RStudio is a separate program but is a useful code editor and interface for R. Install it after installing R -

<https://www.rstudio.com/products/rstudio/download/>



Choose R Installation

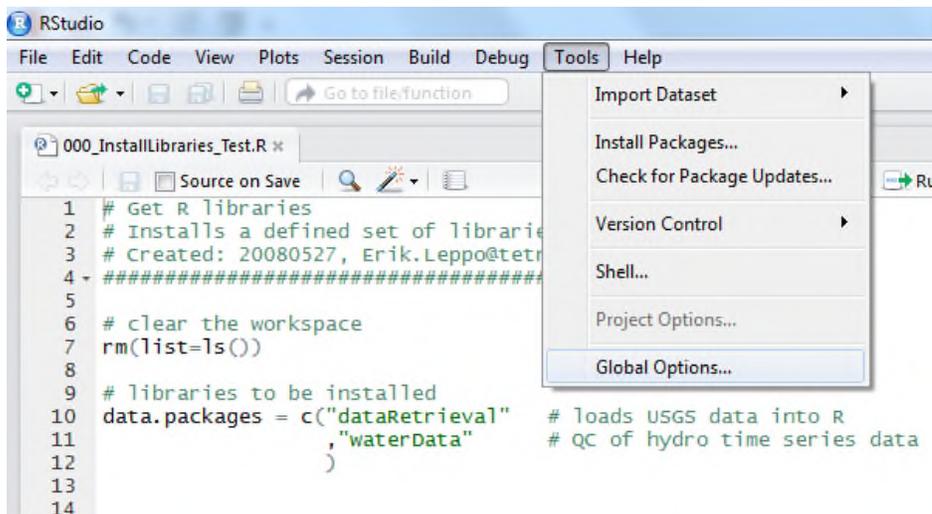
If you download the new version of R and still have an older version on your computer, it's possible that the script is still referencing your older version. If you suspect this is a problem, in RStudio, go to **Tools – Global Options – R General** and make sure it's choosing the correct R Installation.



Getting set up...

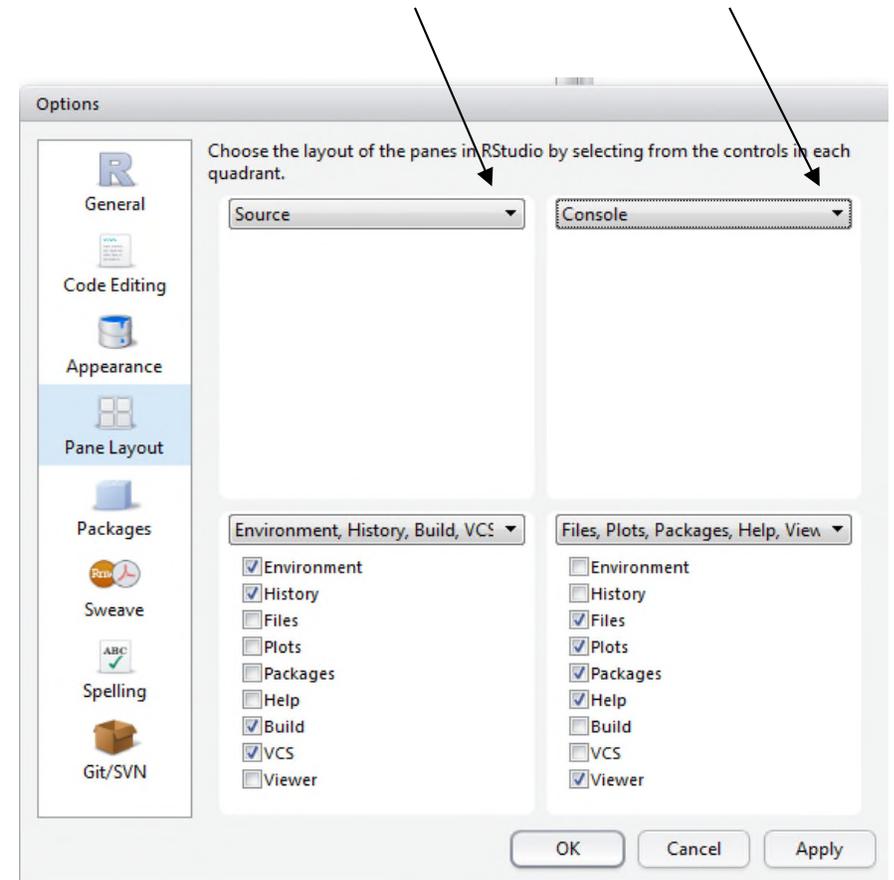
Pane layout - This is a matter of personal preference. We like having 2 panes visible - the **Source** in the left pane (which shows the script that is being run), and **Console** in the right pane (which shows the output from the script). But you can set this up however you like.

1. Go to Tools – Global options



2. Click on 'Pane layout'

3. Use drop-down lists to select Source on the left and Console on the right

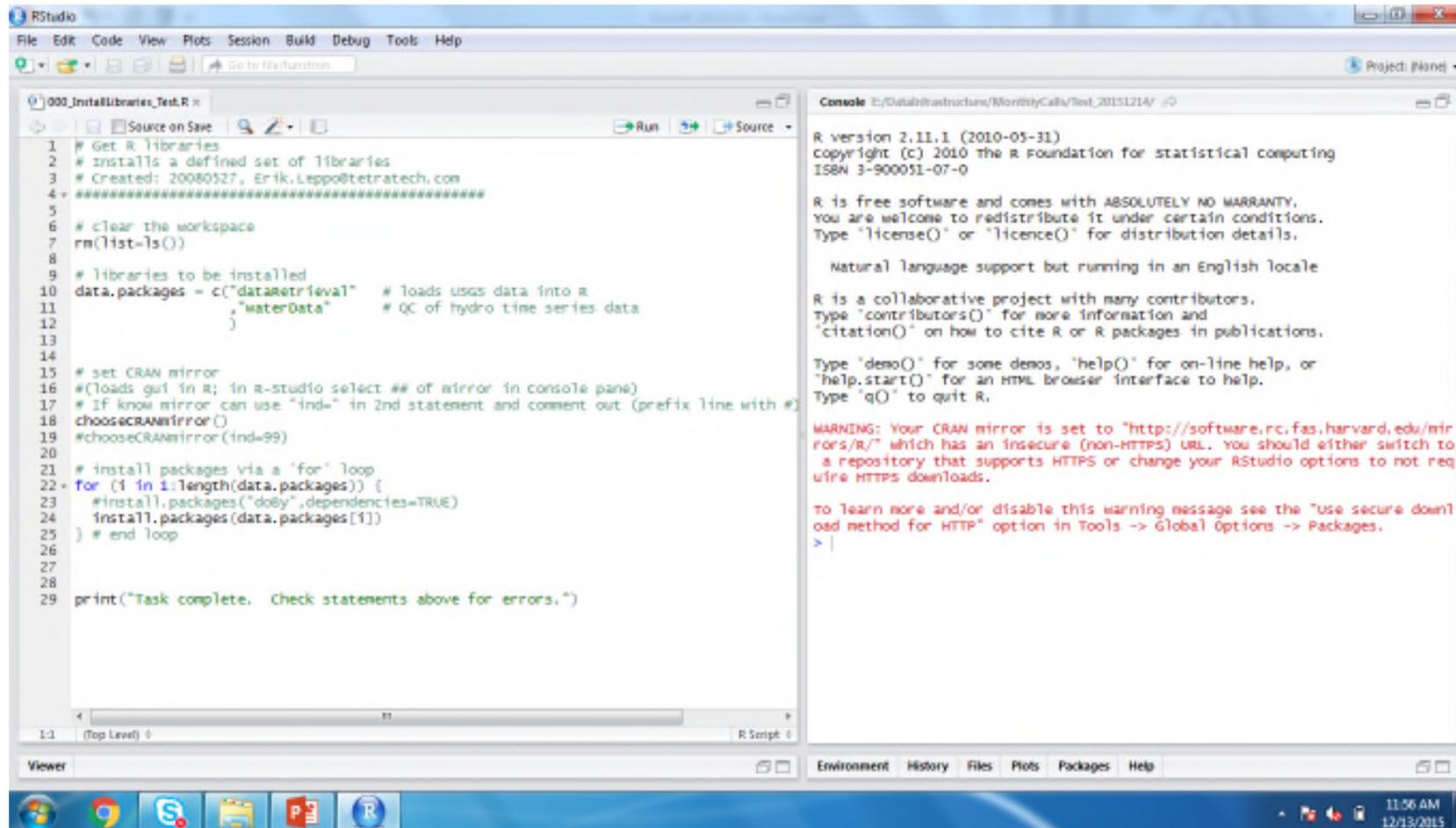


Getting set up...

If you have the **Source** in the left pane, and **Console** in the right pane, the screen should look like this...

(The Source pane shows the script that's being run)

(The Console pane shows the output from the script)



The screenshot shows the RStudio interface. The left pane (Source) contains an R script with the following content:

```
1 # Get R libraries
2 # installs a defined set of libraries
3 # Created: 20080527, Erik.Leppo@tetratech.com
4 #*****
5
6 # clear the workspace
7 rm(list=ls())
8
9 # libraries to be installed
10 data.packages = c("dataRetrieval" # loads usgs data into R
11                  ,"waterData" # QC of hydro time series data
12                  )
13
14
15 # set CRAN mirror
16 # (loads gui in R; in R-studio select ## of mirror in console pane)
17 # If know mirror can use "ind=" in 2nd statement and comment out (prefix line with #)
18 chooseCRANmirror()
19 #chooseCRANmirror(ind=99)
20
21 # install packages via a "for" loop
22 for (i in 1:length(data.packages)) {
23   #install.packages("doBy",dependencies=TRUE)
24   install.packages(data.packages[i])
25 } # end loop
26
27
28
29 print("Task complete. Check statements above for errors.")
```

The right pane (Console) shows the output of the script:

```
R version 2.11.1 (2010-05-31)
copyright (C) 2010 the R Foundation for statistical computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

WARNING: Your CRAN mirror is set to "http://software.rc.fas.harvard.edu/mir
rors/R/" which has an insecure (non-HTTPS) URL. You should either switch to
a repository that supports HTTPS or change your RStudio options to not req
uire HTTPS downloads.

To learn more and/or disable this warning message see the "use secure downl
oad method for HTTP" option in Tools -> Global Options -> Packages.

> |
```

Green = Comments (line starts with "#")

Blue = R command

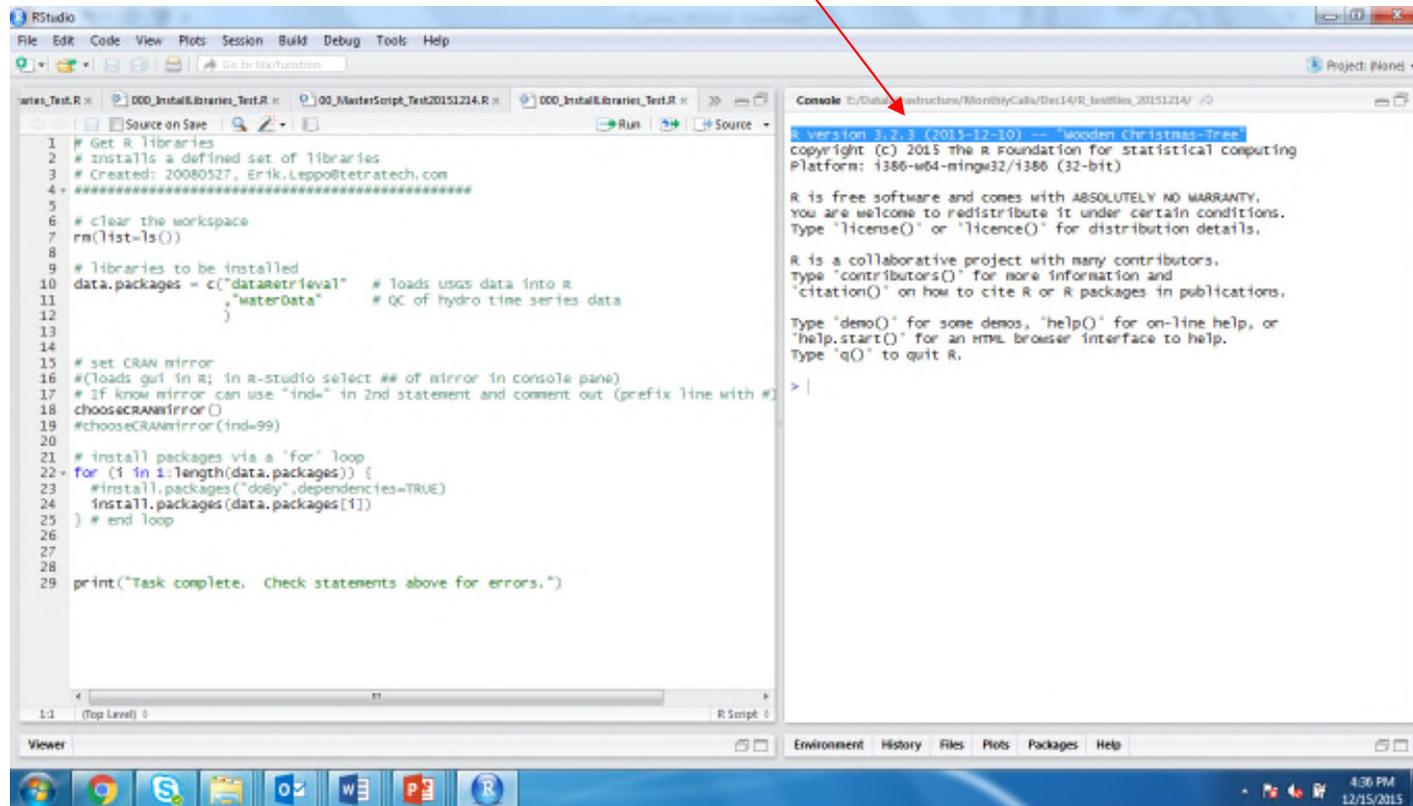
Red = Error and Status messages

Black = Everything else

Getting set up...

Verify which version of R you are using...

Check the top line in the Console (right panel) to see what version you are using.



The screenshot shows the RStudio interface. The left pane contains an R script with the following code:

```
1 # Get R libraries
2 # installs a defined set of 11libraries
3 # Created: 20080527, Erik.Lepko@tetratech.com
4 # =====
5
6 # clear the workspace
7 rm(list=ls())
8
9 # libraries to be installed
10 data.packages = c("dataretrieval" # loads usgs data into R
11                  ,"waterData" # QC of hydro time series data
12                  )
13
14
15 # set CRAN mirror
16 # (Loads gui in R; in R-studio select ## of mirror in console pane)
17 # IF know mirror can use "ind=" in 2nd statement and comment out (prefix line with #)
18 chooseCRANmirror()
19 #chooseCRANmirror(ind=99)
20
21 # install packages via a "for" loop
22 for (i in 1:length(data.packages)) {
23   #install.packages("dplyr",dependencies=TRUE)
24   install.packages(data.packages[i])
25 } # end loop
26
27
28
29 print("Task complete. Check statements above for errors.")
```

The right pane shows the Console output:

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Trees"
copyright (c) 2015 The R Foundation for statistical computing
Platform: i386-m64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY,
you are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

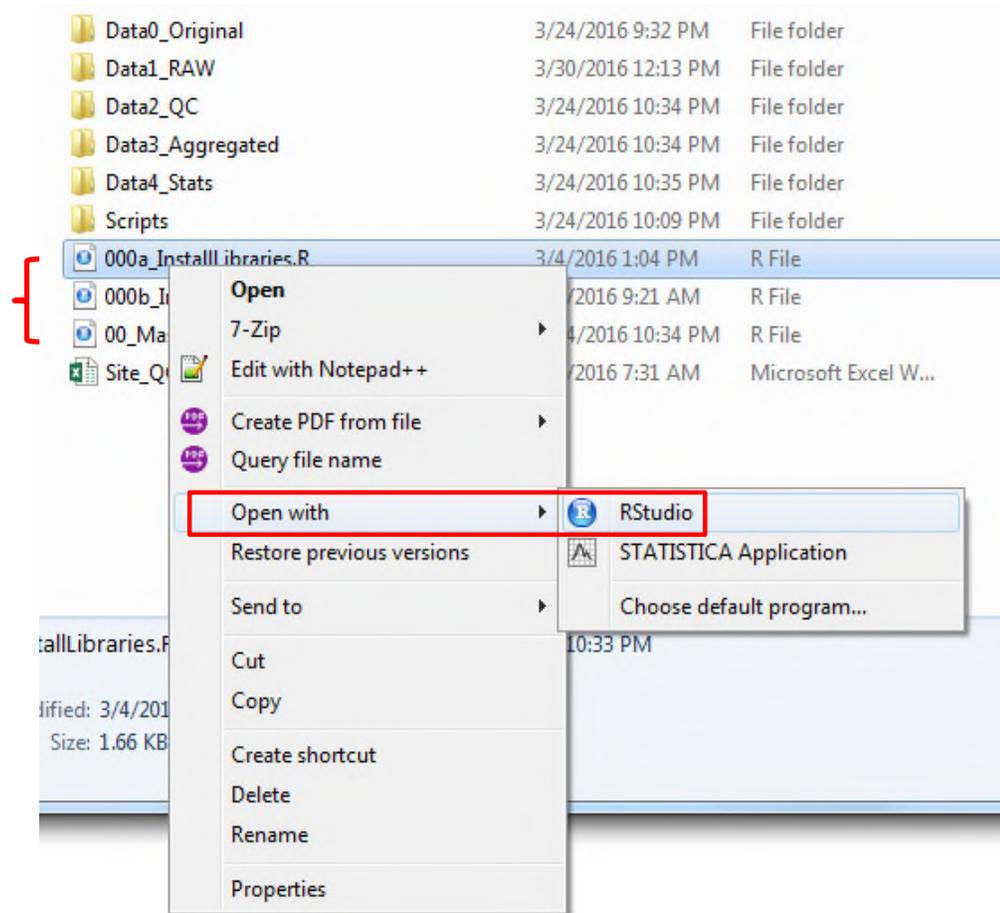
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

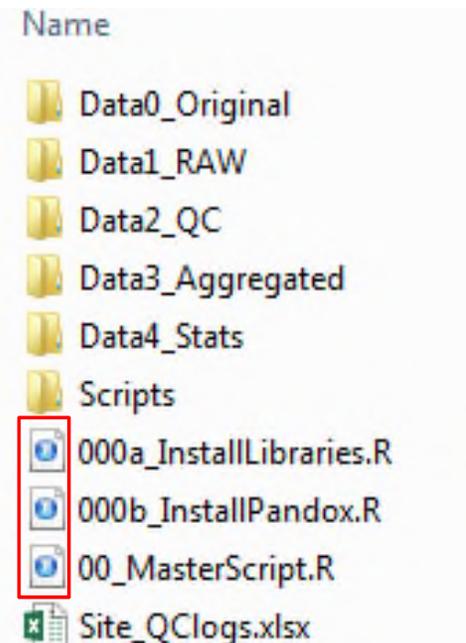
A red arrow points from the text "Check the top line in the Console" to the first line of the console output.

Set R Studio as the default (recommended but not required)

- Right click on one of the R files and a menu will appear
- Go to 'Open with'
- Select 'RStudio' (if it doesn't appear automatically as an option, find it through 'Choose default program')



Once it's the default, you'll see these blue circular symbols in front of the R script files.





Install the data packages

(only need to do this once on your computer)

In order for the R scripts to work, you'll need to run the **Install** scripts (**000a_InstallLibraries** & **000b_InstallPandex**).

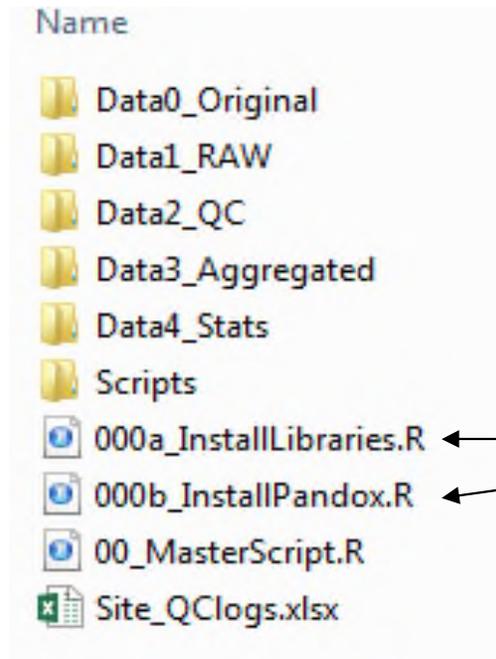
Why do you need to do this?

To retrieve data packages from the online distribution site for R software, which is referred to as the **CRAN mirror**.

You must have internet access to install the data packages. **Packages** are collections of **R** functions, data, and compiled code in a well-defined format. The directory where **packages** are stored is called the **library**.

Running the **Install** scripts

Unzip the 'Rscripts_NWQMC_20150513' file

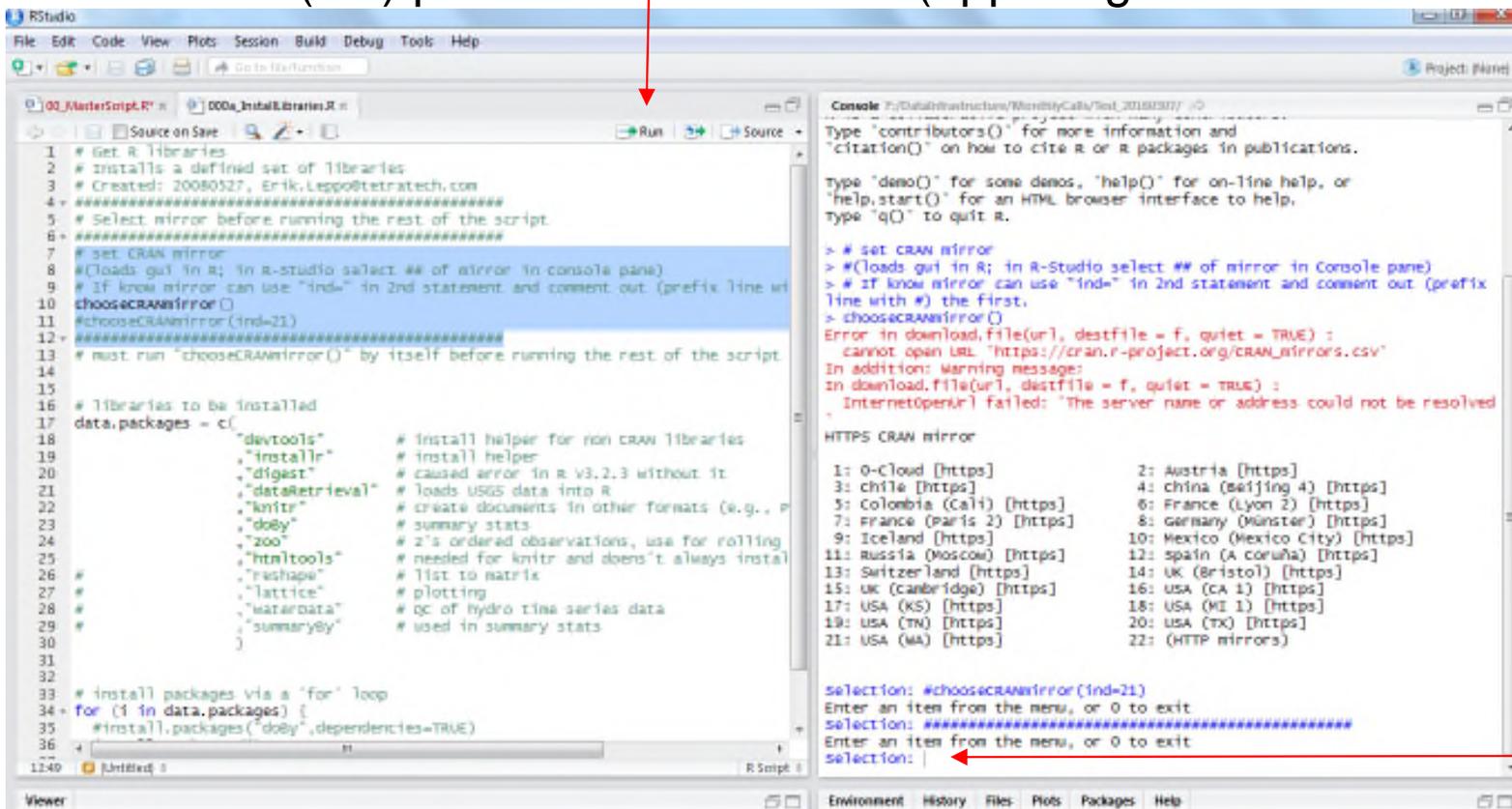


You'll be working with
these 2 Install scripts

Note: you should only have to run these once on your computer (the libraries will then be saved on your computer). If you end up running the InstallLibraries scripts more than once, it is ok (it will just overwrite the existing libraries). If you update the version of R on your computer, you will have to reinstall the libraries (see Slide 28 for a tip on how to do this).

1. Open the **000a_InstallLibraries** script. Perform the following 2-step process -

Step 1 - (using your cursor) highlight lines 7-11 in the Source (left) panel. Then click Run (upper right of



The screenshot shows the RStudio interface with two panels. The left panel is the Source editor, showing the script `000a_InstallLibraries.R`. Lines 7 through 11 are highlighted in blue. The right panel is the Console, showing the output of the script. The output includes a list of CRAN mirrors and a prompt for selection. A red arrow points from the 'Run' button in the Source panel to the Console panel. Another red arrow points from the 'selection:' prompt in the Console panel to the text 'Selection:' in the right-hand text.

```

1 # Get R libraries
2 # installs a defined set of libraries
3 # Created: 20080527, Erik.Lepko@tetratech.com
4 #####
5 # Select mirror before running the rest of the script.
6 #####
7 # set CRAN mirror
8 # (loads gui in R; in R-Studio select ## of mirror in console pane)
9 # If know mirror can use "ind=" in 2nd statement and comment out (prefix line w/
10 chooseCRANmirror())
11 #chooseCRANmirror(ind=21)
12 #####
13 # must run "chooseCRANmirror()" by itself before running the rest of the script.
14
15 # libraries to be installed
16 data.packages = c(
17   "devtools"      # install helper for non CRAN libraries
18   "installr"     # install helper
19   "digest"       # caused error in R v3.2.3 without it
20   "dataretrieval" # loads USGS data into R
21   "knitr"        # create documents in other formats (e.g., PDF)
22   "doby"         # summary stats
23   "zoo"          # z's ordered observations, use for rolling
24   "htmltools"    # needed for knitr and doesn't always install
25   "rshape"       # list to matrix
26   "lattice"      # plotting
27   "waterdata"   # qc of hydro time series data
28   "summaryBy"   # used in summary stats
29 )
30
31
32
33 # install packages via a "for" loop
34 for (i in data.packages) {
35   #install.packages("doby",dependencies=TRUE)
36 }

```

```

Type "contributors()" for more information and
"citation()" on how to cite R or R packages in publications.

Type "demo()" for some demos, "help()" for on-line help, or
"help.start()" for an HTML browser interface to help,
type "q()" to quit R.

> # set CRAN mirror
> # (loads gui in R; in R-Studio select ## of mirror in Console pane)
> # If know mirror can use "ind=" in 2nd statement and comment out (prefix
line with #) the first.
> chooseCRANmirror()
Error in download.file(url, destfile = f, quiet = TRUE) :
cannot open URL 'https://cran.r-project.org/CRAN_mirrors.csv'
In addition: Warning message:
in download.file(url, destfile = f, quiet = TRUE) :
'InternetOpenUrl' failed: 'The server name or address could not be resolved'

HTTPS CRAN mirror
1: 0-Cloud [https]                2: Austria [https]
3: ch1a [https]                  4: china (Beijing 4) [https]
5: Colombia (Cali) [https]       6: France (Lyon 2) [https]
7: France (Paris 2) [https]      8: Germany (Münster) [https]
9: Iceland [https]              10: Mexico (Mexico City) [https]
11: Russia (Moscow) [https]      12: Spain (A Coruña) [https]
13: Switzerland [https]         14: UK (Bristol) [https]
15: UK (Cambridge) [https]       16: USA (CA 1) [https]
17: USA (KS) [https]             18: USA (MI 1) [https]
19: USA (TX) [https]            20: USA (TX) [https]
21: USA (WA) [https]            22: (HTTP mirrors)

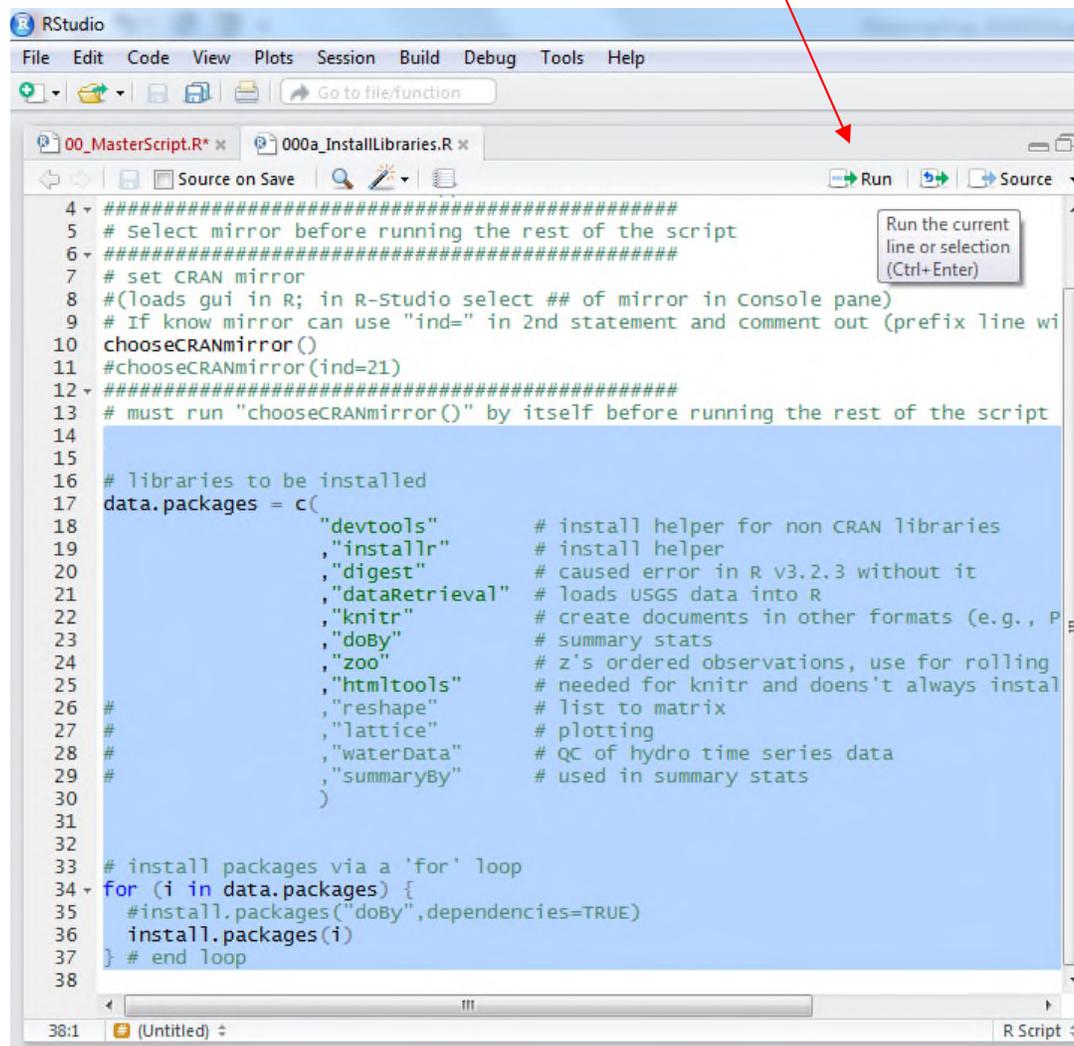
selection: #chooseCRANmirror(ind=21)
Enter an item from the menu, or 0 to exit
selection: #####
Enter an item from the menu, or 0 to exit
selection: |

```

Give it a little time – and a list of locations will appear in the Console (right) panel. In the Console panel, click with your cursor to the right of **Selection:** and enter the number of the closest location. Then hit enter.

Running the **000a_InstallLibraries** script

Step 2 - Then run the next part of the script. (using your cursor) highlight lines 14-37 in the Source (left) panel. Then click Run (upper right of Source panel)

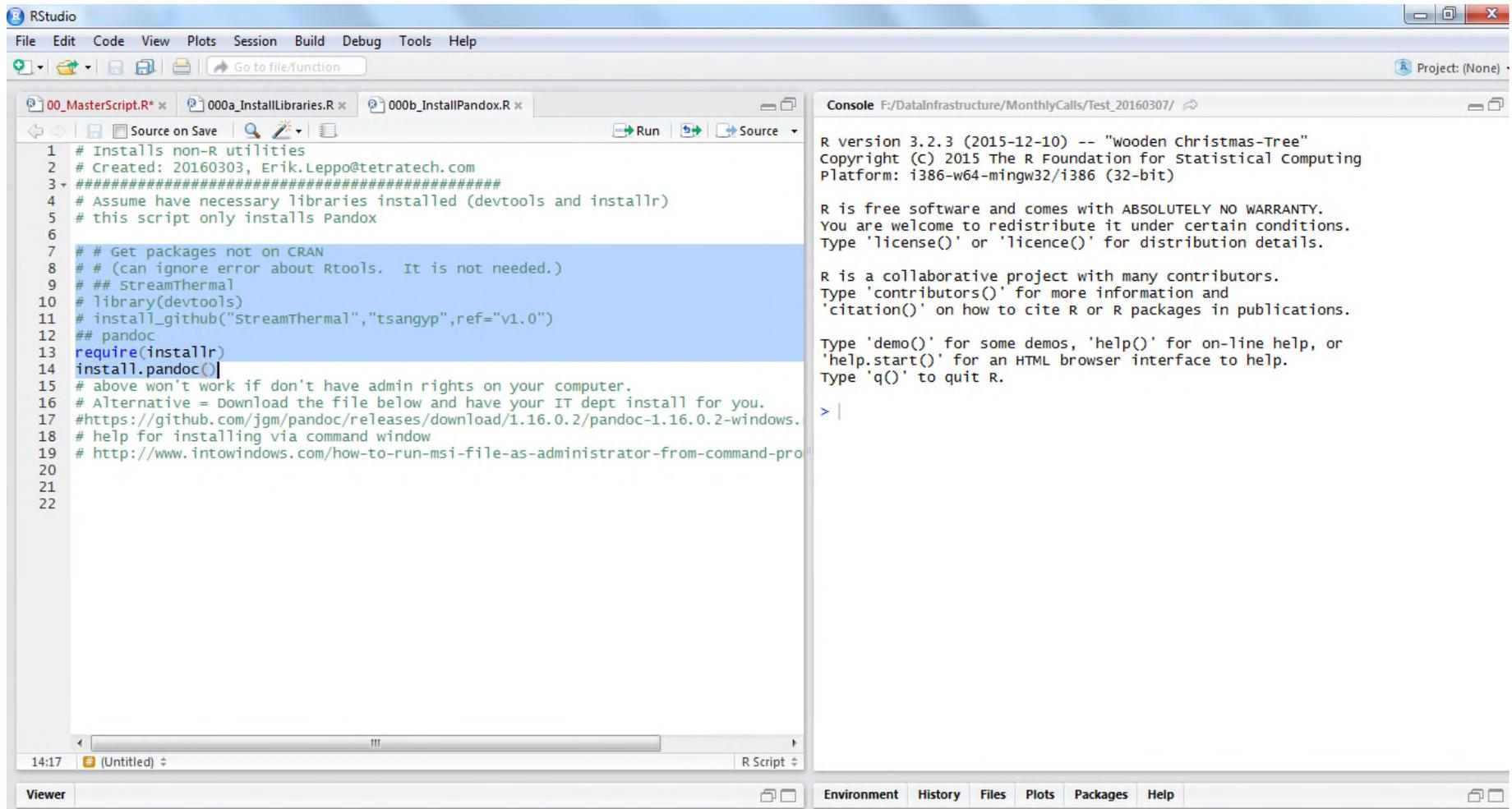


```
4 ~ #####
5 # Select mirror before running the rest of the script
6 ~ #####
7 # set CRAN mirror
8 #(loads gui in R; in R-studio select ## of mirror in console pane)
9 # If know mirror can use "ind=" in 2nd statement and comment out (prefix line wi
10 chooseCRANmirror()
11 #chooseCRANmirror(ind=21)
12 ~ #####
13 # must run "chooseCRANmirror()" by itself before running the rest of the script
14
15
16 # libraries to be installed
17 data.packages = c(
18     "devtools"      # install helper for non CRAN libraries
19     , "installr"    # install helper
20     , "digest"      # caused error in R v3.2.3 without it
21     , "dataRetrieval" # loads USGS data into R
22     , "knitr"       # create documents in other formats (e.g., P
23     , "doBy"        # summary stats
24     , "zoo"         # z's ordered observations, use for rolling
25     , "htmltools"   # needed for knitr and doesn't always instal
26     #
27     , "reshape"    # list to matrix
28     , "lattice"     # plotting
29     , "waterData"  # QC of hydro time series data
30     , "summaryBy"  # used in summary stats
31 )
32
33 # install packages via a 'for' loop
34 ~ for (i in data.packages) {
35     #install.packages("doBy",dependencies=TRUE)
36     install.packages(i)
37 } # end loop
38
```

Give it some time...in the Console (right) panel you'll see a log of the packages being downloaded and installed...

The installations are done when the Console has a ">" by itself on a new line.

Next open and run the **000b_InstallPandox** script



The screenshot shows the RStudio interface. The script editor on the left contains the following code:

```
1 # Installs non-R utilities
2 # Created: 20160303, Erik.Leppo@tetrattech.com
3 #####
4 # Assume have necessary libraries installed (devtools and installr)
5 # this script only installs Pandox
6
7 ## Get packages not on CRAN
8 ## (can ignore error about Rtools. It is not needed.)
9 ## StreamThermal
10 # library(devtools)
11 # install_github("StreamThermal", "tsangyp", ref="v1.0")
12 ## pandoc
13 require(installr)
14 install.pandoc()
15 # above won't work if don't have admin rights on your computer.
16 # Alternative = Download the file below and have your IT dept install for you.
17 #https://github.com/jgm/pandoc/releases/download/1.16.0.2/pandoc-1.16.0.2-windows.
18 # help for installing via command window
19 # http://www.intowindows.com/how-to-run-msi-file-as-administrator-from-command-pro
20
21
22
```

The console window on the right displays the following output:

```
R version 3.2.3 (2015-12-10) -- "wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

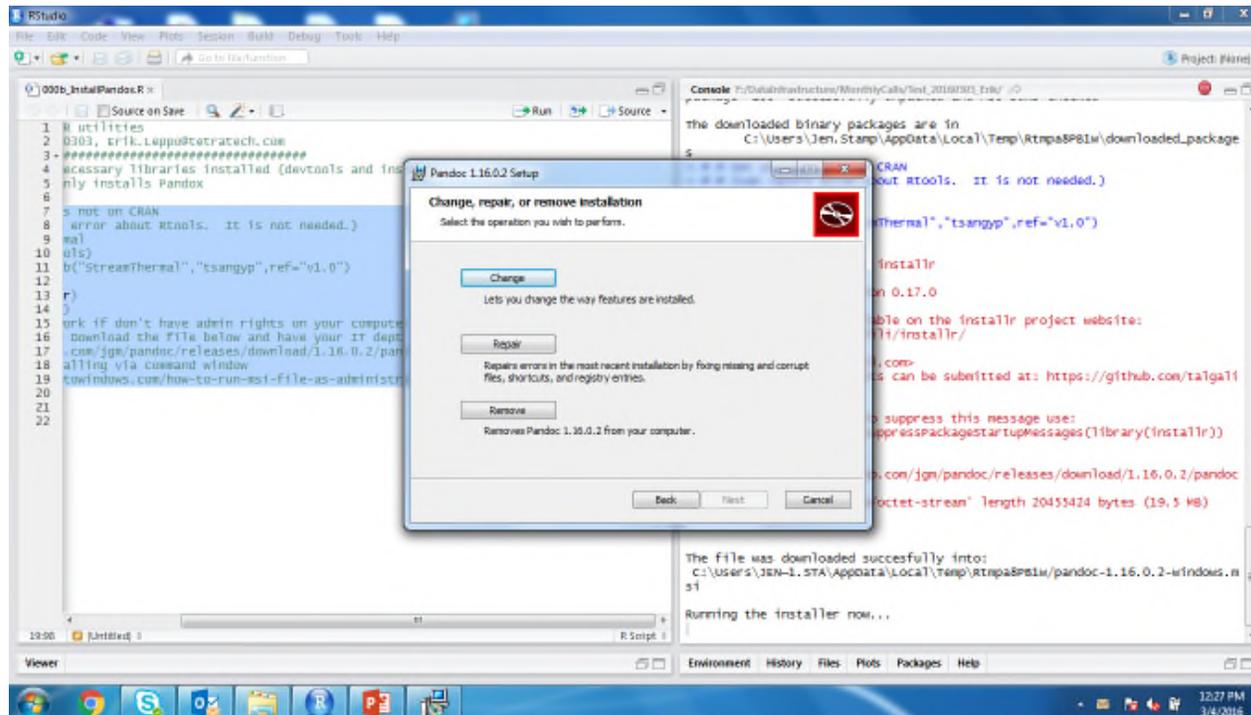
> |
```

000b_InstallPandoc script

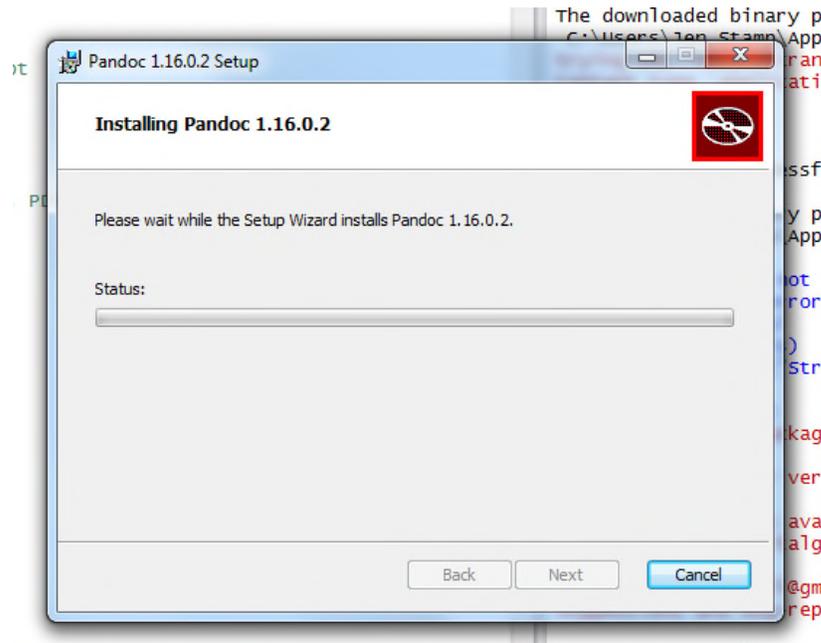
On some computers, a box may come up that gives you 3 choices:

Change, repair or remove Pandoc

If this happens, chose **change** or **repair**.



Pandoc installation



Then you'll be prompted to go through an installation process for Pandoc.

Once that is complete, you'll need to restart your computer in order for Pandoc to work correctly (you'll need this for the QC reports).



Upgrading R to a New Version?

Libraries don't transfer when installing a new version of R.

Here is a 'painless' way to update your libraries -

- After installing the new version, run this in the **old version** -

```
setwd("C:/Temp/")
packages <- installed.packages()[,"Package"]
save(packages, file="Rpackages")
```

- Then run this in the **new version** -

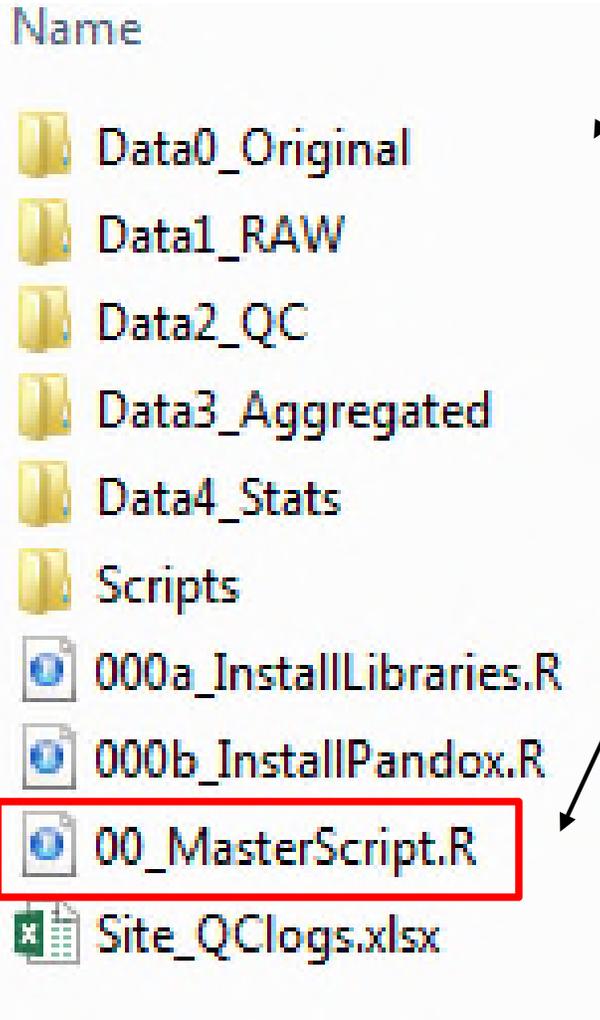
```
setwd("C:/Temp/")
load("Rpackages")
for (p in setdiff(packages, installed.packages()[,"Package"]))
install.packages(p)
```

You should now have updated versions of each library in your new version of R.

Overview of the R scripts

R scripts and directory

Unzip the 'Rscripts_NWQMC_20150422' file –it should look like this...



You'll be doing most of your work with the **MasterScript**.

The MasterScript needs to be located in the same directory where the data files are located.

The MasterScript reads through the files and pulls out data based on file names (siteID, dates, data type).

The MasterScript produces outputs that go into certain folders (so the folders have to be named a certain way)



The MasterScript...

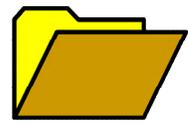
Performs 4 different operations (one at a time).

You specify which operation you want to run and what files you want to run it on.

Operation	Description	Import Directory*	Export Directory**	Notes
GetGageData	Retrieves gage data	USGS NWIS website	Data1_Raw	Optional (if you don't need to bring in USGS gage data, you can skip this step and go right to the QCRaw operation)
QCRaw	Performs QC on Raw data	Data1_Raw	Data2_QC	You need to run this portion of the R script before you can run the Aggregate script
Aggregate	Aggregates QC'd data	Data2_QC	Data3_Aggregated	You need to run this portion of the R script before you can run the SummaryStats script
SummaryStats	Generates summary stats on Aggregated data.	Data3_Aggregated	Data4_Stats	

*The R script looks at files in these folders when the script is run

**Output files will appear in these folders after the script runs



Folder structure

– 0_Original

- Master files. Do not change these! If you make a mistake and need to start over, you'll need these (unaltered) files.

– 1_RAW

- Files are copied from the Original folder and pasted here. These are your working files. Files need to be edited so that they have a standardized format and file naming scheme, and are .csv.

– 2_QC

- When the QC portion of the MasterScript is run, the QC data outputs will go here (2 outputs: Word & .csv).

– 3_Aggregate

- When the Aggregate portion of the MasterScript is run, the outputs go here (2 outputs: Word & .csv).

– 4_Stats

- When the SummaryStats portion of the MasterScript is run, the outputs go here (3 outputs: 2 .csv, 1 PDF)

– Scripts

- R files that you will rarely need to work with, with the possible exception of the fun.UserDefinedValues.R file

Name

- Data0_Original
- Data1_RAW
- Data2_QC
- Data3_Aggregated
- Data4_Stats
- Scripts
- 000a_InstallLibraries.R
- 000b_InstallPadox.R
- 00_MasterScript.R
- Site_QClogs.xlsx



There are only 5 places in the MasterScript where you'll need to make entries...

- 1) Select the **operation**
 - GetGageData, QCRaw, Aggregate or SummaryStats
- 2) Enter the **SiteID**
- 3) Enter the **data/sensor type**
 - Air, water, air/water, gage, air/water/gage, air/gage, water/gage
- 4) Enter a **start date**
- 5) Enter an **end date**

You don't need to write any R code yourself!

All you need to do is **make these entries, highlight the R script and hit Run.**



Game plan

Go through 2 types of datasets

Test Dataset #1

Non-vented pressure transducers

- Water temperature
- Air temperature
- Water level
- Barometric (air) pressure
- Water pressure

2 Files –

- Same site, different date ranges



Test Dataset #2

Standalone temperature sensors

- Water
- Air

2 Files –

- Same site, parameters in separate files, same date range



Work flow

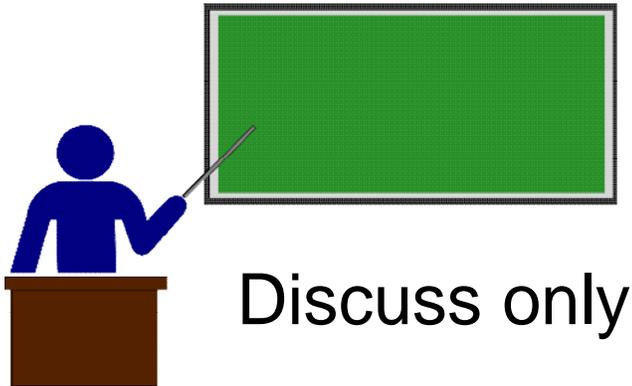
1. Preparing your data
 - Moving it into the correct folders
 - Organizing, formatting and naming the files
2. Running the QC portion of the MasterScript
 - Checking the data, correcting if needed
3. Running the Aggregate portion of the MasterScript
4. Running the SumStats portion of the MasterScript

Note: we QC one file at a time.

Why not combine them first, then do the QC?

We feel it's best to check files before manipulating and combining them. Also it is easier to QC smaller 'chunks' of data.

Two types of slides



When you see this symbol on a slide, please listen and watch.



You do it!

When you see this symbol on a slide, please perform the task.



Questions or problems running the R script?

Raise your hand and we'll send
someone over to try and help you



Test Dataset # 1

Test Dataset #1

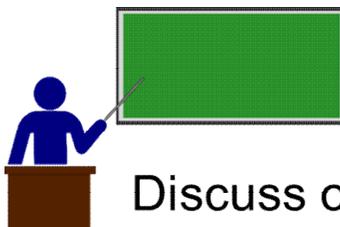
Non-vented pressure transducer data

- Water temperature
- Air temperature
- Water level
- Barometric (air) pressure
- Water pressure



2 Files –

- Same site (ECO66G12)
- Different date ranges
 - April 26, 2013 – July 25, 2013
 - July 25, 2013 – October 15, 2013

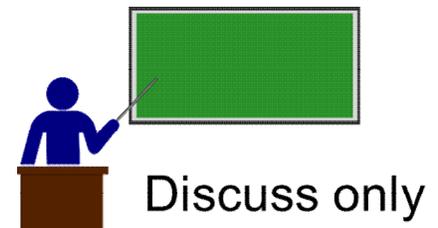


Discuss only

Work flow for Test Dataset #1

- 1. Prepare your data** (demonstrate only*)
 - Move it into the correct folders
 - Organize, format and name the files
- 2. Run the QC** portion of the MasterScript (on both files)
 - Check the data, correct if needed
- 3. Run the Aggregate** portion of the MasterScript to merge the QC'd data into one file (for full date range)
- 4. Run the SumStats** portion of the MasterScript to generate the summary statistics

*We prepared the data files in advance in order to maximize the amount of time we can spend running the actual scripts



Preparing your data



Why is this important?

In order for the R script to work, the files need to be named and formatted a certain way, and located in the appropriate folders.

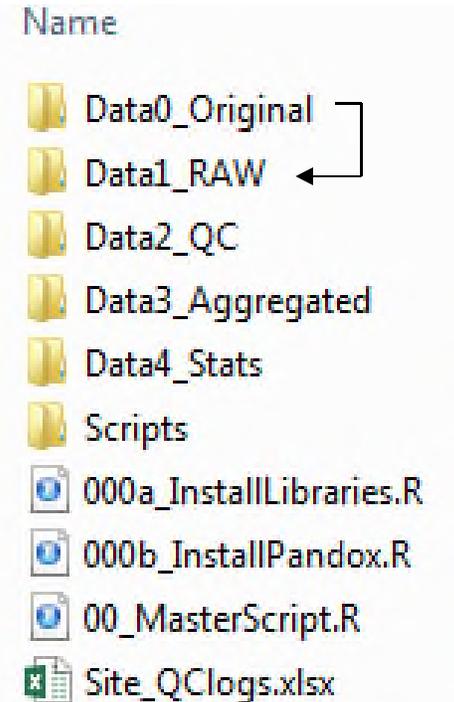
The MasterScript reads through the files in the appropriate folders and pulls out the desired data based on file names (siteID, dates, data type).

This allows for multiple files to be kept in the same directory.



General work flow

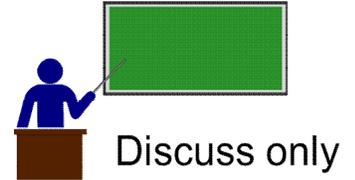
1. Put the 'original' (unaltered) continuous sensor files that you want to run the R scripts on in the Data0_Original folder.
2. Copy and paste these 'original' files into the Data1_RAW folder.
3. Prepare the files in the RAW folder for the R scripts
 - Required & optional fields; naming scheme
 - Check for rows with no data
 - Name files
 - Save as .csv



For today's workshop, to help speed things along, we've already done these steps on the test files. However, we will talk you through these steps so that you'll know how to prepare your own data.



1. Move data into the Data0_original folder



These are your ‘original’ continuous sensor files*



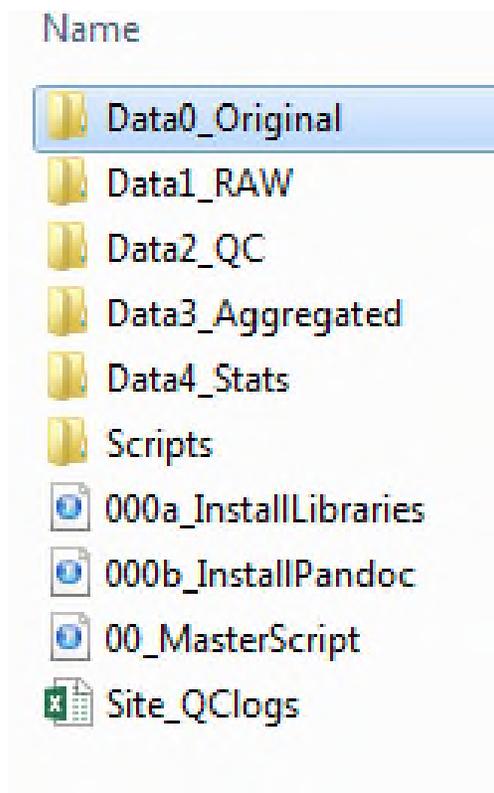
Once the files are in this folder, ***do not change them!*** If you make a mistake, you might need to go back to these (unaltered) files and start over.

The files in the **Data0_Original folder** are *not* used by the R scripts, so the names and formats do not have to be standardized (e.g., it’s ok if the “Original” files are in Excel).

*in some cases, you may need to run the data files through the sensor software (e.g., HOBOWare) before putting them into the Data0_Original folder.

Let's take a look at the test files that are in the Data0_Original folder...

Open the Data0_Original folder



Name	Date modified	Type
 ECO66G12_WATER_4-26-13 TO 7-25-13.xl...	2/3/2015 3:01 PM	Microsoft Excel W...
 ECO66G12_WATER_7-25-13 TO 10-15-13....	2/3/2015 3:01 PM	Microsoft Excel W...
 WSR01__WaterTemp_20140710.xlsx	4/15/2016 6:56 AM	Microsoft Excel W...
 WSR01_AirTemp_20140710.xlsx	4/15/2016 6:57 AM	Microsoft Excel W...

You should see 4 Excel files –

- 2 for site ECO66G12 (test dataset #1)
- 2 for site WSR01 (test dataset #2)

Note: these are examples of typical continuous sensor files. Outputs are for 1 site and for a given time period.

Examples of 'original' continuous sensor files...



Open one of the ECO66G12 files

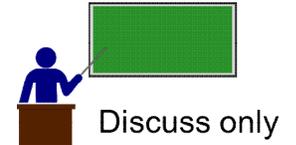


Name	Date modified	Type	Size
ECO66G12_WATER_4-26-13 TO 7-25-13	2016-04-21 19:13	Microsoft Excel W...	223 KB
ECO66G12_WATER_7-25-13 TO 10-15-13	2016-04-21 19:14	Microsoft Excel W...	202 KB
WSR01_WaterTemp_20140710	2016-04-15 06:56	Microsoft Excel W...	93 KB
WSR01_AirTemp_20140710	2016-04-15 06:57	Microsoft Excel W...	97 KB

	A	B	C	D	E	F	G
1	Plot Title: ECO66G12-Water						
2	#	Date Time, GMT-0	Abs Pres, psi	Temp, °C (LGR S	Abs Pres Bar	Water Level, f	Temp, °C (LGI v
3	1	2013-04-26 10:30	14.8024	10.063	14.3638	0.985	15.282
4	2	2013-04-26 11:00	14.7461	10.357	14.3644	0.854	14.134
5	3	2013-04-26 11:30	14.7428	10.748	14.3603	0.855	14.23
6	4	2013-04-26 12:00	14.7441	11.139	14.3627	0.853	15.473
7	5	2013-04-26 12:30	14.7431	11.528	14.3612	0.854	17.284
8	6	2013-04-26 13:00	14.7421	11.916	14.3628	0.848	18.806
9	7	2013-04-26 13:30	14.7453	12.207	14.3629	0.855	19.662

This is an example of a continuous data file from a non-vented pressure transducer (this came out of HOBOWare; other sensor outputs may look slightly different).

2. Copy and paste the files from the Data0_Original folder into the Data1_Raw folder



 Data0_Original folder

*Copy
& paste*

Name	Date modified	Type	Size
 ECO66G12_WATER_4-26-13 TO 7-25-13	2016-04-21 19:13	Microsoft Excel W...	223 KB
 ECO66G12_WATER_7-25-13 TO 10-15-13	2016-04-21 19:14	Microsoft Excel W...	202 KB
 WSR01__WaterTemp_20140710	2016-04-15 06:56	Microsoft Excel W...	93 KB
 WSR01_AirTemp_20140710	2016-04-15 06:57	Microsoft Excel W...	97 KB

 Data1_Raw folder

The files in the Data1_RAW folder are your ‘working’ files.

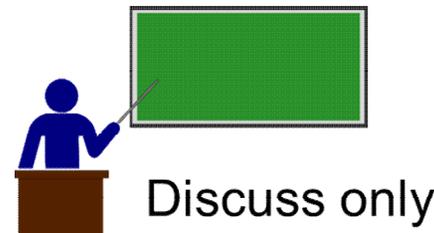
When you run the QC portion of the MasterScript, the R script pulls data from this folder.

In order for the R scripts to run, the RAW data files need to be formatted, named and saved in a specific way.

3. Required fields in the RAW files & naming scheme

Fields do not have to be in any specific order.

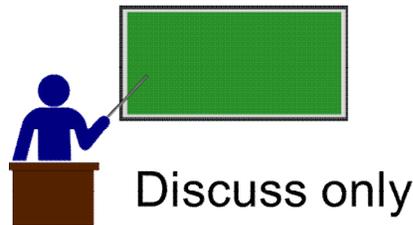
- **SiteID**
- **“Date Time”** (one field) OR **“Date” AND “Time”** (in 2 separate fields) OR **all 3** (in 3 fields)
 - Prefer 24H time (i.e., military). R scripts will assume any time without AM/PM is 24-hr time.
- **Parameters (as available) must be named as follows -**
 - Water Temp C
 - Air Temp C
 - Water BP psi
 - Air BP psi
 - Water Level ft



Optional fields

You can also include additional (optional) fields -

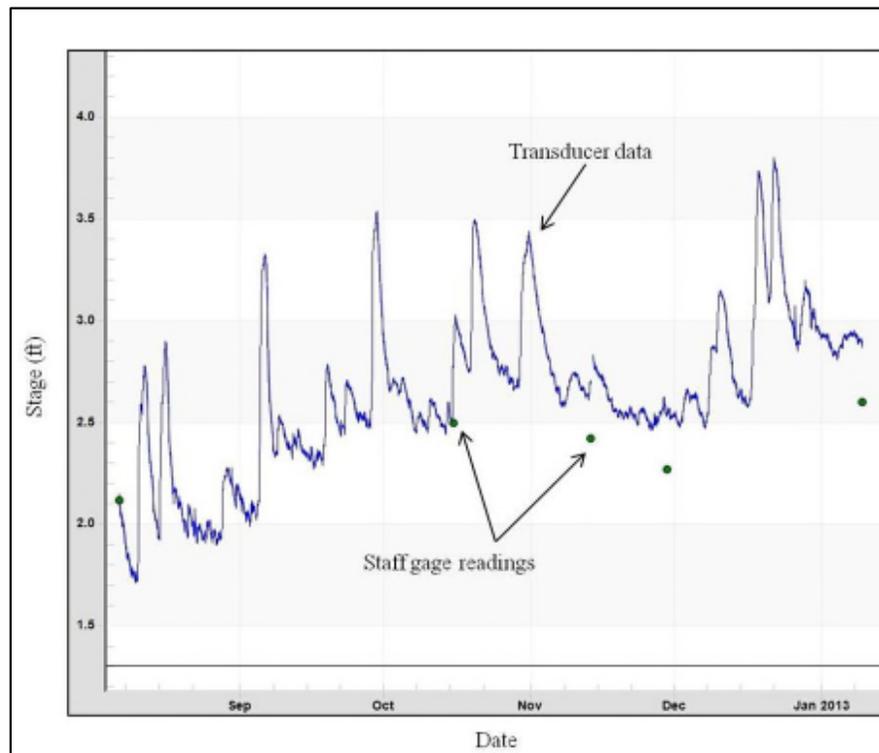
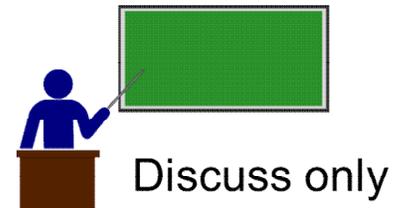
- **Air LoggerID** – unique identifier (e.g., serial number) for an land-based sensor
- **Water LoggerID** – unique identifier (e.g., serial number) for an in-water sensor
- **Air RowID** or **Water RowID** (this shows up as # in the HOBOWare output; you cannot leave it as “#” because R will treat this as a comment and ignore the entire line)



Optional but strongly encouraged - *Discrete measurements*

Discrete measurements are instantaneous measurements taken during field visits that are compared to the closest continuous sensor measurement. They are used for –

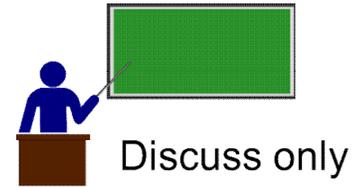
- Accuracy checks
- Making corrections to your data (as needed)
- Validating your data (making them more defensible)



EXAMPLE –

Staff gage readings can provide a quality check of transducer data. In this example, staff gage readings stopped matching transducer readings in November, indicating that the transducer or gage could have changed elevation.

Adding discrete measurements into RAW data files



Enter the discrete value in the sensor measurement row with the **closest date time** (no need to create a new line for just the discrete measurement; in fact, the way the scripts are set up, we prefer that you don't enter a new line with only the discrete measurement).

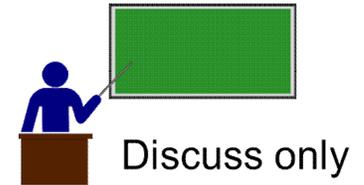
The column heading needs to have the prefix **"Discrete"** (e.g., Discrete Water Temp C).

	A	B	C	D	E	F	G	H
1	Water RowID	Date	Time	Water Temp C	SiteID	Water LoggerID	Date Time	Discrete.Water Temp C
2	3	22-Oct-13	13:00	9.632	HRCC	10406272	10/22/2013 13:00	9.39
3	4	22-Oct-13	13:30	9.632	HRCC	10406272	10/22/2013 13:30	
4	5	22-Oct-13	14:00	9.682	HRCC	10406272	10/22/2013 14:00	
5	6	22-Oct-13	14:30	9.706	HRCC	10406272	10/22/2013 14:30	
6	7	22-Oct-13	15:00	9.78	HRCC	10406272	10/22/2013 15:00	
7	8	22-Oct-13	15:30	9.854	HRCC	10406272	10/22/2013 15:30	
8	9	22-Oct-13	16:00	9.952	HRCC	10406272	10/22/2013 16:00	

IMPORTANT!

MAKE SURE YOUR DISCRETE AND MATCHING CONTINUOUS MEASUREMENT ARE BOTH IN EITHER STANDARD TIME OR DAYLIGHT SAVINGS TIME.

Field naming scheme



Non-vented pressure transducer (after water and air sensor data were combined in HoboWare)

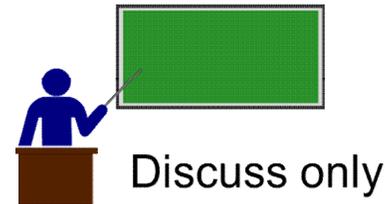
Before...

	A	B	C	D	E	F	G
1	Plot Title: ECO66G12-Water						
2	#	Date Time, GMT-0	Abs Pres, psi	Temp, °C (LGR S	Abs Pres Bar	Water Level, f	Temp, °C (LGI
3	1	2013-04-26 10:30	14.8024	10.063	14.3638	0.985	15.282
4	2	2013-04-26 11:00	14.7461	10.357	14.3644	0.854	14.134
5	3	2013-04-26 11:30	14.7428	10.748	14.3603	0.855	14.23

And after...

A	B	C	D	E	F	G	H	I	J
AW RowID	SiteID	Date Time	Water Temp C	Air Temp C	Water Level ft	Water BP psi	Air BP psi	Water Logger ID	Air Logger ID
1	ECO66G12	4/26/2013 10:30	10.063	15.282	0.985	14.8024	14.3638	10229557	10229571
2	ECO66G12	4/26/2013 11:00	10.357	14.134	0.854	14.7461	14.3644	10229557	10229571
3	ECO66G12	4/26/2013 11:30	10.748	14.23	0.855	14.7428	14.3603	10229557	10229571
4	ECO66G12	4/26/2013 12:00	11.139	15.473	0.853	14.7441	14.3627	10229557	10229571
5	ECO66G12	4/26/2013 12:30	11.528	17.284	0.854	14.7431	14.3612	10229557	10229571
6	ECO66G12	4/26/2013 13:00	11.916	18.806	0.848	14.7421	14.3628	10229557	10229571
7	ECO66G12	4/26/2013 13:30	12.207	19.662	0.855	14.7453	14.3629	10229557	10229571
8	ECO66G12	4/26/2013 14:00	12.594	20.424	0.853	14.7351	14.3537	10229557	10229571

4. Check for rows with no data



FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER Nuance PDF

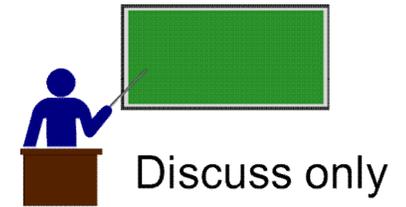
Clipboard Font Alignment Number Styles

B4 : 7/25/2013 12:30:00 PM

	A	B	C	D	E	F	G	H	I	J	K	L
4020	4017	10/17/2013 4:30	14.0234	15.473	13.9275	0.426	15.951					
4021	4018	10/17/2013 5:00	14.0257	15.473	13.9275	0.431	15.951					
4022	4019	10/17/2013 5:30	14.0257	15.473	13.9275	0.431	15.951					
4023	4020	10/17/2013 6:00	14.0234	15.473	13.9233	0.435	16.046					
4024	4021	10/17/2013 6:30	14.0257	15.473	13.9257	0.435	16.046					
4025	4022	10/17/2013 7:00	14.0234	15.473	13.9257	0.43	16.046					
4026	4023	10/17/2013 7:30	14.0234	15.473	13.9233	0.435	16.046					
4027	4024	10/17/2013 8:00	14.0238	15.569	13.9239	0.435	16.141					
4028	4025	10/17/2013 8:30	14.0238	15.569	13.9216	0.44	16.141					
4029	4026	10/17/2013 9:00	14.0285	15.569	13.9243	0.445	16.237					
4030	4027	10/17/2013 9:30	14.0331	15.569	13.9291	0.444	16.237					
4031	4028	10/17/2013 10:00	13.9306	15.664	13.9325	0.2	16.427					
4032	4029	10/17/2013 10:01							Logged			
4033	4030	10/17/2013 10:01								Logged	Logged	
4034												

Remove entries like these...

5. Rename the files in the Data1_RAW folder

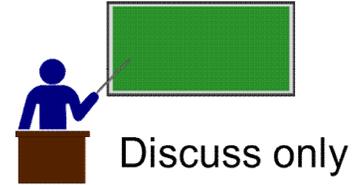


Need to use consistent file naming convention.

Example (non-vented pressure transducer, with all parameters in one file):

- **ECO66G12_AW_20130426_20130725**
 - Site ID (no spaces or underscores) = ECO66G12
 - Data Type (Water/Air/AW/Gage) = AW (air and water)
 - Date, Start (YYYYMMDD) = 20130426
 - Date, End (YYYYMMDD) = 20130725
 - Each element separated by underscore (“_”).
- Allows for multiple files to be kept in the same directory.
- R scripts will pull data from files based on Site ID, data type and date range.

File naming scheme



BEFORE

 ECO66G12_WATER_4-26-13 TO 7-25-13.xlsx
 ECO66G12_WATER_7-25-13 TO 10-15-13.xlsx

AFTER

 ECO66G12_AW_20130426_20130725.csv
 ECO66G12_AW_20130725_20131015.csv

Example: this file is from site ECO66G12

It contains both air and water sensor data (=AW)

Period of record = April 26, 2013 through July 25, 2013

What if this was just water sensor data?

Data type would be ‘_Water’

What if this was just air sensor data?

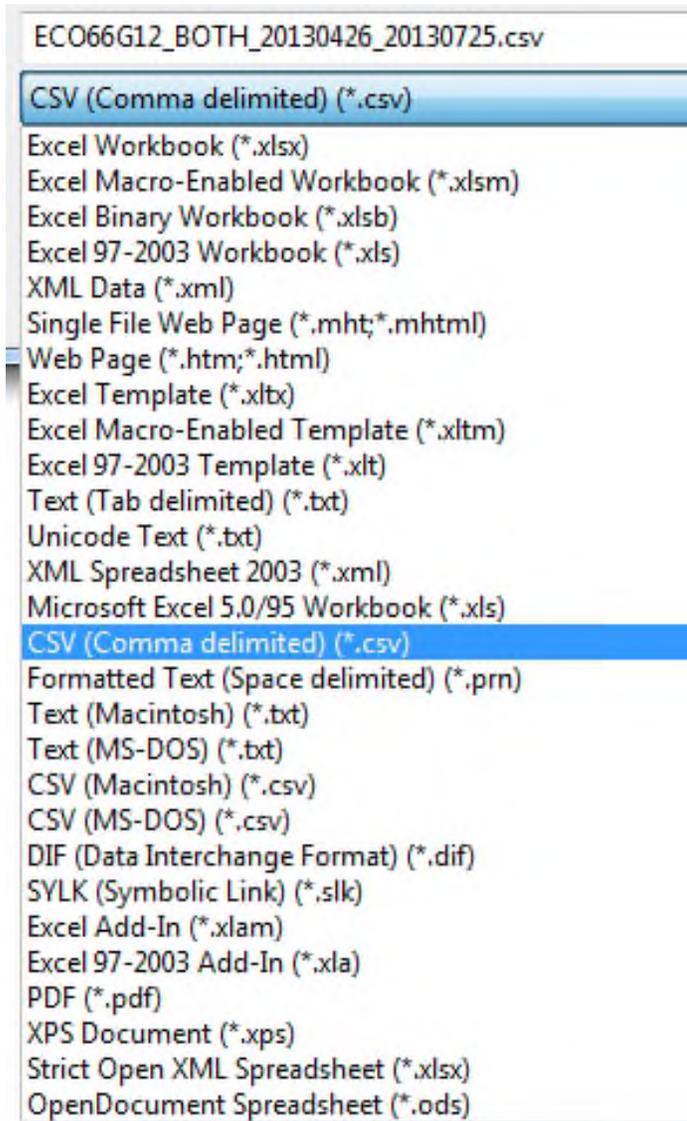
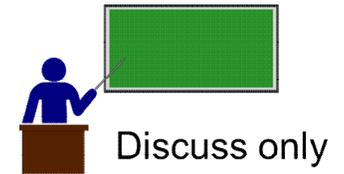
Data type would be ‘_Air’

What if this was USGS gage data?

Data type would be ‘_Gage’

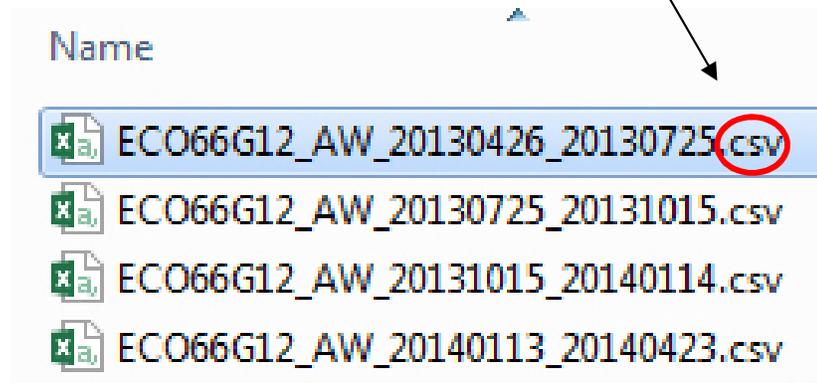
5. Save as .csv file

.csv = comma delimited



File

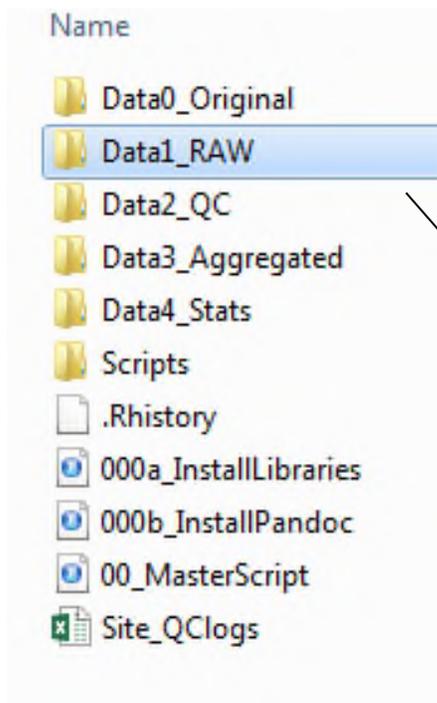
- Save As
 - CSV (Comma delimited) (*.csv)





Let's take a look at the test files that are in the Data1_RAW folder...

**Open the Data1_RAW folder.
Open one of the ECO66G12 files.**



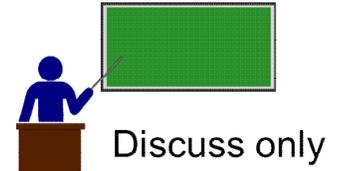
Name	Date modified	Type	Size
01174565_Gage_20140424_20140710	2016-04-15 11:28	Microsoft Excel C...	374 KB
ECO66G12_AW_20130426_20130725	2016-04-21 19:25	Microsoft Excel C...	327 KB
ECO66G12_AW_20130725_20131015	2016-04-21 19:25	Microsoft Excel C...	307 KB
WSR01_Air_20140424_20140710	2016-04-20 07:43	Microsoft Excel C...	125 KB
WSR01_Gage_20140424_20140710	2016-04-20 07:43	Microsoft Excel C...	310 KB
WSR01_Water_20140424_20140710	2016-04-20 07:43	Microsoft Excel C...	158 KB

	A	B	C	D	E	F	G	H	I	J	K
1	AW RowID	SiteID	Date Time	Water BP psi	Water Temp C	Air BP psi	Water Level ft	Air Temp C	Discrete Water Temp C	Discrete Air Temp C	Discrete Water Level ft
2	1	ECO66G12	4/26/2013 10:30	14.8024	10.063	14.3638	0.985	15.282	10.055	15.24	0.98
3	2	ECO66G12	4/26/2013 11:00	14.7461	10.357	14.3644	0.854	14.134			
4	3	ECO66G12	4/26/2013 11:30	14.7428	10.748	14.3603	0.855	14.23			
5	4	ECO66G12	4/26/2013 12:00	14.7441	11.139	14.3627	0.853	15.473			
6	5	ECO66G12	4/26/2013 12:30	14.7431	11.528	14.3612	0.854	17.284			
7	6	ECO66G12	4/26/2013 13:00	14.7421	11.916	14.3628	0.848	18.806			

Running the QC_Raw portion of the MasterScript

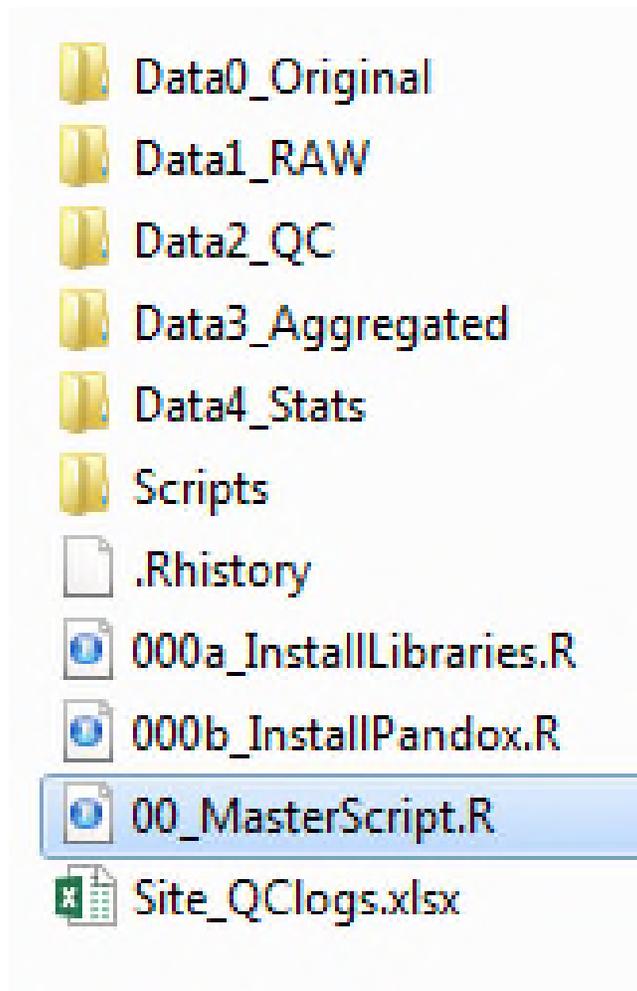


QC process - overview



- 1. Open the MasterScript** (double click on it)
Make the appropriate entries (operation, SiteID, data type, date range)
- 2. Run the script**, which searches the Data1_RAW folder for the appropriate files
- 3. Open the Data2_QC folder** – there should be 2 outputs for each input file – **.csv file & QC report** (Word document)
- 4. Check flagged values** in the QC report and .csv file
- 5. Perform ‘manual’ visual checks** on the plots
- 6. Make corrections as needed**

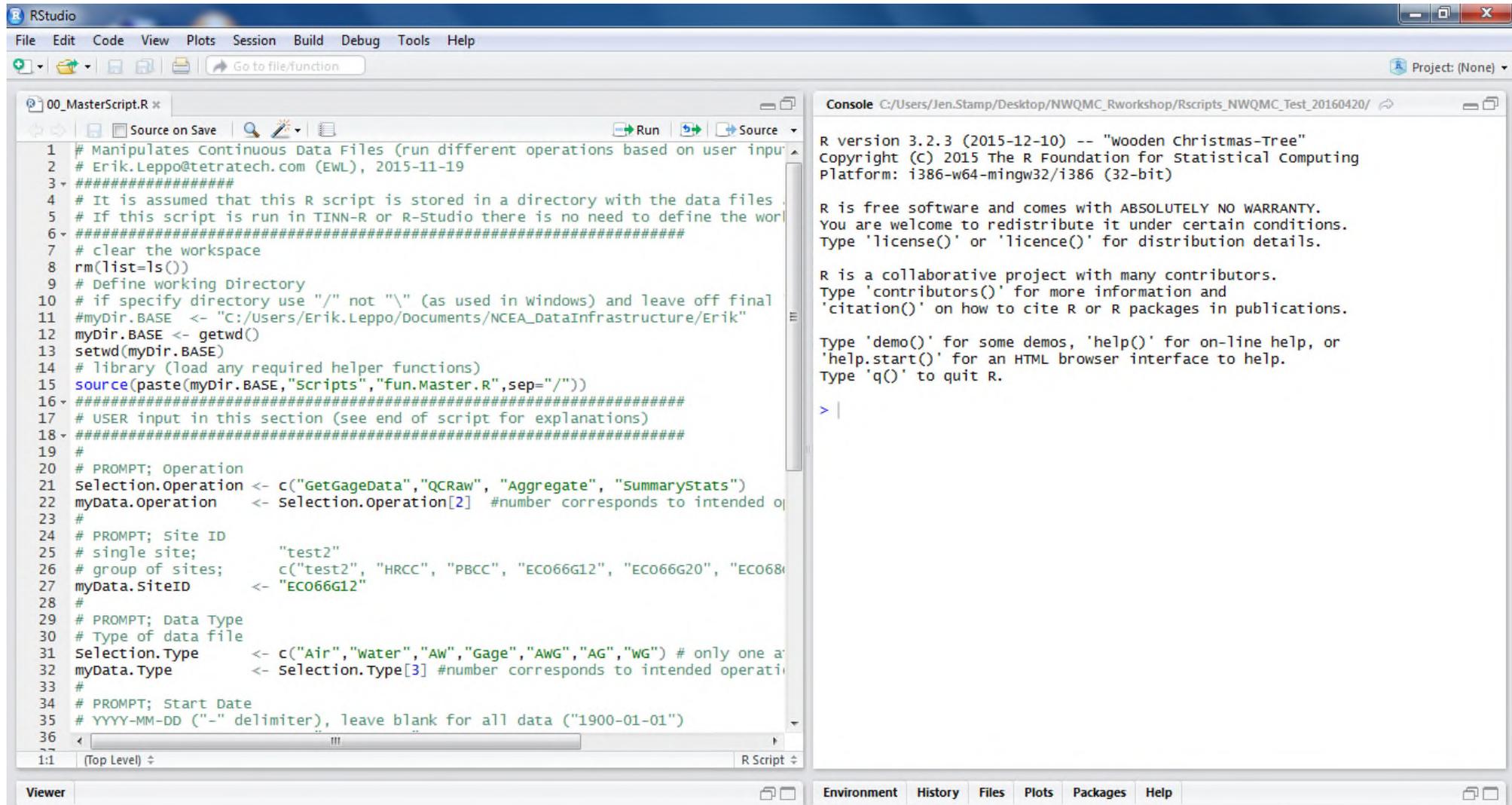
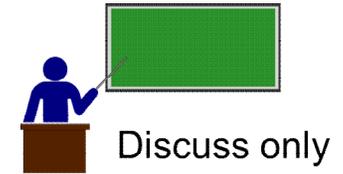
1. Open the MasterScript & make the appropriate entries



Go to the directory & double click on the **MasterScript** to open it.

Use this technique, otherwise you may encounter errors (e.g., sometimes people get error messages when they open the RStudio program then navigate to the file (vs. opening it from the directory)).

The Masterscript will look like this...



The screenshot shows the RStudio interface. The script editor on the left contains the following code:

```
1 # Manipulates Continuous Data Files (run different operations based on user input)
2 # Erik.Leppo@tetratech.com (EWL), 2015-11-19
3 #####
4 # It is assumed that this R script is stored in a directory with the data files
5 # If this script is run in TINN-R or R-Studio there is no need to define the work
6 #####
7 # clear the workspace
8 rm(list=ls())
9 # Define working Directory
10 # if specify directory use "/" not "\" (as used in windows) and leave off final
11 #myDir.BASE <- "C:/Users/Erik.Leppo/Documents/NCEA_DataInfrastructure/Erik"
12 myDir.BASE <- getwd()
13 setwd(myDir.BASE)
14 # library (load any required helper functions)
15 source(paste(myDir.BASE,"Scripts","fun.Master.R",sep="/"))
16 #####
17 # USER input in this section (see end of script for explanations)
18 #####
19 #
20 # PROMPT; Operation
21 Selection.operation <- c("GetGageData","QCraw", "Aggregate", "SummaryStats")
22 myData.operation <- Selection.operation[2] #number corresponds to intended op
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "ECO68
27 myData.SiteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.type <- c("Air","water","AW","Gage","AWG","AG","WG") # only one a
32 myData.Type <- Selection.Type[3] #number corresponds to intended operati
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- delimiter), leave blank for all data ("1900-01-01")
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

The console window on the right shows the R version information and the license text:

```
R version 3.2.3 (2015-12-10) -- "wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

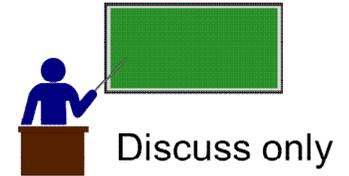
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```



There are 5 places in the MasterScript where you'll need to make entries...



1) Select the **operation**

- GetGageData, QCRaw, Aggregate or SummaryStats

2) Enter the **SiteID**

3) Enter the **data/sensor type**

- Air, water, air/water, gage, air/water/gage, air/gage, water/gage

4) Enter a **start date**

5) Enter an **end date**

```
16 ~~~~~  
17 # USER input in this section (see end of script for explanations)  
18 ~~~~~  
19 #  
20 # PROMPT; operation  
21 Selection.operation <- c("GetGageData","QCRaw", "Aggregate", "SummaryStats")  
22 myData.operation <- Selection.operation[2] #number corresponds to intended op  
23 #  
24 # PROMPT; Site ID  
25 # single site; "test2"  
26 # group of sites; ("test2","HRCC", "PBCC", "ECO66G12", "ECO66G20", "ECO68  
27 myData.SiteID <- "ECO66G12"  
28 #  
29 # PROMPT; Data Type  
30 # type of data file  
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG","WG") # only one at  
32 myData.Type <- Selection.Type[3] #number corresponds to intended operati  
33 #  
34 # PROMPT; Start Date  
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")  
36 myData.DateRange.Start <- "2013-01-01"  
37 #  
38 # PROMPT; End Date  
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)  
40 myData.DateRange.End <- "2013-12-31"  
41 ~~~~~
```

This only shows the Source (left panel)

Select operation



Enter [2]

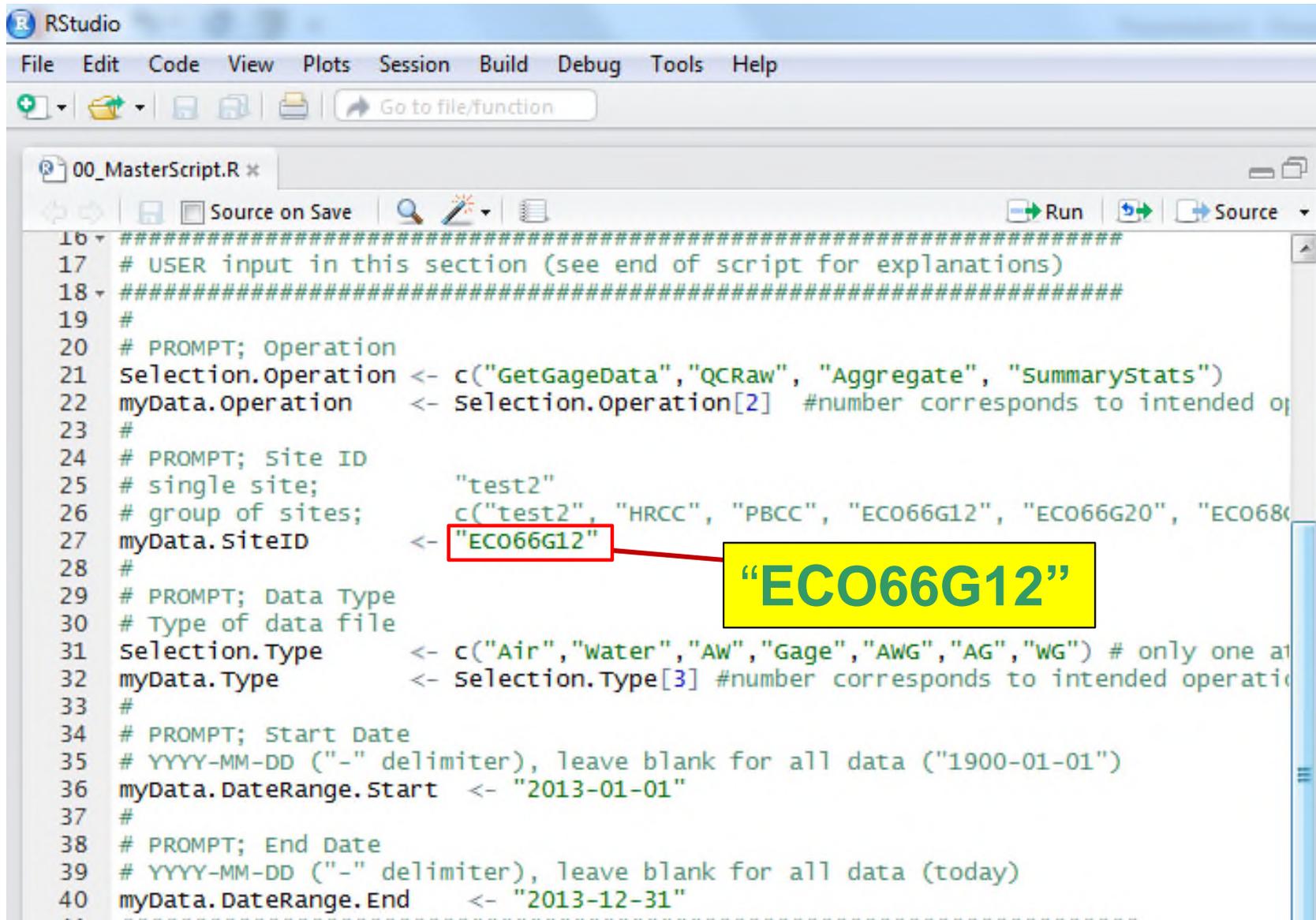
```
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
00_MasterScript.R x
Source on Save Run Source
16 #####
17 # USER input in this section (see end of script for explanations)
18 #####
19 #
20 # PROMPT; operation
21 Selection.Operation <- c("GetGageData", "QCRaw", "Aggregate", "SummaryStats")
22 myData.Operation <- Selection.Operation[2] #number corresponds to intended op
23 #
24 # PROMPT; site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "EC
27 myData.siteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air", "water", "AW", "Gage", "A
32 myData.Type <- Selection.Type[3] #number corr
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter), leave blank for all dat
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 #####
```

Selection.Operation
[1] = GetGageData
[2] = QCRaw
[3] = Aggregate
[4] = SummaryStats

This only shows the Source (left panel)

Enter the SiteID

“ECO66G12”



```
16 #####
17 # USER input in this section (see end of script for explanations)
18 #####
19 #
20 # PROMPT; operation
21 Selection.Operation <- c("GetGageData","QCRaw", "Aggregate", "SummaryStats")
22 myData.Operation    <- Selection.Operation[2] #number corresponds to intended op
23 #
24 # PROMPT; site ID
25 # single site;           "test2"
26 # group of sites;       c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "ECO680
27 myData.SiteID        <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type       <- c("Air","water","AW","Gage","AWG","AG","WG") # only one at
32 myData.Type         <- Selection.Type[3] #number corresponds to intended operati
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
40 myData.DateRange.End   <- "2013-12-31"
41 #####
```

This only shows the Source (left panel)

Enter the data/sensor type



Enter [AW]

```
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
00_MasterScript.R x
Source on Save
16 #####
17 # USER input in this section (see end of script)
18 #####
19 #
20 # PROMPT; Operation
21 Selection.Operation <- c("GetGageData","QCRaw",
22 myData.Operation <- Selection.Operation[2]
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PECC")
27 myData.SiteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG","WG") # only one at
32 myData.Type <- Selection.Type[3] #number corresponds to intended operatio
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 #####
```

- Selection.Type
- [1] = Air
- [2] = water
- [3] = AW (air & water)
- [4] = Gage
- [5] = AWG (air, water, gage)
- [6] = AG (air & gage)
- [7] = WG (water & gage)

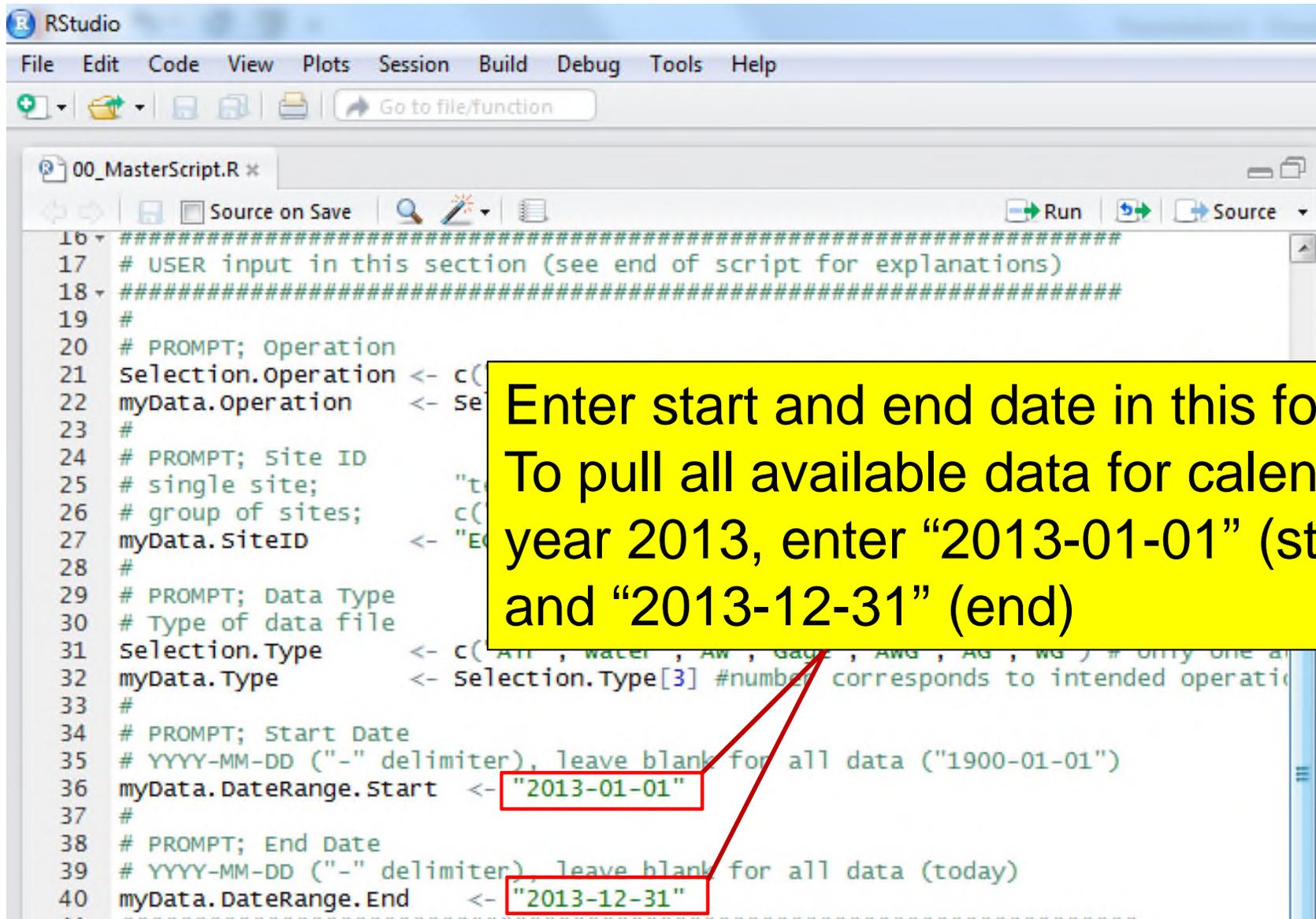
This only shows the Source (left panel)

Enter start & end dates



Start “2013-01-01”; End “2013-12-31”

See next slide for other options

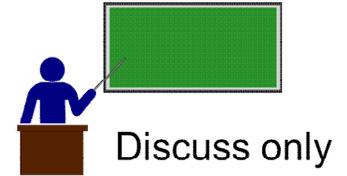


```
16 #####
17 # USER input in this section (see end of script for explanations)
18 #####
19 #
20 # PROMPT; Operation
21 Selection.Operation <- c("All", "Water", "AW", "Gage", "AWG", "AG", "WG") # Only one allowed
22 myData.Operation <- Selection.Operation
23 #
24 # PROMPT; Site ID
25 # single site; "t"
26 # group of sites; c("E", "W", "S")
27 myData.SiteID <- "E"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("All", "Water", "AW", "Gage", "AWG", "AG", "WG") # Only one allowed
32 myData.Type <- Selection.Type[3] #number corresponds to intended operation
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 #####
```

Enter start and end date in this format
To pull all available data for calendar year 2013, enter “2013-01-01” (start) and “2013-12-31” (end)

This only shows the Source (left panel)

Date range



The script looks through the date ranges in the file names and runs QC on any individual files that have dates that overlap with the range that you enter.

For this exercise, let's run both files at once. You can do this several different ways.

- Enter 2013-01-01 (start) and 2013-12-31 (end) ← Let's use this
- Enter 2013-04-26 (start) and 2013-10-15 (end)

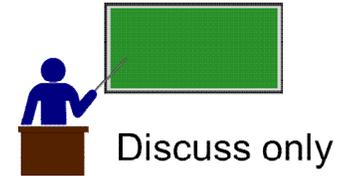
If you want to pull all available data for this site, leave the entry blank (you still need start and end quotes) - ""

What if you only want to run one file at time? Pick a date range that covers only the desired file. You can do this several different ways. One example –

- Enter 2013-04-26 (start) to 2013-07-24 (end)*

*if you entered 2013-07-25 as the end date, it would include the 2nd file because the date ranges overlap.

Entries recap



```
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
00_MasterScript.R x
Source on Save
16 ~~~~~
17 # USER input in this section (see end of script for explanat
18 ~~~~~
19 #
20 # PROMPT; Operation
21 Selection.Operation <- c("GetGageData","QCRaw", "Aggregate",
22 myData.Operation <- Selection.Operation[2] #number corre
23 #
24 # PROMPT; site ID
25 # single site; "test2"
26 # group of sites; c("test2" "HRCC", "PBCC", "ECO66G12"
27 myData.SiteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG
32 myData.Type <- Selection.Type[3] #number corresponds
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("190
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 ~~~~~
```

The 5 entries should read –

- Operation [2]
- SiteID “ECO66G12”
- Data type [3]
- Start date “2013-01-01”
- End date “2013-12-31”

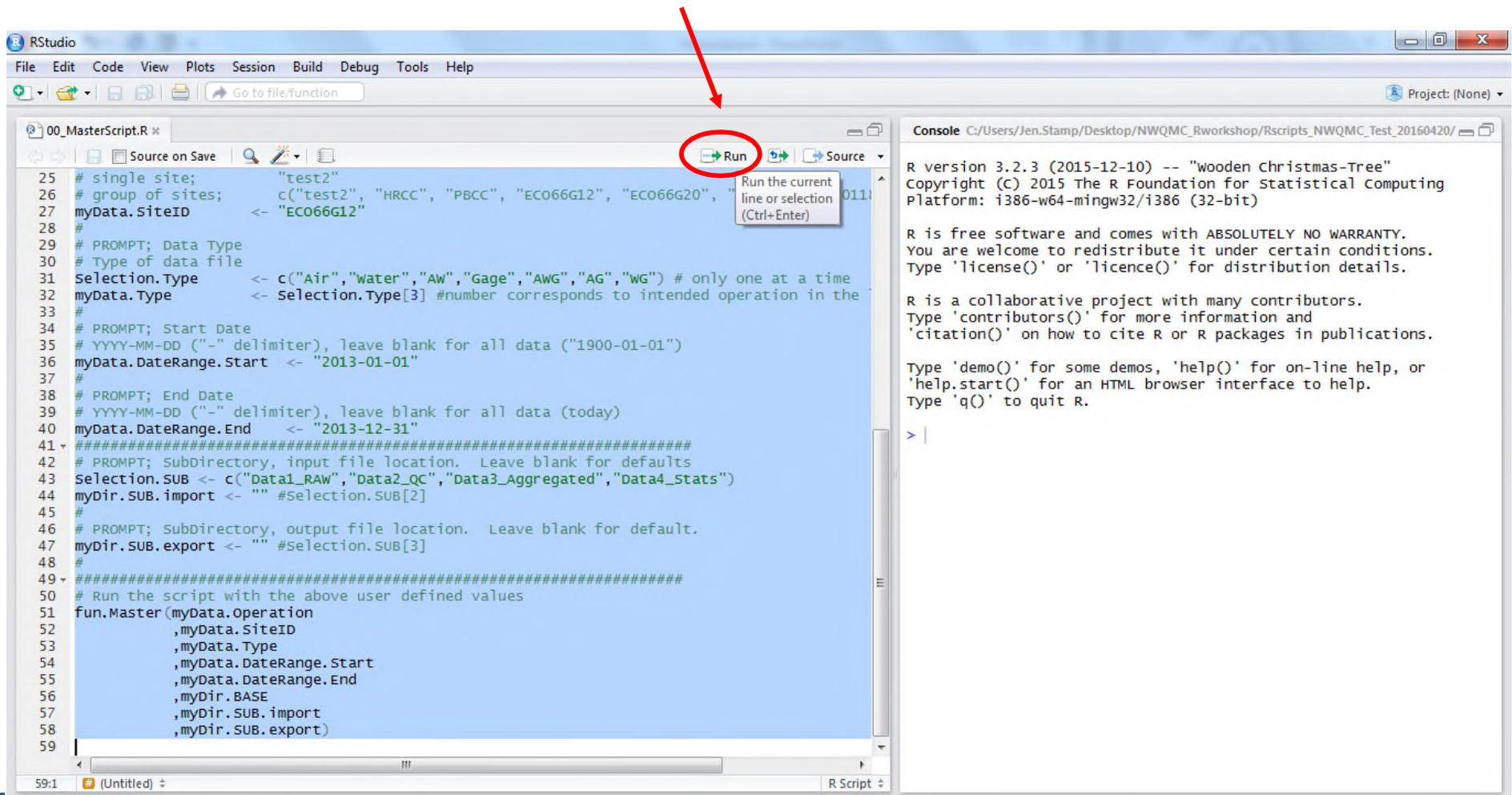
This only shows the Source (left panel)

2. Run the script



Using your cursor, highlight the script from line 7 to the end; it will turn blue)

Then click RUN



```
25 # single site;          "test2"
26 # group of sites;      c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "
27 myData.SiteID         <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type        <- c("Air", "water", "Aw", "Gage", "AWG", "AG", "wg") # only one at a time
32 myData.Type           <- Selection.Type[3] #number corresponds to intended operation in the
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End   <- "2013-12-31"
41 #####
42 # PROMPT; SubDirectory, input file location. Leave blank for defaults
43 Selection.SUB          <- c("Data1_RAW", "Data2_QC", "Data3_Aggregated", "Data4_Stats")
44 myDir.SUB.import       <- "" #Selection.SUB[2]
45 #
46 # PROMPT; SubDirectory, output file location. Leave blank for default.
47 myDir.SUB.export       <- "" #Selection.SUB[3]
48 #
49 #####
50 # Run the script with the above user defined values
51 fun.Master(myData.Operation
52            ,myData.SiteID
53            ,myData.Type
54            ,myData.DateRange.Start
55            ,myData.DateRange.End
56            ,myDir.BASE
57            ,myDir.SUB.import
58            ,myDir.SUB.export)
59
```

Run the current line or selection (Ctrl+Enter)

R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (c) 2015 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |

Check the Console (right) panel



If the script ran successfully, you should see a running log like this...

```
Console C:/Users/Jen.Stamp/Desktop/NWQMC_Rworkshop/Rscripts_NWQMC_Test_20160420/
> #
> #####
> # Run the script with the above user
> fun.Master(myData.operation
+ ,myData.SiteID
+ ,myData.Type
+ ,myData.DateRange.Start
+ ,myData.DateRange.End
+ ,myDir.BASE
+ ,myDir.SUB.import
+ ,myDir.SUB.export)
[1] "Total files to process = 6"
[1] "Processing item 1 of 6, SKIPPED (Non-Match, siteID= 01174565_Gage_
20140424_20140710.csv."
[1] "Processing item 2 of 6, WO
CO66G12_AW_20130426_20130725.cs
[1] "Processing item 2 of 6, WO
66G12_AW_20130426_20130725.csv."
[1] "Processing item 2 of 6, WO
6G12_AW_20130426_20130725.csv."
[1] "Processing item 2 of 6, WO
G12_AW_20130426_20130725.csv."
[1] "Processing item 2 of 6, WO
EC066G12_AW_20130426_20130725.c
[1] "Processing item 2 of 6, WORKING (QC Tests and Flags - Discharge), E
CO66G12_AW_20130426_20130725.csv."

processing file: Report_QC.rmd
|....
6%
ordinary text without R code

|.....
2%
label: unnamed-chunk-1 (with options
List of 3
$ results: chr "asis"
$ echo : logi FALSE
```

First you'll see blue text

Then several rows of "processing items" in black text

Then you'll see references to 'label: unnamed-chunks'

Then you'll see references (red text) to outputs for the first file (April 26-July 25)

```
Console C:/Users/Jen.Stamp/Desktop/NWQMC_Rworkshop/Rscripts_NWQMC_Test_20160420/
ordinary text without R code

|.....
4%
label: unnamed-chunk-8 (with options
List of 3
$ results: chr "asis"
$ echo : logi FALSE
$ warning: logi FALSE

|.....
0%
ordinary text without R code

output file: Report_QC.knit.md

"C:/Users/JEN~1.STA/AppData/Local/Pandoc/pandoc" +RTS -K512m -RTS Report
_QC.utf8.md --to docx --from markdown+autolink_bare_uris+ascii_identifie
rs+tex_math_single_backslash-implicit_figures --output "C:/Users/Jen.Sta
mp/Desktop/NWQMC_Rworkshop/Rscripts_NWQMC_Test_20160420/Data2_QC/QC_EC06
6G12_AW_20130725_20131015_Report_QC.docx" --highlight-style=...

output created: Data2_QC/QC
[1] "Task COMPLETE. QC Repo
[1] "User defined parameter
nge (2013-07-25 to 2013-10-
[1] "Processing item 3 of 6
.
[1] "Processing item 4 of 6
0424_20140710.csv."
[1] "Processing item 5 of 6
40424_20140710.csv."
[1] "Processing item 6 of 6
140424_20140710.csv."
[1] "Task COMPLETE; 0.12 mi
There were 12 warnings (use warnings() to see them)
> |
```

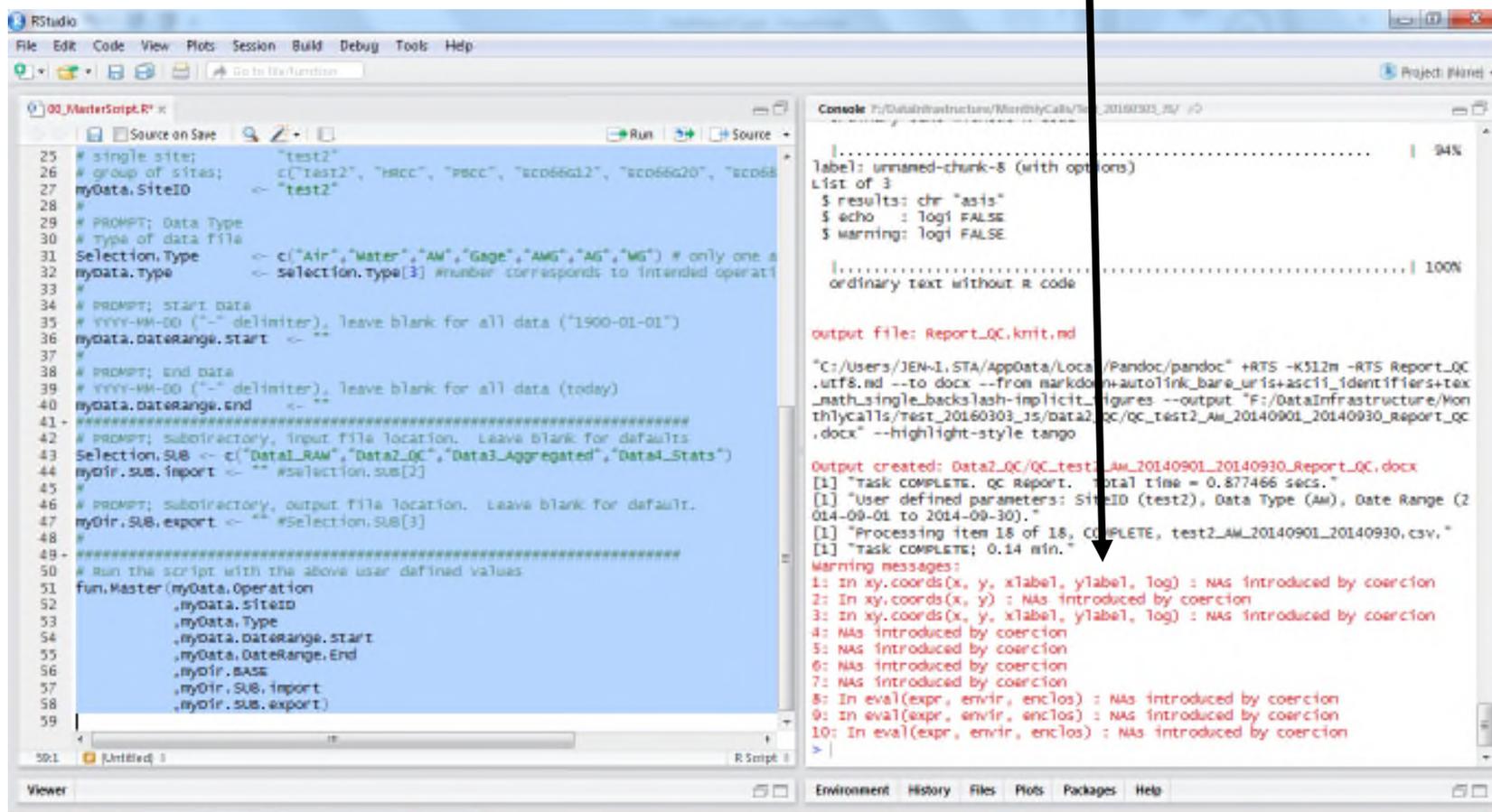
Then you'll see another round of references to 'label: unnamed-chunks'

Finally you'll see references in red text to outputs for the second file (July 25-Oct 15) ending with TASK COMPLETE

Warning Messages

In the Console output you may see some warning messages in red text. These sometimes appear, even if the script runs properly.

If warning messages appear, check your **Data2_QC** folder to make sure the outputs were generated.



```
25 # single site;           "test2"
26 # group of sites;      c("test2", "hccc", "hccc", "sc066c12", "sc066c20", "sc066c20")
27 myData.SiteID         <- "test2"
28
29 # PROMPT: Data Type
30 # Type of data file
31 Selection.Type        <- c("Air","Water","AM","Gage","AMG","AG","WG") # only one s
32 myData.Type           <- Selection.Type[3] #number corresponds to intended operati
33
34 # PROMPT: START DATA
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.start <- ""
37
38 # PROMPT: end DATA
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.end   <- ""
41
42 # PROMPT: subdirectory, input file location. Leave blank for defaults
43 Selection.SUB          <- c("Data_RAW","Data2_QC","Data3_Aggregated","Data4_Stats")
44 myDir.SUB.import       <- "" #selection.sub[2]
45
46 # PROMPT: subdirectory, output file location. Leave blank for default.
47 myDir.SUB.export       <- "" #Selection.SUB[3]
48
49
50 # Run the script with the above user defined values
51 fun.Master(myData.Operation
52            ,myData.SiteID
53            ,myData.Type
54            ,myData.DateRange.start
55            ,myData.DateRange.End
56            ,myDir.SUB.import
57            ,myDir.SUB.export)
58
59
```

```
..... | 94%
label: unnamed-chunk-8 (with options)
List of 3
 $ results: chr "asis"
 $ echo   : log1 FALSE
 $ warning: log1 FALSE

..... | 100%
ordinary text without R code

output file: Report_QC.knit.md

"C:/Users/JEN-I.STA/AppData/Local/Pandoc/pandoc" +RTS -K512m -RTS Report_QC
.utf8.md --to docx --from markdown+autolink_bare_uris+ascii_identifiers+tex
_math_single_backslash-implicit_figures --output "F:/DataInfrastructure/Mon
thlycalls/test_20160303_js/data2_qc/QC_test2_AM_20140901_20140930_Report_QC
.docx" --highlight-style tango

Output created: Data2_QC/QC_test2_AM_20140901_20140930_Report_QC.docx
[1] "task COMPLETE. QC Report. Total time = 0.877466 secs."
[1] "User defined parameters: SiteID (test2), Data Type (AM), Date Range (2
014-09-01 to 2014-09-30)."
[1] "Processing item 18 of 18, COMPLETE, test2_AM_20140901_20140930.csv."
[1] "task COMPLETE; 0.14 min."

Warning messages:
1: In xy.coords(x, y, xlabel, ylabel, log) : NAs introduced by coercion
2: In xy.coords(x, y) : NAs introduced by coercion
3: In xy.coords(x, y, xlabel, ylabel, log) : NAs introduced by coercion
4: NAs introduced by coercion
5: NAs introduced by coercion
6: NAs introduced by coercion
7: NAs introduced by coercion
8: In eval(expr, envir, enclos) : NAs introduced by coercion
9: In eval(expr, envir, enclos) : NAs introduced by coercion
10: In eval(expr, envir, enclos) : NAs introduced by coercion
>
```

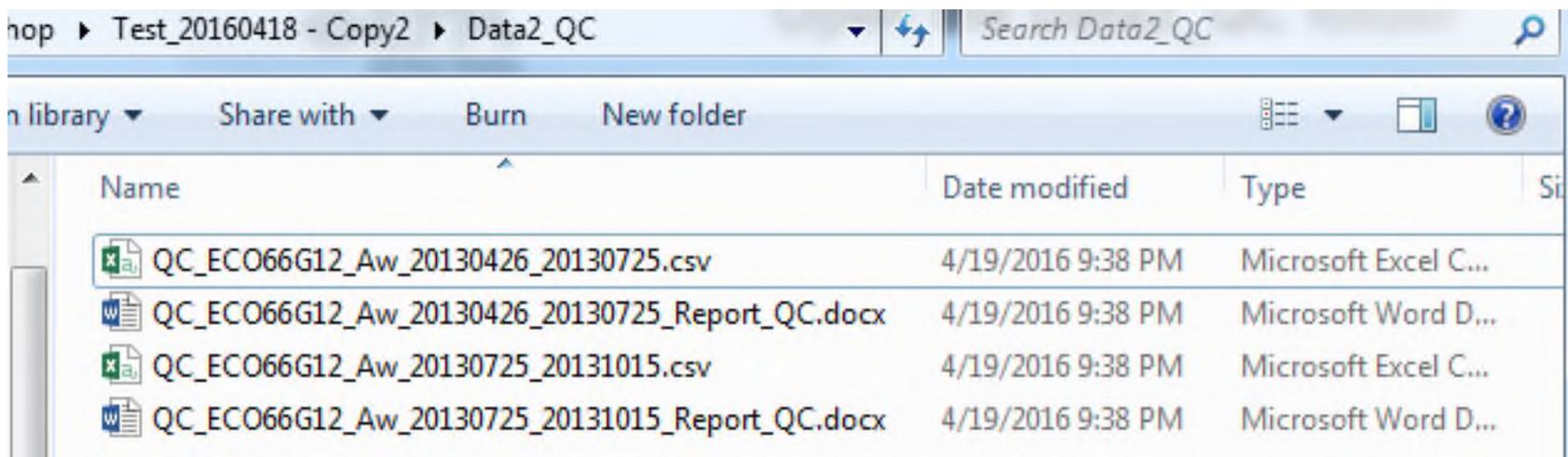
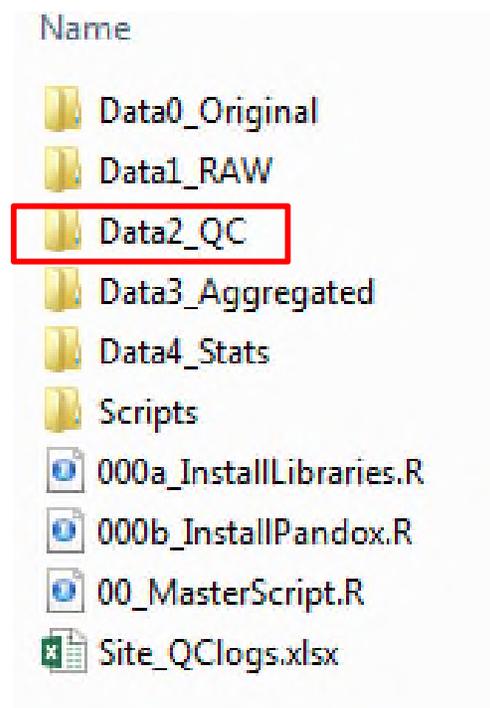
Note: it typically takes less than a minute for the script to run on these test data. If it takes longer, there may be a problem.

3. Open the Data2_QC folder



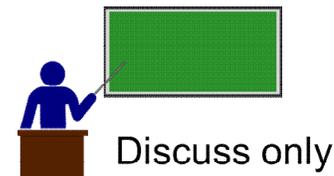
Verify that there are 2 outputs for each file
(4 total files):

- **csv** file (=the original Data1_RAW file with new columns added in – flags, raw + final data, and comments fields)
- **QC report** (Word document) with summary of missing data, flags, time series plots



Overview of QC outputs

2 outputs



QC report (Word document)

.csv file



Data Quality Control Report

Report Date: 2016-02-05

DATA FILE INFORMATION

Filename: QC_test1_Aw_20130725_20131015.csv

SiteID: test1

Period of Record, Requested: 2013-07-25 to 2013-10-15

Period of Record, Actual: 2013-07-25 to 2013-10-15

Recording Interval: 30 minutes

Data Type: Aw

Parameters Included: Water.Temp.C, Air.Temp.C, Water.BP.psi, Air.BP.psi, Water.Level.ft.

DATA SUMMARY, OVERALL

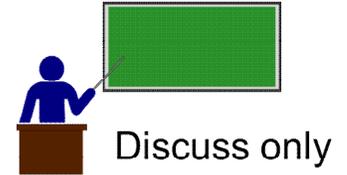
Month and Day versus number of records.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
8	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48
9	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48
10	48	48	48	48	48	48	48	48	48	48	48	48	48	49	48
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	31
7	0	0	0	0	0	0	0	0	0	27	48	48	48	48	48

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
SiteID	Date	Time	Date.Time	Water.Temp.C	Air.Temp.C	Water.BP.psi	Air.BP.psi	Water.Level.ft	Water.Level.ft	Water.Level.ft	Water.Temp.C	Flag	Air.Temp.C	Flag
1	Test1	4/26/2013	10:30:00	10.061	15.282	14.8024	14.3638	0.985	NA	1				
2	Test1	4/26/2013	11:00:00	10.357	14.134	14.7461	14.3684	0.854	NA	2				
3	Test1	4/26/2013	11:30:00	10.748	14.23	14.7428	14.3601	0.855	NA	3				
4	Test1	4/26/2013	12:00:00	11.139	15.473	14.7441	14.3627	0.853	NA	4				
5	Test1	4/26/2013	12:30:00	11.528	17.284	14.7411	14.3612	0.854	NA	5				
6	Test1	4/26/2013	13:00:00	11.916	18.896	14.7421	14.3626	0.948	NA	6				
7	Test1	4/26/2013	13:30:00	12.207	19.662	14.7453	14.3629	0.855	NA	7				
8	Test1	4/26/2013	14:00:00	12.594	20.404	14.7551	14.3537	0.853	NA	8				
9	Test1	4/26/2013	14:30:00	12.883	20.71	14.7515	14.3484	0.857	NA	9				
10	Test1	4/26/2013	15:00:00	13.173	20.901	14.7234	14.3426	0.852	NA	10				
11	Test1	4/26/2013	15:30:00	13.365	20.885	14.7171	14.3331	0.859	NA	11				
12	Test1	4/26/2013	16:00:00	13.558	21.091	14.7109	14.3302	0.852	NA	12				
13	Test1	4/26/2013	16:30:00	13.654	21.989	14.702	14.3215	0.851	NA	13				
14	Test1	4/26/2013	17:00:00	13.75	21.392	14.6978	14.3152	0.856	NA	14				
15	Test1	4/26/2013	17:30:00	13.846	21.473	14.6953	14.3184	0.818	NA	15				
16	Test1	4/26/2013	18:00:00	13.846	20.895	14.7051	14.3239	0.853	NA	16				
17	Test1	4/26/2013	18:30:00	13.846	19.758	14.7119	14.3294	0.856	NA	17				
18	Test1	4/26/2013	19:00:00	13.75	18.426	14.7568	14.3538	0.857	NA	18				
19	Test1	4/26/2013	19:30:00	13.75	17.665	14.7413	14.3564	0.861	NA	19				
20	Test1	4/26/2013	20:00:00	13.75	16.999	14.7392	14.3551	0.859	NA	20				
21	Test1	4/26/2013	20:30:00	13.854	15.854	14.7113	14.3309	0.856	NA	21				
22	Test1	4/26/2013	21:00:00	13.558	15.187	14.6802	14.2999	0.853	NA	22				
23	Test1	4/26/2013	21:30:00	13.558	14.125	14.6994	14.3378	0.854	NA	23				
24	Test1	4/26/2013	22:00:00	13.461	13.75	14.7174	14.3351	0.855	NA	24				
25	Test1	4/26/2013	22:30:00	13.365	13.173	14.7241	14.341	0.857	NA	25				
26	Test1	4/26/2013	23:00:00	13.269	12.594	14.7421	14.3606	0.853	NA	26				
27	Test1	4/26/2013	23:30:00	13.173	12.11	14.7416	14.3558	0.863	NA	27				
28	Test1	4/27/2013	0:00:00	13.371	11.819	14.7325	14.3462	0.867	NA	28				
29	Test1	4/27/2013	0:30:00	13.076	11.722	14.718	14.3515	0.867	NA	29				
30	Test1	4/27/2013	1:00:00	12.98	11.819	14.7457	14.3567	0.866	NA	30				
31	Test1	4/27/2013	1:30:00	12.98	11.819	14.7572	14.3656	0.877	NA	31				
32	Test1	4/27/2013	2:00:00	12.883	11.819	14.7617	14.37	0.881	NA	32				
33	Test1	4/27/2013	2:30:00	12.787	11.819	14.7541	14.3519	0.901	NA	33				
34	Test1	4/27/2013	3:00:00	12.787	11.722	14.7588	14.3447	0.928	NA	34				
35	Test1	4/27/2013	3:30:00	12.69	11.819	14.7447	14.3271	0.936	NA	35				
36	Test1	4/27/2013	4:00:00	12.69	11.722	14.7377	14.3375	0.943	NA	36				
37	Test1	4/27/2013	4:30:00	12.594	11.625	14.7718	14.3374	0.975	NA	37				
38	Test1	4/27/2013	5:00:00	12.594	11.625	14.7656	14.3351	1.012	NA	38				

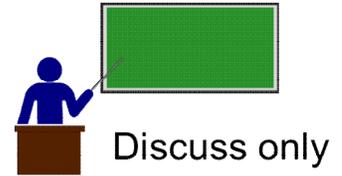
QC report (Word)

4 Main components



- 1. Data file information** (siteID, date range, parameters, recording interval)
- 2. 'Count' tables (# measurements/day)** so that you can find inconsistencies (too few measurements (=missing data) or too many measurements)
- 3. Results from QC tests** - # of entries marked as pass (P), suspect (S), fail (F) or no data (X) (see next slide)
- 4. Time series plots** – for each individual parameter & several combined parameters (water & air temperature; water temperature and water level); includes discrete measurements (if available)

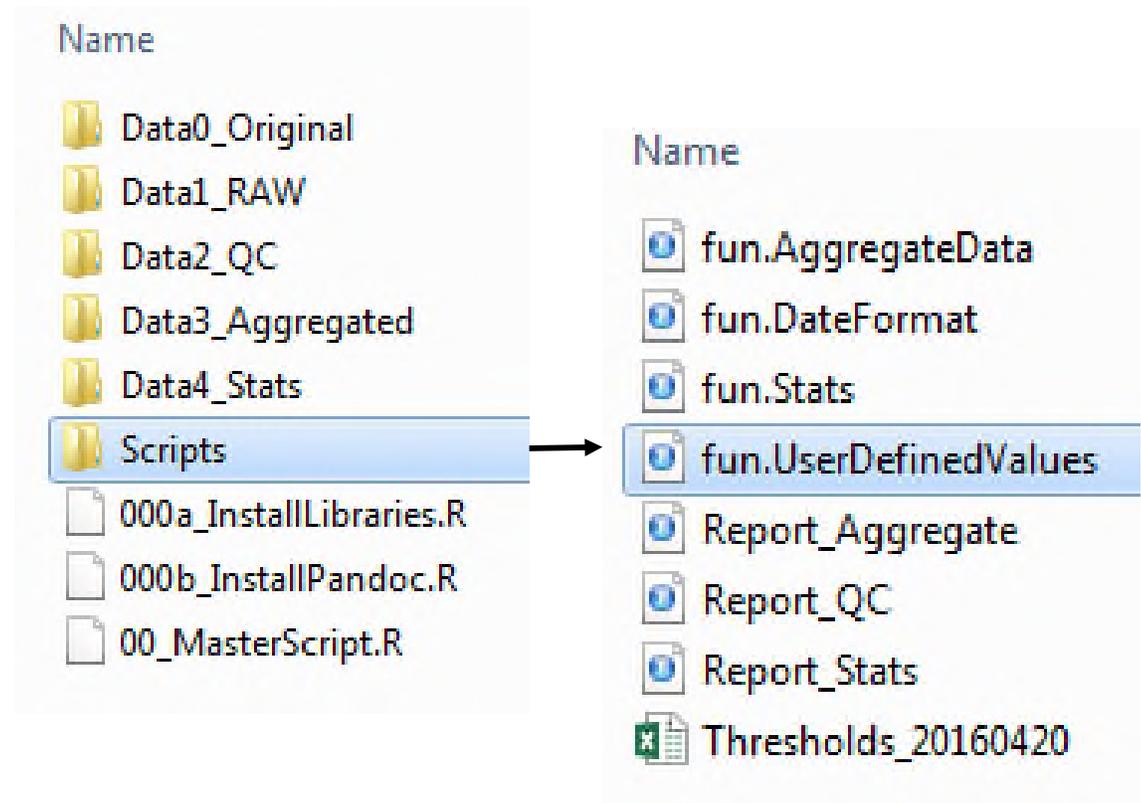
Flags/QC tests



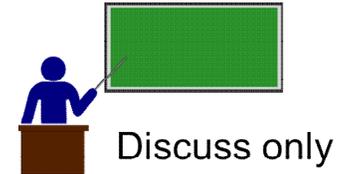
- **Unrealistic values** ('Gross range')
 - Entries are flagged if values are above or below upper and lower limits
- **Spikes**
 - Entries are flagged if adjacent points change by more than 'x' amount
- **Rate of change** (RoC)
 - Entries are flagged if the RoC exceeds a given threshold (e.g., ≥ 3 st dev within 25 hrs)
- **Flat line**
 - Entries are flagged if a certain # of consecutive measurements are within a certain amount of each other (e.g., >10 consecutive temperature measurements are within 0.01 degrees C of one another)

The thresholds for the QC tests (gross, spike, rate of change and flat line) are in the **fun.UserDefinedValues.R** file in the **Scripts** folder.

You can go into this file and change thresholds if desired.



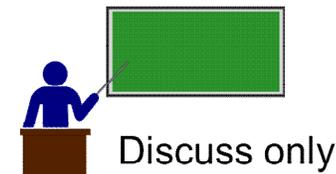
QC report – Organizational scheme



- 1. Data file information** (siteID, date range, parameters, recording interval)
- 2. Overall summary**
 - Count table
 - Overall flag (# entries marked as pass, suspect, fail or missing data, based on the *worst* flag across all of the QC tests)
- 3. Individual parameters**
 - Count table
 - Flags by QC test (gross, spike, rate of change and flat line)
 - Time series plot
- 4. Multi-parameter time series plots** (as available)

QC report – Section 1

Data file information



Data Quality Control Report

Report Date: 2016-04-21

DATA FILE INFORMATION

Filename: QC_ECO66G12_Aw_20130725_20131015.csv

SiteID: ECO66G12

Period of Record, Requested: 2013-07-25 to 2013-10-15

Period of Record, Actual: 2013-07-25 to 2013-10-15

Recording Interval: 30 minutes

Data Type: Aw

Parameters Included: Water.Temp.C, Air.Temp.C, Water.BP.psi, Air.BP.psi, Water.Level.ft, Discrete.Water.Temp.C, Discrete.Air.Temp.C, Discrete.Water.Level.ft,

QC report – Section 2

Overall Summary

'Count' tables (# measurements/day)

DATA SUMMARY, OVERALL

Month and Day versus number of records.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Month	8	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	
	9	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	
	10	48	48	48	48	48	48	48	48	48	48	48	48	49	48	21	
		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Month	7	0	0	0	0	0	0	0	0	0	27	48	48	48	48	48	48
	8	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48
	9	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	0

Expected number of records per day is 48.

Sensor was set to record at 30-minute intervals. This equates to 48 measurements per day.

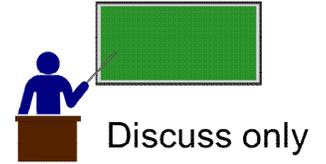
Missing measurements on days of download (July 25 and Oct 15).

October 13 had 1 extra measurement (49).

This is an example from a different data file (not the files you are working with)

QC report – Section 2

Overall Summary



Overall flags by parameter

Water.Temp.C

P	S
3905	31

Air.Temp.C

P
3936

Water.BP.psi

P	S
3934	2

Air.BP.psi

P
3936

Water.Level.ft

P	S
3911	25

For each parameter, # of entries marked as pass (P), suspect (S), fail (F) or missing data (X), for any of the QC tests.

In this example, 3905 of the water temperature entries were marked as pass, and 31 were marked as suspect.

QC report – Section 3

Individual parameters – e.g., water temperature

QC TESTS, Water.Temp.C

Parameter

Flags

Flag summary, overall number of records.

P	S
3905	31

Flag summary (overall)

Month and Day versus number of records.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
8	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	
9	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	
10	48	48	48	48	48	48	48	48	48	48	48	48	48	48	21	
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
7	0	0	0	0	0	0	0	0	0	27	48	48	48	48	48	48
8	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48
9	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	0

Count table

Expected number of records per day is 48.

Flags by QC Test

Flags, Gross

P
3936

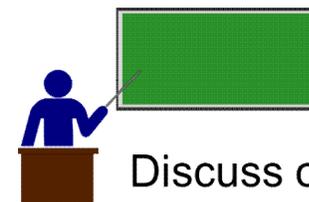
Flags, Spike

P	X
3935	1

Flags, RoC

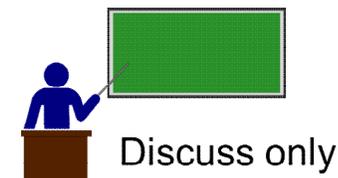
P	X
3935	1

Flag summary, broken down by QC test



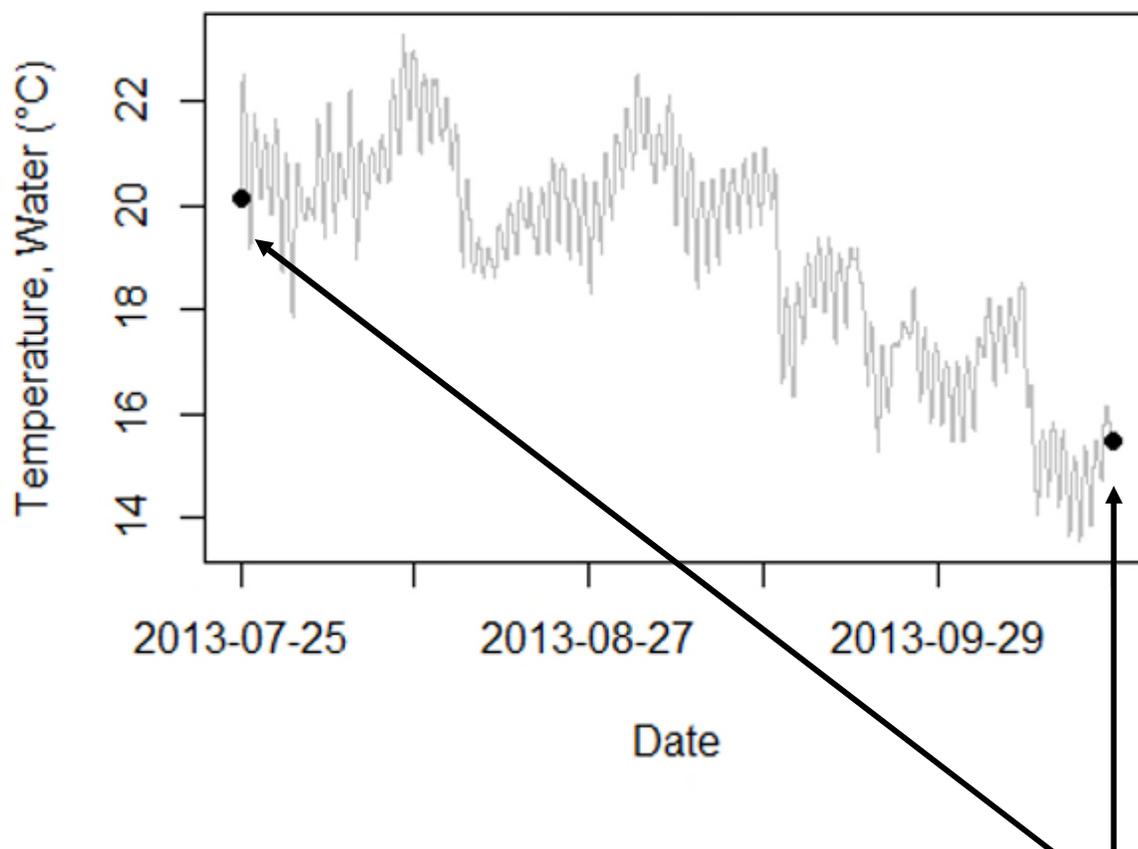
Discuss only

QC report – Section 3



Individual parameters continued...

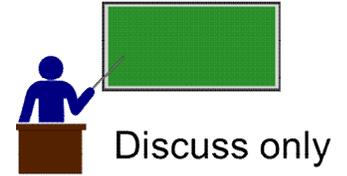
ECO66G12



Time series plot
(all values plotted
vs. date/time) for
water temperature

This example includes discrete measurements

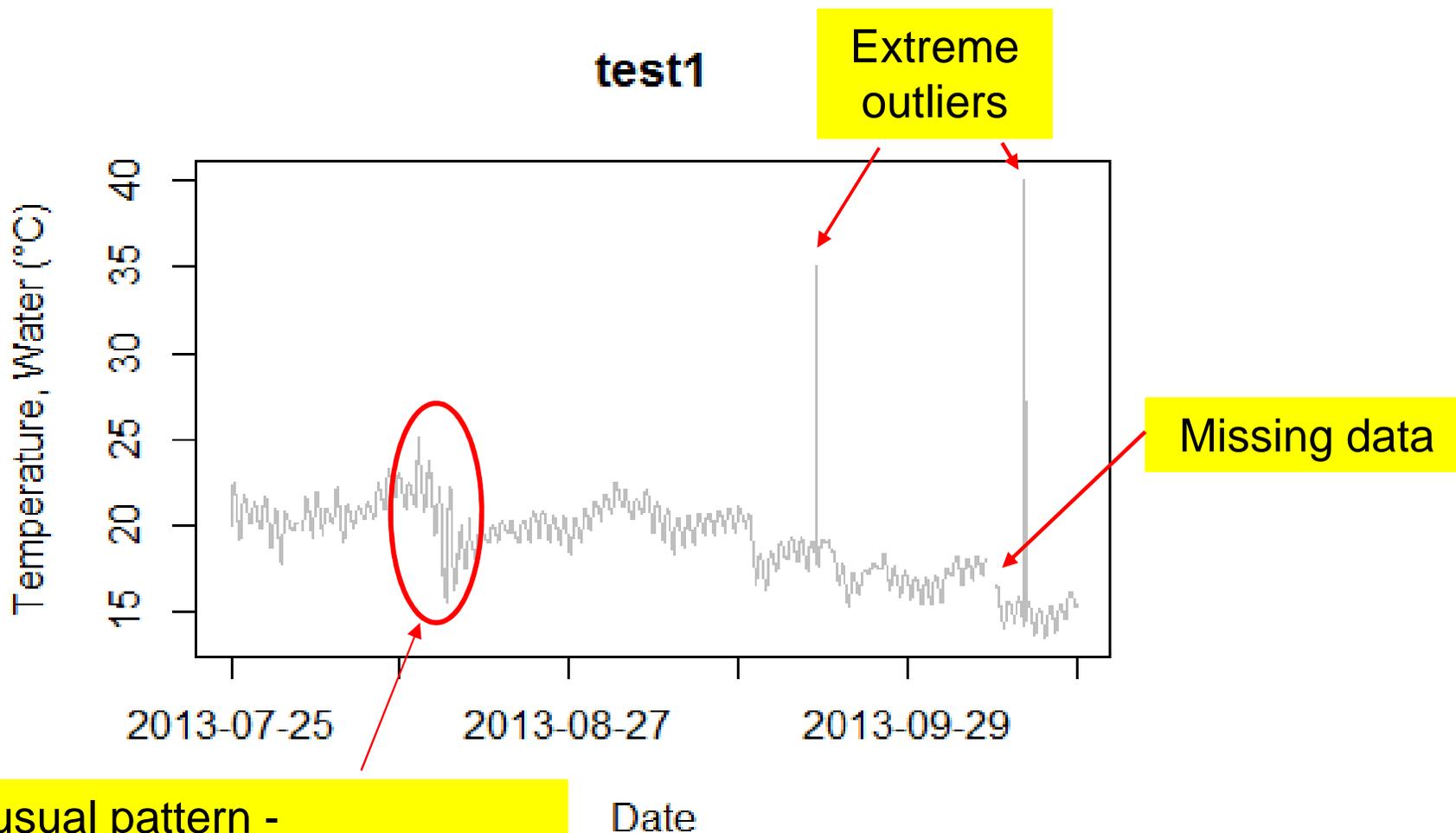
Visual checks on time series plots



- **Missing data**
- **Closely matching water and air temperature** values (this indicates that the stream sensor may have been out of the water)
- **Diel fluxes with flat tops, or “chatter”** in the gage height record (this indicates that the sensor may have been buried in sediment)
- **Water level values of 0** (this could mean that the pressure transducer was dewatered. With vented transducers, another possibility is that moisture got into the cable and caused readings of zero water depth)
- **Water level values that are negative**
- **Outliers or rapidly fluctuating values** (it is possible that the sensor moved, e.g., due to a high flow event or vandalism)
- Values recorded before the sensor was correctly positioned & stabilized, or after the sensor was removed from its location (**“trimming”**)

You can also do these time series checks on plots generated in Hoboware, Aquarius, etc.

Examples of visual checks on time series plots

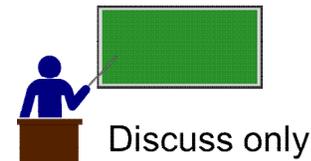


Unusual pattern -
Dewatered? Looks similar to
pattern seen in air
temperature measurements

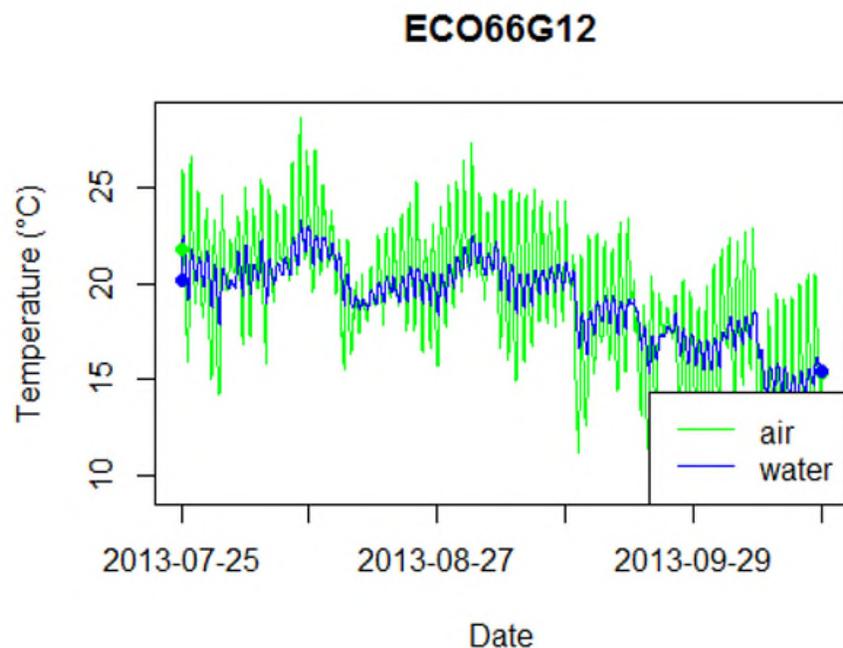
*This is an example from a
different data file (not the files
you are working with)*

QC report – Section 4

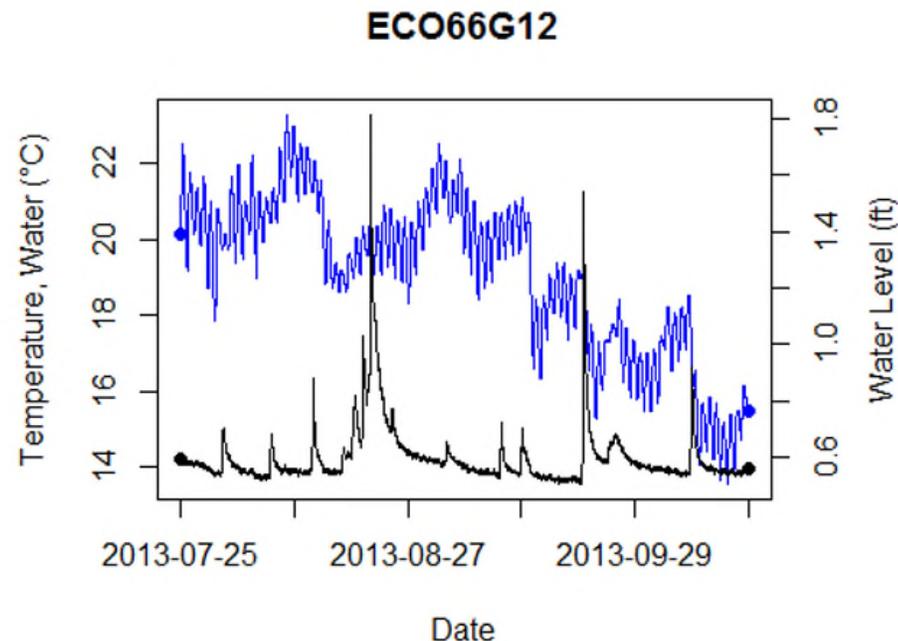
Multi-parameter time series plots



PLOT, Temperature (Air vs. Water)



PLOT, Water Level vs. Water Temperature



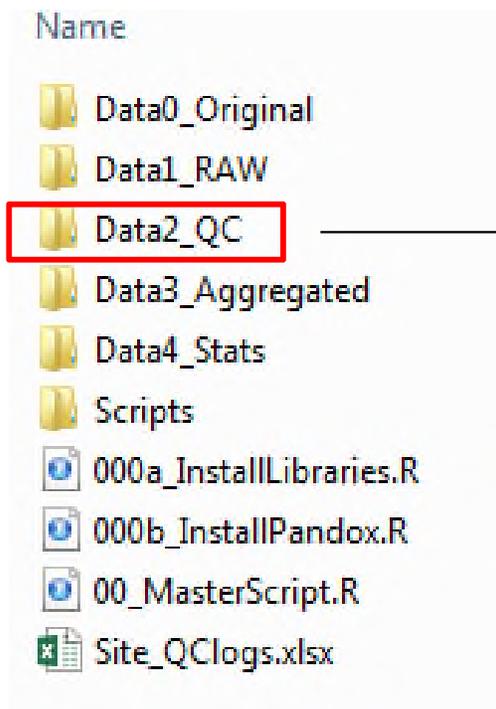
Do the patterns make sense? (e.g., water temperature should fluctuate less than air temperature; increases in water level may correspond with decreases in water temperature)

Your turn!

Open one of the Word documents &
scroll through the QC report



Any questions? Comments?



Name	Date modified	Type	Size
 QC_ECO66G12_Aw_20130426_20130725	2016-04-21 19:27	Microsoft Excel C...	958 KB
 QC_ECO66G12_Aw_20130426_20130725_Report_QC	2016-04-21 19:27	Microsoft Word D...	49 KB
 QC_ECO66G12_Aw_20130725_20131015	2016-04-21 19:27	Microsoft Excel C...	892 KB
 QC_ECO66G12_Aw_20130725_20131015_Report_QC	2016-04-21 19:27	Microsoft Word D...	49 KB

Data Quality Control Report

Report Date: 2016-04-21

DATA FILE INFORMATION

Filename: QC_ECO66G12_Aw_20130426_20130725.csv

SiteID: ECO66G12

Period of Record, Requested: 2013-04-26 to 2013-07-25

Period of Record, Actual: 2013-04-26 to 2013-07-25

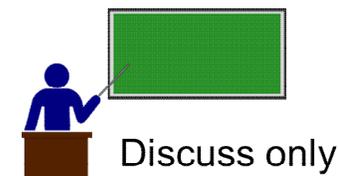
Recording Interval: 30 minutes

Data Type: Aw

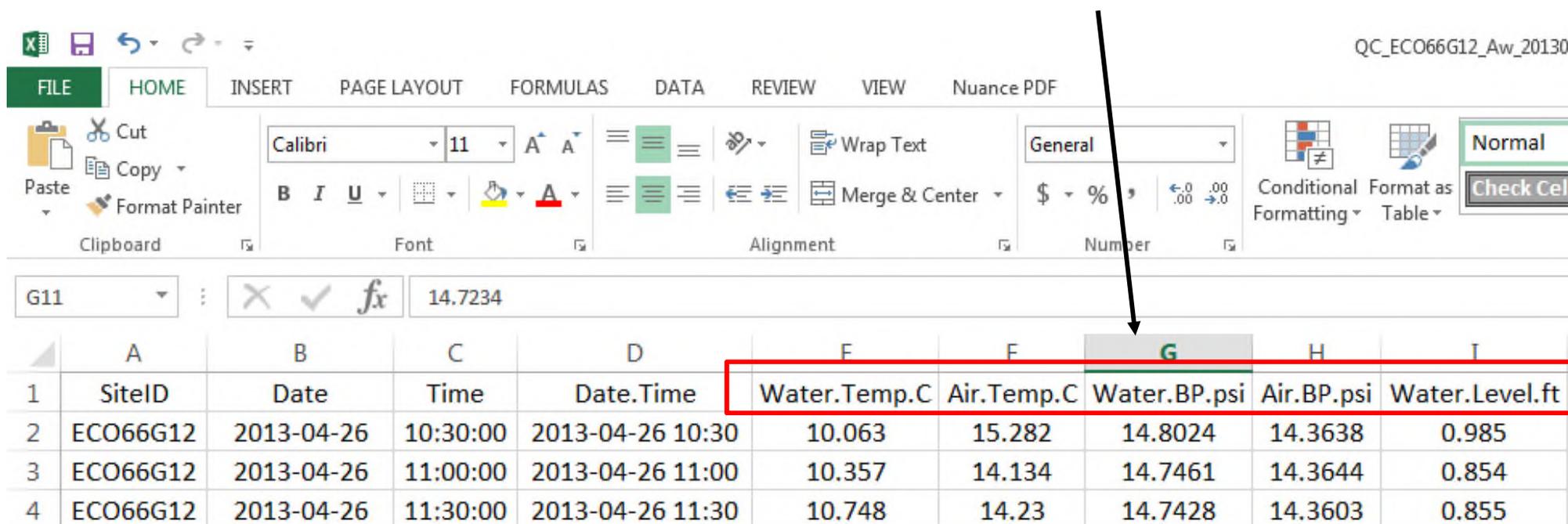
Parameters Included: Water.Temp.C, Air.Temp.C, Water.BP.psi, Air.BP.psi, Water.Level.ft, Discrete.Water.Temp.C, Discrete.Air.Temp.C, Discrete.Water.Level.ft,

Now let's go through the other QC file...

.csv file



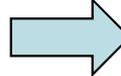
Values (water temp, air temp, water level, etc.)



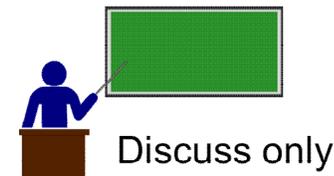
QC_ECO66G12_Aw_20130

	A	B	C	D	E	F	G	H	I
1	SiteID	Date	Time	Date.Time	Water.Temp.C	Air.Temp.C	Water.BP.psi	Air.BP.psi	Water.Level.ft
2	ECO66G12	2013-04-26	10:30:00	2013-04-26 10:30	10.063	15.282	14.8024	14.3638	0.985
3	ECO66G12	2013-04-26	11:00:00	2013-04-26 11:00	10.357	14.134	14.7461	14.3644	0.854
4	ECO66G12	2013-04-26	11:30:00	2013-04-26 11:30	10.748	14.23	14.7428	14.3603	0.855

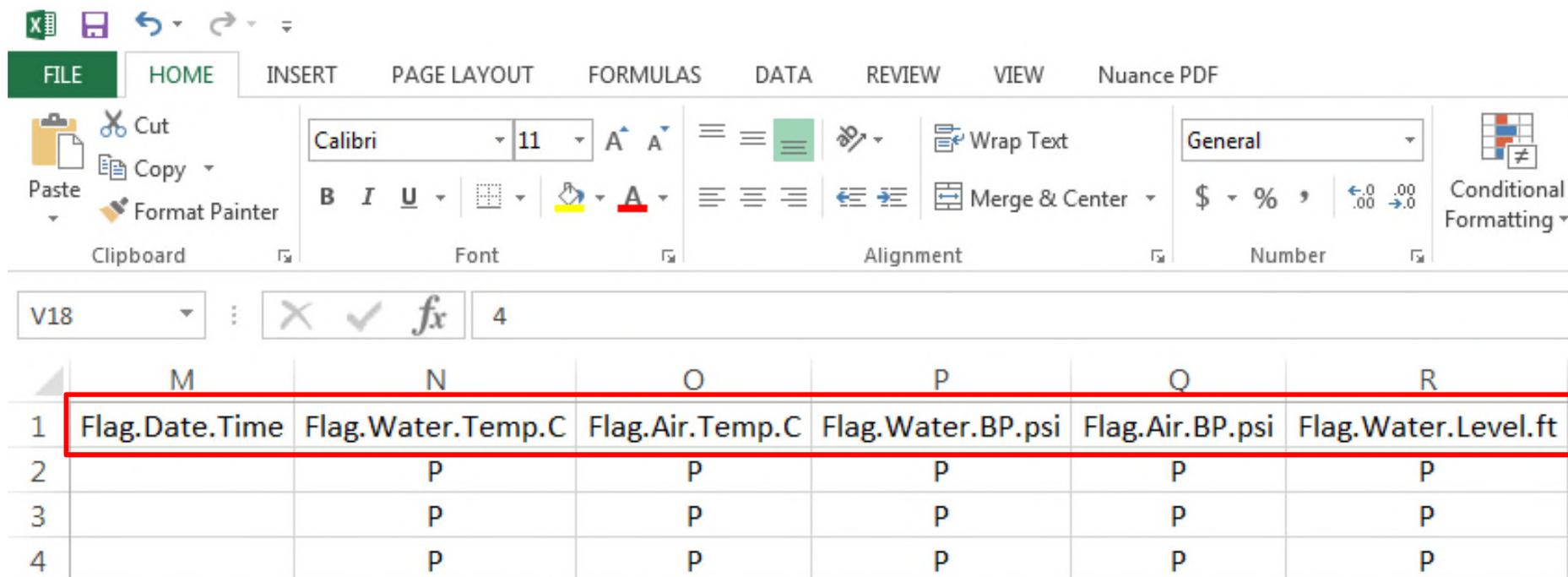
The R scripts add in many additional fields – flags (overall & each parameter/QC test), plus RAW and Comment fields

Scrolling to the right... 

.csv file continued...



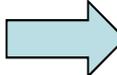
Overall flags (entries marked as pass, suspect, fail or missing data, based on the *worst* flag across all of the QC tests).



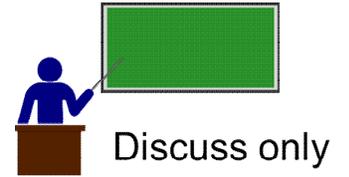
The screenshot shows the Microsoft Excel ribbon with the HOME tab selected. The ribbon includes options for FILE, HOME, INSERT, PAGE LAYOUT, FORMULAS, DATA, REVIEW, VIEW, and Nuance PDF. The ribbon also shows various formatting options such as Cut, Copy, Paste, Format Painter, Font (Calibri, 11), Alignment (Wrap Text, Merge & Center), and Number (General, \$, %, .00, .0). The formula bar shows the active cell is V18 with the value 4. The table below shows columns M through R and rows 1 through 4. The first row is highlighted with a red border.

	M	N	O	P	Q	R
1	Flag.Date.Time	Flag.Water.Temp.C	Flag.Air.Temp.C	Flag.Water.BP.psi	Flag.Air.BP.psi	Flag.Water.Level.ft
2		P	P	P	P	P
3		P	P	P	P	P
4		P	P	P	P	P

F = fail
P = pass
S = suspect
X = missing data/blank

Continue scrolling to the right 

.csv file continued...



Flags for each parameter and each QC test (gross, spike, RoC, flat)

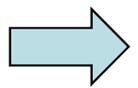
QC_ECO66G12_Aw_20130426_20130725 - Excel

ULAS DATA REVIEW VIEW Nuance PDF

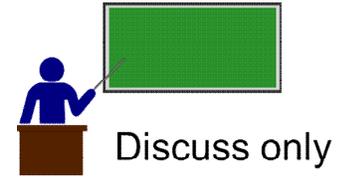


X	Y	Z	AA
Flag.Gross.Water.Temp.C	Flag.Spike.Water.Temp.C	Flag.RoC.Water.Temp.C	Flag.Flat.Water.Temp.C
P	X	X	P
P	P	P	P
P	P	P	P
P	P	P	P
P	P	P	P

Example - water temperature

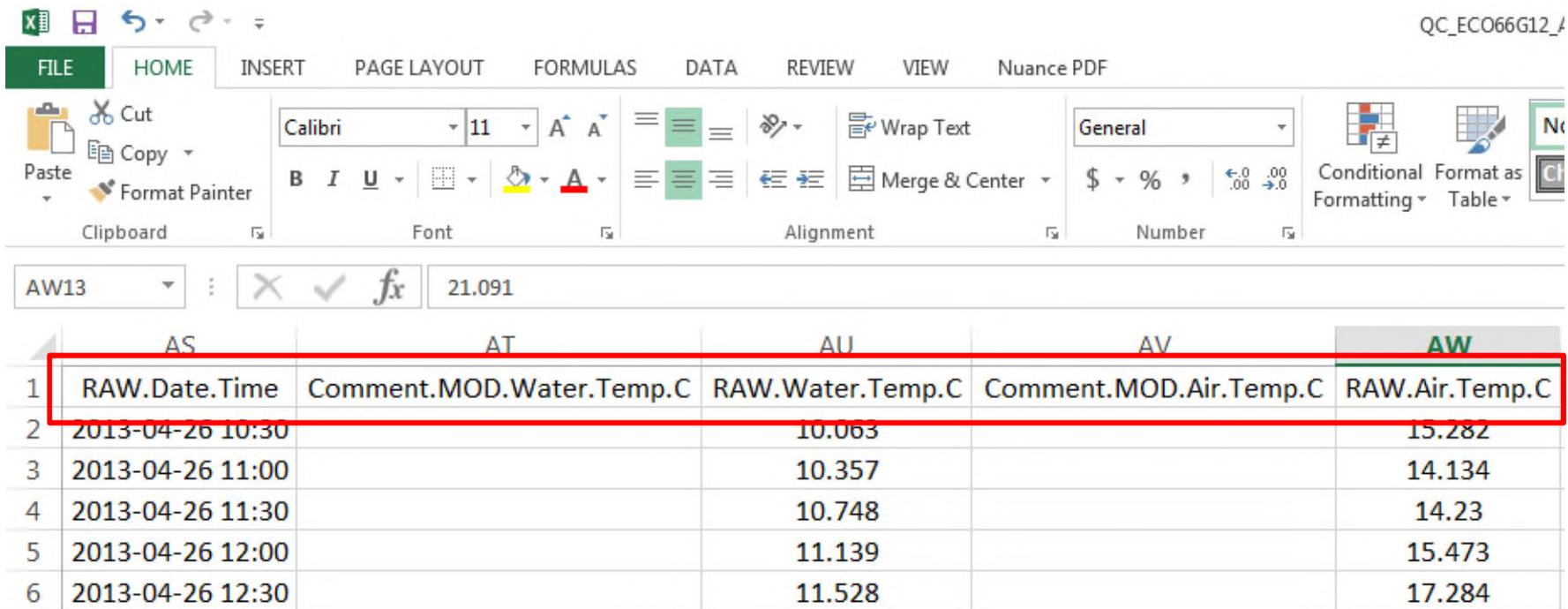
Continue scrolling to the right 

.csv file continued...



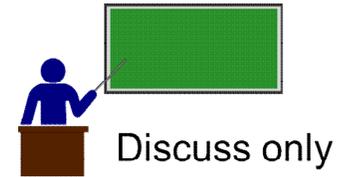
RAW = original values; if you end up making corrections in the primary value fields (in this example, Columns E-I), the RAW fields allow you to retain the original records and track any changes you make.

COMMENT.MOD = allows you to comment on any changes you make.



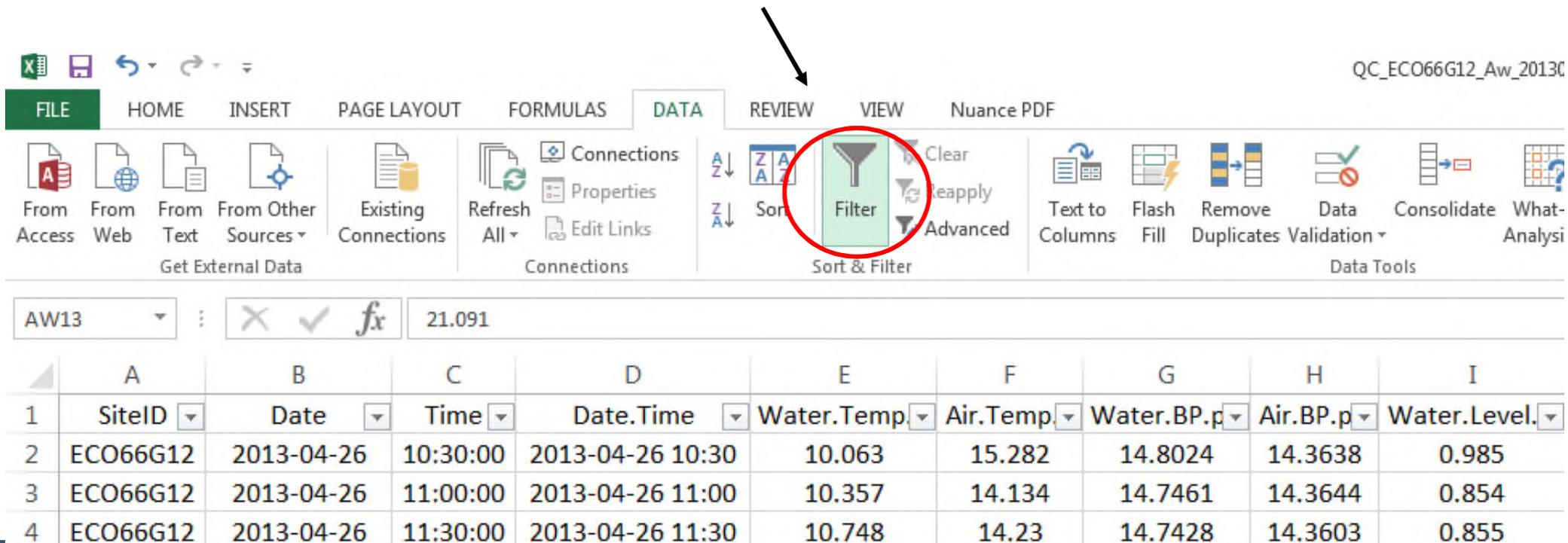
	AS	AT	AU	AV	AW
1	RAW.Date.Time	Comment.MOD.Water.Temp.C	RAW.Water.Temp.C	Comment.MOD.Air.Temp.C	RAW.Air.Temp.C
2	2013-04-26 10:30		10.063		15.282
3	2013-04-26 11:00		10.357		14.134
4	2013-04-26 11:30		10.748		14.23
5	2013-04-26 12:00		11.139		15.473
6	2013-04-26 12:30		11.528		17.284

Tips for easier navigation of the .csv files



Because the .csv files have so many columns, they can be difficult to work with.

Using the **Filter** function can help (e.g., you can filter for entries flagged as 'F' or 'S' and evaluate)



QC_ECO66G12_Aw_2013C

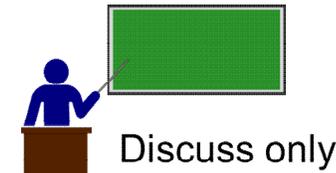
FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW Nuance PDF

From Access From Web From Text From Other Sources Existing Connections Refresh All Connections Properties Edit Links Sort & Filter Filter Sort & Filter Clear Reapply Advanced Text to Columns Flash Fill Remove Duplicates Data Validation Consolidate What-Analysis

AW13 : X ✓ fx 21.091

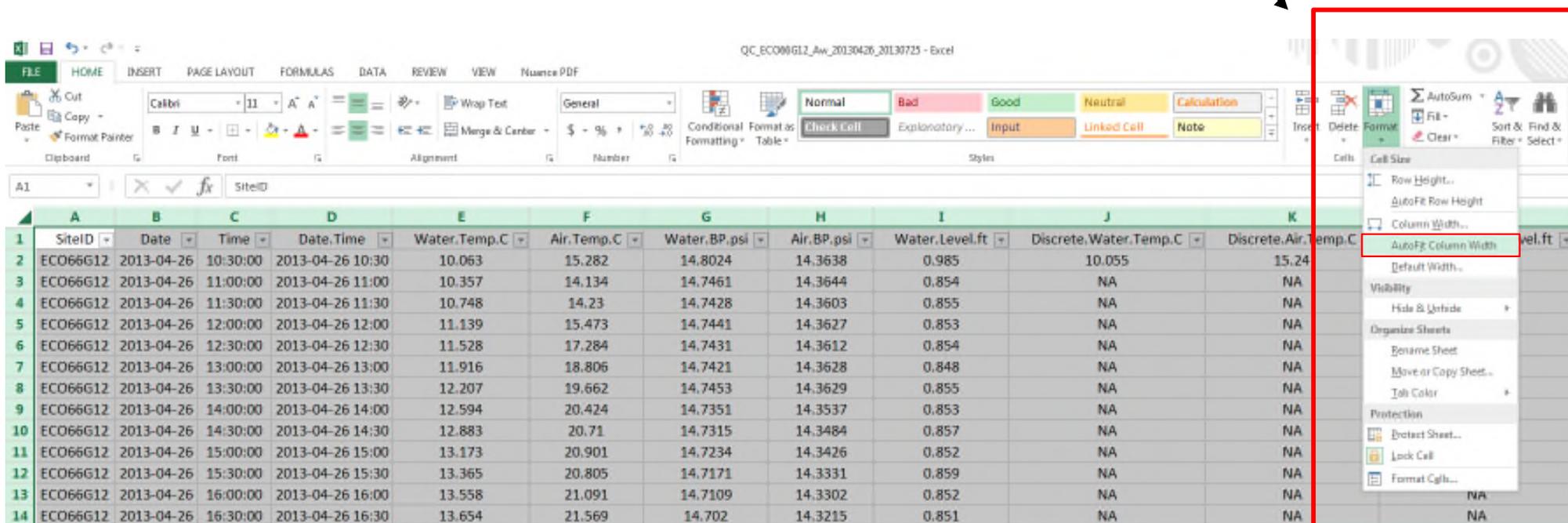
	A	B	C	D	E	F	G	H	I
1	SiteID	Date	Time	Date.Time	Water.Temp	Air.Temp	Water.BP.p	Air.BP.p	Water.Level
2	ECO66G12	2013-04-26	10:30:00	2013-04-26 10:30	10.063	15.282	14.8024	14.3638	0.985
3	ECO66G12	2013-04-26	11:00:00	2013-04-26 11:00	10.357	14.134	14.7461	14.3644	0.854
4	ECO66G12	2013-04-26	11:30:00	2013-04-26 11:30	10.748	14.23	14.7428	14.3603	0.855

Tips for easier navigation of the .csv files



Using the **Autofit Column Width** function can also help.

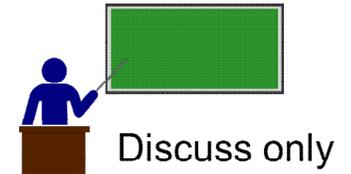
Highlight all the fields, then click **Format – Autofit Column Width**. Excel then automatically adjusts column widths so that you can see the full field names.



The screenshot shows the Microsoft Excel interface with the 'Format' menu open. The 'Autofit Column Width' option is highlighted in red. The spreadsheet below shows a table with columns for SiteID, Date, Time, Date,Time, Water.Temp.C, Air.Temp.C, Water.BP.psi, Air.BP.psi, Water.Level.ft, Discrete.Water.Temp.C, and Discrete.Air.Temp.C. The data rows show various measurements for SiteID ECO66G12 on 2013-04-26.

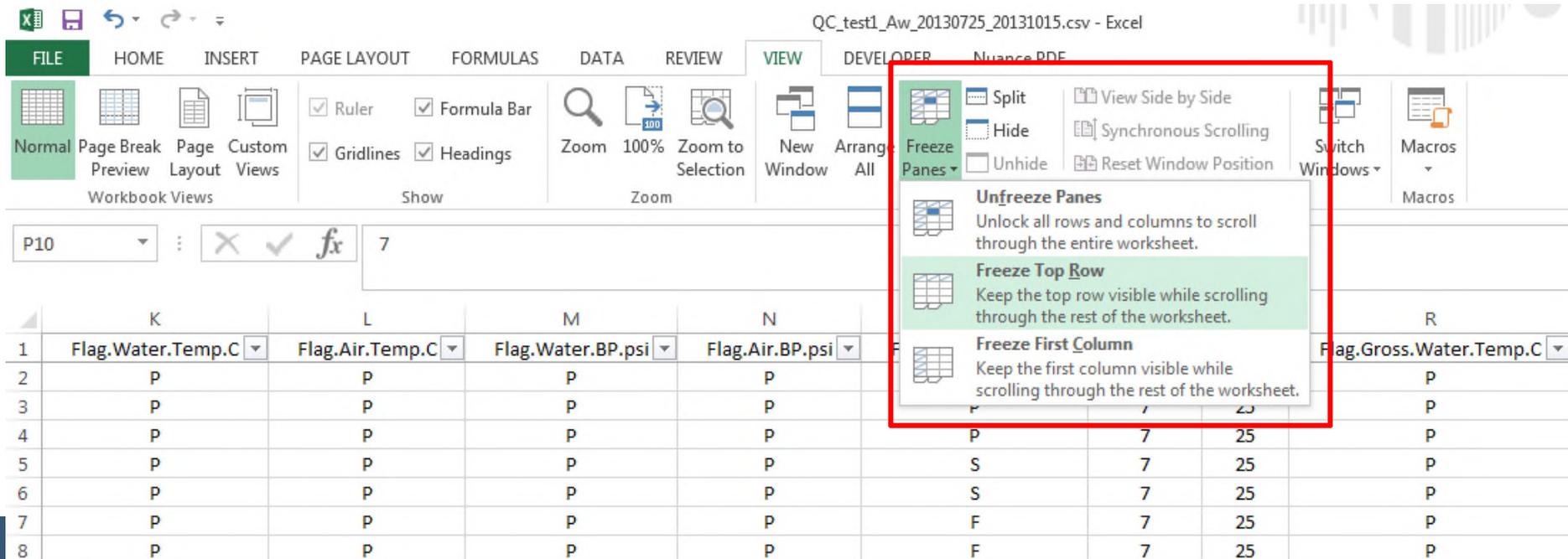
	A	B	C	D	E	F	G	H	I	J	K
	SiteID	Date	Time	Date,Time	Water.Temp.C	Air.Temp.C	Water.BP.psi	Air.BP.psi	Water.Level.ft	Discrete.Water.Temp.C	Discrete.Air.Temp.C
2	ECO66G12	2013-04-26	10:30:00	2013-04-26 10:30	10.063	15.282	14.8024	14.3638	0.985	10.055	15.24
3	ECO66G12	2013-04-26	11:00:00	2013-04-26 11:00	10.357	14.134	14.7461	14.3644	0.854	NA	NA
4	ECO66G12	2013-04-26	11:30:00	2013-04-26 11:30	10.748	14.23	14.7428	14.3603	0.855	NA	NA
5	ECO66G12	2013-04-26	12:00:00	2013-04-26 12:00	11.139	15.473	14.7441	14.3627	0.853	NA	NA
6	ECO66G12	2013-04-26	12:30:00	2013-04-26 12:30	11.528	17.284	14.7431	14.3612	0.854	NA	NA
7	ECO66G12	2013-04-26	13:00:00	2013-04-26 13:00	11.916	18.806	14.7421	14.3628	0.848	NA	NA
8	ECO66G12	2013-04-26	13:30:00	2013-04-26 13:30	12.207	19.662	14.7453	14.3629	0.855	NA	NA
9	ECO66G12	2013-04-26	14:00:00	2013-04-26 14:00	12.594	20.424	14.7351	14.3537	0.853	NA	NA
10	ECO66G12	2013-04-26	14:30:00	2013-04-26 14:30	12.883	20.71	14.7315	14.3484	0.857	NA	NA
11	ECO66G12	2013-04-26	15:00:00	2013-04-26 15:00	13.173	20.901	14.7234	14.3426	0.852	NA	NA
12	ECO66G12	2013-04-26	15:30:00	2013-04-26 15:30	13.365	20.805	14.7171	14.3331	0.859	NA	NA
13	ECO66G12	2013-04-26	16:00:00	2013-04-26 16:00	13.558	21.091	14.7109	14.3302	0.852	NA	NA
14	ECO66G12	2013-04-26	16:30:00	2013-04-26 16:30	13.654	21.569	14.702	14.3215	0.851	NA	NA

Tips for easier navigation of the .csv files



You can also **Freeze Panes – Freeze Top Row** to make viewing easier (this allows you to view the column headings while you scroll down).

You can also use the **Split** function (this allows you to split the screen so that you can see the column headings while scrolling down, and the left-hand columns (e.g., date-time field) while scrolling to the right).



QC_test1_Aw_20130725_20131015.csv - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER

Normal Page Break Preview Page Layout Views Custom Views

Workbook Views

Ruler Formula Bar Gridlines Headings

Show

Zoom 100% Zoom to Selection

New Window Arrange All

Freeze Panes

- Split
- Hide
- Unhide
- View Side by Side
- Synchronous Scrolling
- Reset Window Position

Switch Windows

Macros

Macros

P10

	K	L	M	N		R
1	Flag.Water.Temp.C	Flag.Air.Temp.C	Flag.Water.BP.psi	Flag.Air.BP.psi		Flag.Gross.Water.Temp.C
2	P	P	P	P		P
3	P	P	P	P		P
4	P	P	P	P		P
5	P	P	P	P	S	P
6	P	P	P	P	S	P
7	P	P	P	P	F	P
8	P	P	P	P	F	P

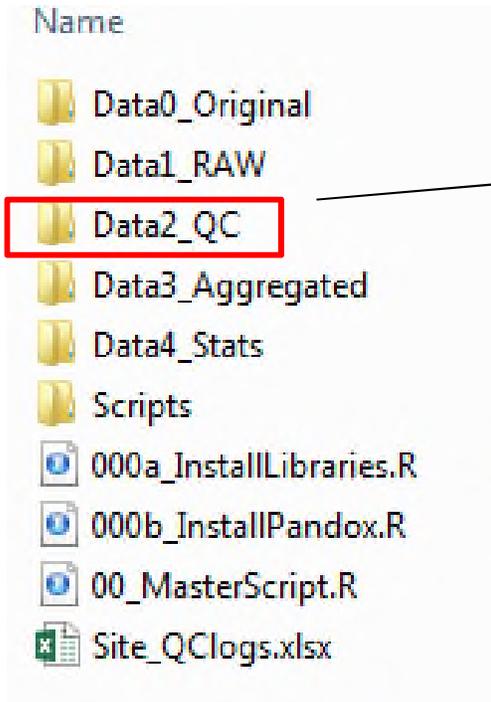


Your turn!

Open & scroll through a .csv file



Any questions? Comments?

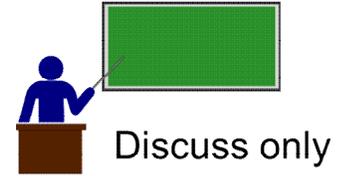


Name	Date modified	Type	Size
QC_ECO66G12_Aw_20130426_20130725	2016-04-21 19:27	Microsoft Excel C...	958 KB
QC_ECO66G12_Aw_20130426_20130725_Report_QC	2016-04-21 19:27	Microsoft Word D...	49 KB
QC_ECO66G12_Aw_20130725_20131015	2016-04-21 19:27	Microsoft Excel C...	892 KB
QC_ECO66G12_Aw_20130725_20131015_Report_QC	2016-04-21 19:27	Microsoft Word D...	49 KB

QC_test2_Aw_20130426_20130725!

	A	B	C	D	E	F	G	H	I	J	K	L
1	SiteID	Date	Time	Date.Time	Water.Temp.C	Air.Temp.C	Water.BP.psi	Air.BP.psi	Water.Level.ft	Water.LoggerID	Water.RowID	Flag.Date.Time
2	Test2	4/26/2013	10:30:00	4/26/2013 10:30	10.063	15.282	14.8024	14.3638	0.985	NA	1	
3	Test2	4/26/2013	11:00:00	4/26/2013 11:00	10.357	14.134	14.7461	14.3644	0.854	NA	2	
4	Test2	4/26/2013	11:30:00	4/26/2013 11:30	10.748	14.23	14.7428	14.3603	0.855	NA	3	
5	Test2	4/26/2013	12:00:00	4/26/2013 12:00	11.139	15.473	14.7441	14.3627	0.853	NA	4	
6	Test2	4/26/2013	12:30:00	4/26/2013 12:30	11.528	17.284	14.7431	14.3612	0.854	NA	5	
7	Test2	4/26/2013	13:00:00	4/26/2013 13:00	11.916	18.806	14.7421	14.3628	0.848	NA	6	
8	Test2	4/26/2013	13:30:00	4/26/2013 13:30	12.207	19.662	14.7453	14.3629	0.855	NA	7	
9	Test2	4/26/2013	14:00:00	4/26/2013 14:00	12.594	20.424	14.7351	14.3537	0.853	NA	8	
10	Test2	4/26/2013	14:30:00	4/26/2013 14:30	12.883	20.71	14.7315	14.3484	0.857	NA	9	

Corrections

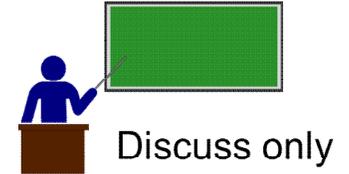


In general, 3 possible actions can be taken:

- Leave data as is
- Apply correction factor
- Remove data

Corrections should not be made unless the cause(s) of error(s) can be validated or explained in the field notes or by comparison with information from nearby stations. Accurate field notes and accuracy check logs are essential in the data correction process. Any discrepancies should be documented in your data file and any actions you take should also be carefully documented.

Corrections



Right now we **don't have any automated corrections programmed into the scripts** (this is a future 'wish list' item).

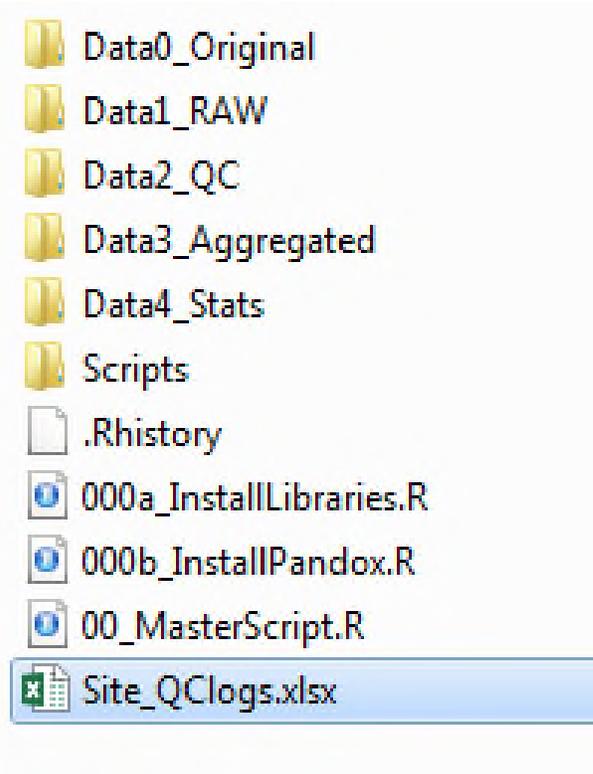
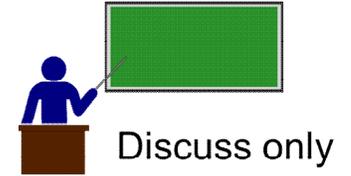
Any corrections you make need to be done **manually**.

Some types of corrections are fairly straightforward and can be done in Excel (e.g., change values by same amount, or basic drift corrections); others are more complex.

Potential references –

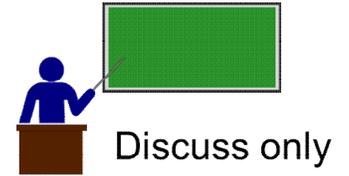
- Wagner et al. 2006 (USGS) (more detailed)
- 'Best Practices' report (more general - has a table with a general summary of different types of problems that can occur (e.g., missing data, failed accuracy check) and recommended actions for addressing them)

Site_QClogs Excel file



We created a QC log file – this is a place where you can log corrections, dates of deployment, sensor replacement, etc. (anything that can affect the quality of the data, or might explain unusual changes in the data)

Site_QClogs Excel file



Site_QClogs - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW Nuance PDF

Clipboard Font Alignment Number Styles

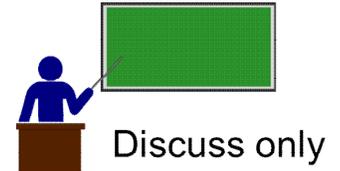
	A	B	C	D	E
	SiteID	Stream name	Person making entry	Date	Action
1	ECO66G12	Sheeds Creek	JS	2013-04-26	deployed Hobo U20 non-vented pressure transducers (one on land, one in the water), water S/N 10229557, long -72.38454, lat 42.46471; air S/N 10229557, long -72.38450, lat 42.46470
2	ECO66G12	Sheeds Creek	JS	2013-07-25	site visit, downloaded data; maintenance check - everything looked good!
3	ECO66G12	Sheeds Creek	JS	2013-10-15	site visit, downloaded data; maintenance check - cleaned the sensor
4	ECO66G12	Sheeds Creek	JS	2013-11-15	ran QC checks on files from 4/26-10/15/2013; water temp - deleted 2 extreme outliers; aggregated
5					

Example entries...this is still a work in progress...we welcome your comments and suggestions!

Running the Aggregate script

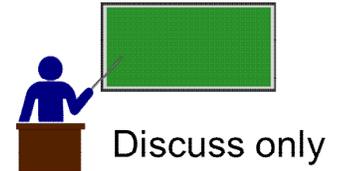


Aggregate process - overview



1. **Update entries in the MasterScript** (in this example, you only need to change 1 entry - Operation needs to be changed from [2] to [3])
2. **Run the script**, which searches the Data2_QC folder for the appropriate files and, in this example, combines the 2 QC'd files into a single file that spans the full date range (April 26 through October 15)
3. Open the **Data3_Aggregated folder**. There should be 2 outputs for each input file:
 - **.csv** file
 - **QC report** (Word document)

1. Update entries in the MasterScript



In this example, you only need to change 1 entry -

Operation needs to be changed from [2] to [3]

Selection.Operation

[1] = GetGageData

[2] = QCRaw

[3] = Aggregate

[4] = SummaryStats

1. Update entries in the MasterScript



```
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
00_MasterScript.R*
Source on Save
12 myDir.BASE <- getwd()
13 setwd(myDir.BASE)
14 # library (load any required helper functions)
15 source(paste(myDir.BASE,"scripts","fun.Master.R",sep="/"))
16 #####
17 # USER input in this section (see end of script for explanatio
18 #####
19 #
20 # PROMPT; Operation
21 Selection.Operation <- c("GetGageData","OCRAW", "Aggregate", "
22 myData.Operation <- Selection.Operation[3] #number corresp
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2" "HRCC", "PBCC", "ECO66G12",
27 myData.SiteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG",
32 myData.Type <- Selection.Type[3] #number corresponds to intended operatio
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 #####
```

The 5 entries should read –

Operation [3]

SiteID **"ECO66G12"**

Data type [3]

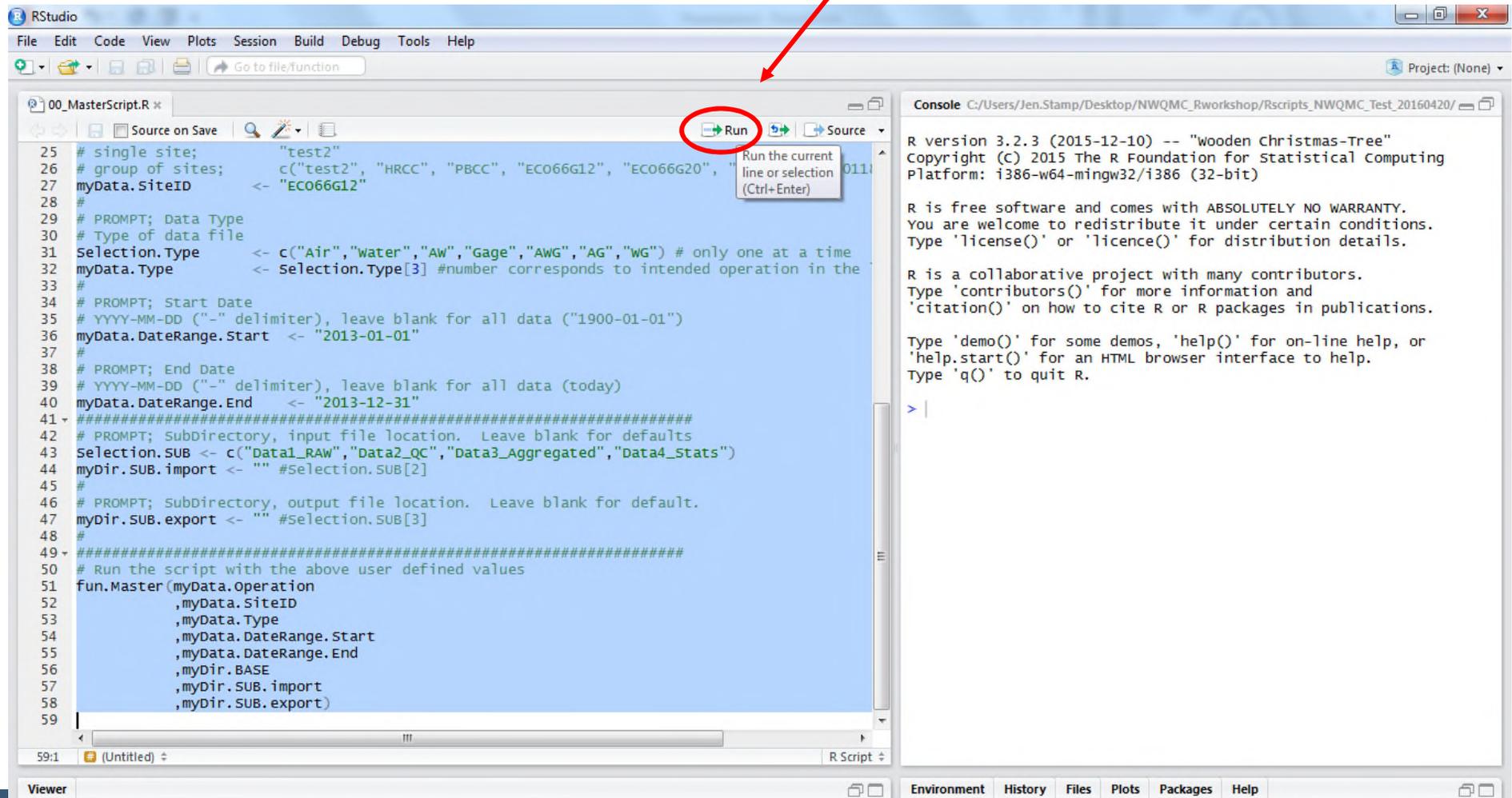
Start date **"2013-01-01"**

End date **"2013-12-31"**

2. Run the script



Using your cursor, highlight the script
and click RUN



The screenshot shows the RStudio interface with the following elements:

- Source Editor:** Contains an R script named `00_MasterScript.R`. The script is highlighted in blue. The `Run` button (a green play icon) is circled in red. A tooltip for this button reads: "Run the current line or selection (Ctrl+Enter)".
- Console:** Displays the R version `3.2.3 (2015-12-10)` and copyright information: "Copyright (C) 2015 The R Foundation for Statistical Computing Platform: i386-w64-mingw32/i386 (32-bit)". It also includes the standard R disclaimer and help instructions.
- Code:** The script defines variables for site information, data type, dates, and directory paths, and then calls a function `fun.Master` with these variables.

If the script ran successfully, you should see a running log like this...

```
Console C:/Users/Jen.Stamp/Desktop/NWQMC_Rworkshop/Rscripts_NWQMC_Test_20160420/
|.....| 89%
Label: unnamed-chunk-4 (with options)
List of 3
$ results: chr "asis"
$ echo : logi FALSE
$ warning: logi FALSE

|.....| 100%
ordinary text without R code

output file: Report_Aggregate.knit.md

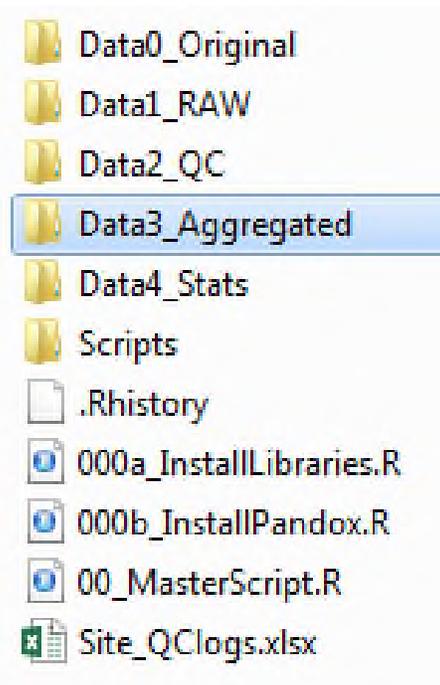
"C:/Users/JEN-1.STA/AppData/Local/Pandoc/pandoc" +
ggregate.utf8.md --to docx --from markdown+autolink
fiers+tex_math_single_backslash-implicit_figures -
amp/Desktop/NWQMC_Rworkshop/Rscripts_NWQMC_Test_20
DATA_ECO66G12_Aw_20130101_20131231_Report_Aggregate.docx" --highlight-styl
e tango

Output created: Data3_Aggregated/DATA_ECO66G12_Aw_20130101_20131231_Report
_Aggregate.docx
[1] "Task COMPLETE. QC Report. Total time = 2.142006 secs."
[1] "User defined parameters: SiteID (ECO66G12), Data Type (Aw), Date Rang
e (2013-01-01 to 2013-12-31)."
```

**Output created...
Task COMPLETE**

```
[1] "Processing item 2 of 2, COMPLETE, QC_ECO66G12_Aw_20130725_20131015.cs
v."
[1] "Processing of 2 of 2 files complete."
[1] "Processing of items (n=2) finished. Total time = 5.025429 secs."
[1] "Items COMPLETE = 2."
[1] "Items SKIPPED = 0."
[1] "User defined parameters: SiteID (ECO66G12), Data Type (Aw), Date Rang
e (2013-01-01 to 2013-12-31)."
```

3. Open the Data3_Aggregated folder



There should be 2 files –

- **.csv** file (=merged, QC'd files for site and desired time period; all fields in the QC .csv file are carried over)
- **Aggregate report** (Word document) – contains time series plots

 DATA_ECO66G12_Aw_20130426_20141013.csv DATA_ECO66G12_Aw_20130426_20141013_Report_Aggregate.docx



Open the .csv file

- Does the file cover the full date range?
- Does everything else look correct?

Open the Aggregate report (Word document)

- Do the time series plots look correct?

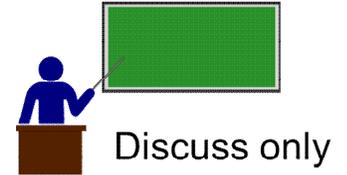
 [DATA_ECO66G12_Aw_20130426_20141013.csv](#)

 [DATA_ECO66G12_Aw_20130426_20141013_Report_Aggregate.docx](#)

Running the SummaryStats script



SummaryStats script



Setting expectations...

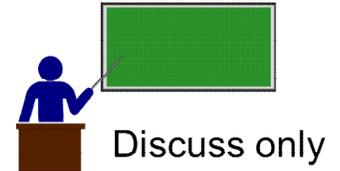
Right now the summary script is very basic (this is not going to 'wow' you).

The main intent is to get all the data (which come from different sources) QC'd and into the same format.

Once that happens, there are many possibilities for better visualization tools, more summary statistics, etc.

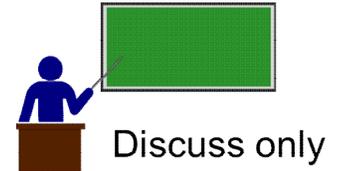


SummaryStats process - overview



- 1. Update entries in the MasterScript** (in this example, you only need to change 1 entry - Operation needs to be changed from [3] to [4])
- 2. Run the script**, which searches the Data3_Aggregated folder for the appropriate files
- 3. Open the Data4_Stats folder.** There should be 3 outputs *for each parameter*.
 - **DV** .csv file (daily average)
 - **STATS** .csv file (summary statistics)
 - **PDF** files with various time series plots

1. Update entries in the MasterScript



In this example, you only need to change 1 entry -

Operation needs to be changed from [3] to [4]

Selection.Operation
[1] = GetGageData
[2] = QCRaw
[3] = Aggregate
[4] = SummaryStats

1. Update entries in the MasterScript



```
16 ▾ #####
17 # USER input in this section (see end of script for e
18 ▾ #####
19 #
20 # PROMPT; Operation
21 Selection.Operation <- c("GetGageData","QCRaw", "Aggr
22 myData.Operation <- Selection.Operation[4] #number
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "EC
27 myData.SiteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","A
32 myData.Type <- Selection.Type[3] #number corresponds to intended operatio
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter). leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter). leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 ▾ #####
```

The 5 entries should read –

Operation [4]

SiteID **"ECO66G12"**

Data type **[3]**

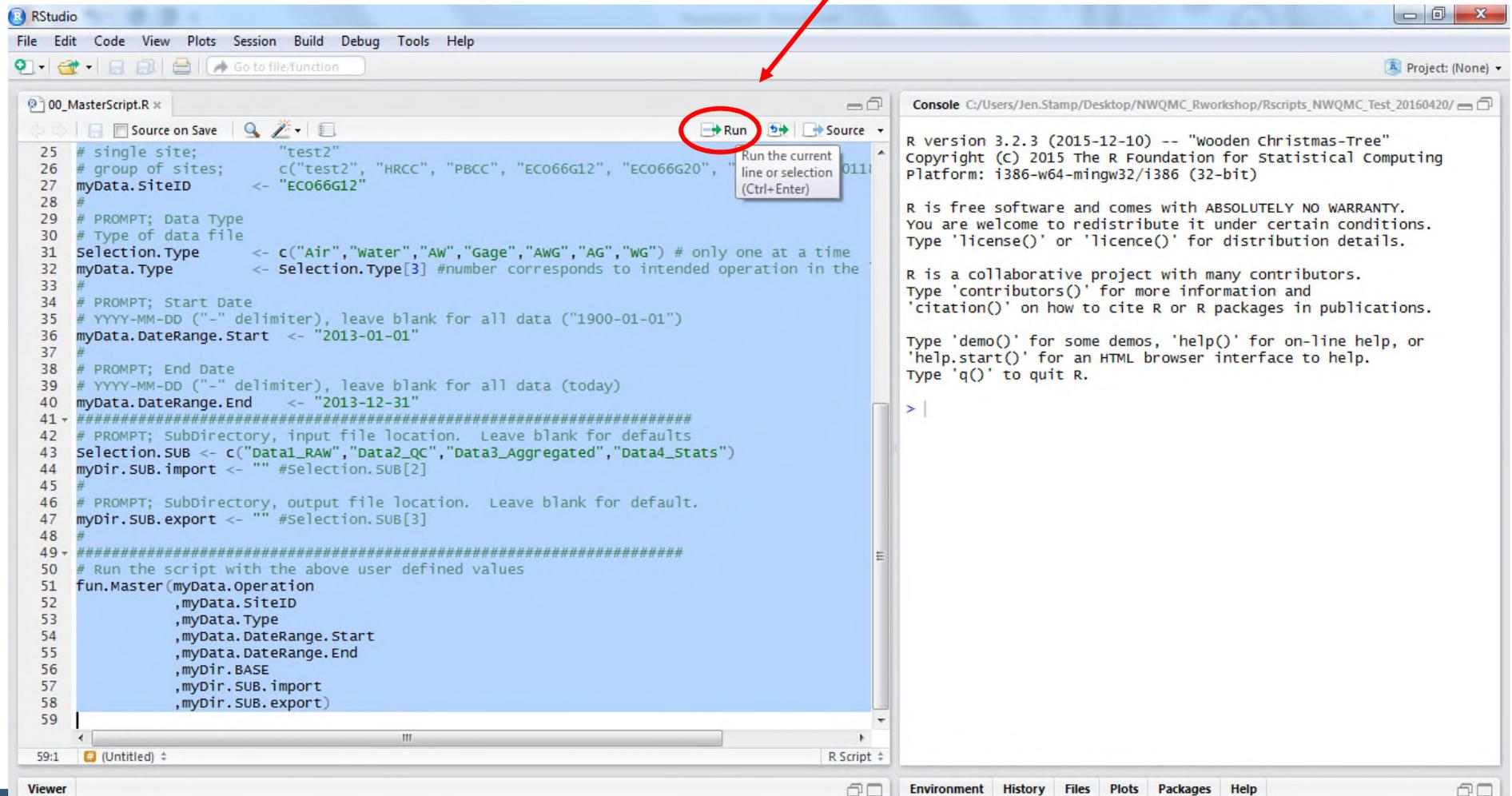
Start date **"2013-01-01"**

End date **"2013-12-31"**

2. Run the script



Using your cursor, highlight the script and click RUN



The screenshot shows the RStudio interface with the following elements:

- Source Editor:** Contains an R script named "00_MasterScript.R" with the following code:

```
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "011
27 myData.SiteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG","WG") # only one at a time
32 myData.Type <- Selection.Type[3] #number corresponds to intended operation in the
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 #####
42 # PROMPT; Subdirectory, input file location. Leave blank for defaults
43 Selection.SUB <- c("Data1_RAW","Data2_QC","Data3_Aggregated","Data4_Stats")
44 myDir.SUB.import <- "" #Selection.SUB[2]
45 #
46 # PROMPT; Subdirectory, output file location. Leave blank for default.
47 myDir.SUB.export <- "" #Selection.SUB[3]
48 #
49 #####
50 # Run the script with the above user defined values
51 fun.Master(myData.Operation
52 ,myData.SiteID
53 ,myData.Type
54 ,myData.DateRange.Start
55 ,myData.DateRange.End
56 ,myDir.BASE
57 ,myDir.SUB.import
58 ,myDir.SUB.export)
59
```
- Run Button:** A red arrow points to the "Run" button (a green play icon) in the top toolbar, which is circled in red. A tooltip is visible over it: "Run the current line or selection (Ctrl+Enter)".
- Console:** Shows the R version and license information:

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```
- Viewer:** Shows "59:1 (Untitled)" and "R Script".
- Bottom Panel:** Includes tabs for Environment, History, Files, Plots, Packages, and Help.

If the script ran successfully, you should see a running log like this...

```
Console C:/Users/Jen.Stamp/Desktop/NWQMC_Rworkshop/Rscripts_NWQMC_Test_20160420/
> myData.SiteID      <- "ECO66G12"
> #
> # PROMPT; Data Type
> # Type of data file
> Selection.Type     <- c("Air","water","Aw","Gage","AWG","AG","WG") # on
ly one at a time
> myData.Type        <- selection.Type[3] #number corresponds to intended
operation in the line above
> #
> # PROMPT; Start Date
> # YYYY-MM-DD ("- " delimiter), leave blank for all data ("1900-01-01")
> myData.DateRange.start <- "2013-01-01"
> #
> # PROMPT; End Date
> # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
> myData.DateRange.End   <- "2013-12-31"
> #####
> # PROMPT; subDirectory, input file location. Leave blank for defaults
> Selection.SUB <- c("Data1_RAW","Data2_QC","Data3_Aggregated"
s")
> myDir.SUB.import <- "" #selection.SUB[2]
> #
> # PROMPT; subDirectory, output file location. Leave blank
> myDir.SUB.export <- "" #selection.SUB[3]
> #
> #####
> # Run the script with the above user defined values
> fun.Master(myData.Operation
+           ,myData.SiteID
+           ,myData.Type
+           ,myData.DateRange.start
+           ,myData.DateRange.End
+           ,myDir.BASE
+           ,myDir.SUB.import
+           ,myDir.SUB.export)
[1] "Total items to process = 3:"
[1] "Water.Temp.C"   "Air.Temp.C"   "Water.Level.ft"
> |
```

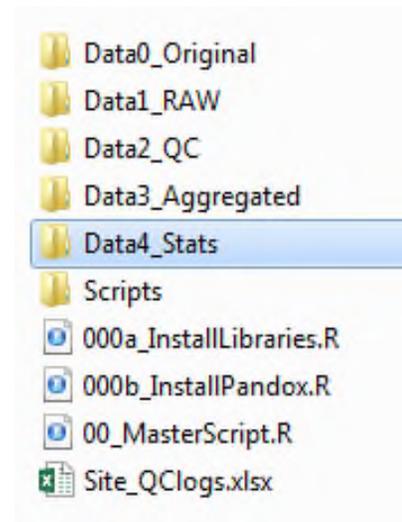
Total items to process = 3
Water.Temp.C
Air.Temp.C
Water.Level.ft

3. Open the Data4_Stats folder



There should be 3 files (each) for air temperature, water temperature and water level*

- **DV** .csv file (daily average)
- **STATS** .csv file (summary statistics for different time periods)
- **PDF** files with various time series plots

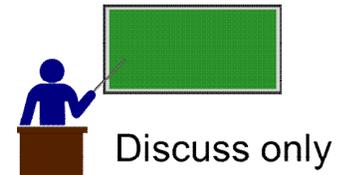


Name	Date modified	Type
DV_ECO66G12_Aw_20130101_20131231_Air.Temp.C.csv	4/19/2016 9:55 PM	Microsoft Excel
DV_ECO66G12_Aw_20130101_20131231_Water.Level.ft.csv	4/19/2016 9:55 PM	Microsoft Excel
DV_ECO66G12_Aw_20130101_20131231_Water.Temp.C.csv	4/19/2016 9:55 PM	Microsoft Excel
STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C.csv	4/19/2016 9:55 PM	Microsoft Excel
STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C.pdf	4/19/2016 9:55 PM	Adobe Acrobat
STATS_ECO66G12_Aw_20130101_20131231_Water.Level.ft.csv	4/19/2016 9:55 PM	Microsoft Excel
STATS_ECO66G12_Aw_20130101_20131231_Water.Level.ft.pdf	4/19/2016 9:55 PM	Adobe Acrobat
STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C.csv	4/19/2016 9:55 PM	Microsoft Excel
STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C.pdf	4/19/2016 9:55 PM	Adobe Acrobat

*we're working on adding in discharge and potentially other parameters

SummaryStats - DV (daily) output

Example - air temperature



	A	B	C	D	E	F	G	H	I	J	K	L
1	SiteID	Parameter	mean	Date	Year	YearMonth	MonthDay	Month	JulianDay	Season	YearSeason	
2	ECO66G12	Air.Temp.C	17.5507037	4/26/2013	2013	201304	426	4	115	Spring	2013Spring	
3	ECO66G12	Air.Temp.C	12.03854167	4/27/2013	2013	201304	427	4	116	Spring	2013Spring	
4	ECO66G12	Air.Temp.C	13.18835417	4/28/2013	2013	201304	428	4	117	Spring	2013Spring	
5	ECO66G12	Air.Temp.C	14.0801875	4/29/2013	2013	201304	429	4	118	Spring	2013Spring	
6	ECO66G12	Air.Temp.C	14.87283333	4/30/2013	2013	201304	430	4	119	Spring	2013Spring	
7	ECO66G12	Air.Temp.C	17.14391667	5/1/2013	2013	201305	501	5	120	Spring	2013Spring	
8	ECO66G12	Air.Temp.C	17.33864583	5/2/2013	2013	201305	502	5	121	Spring	2013Spring	
9	ECO66G12	Air.Temp.C	16.09804167	5/3/2013	2013	201305	503	5	122	Spring	2013Spring	
10	ECO66G12	Air.Temp.C	12.669375	5/4/2013	2013	201305	504	5	123	Spring	2013Spring	
11	ECO66G12	Air.Temp.C	10.71225	5/5/2013	2013	201305	505	5	124	Spring	2013Spring	
12	ECO66G12	Air.Temp.C	9.452270833	5/6/2013	2013	201305	506	5	125	Spring	2013Spring	
13	ECO66G12	Air.Temp.C	12.2388125	5/7/2013	2013	201305	507	5	126	Spring	2013Spring	
14	ECO66G12	Air.Temp.C	14.0598125	5/8/2013	2013	201305	508	5	127	Spring	2013Spring	
15	ECO66G12	Air.Temp.C	16.40225	5/9/2013	2013	201305	509	5	128	Spring	2013Spring	
16	ECO66G12	Air.Temp.C	16.02366667	5/10/2013	2013	201305	510	5	129	Spring	2013Spring	
17	ECO66G12	Air.Temp.C	16.1740417	5/11/2013	2013	201305	511	5	130	Spring	2013Spring	

Check a DV file

Example - air temperature

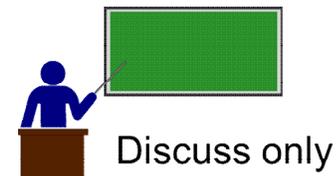


Name	Date modified	Type	Size
 DV_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Microsoft Excel C...	15 KB
 DV_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Microsoft Excel C...	16 KB
 DV_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Microsoft Excel C...	16 KB
 STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Microsoft Excel C...	83 KB
 STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Adobe Acrobat D...	15 KB
 STATS_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Microsoft Excel C...	85 KB
 STATS_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Adobe Acrobat D...	15 KB
 STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Microsoft Excel C...	82 KB
 STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Adobe Acrobat D...	15 KB

Any comments? Questions?

SummaryStats - STATs output

Example - air temperature



Statistics (for various timeframes) include –

- Mean
- Median
- Min
- Max
- Range
- St dev
- Variance
- Count (n)
- Percentiles (01, 05, 10, 25, 50, 75, 90, 95, 99)

STATS_ECO06GL2_Air_Temp_Cov_20130426_20141013_Air_Temp_C.cov - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	SiteID	Parameter	TimeFrame	TimeValue	mean	median	min	max	range	sd	var	cv	n	q01	q05	q10	q25	q50	q75	q90	q95	q99
1	ECO66G12	Air.Temp.C	year	2013	15.29215931	17.92777083	-2.385479167	23.6849375	26.07041667	7.001553732	49.02175467	0.457852524	250	-0.833736042	0.284664583	3.587795833	11.92061979	17.92777083	20.68778646	22.048775	22.50769479	23.23991146
2	ECO66G12	Air.Temp.C	year	2014	13.94472461	17.48488542	-11.2189375	23.74520833	34.96414583	8.251038601	68.079638	0.591696059	286	-8.541278125	-1.671328125	0.84065625	8.275692708	17.48488542	20.53886458	21.74290625	22.28585417	23.11208958
3	ECO66G12	Air.Temp.C	season	Fall	14.58649684	16.2456875	-2.385479167	22.9605	25.34597917	6.207228908	38.52969071	0.425546241	134	-0.686075625	0.450930208	4.510489583	11.07234375	16.2456875	18.8059375	21.35731458	21.88353646	22.55307313
4	ECO66G12	Air.Temp.C	season	Spring	14.01217389	14.28185417	0.9399375	21.6948125	20.754875	4.759621086	22.65399289	0.339677564	128	2.003454375	5.804933333	7.595283333	10.78840625	14.28185417	17.98004688	19.68673958	20.29194479	20.89141021
5	ECO66G12	Air.Temp.C	season	Summer	20.90715863	20.885625	16.91983333	23.74520833	6.825375	1.343190951	1.804161125	0.06424549	184	17.54123093	18.59349792	19.10801458	20.00013021	20.885625	21.83521354	22.52059792	23.07896771	23.39032208
6	ECO66G12	Air.Temp.C	season	Winter	2.411824255	1.997791667	-11.2189375	16.04572917	27.26466667	5.604208695	31.4071551	2.323638914	89	-9.034630833	-6.526554167	-4.9513125	-0.426145833	1.997791667	5.505	9.149904167	13.82639583	15.19145083
7	ECO66G12	Air.Temp.C	season	NA	1.5126875	1.5126875	-1.5126875	1.5126875	0	NA	NA	NA	1	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875
8	ECO66G12	Air.Temp.C	year_season	2013Fall	12.76169368	14.45866667	-2.385479167	22.11170833	24.4971875	6.4446708	41.53378172	0.505001214	91	-0.937622917	0.139885417	3.613854167	8.130135417	14.45866667	17.86986458	20.0435	20.82810417	21.68055208
9	ECO66G12	Air.Temp.C	year_season	2013NA	1.5126875	1.5126875	-1.5126875	1.5126875	0	NA	NA	NA	1	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875	1.5126875
10	ECO66G12	Air.Temp.C	year_season	2013Spring	16.1027635	16.28842708	9.339666667	21.6948125	12.35514583	3.326153604	11.00329753	0.206557909	36	9.379078125	10.39725521	12.05963542	13.68616667	16.28842708	18.07420833	20.48491667	20.8448021	21.42278229
11	ECO66G12	Air.Temp.C	year_season	2013Summer	21.04018524	20.88833333	18.06297917	23.6849375	5.621958333	1.266412702	1.603801132	0.060190188	92	18.58258917	19.00181354	19.46037708	20.03010938	20.88833333	22.08027083	22.72566458	23.10502396	23.36192542
12	ECO66G12	Air.Temp.C	year_season	2013Winter	4.827213889	3.571979167	-1.903479167	16.04572917	17.94920833	5.297611175	28.06468416	1.097446953	30	-1.605341042	-0.837138542	-0.552810417	0.493989583	3.571979167	7.54976525	14.08768958	15.07400208	15.76420563
13	ECO66G12	Air.Temp.C	year_season	2014Fall	18.44828958	18.37429167	7.732604167	22.9605	15.22789583	3.244441973	10.52640372	0.175866817	43	8.817892917	12.4576042	15.57963333	17.14003125	18.37429167	21.38375	22.07462917	22.40546667	22.81552125
14	ECO66G12	Air.Temp.C	year_season	2014Spring	13.1941163	13.07396875	0.9399375	20.69829167	19.75835417	4.995528736	24.95530735	0.37861791	92	1.361551875	5.192844792	6.8350125	9.370885417	13.07396875	17.88082813	19.46016875	19.81861042	20.41969896
15	ECO66G12	Air.Temp.C	year_season	2014Summer	20.77413202	20.885625	16.91983333	23.74520833	6.825375	1.410165232	1.988565982	0.067880826	92	17.34165625	18.46439583	18.72682292	19.84068229	20.885625	21.72229688	22.34431458	22.70114688	23.21388208
16	ECO66G12	Air.Temp.C	year_season	2014Winter	1.183660035	1.074208333	-11.2189375	14.142625	25.3615625	5.391461547	29.06785761	4.554907142	59	-9.779280833	-8.53308125	-6.302875	-1.059239583	1.074208333	4.576429366	7.8396625	9.432833333	13.82711708
17	ECO66G12	Air.Temp.C	month	1	-1.588124181	-0.693541667	-11.2189375	8.733770833	19.95270833	4.833797143	23.36559482	-3.043714844	31	-10.4742875	-8.708697917	-8.5166875	-5.90739583	-0.693541667	1.987302083	4.099	4.576429366	7.497140993
18	ECO66G12	Air.Temp.C	month	2	4.252421131	3.82209375	-1.683416667	14.142625	15.82604167	4.230039079	17.89323061	0.994736633	28	-1.517541042	-0.780254167	-0.231835417	0.279635417	3.82209375	6.373005208	9.541791667	12.31222083	13.99575083
19	ECO66G12	Air.Temp.C	month	3	8.197885735	8.454583333	0.9399375	14.44208333	13.50214583	3.198105625	10.22787959	0.390113466	31	1.07893125	2.514739583	3.800875	6.442489583	8.454583333	10.29227083	11.8958125	12.64263542	14.04524583



Check a STATS file



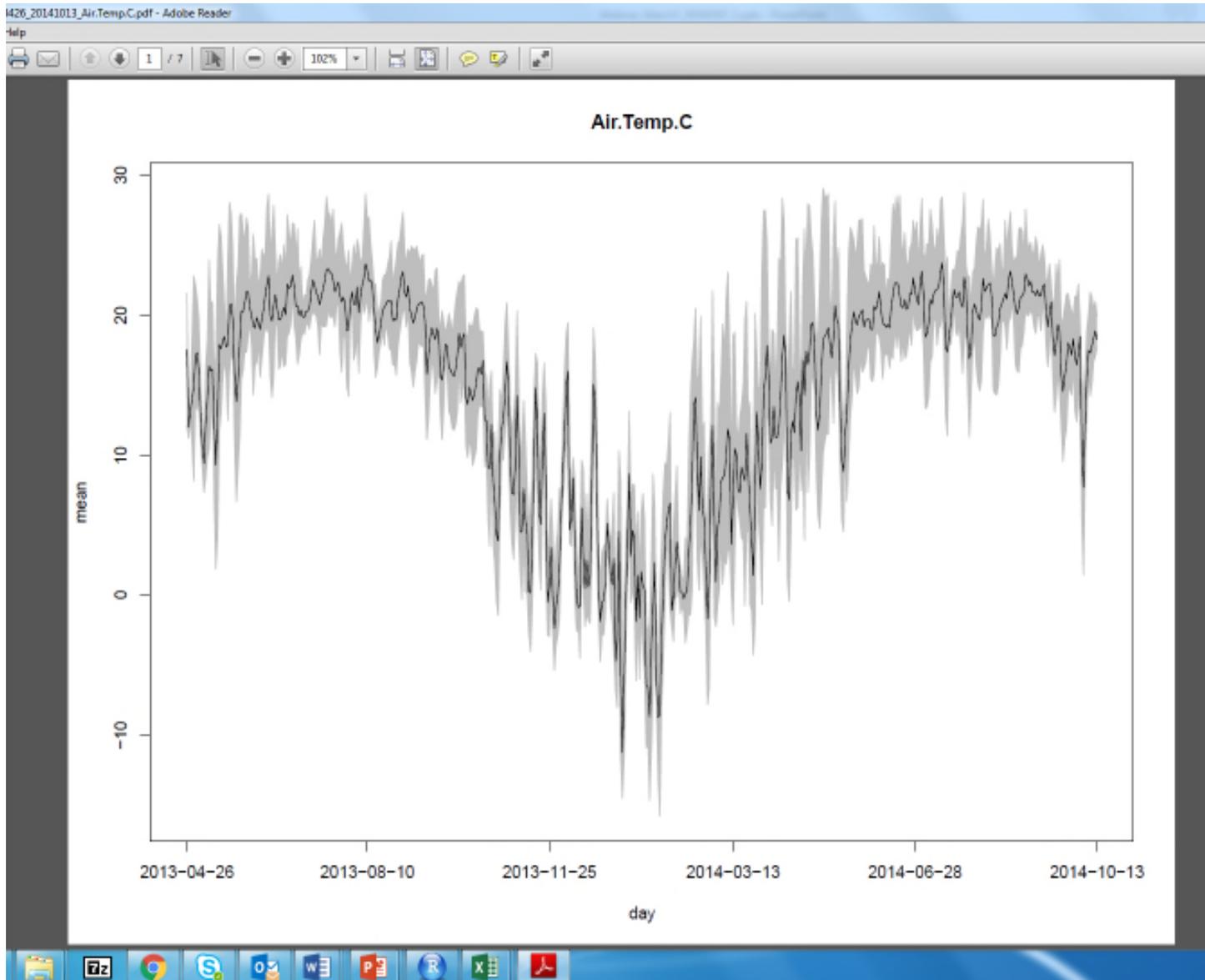
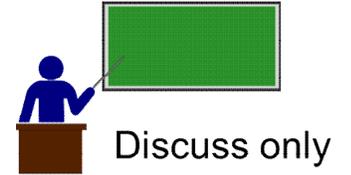
Example - air temperature

Name	Date modified	Type	Size
 DV_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Microsoft Excel C...	15 KB
 DV_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Microsoft Excel C...	16 KB
 DV_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Microsoft Excel C...	16 KB
 STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Microsoft Excel C...	83 KB
 STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Adobe Acrobat D...	15 KB
 STATS_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Microsoft Excel C...	85 KB
 STATS_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Adobe Acrobat D...	15 KB
 STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Microsoft Excel C...	82 KB
 STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Adobe Acrobat D...	15 KB

Any comments? Questions?

SummaryStats - PDF output

Example - air temperature, mean daily





Check a PDF file



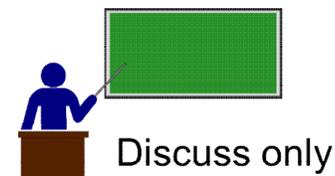
Example - air temperature

Name	Date modified	Type	Size
DV_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Microsoft Excel C...	15 KB
DV_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Microsoft Excel C...	16 KB
DV_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Microsoft Excel C...	16 KB
STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Microsoft Excel C...	83 KB
STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Adobe Acrobat D...	15 KB
STATS_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Microsoft Excel C...	85 KB
STATS_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Adobe Acrobat D...	15 KB
STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Microsoft Excel C...	82 KB
STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Adobe Acrobat D...	15 KB

Any comments? Questions?



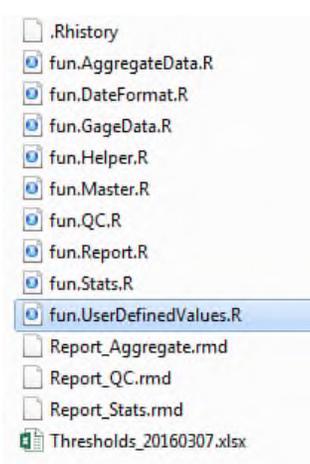
What if data are flagged as fail (F)? Are they included in the SummaryStat calculations?



The scripts are written in a way that gives you flexibility on this. In the last line (line 233) of the **fun.UserDefinedValues.R** file in the **Scripts** folder, if you enter –

- TRUE – the script will exclude values flagged as ‘F’
- FALSE – the script will include values flagged as ‘F’

Scripts
folder



```
221 myTimeFrame.Season.Winter.Start <- "1201"  
222 #myTimeFrame.Season.Winter.End <- "0228" #but 0229 in leap year, use start dat  
223 # Time Frame Names  
224 myName.Yr <- "Year"  
225 myName.YrMo <- "YearMonth"  
226 myName.Mo <- "Month"  
227 myName.MoDa <- "MonthDay"  
228 myName.JuDa <- "JulianDay"  
229 myName.Season <- "Season"  
230 myName.YrSeason <- "YearSeason"  
231 - #####  
232 # Trigger for Stats to exclude (TRUE) or include (FALSE) where flag = "fail"  
233 myStats.Fails.Exclude <- TRUE #FALSE #TRUE  
234 - #####  
235
```

Right now the script you are working with is set to ‘TRUE’

Congratulations!

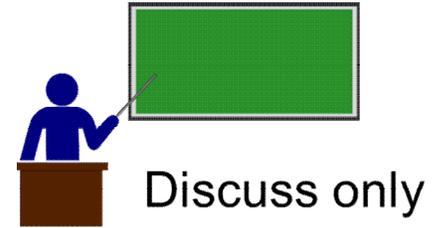


You have successfully run
through test dataset #1!



Test Dataset # 2

Test Dataset #2



Standalone temperature sensor data

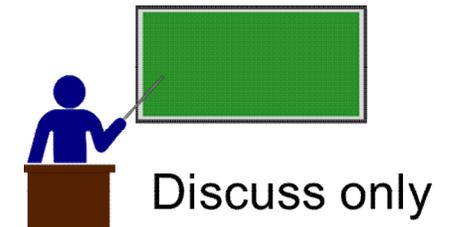
- Water temperature
- Air temperature

2 Files –

- Same site (WSR01)
- Different parameters
- Same date range
 - April 24, 2014 – July 10, 2014



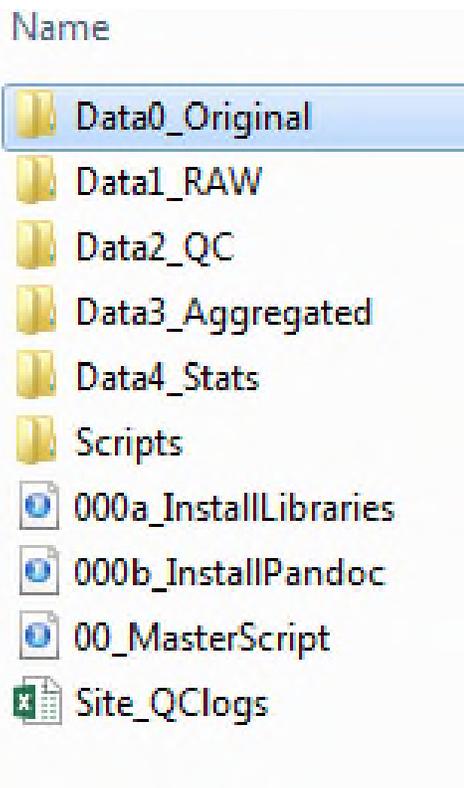
Work flow for Test Dataset #2



- 1. Prepare your data** (review only*)
 - Move it into the correct folders
 - Organize, format and name the files
- 2. Run the QC portion** of the MasterScript (on both files) – we'll run air temperature first, then water temperature
 - Check the data, correct if needed
- 3. Run the Aggregate** portion of the MasterScript to get both parameters into the same file
 - we'll run air temperature first, then water temperature, then will aggregate them as an AW (air-water) file
- 4. Run the SumStats** portion of the MasterScript to generate the summary statistics

*We prepared the data files in advance in order to maximize the amount of time we can spend running the actual scripts

Open the Data0_Original folder



Name	Date modified	Type
ECO66G12_WATER_4-26-13 TO 7-25-13.xl...	2/3/2015 3:01 PM	Microsoft Excel W...
ECO66G12_WATER_7-25-13 TO 10-15-13....	2/3/2015 3:01 PM	Microsoft Excel W...
WSR01__WaterTemp_20140710.xlsx	4/15/2016 6:56 AM	Microsoft Excel W...
WSR01_AirTemp_20140710.xlsx	4/15/2016 6:57 AM	Microsoft Excel W...

To save time, we've already put the WSR01 files into the Data0_Original folder.

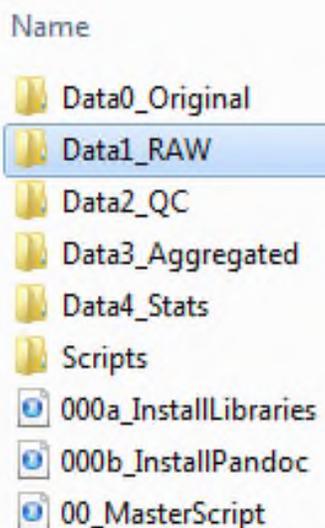
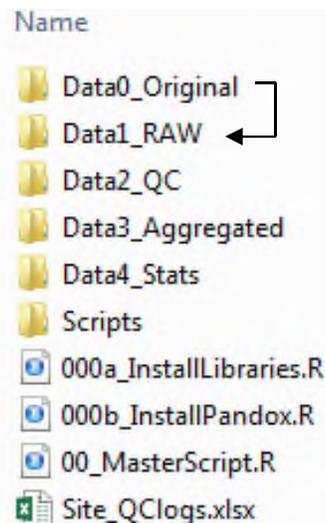
If you were working with your own files, you would put the originals into this folder (and would not alter them).

Open the Data1_RAW folder



To save time, we've already copied and pasted the WSR01 files from the Data0_Original into the Data1_RAW folder, and prepared them (reformatted, renamed, saved as .csv).

For a quick refresher, open the 2 files and review them.



Name	Date modified	Type	Size
01174565_Gage_20140424_20140710	2016-04-15 11:28	Microsoft Excel Comma Separated Values File	374 KB
ECO66G12_AW_20130426_20130725	2016-04-21 19:25	Microsoft Excel Comma Separated Values File	327 KB
ECO66G12_AW_20130725_20131015	2016-04-21 19:25	Microsoft Excel Comma Separated Values File	307 KB
WSR01_Air_20140424_20140710	2016-04-20 07:43	Microsoft Excel Comma Separated Values File	125 KB
WSR01_Gage_20140424_20140710	2016-04-20 07:43	Microsoft Excel Comma Separated Values File	310 KB
WSR01_Water_20140424_20140710	2016-04-20 07:43	Microsoft Excel Comma Separated Values File	158 KB

Note: If you were working with your own files, you would have to perform these steps manually.

QC



1. Update entries in the MasterScript for the **air temperature** file



```
16 # USER input in this section (see end of script for explanations)
17 # USER input in this section (see end of script for explanations)
18 # USER input in this section (see end of script for explanations)
19 #
20 # PROMPT; operation
21 Selection.operation <- c("GetGageData","QCRaw", "Aggregate", "SummaryStats")
22 myData.operation <- Selection.operation[2] #number corresponds to intended op
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "ECO66G12",
27 myData.SiteID <- "WSR01"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG")
32 myData.Type <- Selection.Type[1] #number corresponds
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter). leave blank for all data ("1900
36 myData.DateRange.Start <- "2014-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter). leave blank for all data (today
40 myData.DateRange.End <- "2014-12-31"
41 #
42 # PROMPT; SubDirectory, input file location. Leave blank for defaults
43 Selection.SUB <- c("Data1_RAW","Data2_QC","Data3_Aggregated","Data4_Stats")
44 myDir.SUB.import <- "" #Selection.SUB[2]
45 #
46 # PROMPT; SubDirectory, output file location. Leave blank for default.
47 myDir.SUB.export <- "" #Selection.SUB[3]
48
```

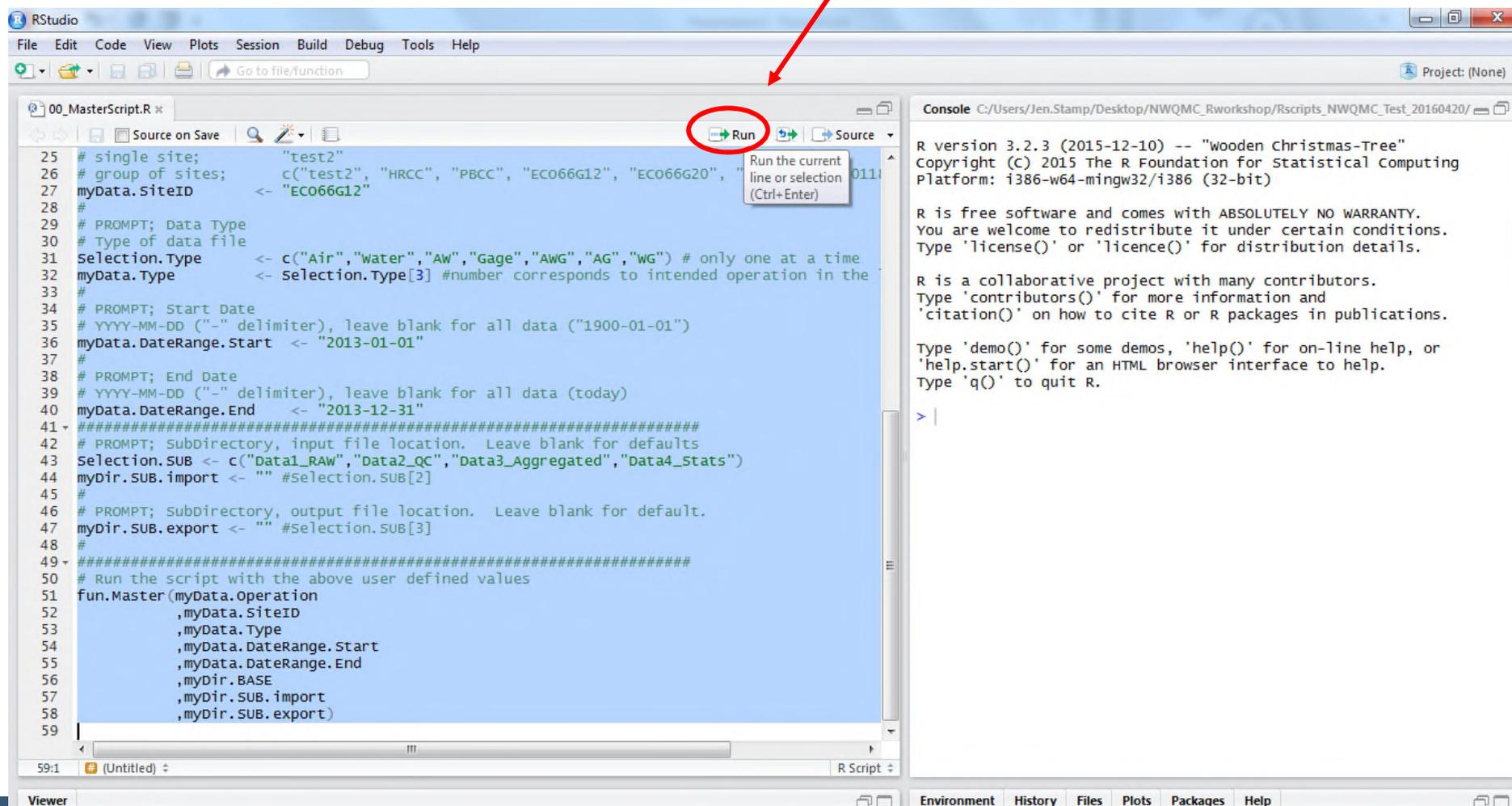
The 5 entries should read –

- Operation [2] = QCRaw
- SiteID “WSR01”
- Data type [1] = Air
- Start date “2014-01-01”
- End date “2014-12-31”

There are numerous possibilities for date entries...in this example, we used calendar year 2014. Feel free to experiment with other entries.

2. Run the script

Using your cursor, highlight the script and click RUN



The screenshot shows the RStudio interface with the following elements:

- Script Editor:** Contains R code for a function named `fun.Master`. The code includes prompts for site ID, data type, date range, and subdirectory. The `Run` button in the toolbar is circled in red, and a red arrow points to it from the text above. A tooltip for the `Run` button is visible, stating "Run the current line or selection (Ctrl+Enter)".
- Console:** Displays the R version (3.2.3) and copyright information (© 2015 The R Foundation for Statistical Computing). It also shows the R license text and instructions for using the help system.
- Environment/Plots/Files/Help:** The bottom of the window shows the Environment, History, Files, Plots, Packages, and Help tabs.

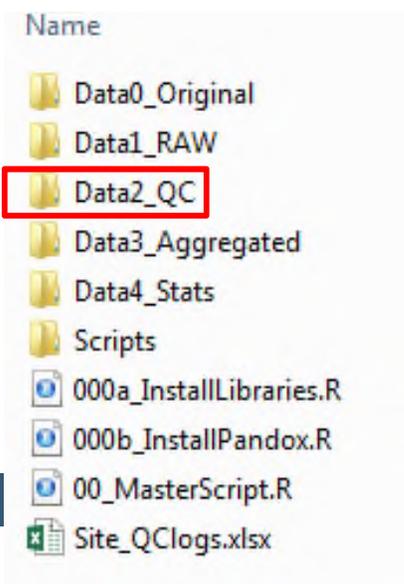
3. Verify that the script ran properly – **air temperature**



Check your Console (right) panel –
should end with **‘Task COMPLETE’**

Open the Data2_QC folder – verify that there are **2 outputs** (.csv & Word) for *air temperature*

Open & check the files (if time permits) – missing data, flagged values, time series plots



Name	Date modified	Type	Size
 QC_ECO66G12_Aw_20130426_20130725	2016-04-21 19:27	Microsoft Excel C...	958 KB
 QC_ECO66G12_Aw_20130426_20130725_Report_QC	2016-04-21 19:27	Microsoft Word D...	49 KB
 QC_ECO66G12_Aw_20130725_20131015	2016-04-21 19:27	Microsoft Excel C...	892 KB
 QC_ECO66G12_Aw_20130725_20131015_Report_QC	2016-04-21 19:27	Microsoft Word D...	49 KB
 QC_WSR01_Air_20140424_20140710	2016-04-21 19:38	Microsoft Excel C...	400 KB
 QC_WSR01_Air_20140424_20140710_Report_QC	2016-04-21 19:38	Microsoft Word D...	17 KB

4. Update entries in the MasterScript for the **water temperature** file



```

#####
# USER input in this section (see end of script for explanations)
#####
#
# PROMPT; Operation
Selection.Operation <- c("GetGageData","QCRaw", "Aggregate", "SummaryStats")
myData.Operation   <- Selection.Operation[2] #number corresponds to intended op
#
# PROMPT; Site ID
# single site;           "test2"
# group of sites;       c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "ECO68C
myData.SiteID       <- "WSR01"
#
# PROMPT; Data Type
# Type of data file
Selection.Type       <- c("Air","water","Aw","Gage","Av")
myData.Type         <- Selection.Type[2] #number corresponds to intended op
#
# PROMPT; Start Date
# YYYY-MM-DD ("- " delimiter) leave blank for all data
myData.DateRange.Start <- "2014-01-01"
#
# PROMPT; End Date
# YYYY-MM-DD ("- " delimiter) leave blank for all data
myData.DateRange.End   <- "2014-12-31"
#####
# PROMPT; subDirectory, input file location. Leave blank for all data
Selection.SUB <- c("Data1_RAW","Data2_QC","Data3_Aggregate")
myDir.SUB.import <- "" #Selection.SUB[2]
#
# PROMPT; subDirectory, output file location. Leave blank for all data
myDir.SUB.export <- "" #Selection.SUB[3]

```

The only entry that you need to change is Data type (from [1] to [2])

The 5 entries should read –

Operation [2] = QCRaw

SiteID “WSR01”

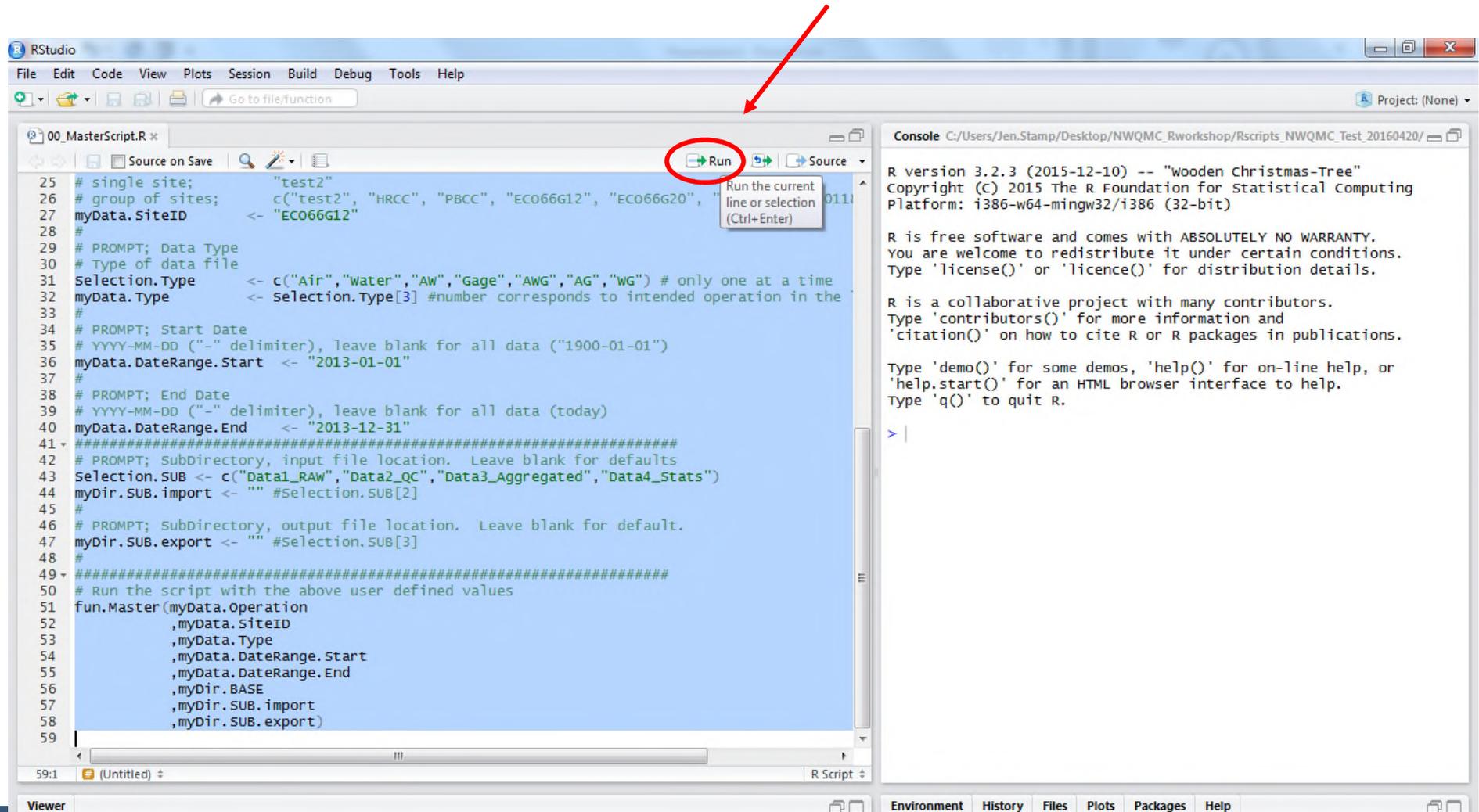
Data type [2] = Water

Start date “2014-01-01”

End date “2014-12-31”

5. Run the script

Using your cursor, highlight the script
and click RUN



The screenshot shows the RStudio interface with the following elements:

- Code Editor:** A script named `00_MasterScript.R` is open. The code is highlighted in blue. The script defines variables for site information, data type, dates, and directory paths, and then calls a function `fun.Master`.
- Run Button:** The `Run` button (a green play icon) is circled in red. A tooltip is visible over it, stating: "Run the current line or selection (Ctrl+Enter)". A red arrow points to this button from the text above.
- Console:** The console window on the right shows the R version (3.2.3) and the license text for R.

```
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "011
27 myData.SiteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG","WG") # only one at a time
32 myData.Type <- Selection.Type[3] #number corresponds to intended operation in the
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 #####
42 # PROMPT; Subdirectory, input file location. Leave blank for defaults
43 Selection.SUB <- c("Data1_RAW","Data2_QC","Data3_Aggregated","Data4_Stats")
44 myDir.SUB.import <- "" #Selection.SUB[2]
45 #
46 # PROMPT; Subdirectory, output file location. Leave blank for default.
47 myDir.SUB.export <- "" #Selection.SUB[3]
48 #
49 #####
50 # Run the script with the above user defined values
51 fun.Master(myData.Operation
52           ,myData.SiteID
53           ,myData.Type
54           ,myData.DateRange.Start
55           ,myData.DateRange.End
56           ,myDir.BASE
57           ,myDir.SUB.import
58           ,myDir.SUB.export)
59
```

Console C:/Users/Jen.Stamp/Desktop/NWQMC_Rworkshop/Rscripts_NWQMC_Test_20160420/

R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |

QC



6. Verify that the script ran properly – **water temperature**



Check your Console (right) panel –
should end with **‘Task COMPLETE’**

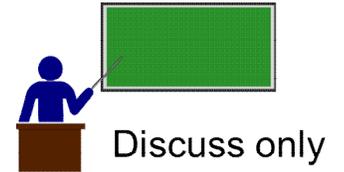
Open the Data2_QC folder – verify that there are 2 outputs (.csv & Word) for *water temperature*

Open & check the files (if time permits) – missing data, flagged values, time series plots

The screenshot shows a file explorer interface. On the left, a list of folders is displayed, with 'Data2_QC' highlighted by a red box. On the right, a table lists the contents of the 'Data2_QC' folder, including file names, modification dates, types, and sizes.

Name	Date modified	Type	Size
QC_ECO66G12_Aw_20130426_20130725	2016-04-21 19:27	Microsoft Excel C...	958 KB
QC_ECO66G12_Aw_20130426_20130725_Report_QC	2016-04-21 19:27	Microsoft Word D...	49 KB
QC_ECO66G12_Aw_20130725_20131015	2016-04-21 19:27	Microsoft Excel C...	892 KB
QC_ECO66G12_Aw_20130725_20131015_Report_QC	2016-04-21 19:27	Microsoft Word D...	49 KB
QC_WSR01_Air_20140424_20140710	2016-04-21 19:38	Microsoft Excel C...	400 KB
QC_WSR01_Air_20140424_20140710_Report_QC	2016-04-21 19:38	Microsoft Word D...	17 KB
QC_WSR01_Water_20140424_20140710_Report_QC	2016-04-21 19:38	Microsoft Word D...	17 KB
QC_WSR01_Water_20140424_20140710	2016-04-21 19:38	Microsoft Excel C...	401 KB

Aggregate – review steps



For each file (let's do air temperature first, then water temperature) -

1. **Update entries** in the MasterScript
2. **Run the script**
3. **Verify that the script ran properly**
4. **Open and check the files**

Then **change the Data type in the MasterScript to AW** (=air-water) & follow the same steps to generate an output that has both parameters (air + water temperature) in one file

Note: in this example, the individual air and water temperature files have to be run through the Aggregate script before the parameters can be combined into one 'AW' file. Ideally we could skip steps 1 & 2 and go right to the AW file, but the scripts are not currently written in a way that allows this (this is a future "wish list" item).

Aggregate

1. Update entries in the MasterScript for the **air temperature** file



```
16 ▾ #####
17 # USER input in this section (see end of script for explanations)
18 ▾ #####
19 #
20 # PROMPT; Operation
21 Selection.operation <- c("GetGageData","QCRaw", "A", "W", "G")
22 myData.operation <- Selection.operation[3] #number of operations
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "WCC")
27 myData.siteID <- "WSR01"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air", "Water", "AW", "Gage")
32 myData.Type <- Selection.Type[1] #number of data types
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
36 myData.DateRange.Start <- "2014-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2014-12-31"
41 ▾ #####
```

The 5 entries should read –

Operation [3] = Aggregate
SiteID “**WSR01**”

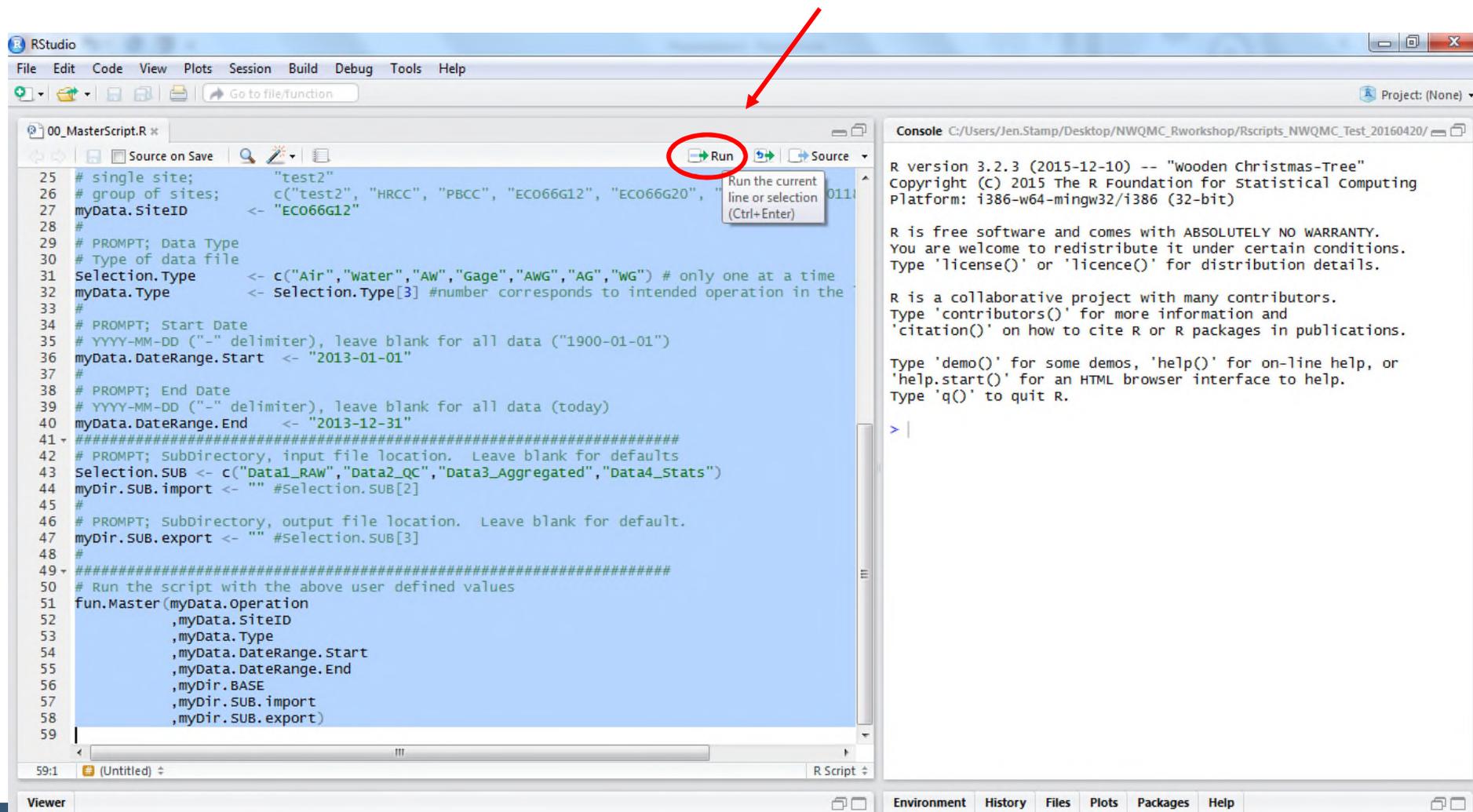
Data type [1] = Air

Start date “**2014-01-01**”

End date “**2014-12-31**”

2. Run the script

Using your cursor, highlight the script and click RUN



The screenshot shows the RStudio interface with the following elements:

- Code Editor:** Contains an R script named `00_MasterScript.R`. The script defines variables for site information, data type, date range, and directory paths, and then calls a function `fun.Master`.
- Run Button:** Located in the top toolbar, circled in red. A tooltip indicates it runs the current line or selection (Ctrl+Enter).
- Console:** Displays the R version (3.2.3) and license information.
- Viewer:** Shows the script content at line 59.

```
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "011
27 myData.SiteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG","WG") # only one at a time
32 myData.Type <- Selection.Type[3] #number corresponds to intended operation in the
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 #####
42 # PROMPT; Subdirectory, input file location. Leave blank for defaults
43 Selection.SUB <- c("Data1_RAW","Data2_QC","Data3_Aggregated","Data4_Stats")
44 myDir.SUB.import <- "" #Selection.SUB[2]
45 #
46 # PROMPT; Subdirectory, output file location. Leave blank for default.
47 myDir.SUB.export <- "" #Selection.SUB[3]
48 #
49 #####
50 # Run the script with the above user defined values
51 fun.Master(myData.Operation
52           ,myData.SiteID
53           ,myData.Type
54           ,myData.DateRange.Start
55           ,myData.DateRange.End
56           ,myDir.BASE
57           ,myDir.SUB.import
58           ,myDir.SUB.export)
59
```

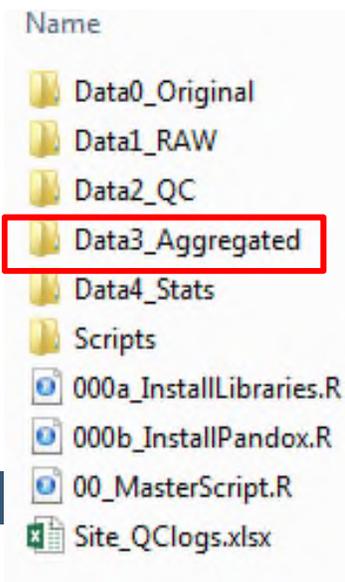


3. Verify that the script ran properly – **air temperature**

Check your Console (right) panel – log should include “Output created” and “Task COMPLETE”

Open the Data3_Aggregated folder – verify that there are 2 outputs for the WSR01 *air temperature* file

- .CSV
- Aggregate report (Word)



Name	Date modified	Type	Size
 DATA_ECO66G12_Aw_20130101_20131231	2016-04-21 19:28	Microsoft Excel C...	2,070 KB
 DATA_ECO66G12_Aw_20130101_20131231_Report_Aggregate	2016-04-21 19:28	Microsoft Word D...	54 KB
 DATA_WSR01_Air_20140101_20141231	2016-04-21 19:39	Microsoft Excel C...	437 KB
 DATA_WSR01_Air_20140101_20141231_Report_Aggregate	2016-04-21 19:39	Microsoft Word D...	19 KB

4. Update entries in the MasterScript for the **water temperature** file

```
#####  
# USER input in this section (see end of script for explanations)  
#####  
#  
# PROMPT; Operation  
Selection.Operation <- c("GetGageData","QCRaw", "Aggregate", "SummaryStats")  
myData.Operation <- Selection.Operation[3] #number corresponds to intended op  
#  
# PROMPT; Site ID  
# single site; "test2"  
# group of sites; c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "ECO68C  
myData.SiteID <- "WSR01"  
#  
# PROMPT; Data Type  
# Type of data file  
Selection.Type <- c("Air","water","AW","Gage","AWG")  
myData.Type <- Selection.Type[2] #number corresponds to intended op  
#  
# PROMPT; Start Date  
# YYYY-MM-DD ("- " delimiter) leave blank for all data ( )  
myData.DateRange.Start <- "2014-01-01"  
#  
# PROMPT; End Date  
# YYYY-MM-DD ("- " delimiter) leave blank for all data ( )  
myData.DateRange.End <- "2014-12-31"  
#####  
# PROMPT; subDirectory, input file location. Leave blank for default.  
Selection.SUB <- c("Data1_RAW","Data2_QC","Data3_Aggregate")  
myDir.SUB.import <- "" #Selection.SUB[2]  
#  
# PROMPT; subDirectory, output file location. Leave blank for default.  
myDir.SUB.export <- "" #Selection.SUB[3]
```

The only entry that
you need to
change is Data
type (from [1] to [2])

The 5 entries should read –

Operation [3] = Aggregate

SiteID "WSR01"

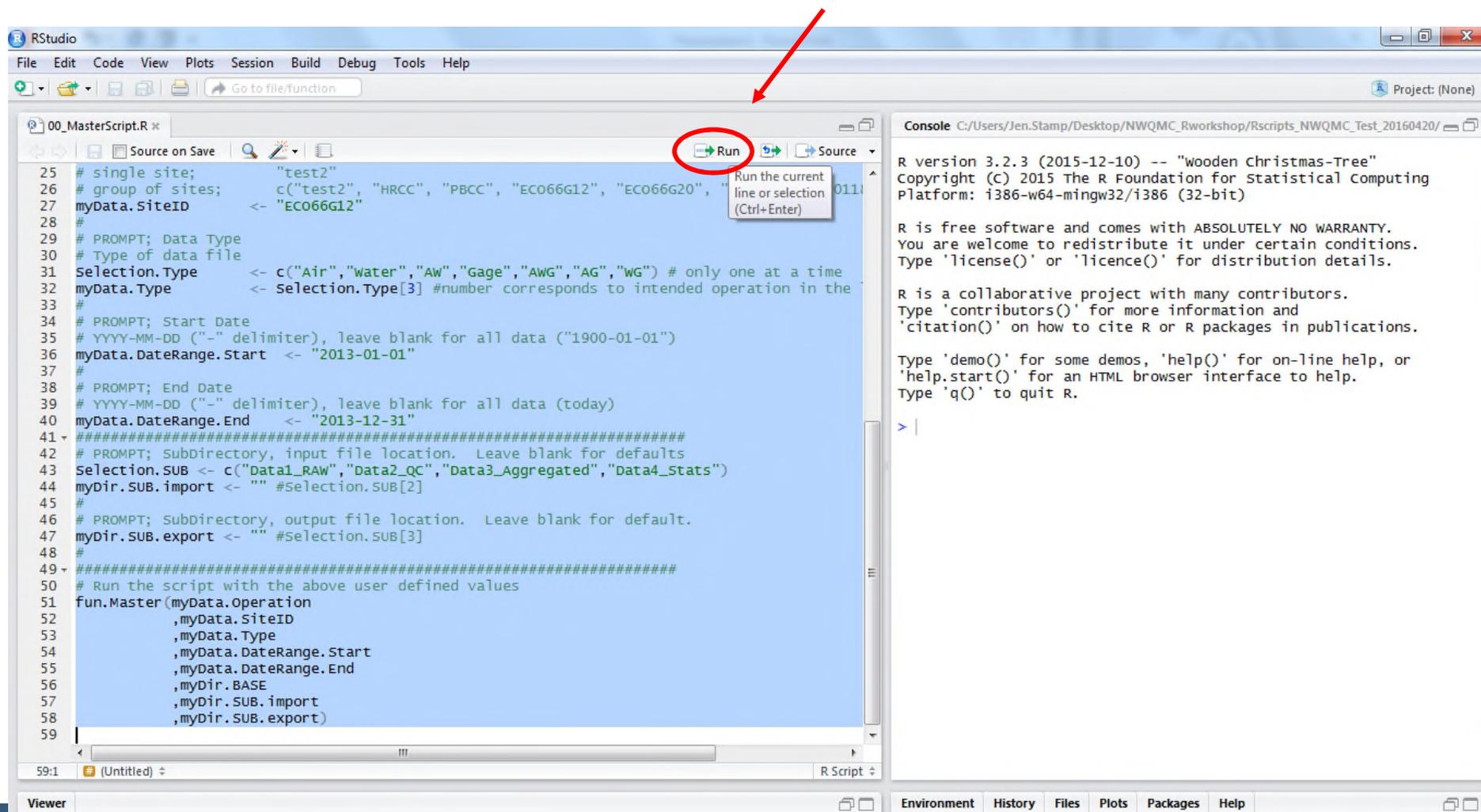
Data type [2] = Water

Start date "2014-01-01"

End date "2014-12-31"

5. Run the script

Using your cursor, highlight the script
and click RUN



```
00_MasterScript.R *
Source on Save
25 # single site;      "test2"
26 # group of sites;  c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "011
27 myData.SiteID      <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type     <- c("Air","water","AW","Gage","AWG","AG","WG") # only one at a time
32 myData.Type        <- Selection.Type[3] #number corresponds to intended operation in the
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End   <- "2013-12-31"
41 #####
42 # PROMPT; Subdirectory, input file location. Leave blank for defaults
43 Selection.SUB <- c("Data1_RAW","Data2_QC","Data3_Aggregated","Data4_Stats")
44 myDir.SUB.import <- "" #Selection.SUB[2]
45 #
46 # PROMPT; Subdirectory, output file location. Leave blank for default.
47 myDir.SUB.export <- "" #Selection.SUB[3]
48 #
49 #####
50 # Run the script with the above user defined values
51 fun.Master(myData.Operation
52            ,myData.SiteID
53            ,myData.Type
54            ,myData.DateRange.Start
55            ,myData.DateRange.End
56            ,myDir.BASE
57            ,myDir.SUB.import
58            ,myDir.SUB.export)
59
```

Run the current line or selection (Ctrl+Enter)

Console C:/Users/Jen.Stamp/Desktop/NWQMC_Rworkshop/Rscripts_NWQMC_Test_20160420/

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

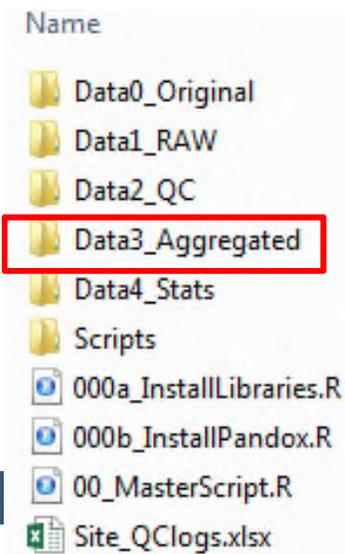


6. Verify that the script ran properly – **water temperature**

Check your Console (right) panel – log should include
“Output created” and “Task COMPLETE”

**Open the Data3_Aggregated folder – verify that there
are 2 outputs for the WSR01 *water temperature* file**

- .CSV
- Aggregate report (Word)



Name	Date modified	Type	Size
DATA_ECO66G12_Aw_20130101_20131231	2016-04-21 19:28	Microsoft Excel C...	2,070 KB
DATA_ECO66G12_Aw_20130101_20131231_Report_Aggregate	2016-04-21 19:28	Microsoft Word D...	54 KB
DATA_WSR01_Air_20140101_20141231	2016-04-21 19:39	Microsoft Excel C...	437 KB
DATA_WSR01_Air_20140101_20141231_Report_Aggregate	2016-04-21 19:39	Microsoft Word D...	19 KB
DATA_WSR01_Water_20140101_20141231_Report_Aggregate	2016-04-21 19:39	Microsoft Word D...	16 KB
DATA_WSR01_Water_20140101_20141231	2016-04-21 19:39	Microsoft Excel C...	437 KB

7. Update entries in the MasterScript for the **AW (air-water)** file

```
16 # *****  
17 # USER input in this section (see end of script for explanations)  
18 # *****  
19 #  
20 # PROMPT; Operation  
21 Selection.Operation <- c("GetGageData", "QC Raw", "Aggregate", "SummaryStats")  
22 myData.operation    <- Selection.Operation[3] #number corresponds to intended  
23 #  
24 # PROMPT; Site ID  
25 # single site;          "test2"  
26 # group of sites;     c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "ECO  
27 myData.SiteID        <- "WSR01"  
28 #  
29 # PROMPT; Data Type  
30 # Type of data file  
31 Selection.Type       <- c("Air", "water", "AW", "Gage", "true", "false", "no")  
32 myData.Type         <- Selection.Type[3] #number corresponds to intended  
33 #  
34 # PROMPT; Start Date  
35 # YYYY-MM-DD ("-" delimiter), leave blank for all d  
36 myData.DateRange.Start <- "2014-01-01"  
37 #  
38 # PROMPT; End Date  
39 # YYYY-MM-DD ("-" delimiter), leave blank for all d  
40 myData.DateRange.End   <- "2014-12-31"  
41 # *****
```

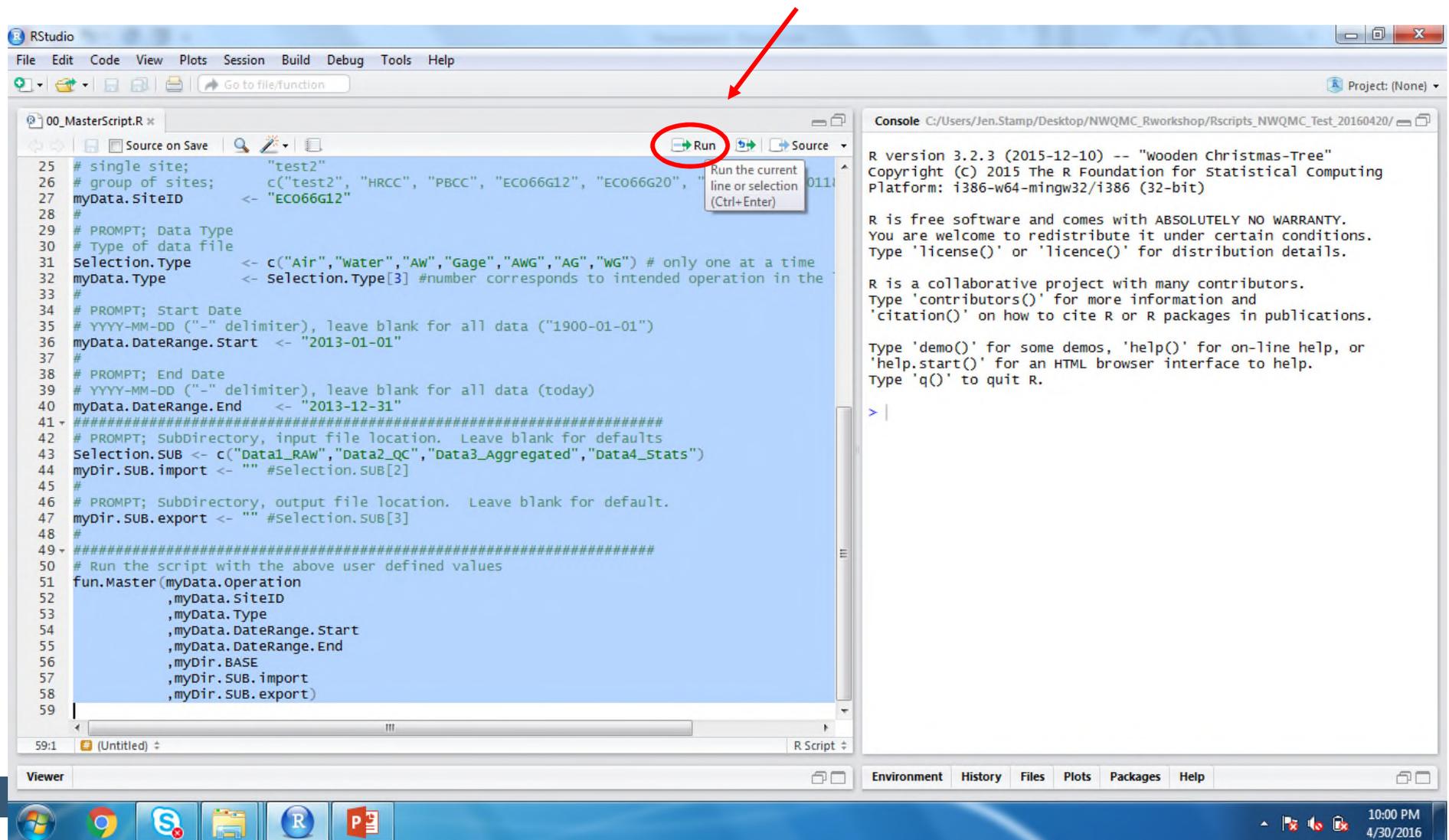
The only entry that you need to change is Data type (from [2] to [3])

The 5 entries should read –

- Operation [3] = Aggregate
- SiteID "WSR01"
- Data type [3] = AW
- Start date "2014-01-01"
- End date "2014-12-31"

8. Run the script

Using your cursor, highlight the script
and click RUN



The screenshot shows the RStudio interface with the following elements:

- Code Editor:** Contains an R script named `00_MasterScript.R`. The script defines variables for site information, data type, date range, and directory paths, and then calls a function `fun.Master`.
- Run Button:** Located in the top toolbar, circled in red. A tooltip indicates it runs the current line or selection (Ctrl+Enter).
- Console:** Displays the R version (3.2.3), copyright information, and license details.
- Environment/Plots/Files/Package/Help:** The bottom status bar shows these panels are available.

```
25 # single site;      "test2"
26 # group of sites;  c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "
27 myData.SiteID      <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type     <- c("Air", "water", "AW", "Gage", "AWG", "AG", "WG") # only one at a time
32 myData.Type        <- Selection.Type[3] #number corresponds to intended operation in the
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End   <- "2013-12-31"
41 #####
42 # PROMPT; Subdirectory, input file location. Leave blank for defaults
43 Selection.SUB <- c("Data1_RAW", "Data2_QC", "Data3_Aggregated", "Data4_Stats")
44 myDir.SUB.import <- "" #Selection.SUB[2]
45 #
46 # PROMPT; Subdirectory, output file location. Leave blank for default.
47 myDir.SUB.export <- "" #Selection.SUB[3]
48 #
49 #####
50 # Run the script with the above user defined values
51 fun.Master(myData.Operation
52            ,myData.SiteID
53            ,myData.Type
54            ,myData.DateRange.Start
55            ,myData.DateRange.End
56            ,myDir.BASE
57            ,myDir.SUB.import
58            ,myDir.SUB.export)
59
```

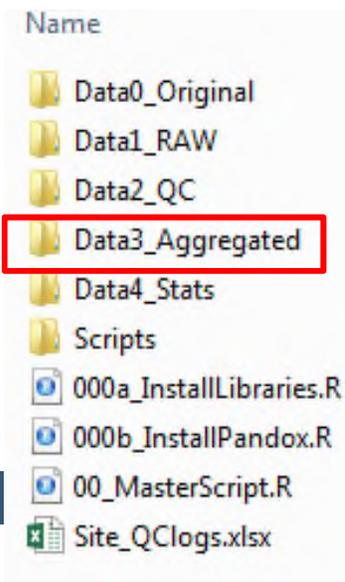


9. Verify that the script ran properly – **AW**

Check your Console (right) panel – log should include
“Output created” and “Task COMPLETE”

**Open the Data3_Aggregated folder – verify that there
are 2 outputs for the WSR01 AW file**

- .CSV
- Aggregate report (Word)

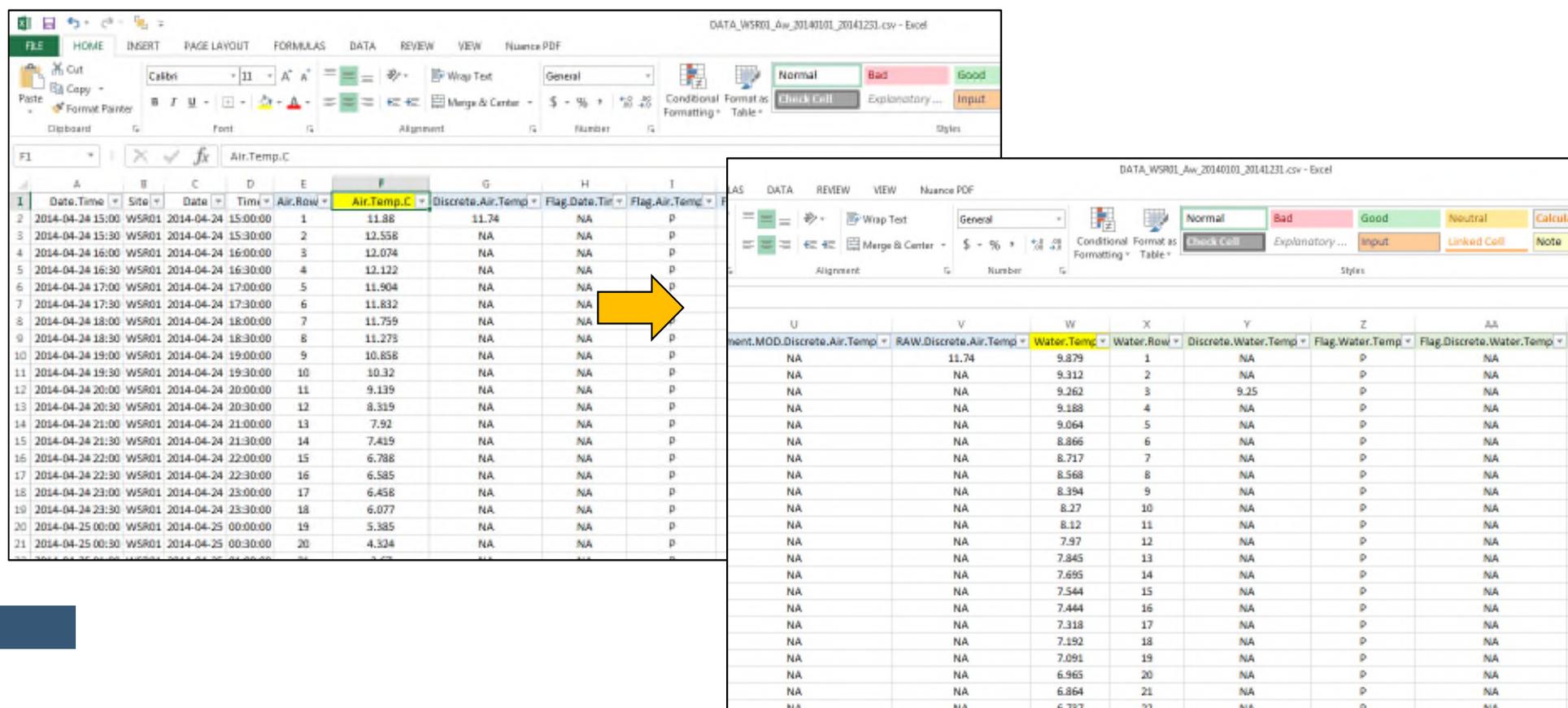


Name	Date modified	Type	Size
 DATA_ECO66G12_Aw_20130101_20131231	2016-04-21 19:28	Microsoft Excel C...	2,070 KB
 DATA_ECO66G12_Aw_20130101_20131231_Report_Aggregate	2016-04-21 19:28	Microsoft Word D...	54 KB
 DATA_WSR01_Air_20140101_20141231	2016-04-21 19:39	Microsoft Excel C...	437 KB
 DATA_WSR01_Air_20140101_20141231_Report_Aggregate	2016-04-21 19:39	Microsoft Word D...	19 KB
 DATA_WSR01_Water_20140101_20141231	2016-04-21 19:39	Microsoft Excel C...	437 KB
 DATA_WSR01_Water_20140101_20141231_Report_Aggregate	2016-04-21 19:39	Microsoft Word D...	16 KB
 DATA_WSR01_Aw_20140101_20141231_Report_Aggregate	2016-04-21 20:02	Microsoft Word D...	33 KB
 DATA_WSR01_Aw_20140101_20141231	2016-04-21 20:02	Microsoft Excel C...	592 KB

10. Open the AW files – give them a quick review (time-permitting)

Verify that both parameters are in the file.

Note: the air temperature fields come first (including flags, etc.); you'll need to scroll to the right to find the water temperature fields



The image displays two screenshots of an Excel spreadsheet. The left screenshot shows a table with columns: Date, Time, Site, Air.Temp.C, Discrete.Air.Temp, Flag.Date, and Flag.Air.Temp. The right screenshot shows a table with columns: Water.Temp, Water.Row, Discrete.Water.Temp, Flag.Water.Temp, and Flag.Discrete.Water.Temp. A yellow arrow points from the right side of the first table to the left side of the second table, indicating the need to scroll to find water temperature fields.

Date	Time	Site	Air.Temp.C	Discrete.Air.Temp	Flag.Date	Flag.Air.Temp
2014-04-24	15:00	WSR01	11.88	11.74	NA	P
2014-04-24	15:30	WSR01	12.558	NA	NA	P
2014-04-24	16:00	WSR01	12.074	NA	NA	P
2014-04-24	16:30	WSR01	12.122	NA	NA	P
2014-04-24	17:00	WSR01	11.904	NA	NA	P
2014-04-24	17:30	WSR01	11.832	NA	NA	P
2014-04-24	18:00	WSR01	11.759	NA	NA	P
2014-04-24	18:30	WSR01	11.273	NA	NA	P
2014-04-24	19:00	WSR01	10.858	NA	NA	P
2014-04-24	19:30	WSR01	10.32	NA	NA	P
2014-04-24	20:00	WSR01	9.139	NA	NA	P
2014-04-24	20:30	WSR01	8.319	NA	NA	P
2014-04-24	21:00	WSR01	7.92	NA	NA	P
2014-04-24	21:30	WSR01	7.419	NA	NA	P
2014-04-24	22:00	WSR01	6.798	NA	NA	P
2014-04-24	22:30	WSR01	6.585	NA	NA	P
2014-04-24	23:00	WSR01	6.458	NA	NA	P
2014-04-24	23:30	WSR01	6.077	NA	NA	P
2014-04-25	00:00	WSR01	5.385	NA	NA	P
2014-04-25	00:30	WSR01	4.324	NA	NA	P

Water.Temp	Water.Row	Discrete.Water.Temp	Flag.Water.Temp	Flag.Discrete.Water.Temp
NA	1	NA	P	NA
9.312	2	NA	P	NA
9.262	3	9.25	P	NA
9.188	4	NA	P	NA
9.064	5	NA	P	NA
8.866	6	NA	P	NA
8.717	7	NA	P	NA
8.568	8	NA	P	NA
8.394	9	NA	P	NA
8.27	10	NA	P	NA
8.12	11	NA	P	NA
7.97	12	NA	P	NA
7.845	13	NA	P	NA
7.695	14	NA	P	NA
7.544	15	NA	P	NA
7.444	16	NA	P	NA
7.318	17	NA	P	NA
7.192	18	NA	P	NA
7.091	19	NA	P	NA
6.965	20	NA	P	NA
6.864	21	NA	P	NA
6.737	22	NA	P	NA

1. Update entries in the MasterScript



```
16 ▾ #####
17 # USER input in this section (see end of script for explanations)
18 ▾ #####
19 #
20 # PROMPT; Operation
21 Selection.Operation <- c("GetGageData","QCraw", "Aggregate", "SummaryStats")
22 myData.Operation <- Selection.Operation[4] #number corresponds to intended
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "E
27 myData.SiteID <- "WSR01"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG","WG") # only o
32 myData.Type <- Selection.Type[3] #number corresponds to intended oper
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for
36 myData.DateRange.Start <- "2014-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for
40 myData.DateRange.End <- "2014-12-31"
41 ▾ #####
```

The only entry that you need to change is Operation from [3] to [4])

The 5 entries should read –

Operation [4] = SummaryStats

SiteID **"WSR01"**

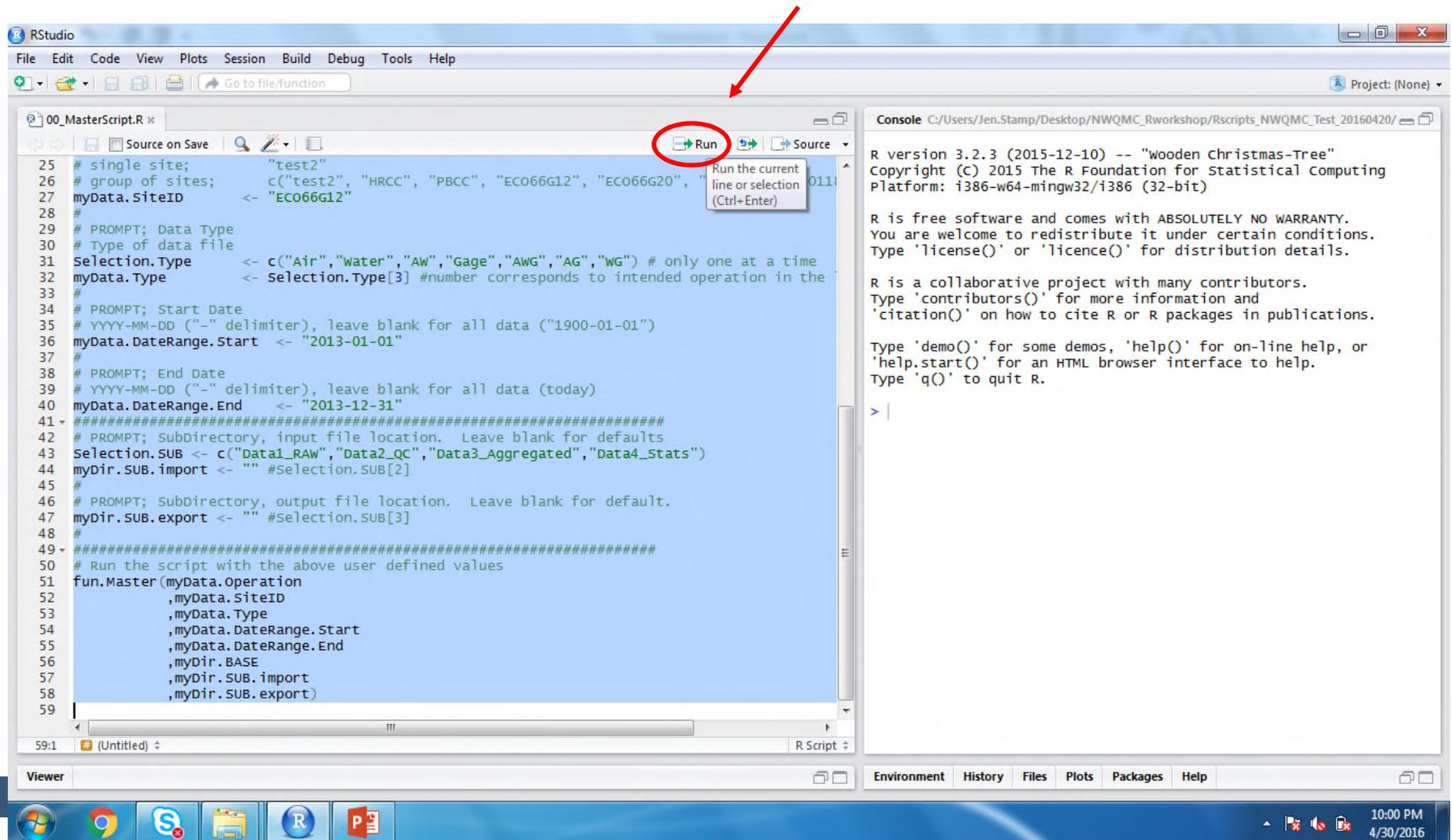
Data type **[3]** = Air + water

Start date **"2014-01-01"**

End date **"2014-12-31"**

2. Run the script

Using your cursor, highlight the script
and click RUN



The screenshot shows the RStudio interface with the following components:

- Code Editor:** Contains an R script named `00_MasterScript.R`. The script defines variables for site information, data type, date range, and directory paths, and then calls a function `fun.Master` with these variables. The entire script is highlighted in blue.
- Toolbar:** The `Run` button (represented by a play icon) is circled in red, with a red arrow pointing to it from the text above. A tooltip over the button reads: "Run the current line or selection (Ctrl+Enter)".
- Console:** Displays the R startup message: "R version 3.2.3 (2015-12-10) -- 'Wooden Christmas-Tree' Copyright (C) 2015 The R Foundation for Statistical Computing Platform: i386-w64-mingw32/i386 (32-bit)". It also shows the R license text and instructions for using `demo()`, `help()`, `help.start()`, and `q()`.
- Viewer:** Shows the Environment, History, Files, Plots, Packages, and Help panes.
- Taskbar:** Shows the Windows taskbar with icons for Internet Explorer, Chrome, Slack, File Explorer, RStudio, and PowerPoint. The system clock shows 10:00 PM on 4/30/2016.

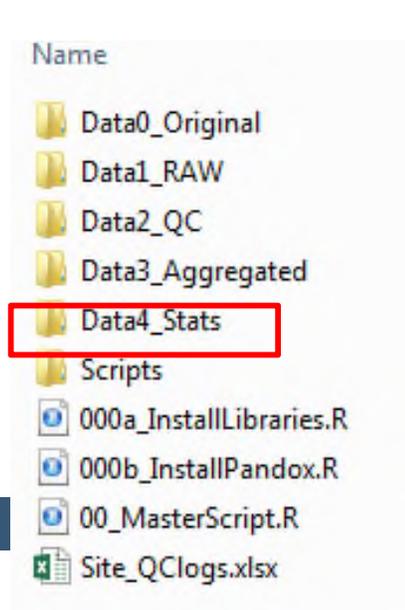
3. Verify that the script ran properly

Check your Console (right) panel –

Total items to process = 2 (Water.Temp.C, Air.Temp.C)

Open the Data4_Stats folder – verify that there are 3 outputs for each parameter (*air temperature, water temperature*)

- **DV** .csv file (daily average)
- **STATS** .csv file (summary statistics)
- **PDF** files with various time series plots



Name	Date modified	Type	Size
DV_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Microsoft Excel C...	15 KB
DV_ECO66G12_Aw_20130101_20131231_Water.Level.ft	2016-04-21 19:29	Microsoft Excel C...	16 KB
DV_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Microsoft Excel C...	16 KB
DV_WSR01_Aw_20140101_20141231_Air.Temp.C	2016-04-21 20:05	Microsoft Excel C...	7 KB
DV_WSR01_Aw_20140101_20141231_Water.Temp.C	2016-04-21 20:05	Microsoft Excel C...	7 KB
STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Microsoft Excel C...	83 KB
STATS_ECO66G12_Aw_20130101_20131231_Air.Temp.C	2016-04-21 19:29	Adobe Acrobat D...	15 KB

If time permits, open the files and give them a quick review

STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Microsoft Excel C...	82 KB
STATS_ECO66G12_Aw_20130101_20131231_Water.Temp.C	2016-04-21 19:29	Adobe Acrobat D...	15 KB
STATS_WSR01_Aw_20140101_20141231_Air.Temp.C	2016-04-21 20:05	Microsoft Excel C...	38 KB
STATS_WSR01_Aw_20140101_20141231_Air.Temp.C	2016-04-21 20:05	Adobe Acrobat D...	12 KB
STATS_WSR01_Aw_20140101_20141231_Water.Temp.C	2016-04-21 20:05	Microsoft Excel C...	38 KB
STATS_WSR01_Aw_20140101_20141231_Water.Temp.C	2016-04-21 20:05	Adobe Acrobat D...	11 KB

Congratulations!



You have successfully run
through test dataset #2!



Seeking feedback

- Wish list – what additional features and functionality would you like these scripts to have?
- Are the scripts working properly? If not, please describe the errors you are getting (screenshots are great).
- Are the **instruction slides** detailed enough? How can we improve them?



QUESTIONS? COMMENTS?



Britta Bierwagen (bierwagen.britta@epa.gov)

Erik W. Leppo (Erik.Leppo@tetratech.com)

Jen Stamp (Jen.Stamp@tetratech.com)

Anna Hamilton (Anna.Hamilton@tetratech.com)



EXTRAS

Additional Background

RMN and “Best Practices” reports

EPA/600/R-12/250 | February 2013 | www.epa.gov/research



**Regional Monitoring Networks (RMNs)
to Detect Changing Baselines in
Freshwater Wadeable Streams**



Support Clean Water Act programs

Detect trends attributable to climate change

1-2 years 3-5 years 5-10 years 10+ years

Establish current conditions

Detect trends in high quality waters

EPA/600/R-13/117 | September 2014 | www.epa.gov/oa



**Best Practices for Continuous
Monitoring of Temperature and Flow
in Wadeable Streams**



National Center for Environmental Assessment
Office of Research and Development



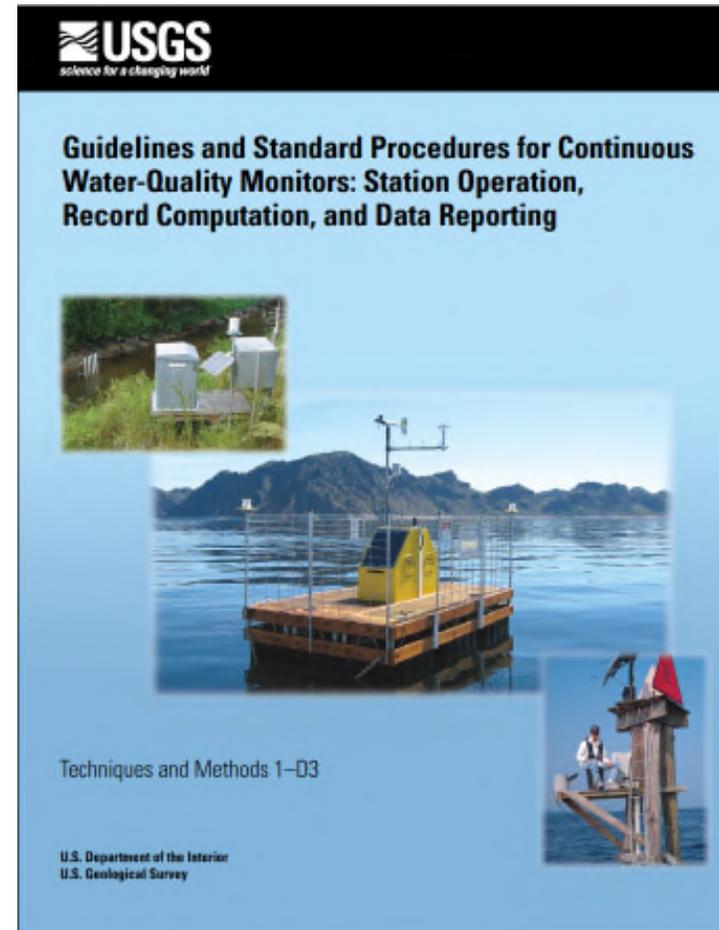
Additional R Resources (not required, just FYI)

- The R Book
 - Michael Crawley. 2007. Wiley and Sons.
- R Programming – Johns Hopkins / Coursera
 - <https://www.coursera.org/course/rprog>
- An Introduction to R
 - W. N. Venables, D. M. Smith and the R Development Core Team. 2012.
 - <http://cran.r-project.org/doc/manuals/R-intro.pdf>
- Statistical Training and Consulting Using R
 - Paul Geissler
 - <http://paulrstat.com/Reference.aspx>
- Introductory Statistics with R
 - Peter Dalgaard. 2008. Springer, New York

Automated QC report - sources/ideas

USGS Data Processing Procedures

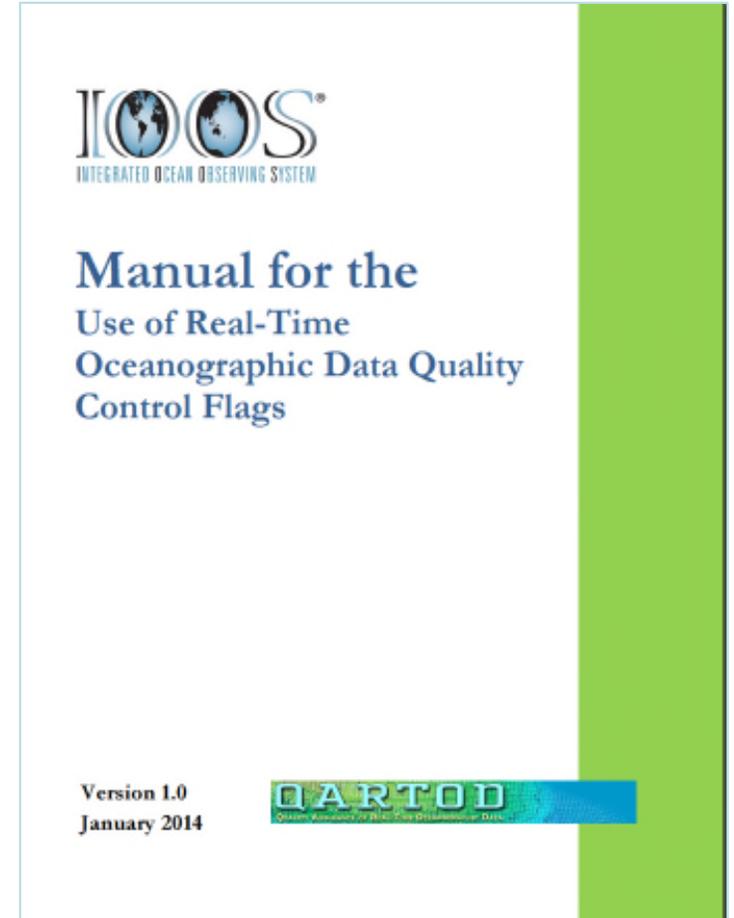
- Initial data evaluation
- Application of data corrections
- Application and evaluation of cross-section corrections
- Final data evaluation
- Record computation
- Final record review



Automated QC report - sources/ideas

NOAA QARTOD

- Timing/gap test
- Gross range test
 - High and low values
 - Real-world and Equipment
- Spike test
 - High and low thresholds
 - Compare adjacent points
- Rate of change test
 - N_DEV (default 3)
 - Time_Dev (default 25 hour)
 - Compare adjacent points
- Flat line test
 - Adjacent points
 - Repeat count (within a given tolerance)
 - Fail (5) and suspect (3)





More advanced material

- Running the GetGageData script to get gage data for site WSR01
- Aggregating gage data with the WSR01 temperature data
- Resetting Date Time on your computer (may be necessary if you make corrections and resave .csv files in the QC folder)
- Changing thresholds



Running the GetGageData script

Pulls data from the USGS NWIS website (thus requires an internet connection)

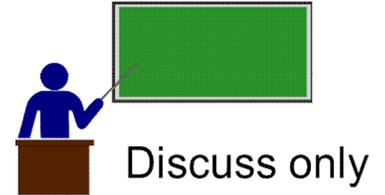
Example scenario

We need to obtain gage data for the West Branch Swift site in Massachusetts for the time period that we have MA DEP temperature sensor data for (2014-04-24 to 2014-07-10).

The USGS gage identifier is **01174565**.



Work flow for GetGageData



1. Make the appropriate entries in the MasterScript
2. Run the GetGageData portion of the script
3. Verify that the script ran properly
 - Check Console panel (should see gage information, then Task COMPLETE)
 - Open the Data1_RAW folder – there should be a gage file (.csv)
4. Open the gage file and add a new column with the SiteID that you want to match the gage data with
5. Change the file name so that it references the appropriate SiteID

1. Make appropriate entries into the MasterScript



```
10 ~~~~~
17 # USER input in this section (see end of script for explanations)
18 ~~~~~
19 #
20 # PROMPT; Operation
21 Selection.Operation <- c("GetGageData","QCRaw", "Aggregate", "SummaryStats")
22 myData.Operation <- Selection.Operation[1] #number corresponds to intended op
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC")
27 myData.SiteID <- "01174565"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage")
32 myData.Type <- Selection.Type[4] #number
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
36 myData.DateRange.Start <- "2014-04-24"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2014-07-10"
41 ~~~~~
```

The 5 entries should read –

Operation [1] = GetGageData

SiteID "01174565"

Data type [4] = Gage

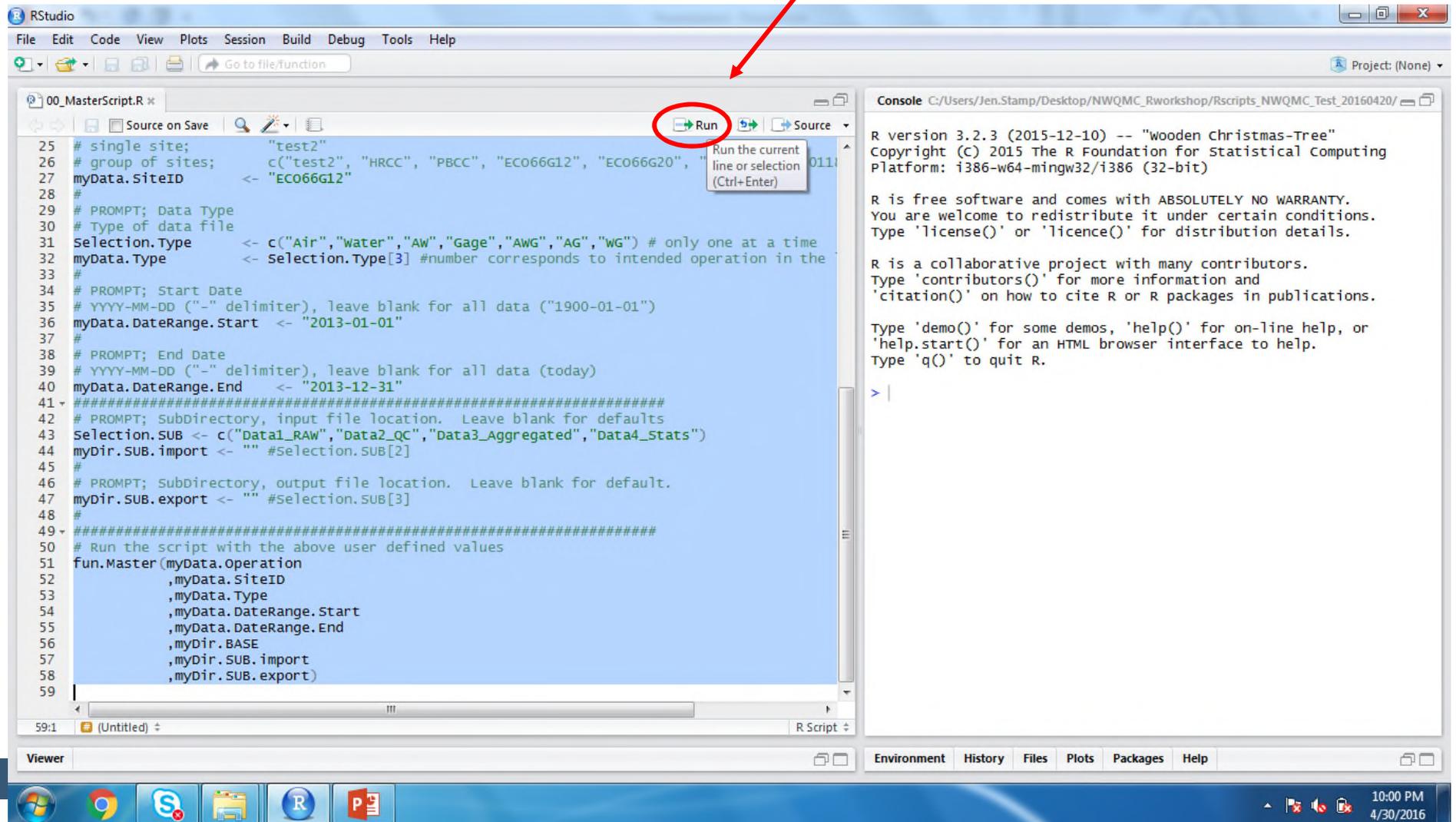
Start date "2014-04-24"

End date "2014-07-10"

2. Run the script



Using your cursor, highlight the script and click RUN



The screenshot shows the RStudio interface. The main editor window displays an R script named "00_MasterScript.R". The script contains several lines of code, including comments and assignments. A red arrow points to the "Run" button (a green play icon) in the top toolbar, which is circled in red. A tooltip over the button reads "Run the current line or selection (Ctrl+Enter)". The console window on the right shows the output of the script, including the R version (3.2.3) and copyright information. The status bar at the bottom shows the current line (59:1) and the script name (R Script).

```
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "ECO66G12", "ECO66G20", "011
27 myData.SiteID <- "ECO66G12"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air", "water", "AW", "Gage", "AWG", "AG", "WG") # only one at a time
32 myData.Type <- Selection.Type[3] #number corresponds to intended operation in the
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2013-01-01"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2013-12-31"
41 #####
42 # PROMPT; Subdirectory, input file location. Leave blank for defaults
43 Selection.SUB <- c("Data1_RAW", "Data2_QC", "Data3_Aggregated", "Data4_Stats")
44 myDir.SUB.import <- "" #Selection.SUB[2]
45 #
46 # PROMPT; Subdirectory, output file location. Leave blank for default.
47 myDir.SUB.export <- "" #Selection.SUB[3]
48 #
49 #####
50 # Run the script with the above user defined values
51 fun.Master(myData.Operation
52           ,myData.SiteID
53           ,myData.Type
54           ,myData.DateRange.Start
55           ,myData.DateRange.End
56           ,myDir.BASE
57           ,myDir.SUB.import
58           ,myDir.SUB.export)
59
```

Console C:/Users/Jen.Stamp/Desktop/NWQMC_Rworkshop/Rscripts_NWQMC_Test_20160420/

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

3. Verify that the script ran properly



Check the Console (right) panel

```

Console F:/NWQMC_2016/R_workshop/Rscripts/Rscripts_NWQMC_20150422 with outputs/
+      ,myData.SiteID
+      ,myData.Type
+      ,myData.DateRange.Start
+      ,myData.DateRange.End
+      ,myDir.BASE
+      ,myDir.SUB.import
+      ,myDir.SUB.export)
[1] "Total items to process = 1"

[1] "Getting available data; 01174565."

  parm_cd agency_cd  site_no
10  00060      USGS 01174565
12  00065      USGS 01174565

      station_nm site_tp_cd dec_lat_va
10 WEST BRANCH SWIFT RIVER NEAR SHUTESBURY, MA      ST  42.45509
12 WEST BRANCH SWIFT RIVER NEAR SHUTESBURY, MA      ST  42.45509
  dec_long_va coord_acy_cd dec_coord_datum_cd alt_va alt_acy_va
10  -72.38175          S          NAD83      540      4.3
12  -72.38175          S          NAD83      540      4.3
  alt_datum_cd  huc_cd data_type_cd  stat_cd dd_nu loc_web_ds
10      NAVD88 01080204          uv          1
12      NAVD88 01080204          uv          2
  medium_grp_cd parm_grp_cd  srs_id access_cd begin_date  end_date
10      wat          1645423          0 2007-10-01 2016-05-02
12      wat          17164583          0 2007-10-01 2016-05-02
  count_nu parameter_group_nm          parameter_nm casrn
10      3136          Physical Discharge, cubic feet per second
12      3136          Physical          Gage height, feet

      srsname parameter_units
10 stream flow, mean. daily          ft3/s
12      Height, gage          ft

[1] "Processing item 1 of 1, COMPLETE, 01174565."
[1] "Task COMPLETE; 0.08 min."

```

Gage information
will appear
immediately

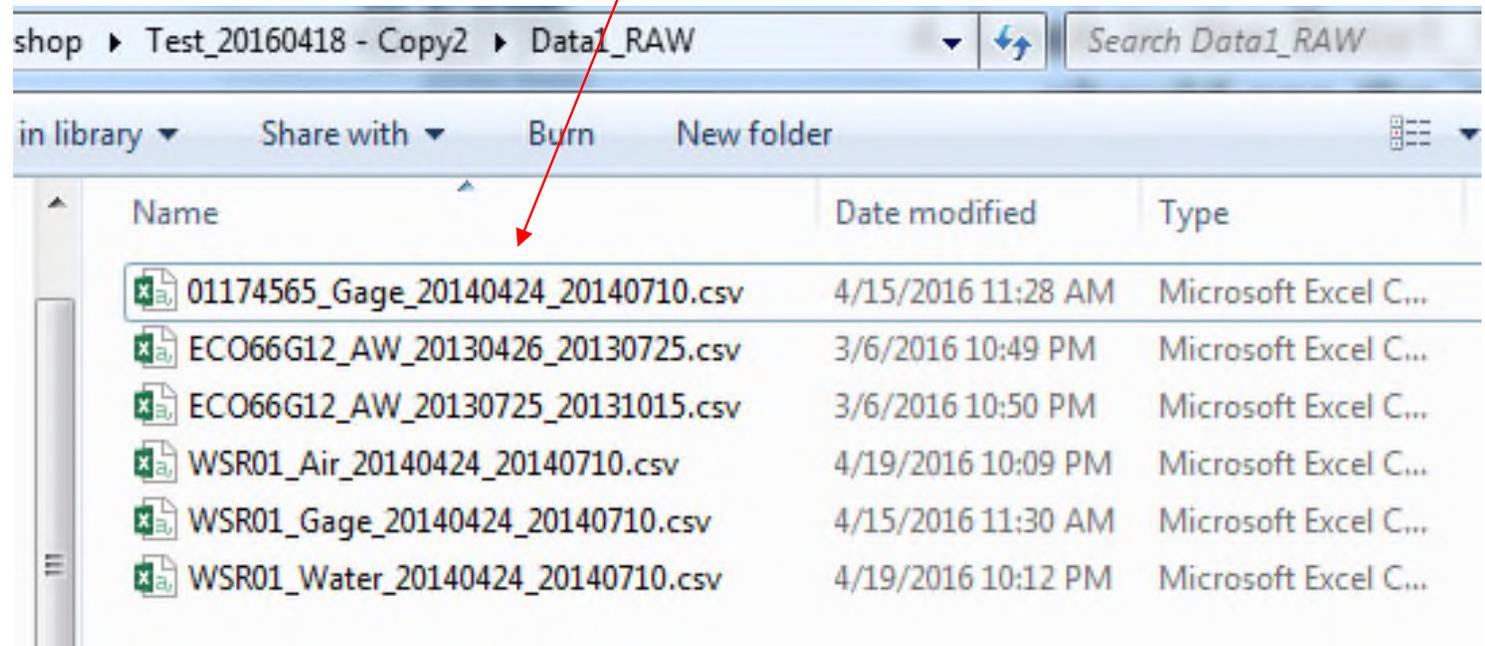
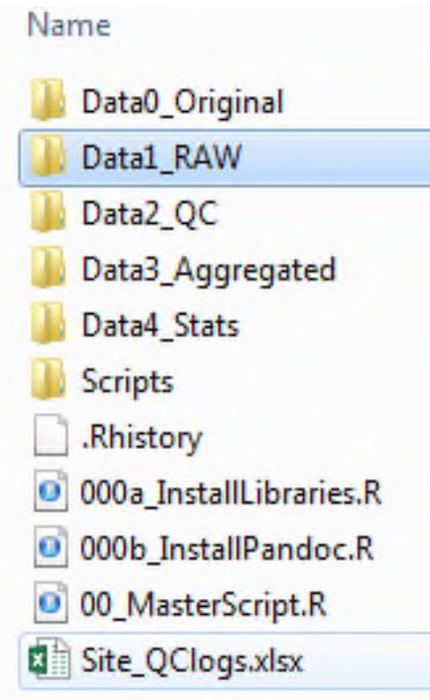
When it's finished
running, it will say
'Task COMPLETE'

3. Verify that the script ran properly



Look in the Data1_RAW folder –
you should see the gage .csv file

File name: 01174565_Gage_20140424_20140710



4. Open the gage .csv file and add in a SiteID column, then save the updated file (still as a .csv)

Fill in the SiteID that you want to match the gage data with – *in this example, WSR01*

The script formats the column headings for you

	A	B	C	D	E	F	G
1	GageID	SiteID	Date.Time	Discharge.ft3/s	Water.Level.ft	Discharge.ft3/s_cd	Water.Level.ft_cd
2	1174565	WSR01	4/24/2014 0:00	38	1.77	A	A
3	1174565	WSR01	4/24/2014 0:15	37	1.75	A	A
4	1174565	WSR01	4/24/2014 0:30	38	1.76	A	A
5	1174565	WSR01	4/24/2014 0:45	38	1.76	A	A
6	1174565	WSR01	4/24/2014 1:00	38	1.76	A	A
7	1174565	WSR01	4/24/2014 1:15	38	1.77	A	A
8	1174565	WSR01	4/24/2014 1:30	37	1.75	A	A
9	1174565	WSR01	4/24/2014 1:45	37	1.75	A	A

The output includes USGS codes (P=provisional; A=approved)



5. Change the file name so that it references the appropriate SiteID (=the SiteID that you want to match the temperature data files with)



Change the file name to 'WSR01_Gage_20140424_20140710'

Name	Date modified	Type
 01174565_Gage_20140424_20140710.csv	4/15/2016 11:28 AM	Microsoft Excel C...
 ECO66G12_AW_20130426_20130725.csv	4/21/2016 7:25 PM	Microsoft Excel C...
 ECO66G12_AW_20130725_20131015.csv	4/21/2016 7:25 PM	Microsoft Excel C...
 WSR01_Air_20140424_20140710.csv	4/20/2016 7:43 AM	Microsoft Excel C...
 WSR01_Gage_20140424_20140710.csv	4/20/2016 7:43 AM	Microsoft Excel C...
 WSR01_Water_20140424_20140710.csv	4/20/2016 7:43 AM	Microsoft Excel C...



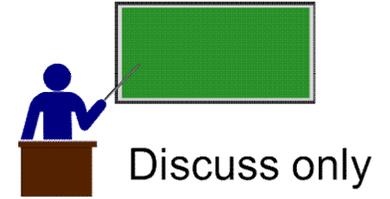
Congrats! You have successfully brought in USGS gage data.





Aggregating gage data with the WSR01 temperature data

Work flow for aggregating the gage data with the temperature data



1. **Run the QC** portion of the script on the gage data
2. **Verify that the QC script ran properly**, make corrections if needed
3. **Run the Aggregate** portion of the script on the **gage file (alone)**
4. **Run the Aggregate** portion of the script on **Data type AWG (air-water-gage)**
5. **Verify that the Aggregate script ran properly**

1. Run the QC portion of the script on the **gage** file



Make appropriate entries into the MasterScript

```
17 # USER input in this section (see end of script for explanations)
18 #####
19 #
20 # PROMPT; Operation
21 Selection.Operation <- c("GetGageData","QCRaw", "Aggregate", "SummaryStats")
22 myData.Operation <- Selection.Operation[2] #number
23 #
24 # PROMPT; site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "
27 myData.SiteID <- "WSR01"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water", "AW", "Gage",
32 myData.Type <- Selection.Type[4] #number co
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter), leave blank for all d
36 myData.DateRange.Start <- "2014-04-24"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2014-07-10"
41 #####
```

The 5 entries should read –

Operation [2] = QCRaw

SiteID "WSR01"

Data type [4] = Gage

Start date "2014-04-24"

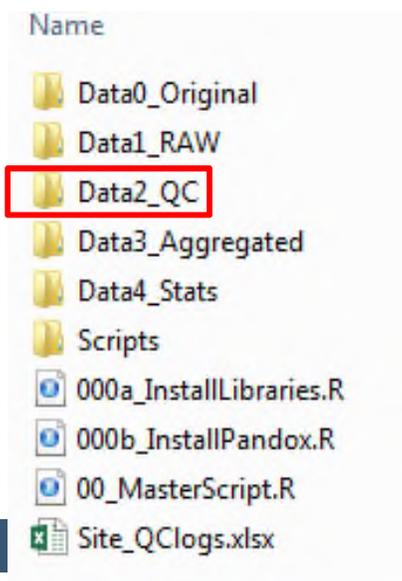
End date "2014-07-10"

Highlight the script in the Source panel and click RUN

2. Verify that the QC script ran properly



- **Check the Console** (right) panel –
 - should see ‘Task COMPLETE’
- **Open the Data2_QC folder**
 - Verify that there are 2 outputs for the WSR01 *gage* file (.csv and Word files)



Name	Date modified	Type
QC_ECO66G12_Aw_20130426_20130725.csv	4/21/2016 7:27 PM	Microsoft Excel
QC_ECO66G12_Aw_20130426_20130725_Report_QC.docx	4/21/2016 7:27 PM	Microsoft Word
QC_ECO66G12_Aw_20130725_20131015.csv	4/21/2016 7:27 PM	Microsoft Excel
QC_ECO66G12_Aw_20130725_20131015_Report_QC.docx	4/21/2016 7:27 PM	Microsoft Word
QC_WSR01_Air_20140424_20140710.csv	4/21/2016 7:38 PM	Microsoft Excel
QC_WSR01_Air_20140424_20140710_Report_QC.docx	4/21/2016 7:38 PM	Microsoft Word
QC_WSR01_Gage_20140424_20140710.csv	4/21/2016 7:38 PM	Microsoft Excel
QC_WSR01_Gage_20140424_20140710_Report_QC.docx	4/21/2016 7:38 PM	Microsoft Word
QC_WSR01_Water_20140424_20140710.csv	4/21/2016 7:38 PM	Microsoft Excel
QC_WSR01_Water_20140424_20140710_Report_QC.docx	4/21/2016 7:38 PM	Microsoft Word

3. Run the Aggregate portion of the script on the **gage** file



Make appropriate entries into the MasterScript

```
16 ▾ #####
17 # USER input in this section (see end of script for explanations)
18 ▾ #####
19 #
20 # PROMPT; Operation
21 Selection.Operation <- c("GetGageData","QCRAW", "Aggregate" "SummaryStats")
22 myData.Operation <- Selection.Operation [3] #number
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "EC
27 myData.SiteID <- "WSR01"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","A
32 myData.Type <- Selection.Type [4] #number corr
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("-" delimiter), leave blank for all dat
36 myData.DateRange.Start <- "2014-04-24"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("-" delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2014-07-10"
41 ▾ #####
```

The 5 entries should read –

Operation [3] = Aggregate
SiteID **“WSR01”**

Data type [4] = Gage

Start date **“2014-04-24”**

End date **“2014-07-10”**

Highlight the script in the Source panel and click RUN

4. Run the Aggregate portion of the script for **AWG** (AWG= air/water/gage)



Make appropriate entries into the MasterScript

The 5 entries should read –

Operation [3] = Aggregate
SiteID **WSR01**
Data type [5] = AWG
Start date **2014-04-24**
End date **2014-07-10**

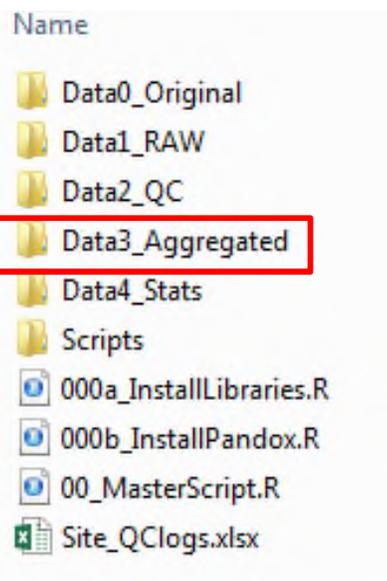
```
19 #
20 # PROMPT; operation
21 Selection.Operation <- c("GetGageData","QCraw", "Aggregate")
22 myData.Operation <- Selection.Operation[3] #number
23 #
24 # PROMPT; Site ID
25 # single site; "test2"
26 # group of sites; c("test2", "HRCC", "PBCC", "ECOC")
27 myData.SiteID <- "WSR01"
28 #
29 # PROMPT; Data Type
30 # Type of data file
31 Selection.Type <- c("Air","water","AW","Gage","AWG","AG","WG") # only one at
32 myData.Type <- Selection.Type[5] #number corresponds to intended operatio
33 #
34 # PROMPT; Start Date
35 # YYYY-MM-DD ("- " delimiter), leave blank for all data ("1900-01-01")
36 myData.DateRange.Start <- "2014-04-24"
37 #
38 # PROMPT; End Date
39 # YYYY-MM-DD ("- " delimiter), leave blank for all data (today)
40 myData.DateRange.End <- "2014-07-10"
41 #####
```

Highlight the script in the Source panel and click RUN

5. Verify that the Aggregate script ran properly



- **Check the Console** (right) panel –
 - log should include “Output created” and “Task COMPLETE”
- **Open the Data3_Aggregated folder**
 - Verify that there are 2 outputs for the WSR01 AWG file (.csv and Word files)
- **Open & check the AWG files**
 - Verify that all the 3 parameters are in the file (scroll to the right!)
 - Note the different recording intervals (gage 15-minute, temperature 30-minute)



Name	Date modified	Type
DATA_ECO66G12_Aw_20130101_20131231.csv	4/21/2016 7:28 PM	Micro
DATA_ECO66G12_Aw_20130101_20131231_Report_Aggregate.docx	4/21/2016 7:28 PM	Micro
DATA_WSR01_Air_20140101_20141231.csv	4/21/2016 7:39 PM	Micro
DATA_WSR01_Air_20140101_20141231_Report_Aggregate.docx	4/21/2016 7:39 PM	Micro
DATA_WSR01_Aw_20140101_20141231.csv	4/21/2016 8:02 PM	Micro
DATA_WSR01_Aw_20140101_20141231_Report_Aggregate.docx	4/21/2016 8:02 PM	Micro
DATA_WSR01_Awg_20140101_20141231.csv	4/21/2016 8:04 PM	Micro
DATA_WSR01_Awg_20140101_20141231_Report_Aggregate.docx	4/21/2016 8:04 PM	Micro
DATA_WSR01_Gage_20140101_20141231.csv	4/21/2016 8:04 PM	Micro
DATA_WSR01_Gage_20140101_20141231_Report_Aggregate.docx	4/21/2016 8:04 PM	Micro
DATA_WSR01_Water_20140101_20141231.csv	4/21/2016 7:39 PM	Micro
DATA_WSR01_Water_20140101_20141231_Report_Aggregate.docx	4/21/2016 7:39 PM	Micro



Congratulations!

You have successfully aggregated the
USGS gage data with the temperature data



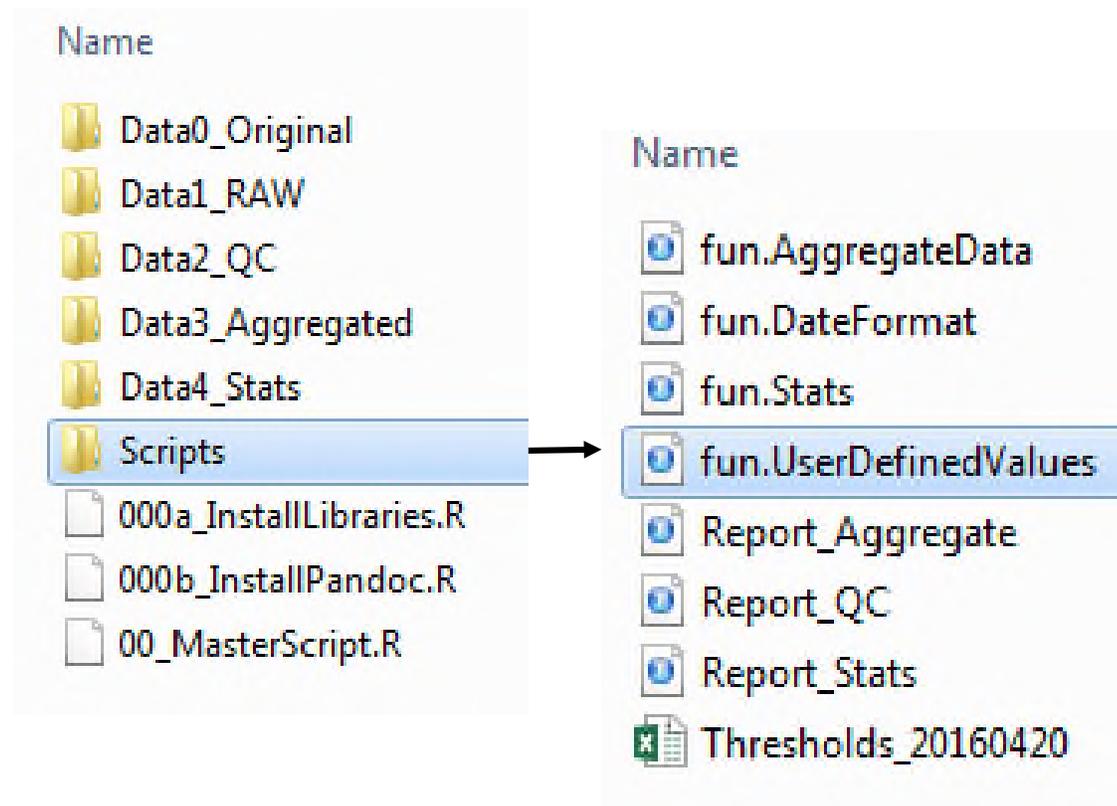


Changing thresholds for the QC tests

Changing thresholds - QC

The thresholds for the tests (gross, spikes, rate of change (RoC) and flat line) are in the **fun.UserDefinedValues.R** file in the **Scripts** folder.

You can go into this file and change thresholds if desired (we're still trying to figure out appropriate thresholds for each parameter).



Thresholds for Gross test

Gross Min/Max - Fail flag

- Extreme, unrealistic, seemingly impossible values
- Example - water temp values flagged 'F' if ≥ 30 or ≤ -2

Gross Min/Max - Suspect flag

- Highly unlikely, but not as extreme as above
- Example - water temp values flagged 'S' if ≥ 25 or ≤ -1

```

100 # *****
101 # QC Tests and Calculations
102 #http://stackoverflow.com/questions/16143700/pa
103 #myNames.QCTests.Calcs.combo <- as.vector(t(out
104 # combine so can check for and remove later.
105 #myNames.DataFields.QCTests.Calcs.combo <- as.v
106 # Data Quality Flag Thresholds
107 ## Gross Min/Max, Fail (equipment)
108 myThresh.Gross.Fail.Hi.WaterTemp <- 30
109 myThresh.Gross.Fail.Lo.WaterTemp <- -2
110 myThresh.Gross.Fail.Hi.AirTemp <- 35
111 myThresh.Gross.Fail.Lo.AirTemp <- -25
112 myThresh.Gross.Fail.Hi.WaterBP <- 17
113 myThresh.Gross.Fail.Lo.WaterBP <- 11
114 myThresh.Gross.Fail.Hi.AirBP <- 17
115 myThresh.Gross.Fail.Lo.AirBP <- 11
116 myThresh.Gross.Fail.Hi.WaterLevel <- 6
117 myThresh.Gross.Fail.Lo.WaterLevel <- -1
118 ## Gross Min/Max, Suspect (extreme)
119 myThresh.Gross.Suspect.Hi.WaterTemp <- 25
120 myThresh.Gross.Suspect.Lo.WaterTemp <- -1
121 myThresh.Gross.Suspect.Hi.AirTemp <- 30
122 myThresh.Gross.Suspect.Lo.AirTemp <- -20
123 myThresh.Gross.Suspect.Hi.WaterBP <- 16
124 myThresh.Gross.Suspect.Lo.WaterBP <- 12
125 myThresh.Gross.Suspect.Hi.AirBP <- 16
126 myThresh.Gross.Suspect.Lo.AirBP <- 12
127 myThresh.Gross.Suspect.Hi.WaterLevel <- 5
128 myThresh.Gross.Suspect.Lo.WaterLevel <- 0
129 ## spike thresholds (absolute change)

```

Screenshot showing where you can adjust thresholds for the Gross test in the **fun.UserDefinedValues.R** file (in the Scripts folder)

Thresholds for Spike test (absolute change)

- Difference from one measurement to the next
- Example – water temp values flagged ‘F’ if absolute change is ≥ 10 and flagged ‘S’ if absolute change is ≥ 5

```
129 ## spike thresholds (absolute change)
130 myThresh.Spike.Hi.WaterTemp <- 10
131 myThresh.Spike.Lo.WaterTemp <- 5
132 myThresh.Spike.Hi.AirTemp <- 10
133 myThresh.Spike.Lo.AirTemp <- 5
134 myThresh.Spike.Hi.WaterBP <- 5
135 myThresh.Spike.Lo.WaterBP <- 3
136 myThresh.Spike.Hi.AirBP <- 5
137 myThresh.Spike.Lo.AirBP <- 3
138 myThresh.Spike.Hi.WaterLevel <- 5
139 myThresh.Spike.Lo.WaterLevel <- 3
```

Screenshot showing where you can adjust thresholds for the Spike test in the **fun.UserDefinedValues.R** file (in the Scripts folder)

Thresholds for Rate of Change (RoC) (relative change) test

- Calculates standard deviation (st dev) over 25-hour period, if values differ by ≥ 3 st dev over this period, values flagged 'S'

```
140 ## Rate of Change (relative change)
141 myDefault.RoC.SD.number <- 3
142 myDefault.RoC.SD.period <- 25 #hours
143 myThresh.RoC.SD.number.WaterTemp <- myDefault.RoC.SD.number
144 myThresh.RoC.SD.period.WaterTemp <- myDefault.RoC.SD.period
145 myThresh.RoC.SD.number.AirTemp <- myDefault.RoC.SD.number
146 myThresh.RoC.SD.period.AirTemp <- myDefault.RoC.SD.period
147 myThresh.RoC.SD.number.WaterBP <- myDefault.RoC.SD.number
148 myThresh.RoC.SD.period.WaterBP <- myDefault.RoC.SD.period
149 myThresh.RoC.SD.number.AirBP <- myDefault.RoC.SD.number
150 myThresh.RoC.SD.period.AirBP <- myDefault.RoC.SD.period
151 myThresh.RoC.SD.number.WaterLevel <- myDefault.RoC.SD.number
152 myThresh.RoC.SD.period.WaterLevel <- myDefault.RoC.SD.period
153 ## No Change (flat-line)
```

Screenshot showing where you can adjust thresholds for the RoC test in the **fun.UserDefinedValues.R** file (in the Scripts folder)



Thresholds for Flat-line (no change) test

- Looks across **consecutive measurements**
- If 30 measurements in a row are within ≤ 0.01 or 0.001 of one another (depending on the parameter), values are flagged 'F'
- If 10 measurements in a row are within ≤ 0.01 or 0.001 of one another (depending on the parameter), values are flagged 'S'

```
152 myThresh.RoC.SD.number.AirBP <- myDefault.RoC.SD.number
153 myThresh.RoC.SD.period.AirBP <- myDefault.RoC.SD.period
154 myThresh.RoC.SD.number.WaterLevel <- myDefault.RoC.SD.number
155 myThresh.RoC.SD.period.WaterLevel <- myDefault.RoC.SD.period
156 ## No Change (flat-line)
157 myDefault.Flat.Hi <- 30 # maximum is myThresh.Flat.Max
158 myDefault.Flat.Lo <- 10
159 myDefault.Flat.Tolerance <- 0.01 # set to one sigdig less than
160 myThresh.Flat.Hi.WaterTemp <- myDefault.Flat.Hi
161 myThresh.Flat.Lo.WaterTemp <- myDefault.Flat.Lo
162 myThresh.Flat.Tolerance.WaterTemp <- 0.01
163 myThresh.Flat.Hi.AirTemp <- myDefault.Flat.Hi
164 myThresh.Flat.Lo.AirTemp <- myDefault.Flat.Lo
165 myThresh.Flat.Tolerance.AirTemp <- 0.01
166 myThresh.Flat.Hi.WaterP <- myDefault.Flat.Hi
167 myThresh.Flat.Lo.WaterP <- myDefault.Flat.Lo
168 myThresh.Flat.Tolerance.WaterP <- 0.001
169 myThresh.Flat.Hi.AirBP <- myDefault.Flat.Hi
170 myThresh.Flat.Lo.AirBP <- myDefault.Flat.Lo
171 myThresh.Flat.Tolerance.AirBP <- 0.001
172 myThresh.Flat.Hi.WaterLevel <- myDefault.Flat.Hi * 2
173 myThresh.Flat.Lo.WaterLevel <- myDefault.Flat.Lo * 2
174 myThresh.Flat.Tolerance.WaterLevel <- 0.0
175 myThresh.Flat.MaxComp <- max(myThresh.Flat.Hi.WaterTemp
176 ,myThresh.Flat.Hi.AirTemp
177 ,myThresh.Flat.Hi.WaterP
178 ,myThresh.Flat.Hi.AirBP
179 ,myThresh.Flat.Hi.WaterLevel)
```

Screenshot showing where you can adjust thresholds for the flat-line test in the **fun.UserDefinedValues.R** file (in the Scripts folder)

Thresholds 'translator' file

We added an Excel file into the Scripts folder to help make it easier to interpret the settings in the fun.UserDefinedValues.R script.

Name		Water Temp		Air Temp		Water Level	
		Fail	Suspect	Fail	Suspect	Fail	Suspect
1	Adjusted Table uses formulas on the next page						
2	So can modify the units and the table auto updates						
3							
4							
5	Gross	>30°C or <-2°C	>30°C or <-2°C	>35°C or <-25°C	>35°C or <-25°C	>6 ft or <-1 ft	>6 ft or <-1 ft
6	Spike	≥10°C (±)	≥5°C (±)	≥10°C (±)	≥5°C (±)	≥5 ft (±)	≥3 ft (±)
7	Rate of Change	NA	≥3 stdev within 25 hrs	NA	≥3 stdev within 25 hrs	NA	≥3 stdev within 25 hrs
8	Flat Line	>30 consecutive measurements within 0.01 units of one another	>10 consecutive measurements within 0.01 units of one another	>30 consecutive measurements within 0.01 units of one another	>10 consecutive measurements within 0.01 units of one another	>60 consecutive measurements within 0 units of one another	>20 consecutive measurements within 0 units of one another
9							
10	*gross test for water level would need to be adjusted for each site						

This file has 2 worksheets. Shown here is the Table worksheet, which 'translates' the thresholds. If you make changes in the fun.UserDefinedValues.R file, enter the changes in the 'ScriptValues' worksheet, and it will automatically update the 'Table' worksheet with the new 'translated' thresholds.

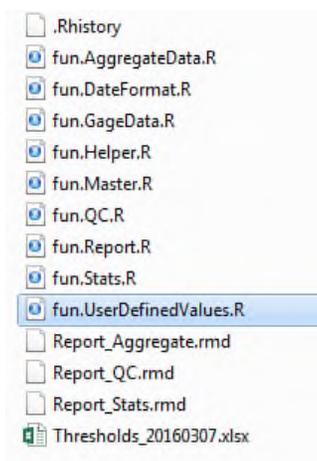


What if data are flagged as fail (F)? Are they included in the SummaryStat calculations?

The scripts are written in a way that gives you flexibility on this. In the last line (line 233) of the **fun.UserDefinedValues.R** file in the **Scripts folder**, if you enter –

- TRUE – the script will exclude values flagged as ‘F’
- FALSE – the script will include values flagged as ‘F’

Scripts folder



```
221 myTimeFrame.Season.Winter.Start <- "1201"
222 #myTimeFrame.Season.Winter.End <- "0228" #but 0229 in leap year, use start dat
223 # Time Frame Names
224 myName.Yr <- "Year"
225 myName.YrMo <- "YearMonth"
226 myName.Mo <- "Month"
227 myName.MoDa <- "MonthDay"
228 myName.JuDa <- "JulianDay"
229 myName.Season <- "Season"
230 myName.YrSeason <- "YearSeason"
231 - #####
232 # Trigger for Stats to exclude (TRUE) or include (FALSE) where flag = "fail"
233 myStats.Fails.Exclude <- TRUE #FALSE #TRUE
234 - #####
235
```

Right now the script you are working with is set to ‘TRUE’



Having trouble getting the Aggregate script to run after you make edits and resave the files in the Data2_QC folder?



Troubleshooting potential problem after resaving .csv files in the Data2_QC

We discovered that on some computers, Excel changes the date format in the .csv file if you make a change to the file in the QC folder and resave it. We believe this has to do with the settings on Erik's computer; he wrote the scripts, and the scripts utilize his date time settings (which aren't the same across computers).

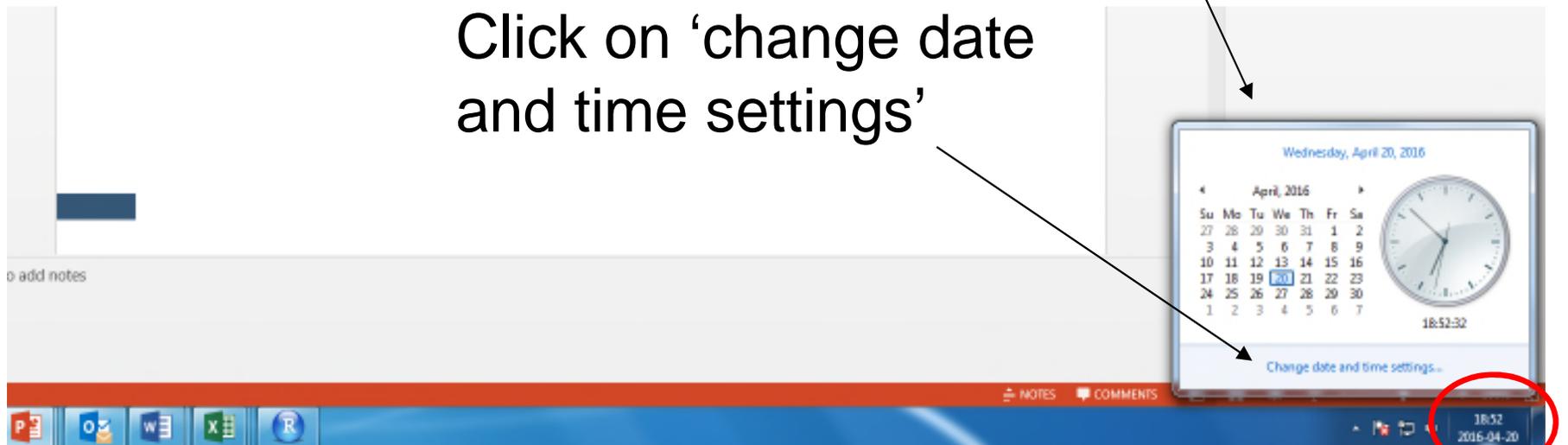
Unfortunately this issue prevents the Aggregate script from running.

The best solution we've come up with at this point is to reset the date-time setting on your computer to match with his. The next slides provide instructions on how to do this.

Left click on the time-date in the lower right hand corner of your computer screen.

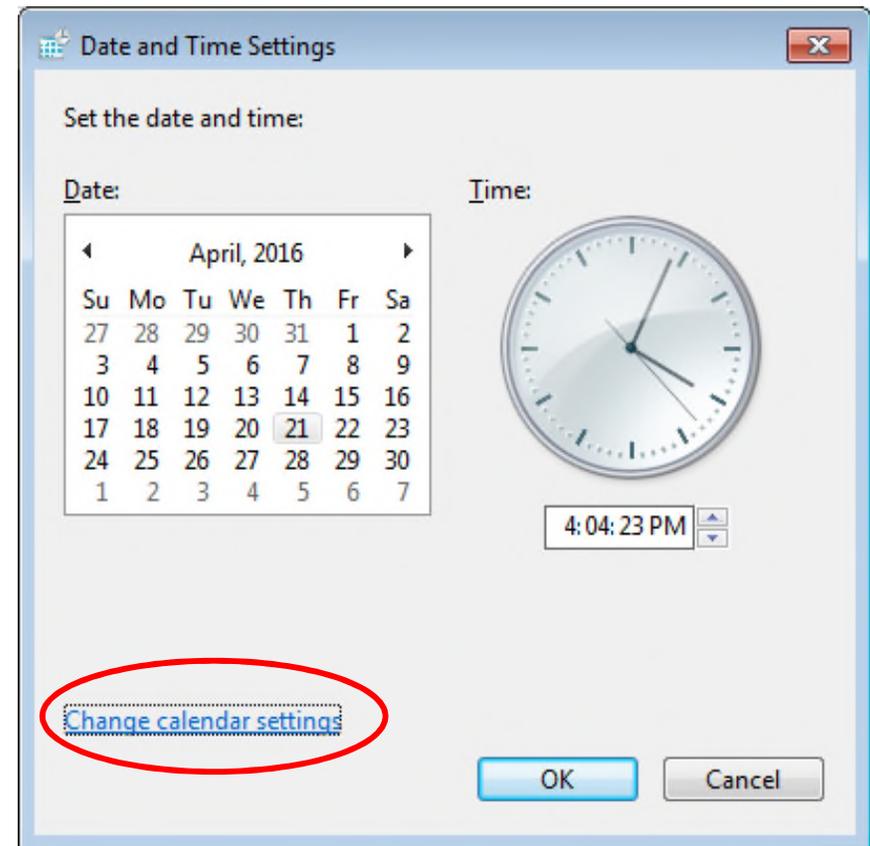
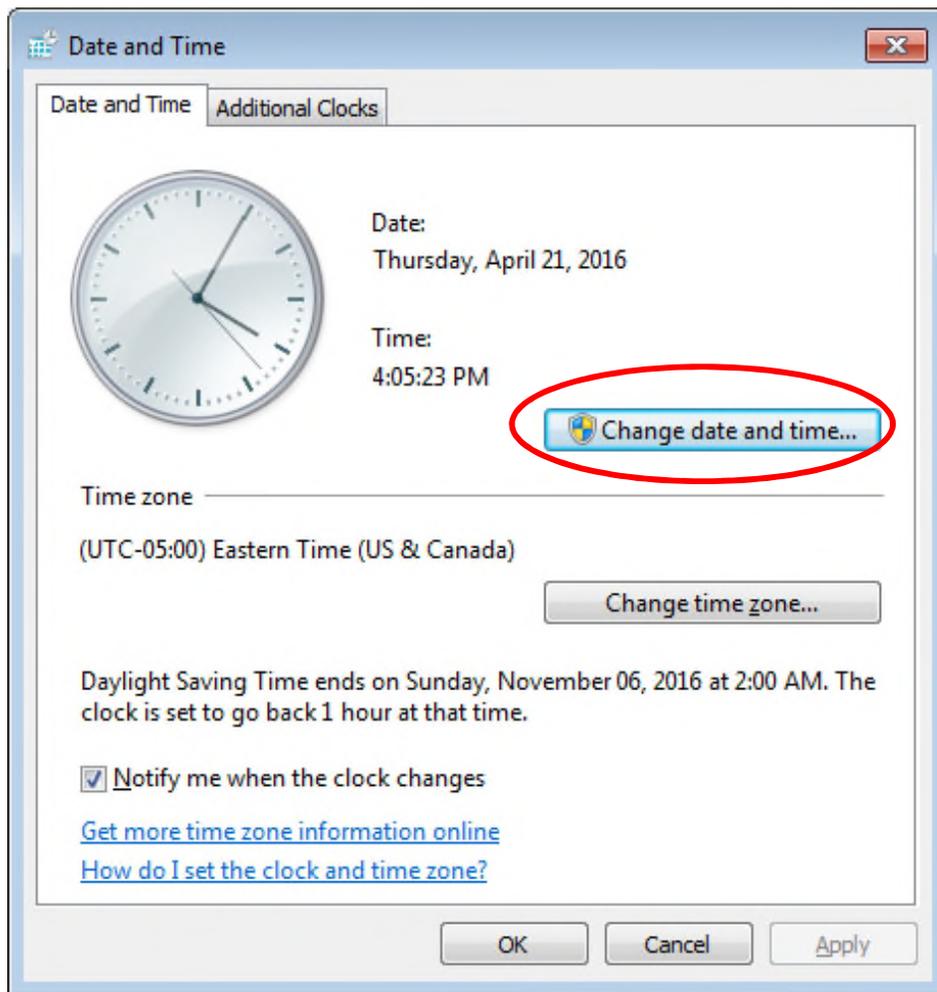
This box will then appear...

Click on 'change date and time settings'



Left click here

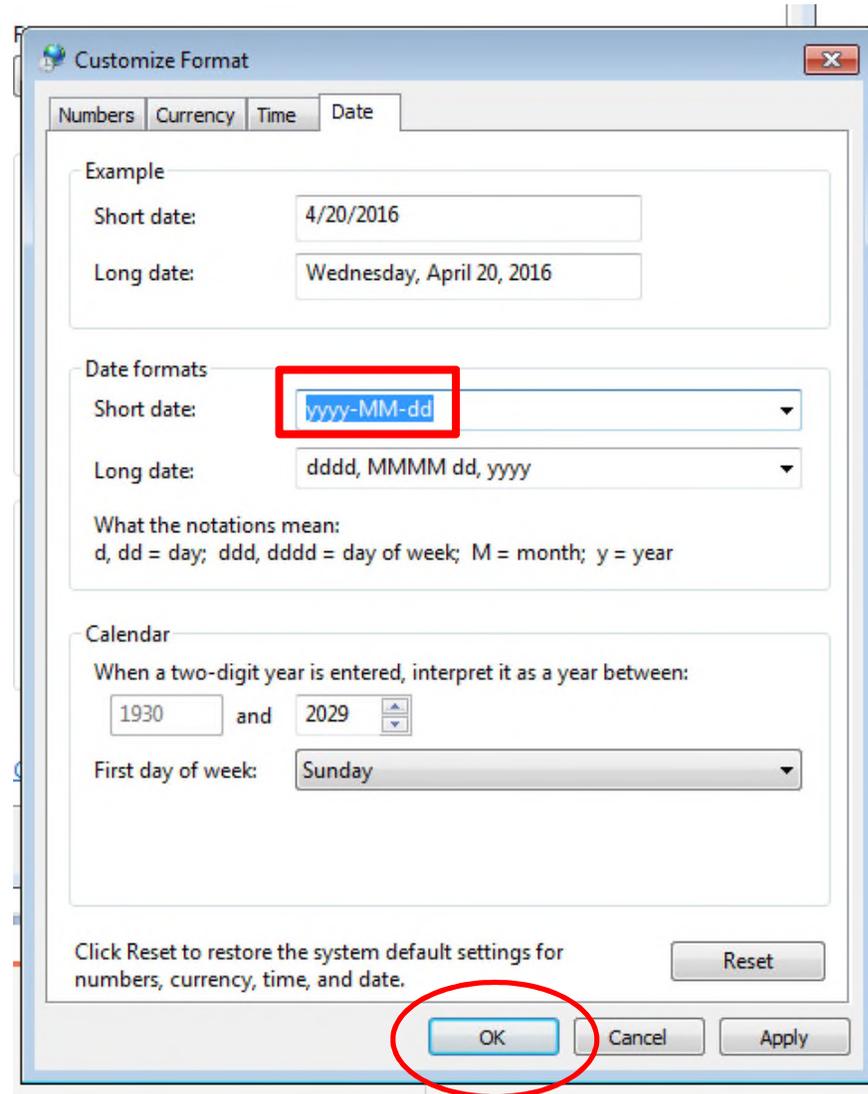
Click 'Change date and time...' Then click 'Change calendar settings'



Customize Format

- Click on the **Date** tab
- For 'Short date' format enter **yyyy-MM-dd**
- Leave 'Long date' format as is

Note: you may need administrator privileges to do this...

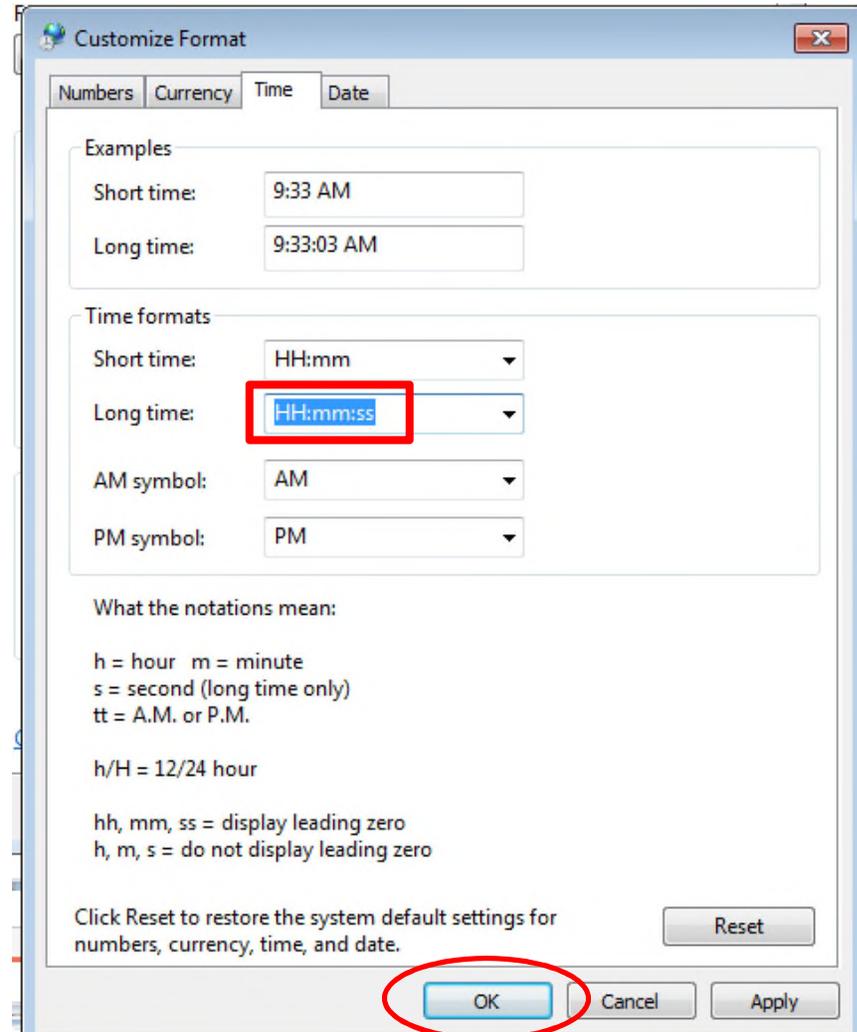


Then click 'OK'

Customize Format

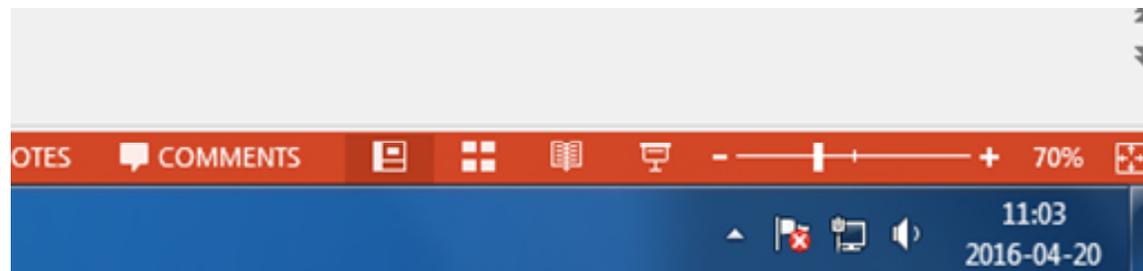
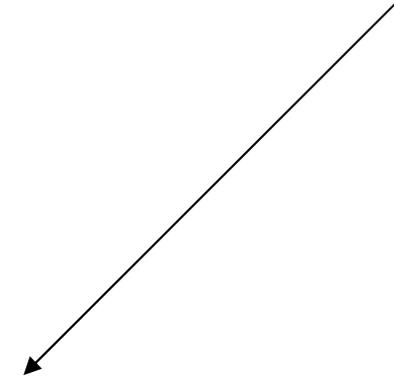
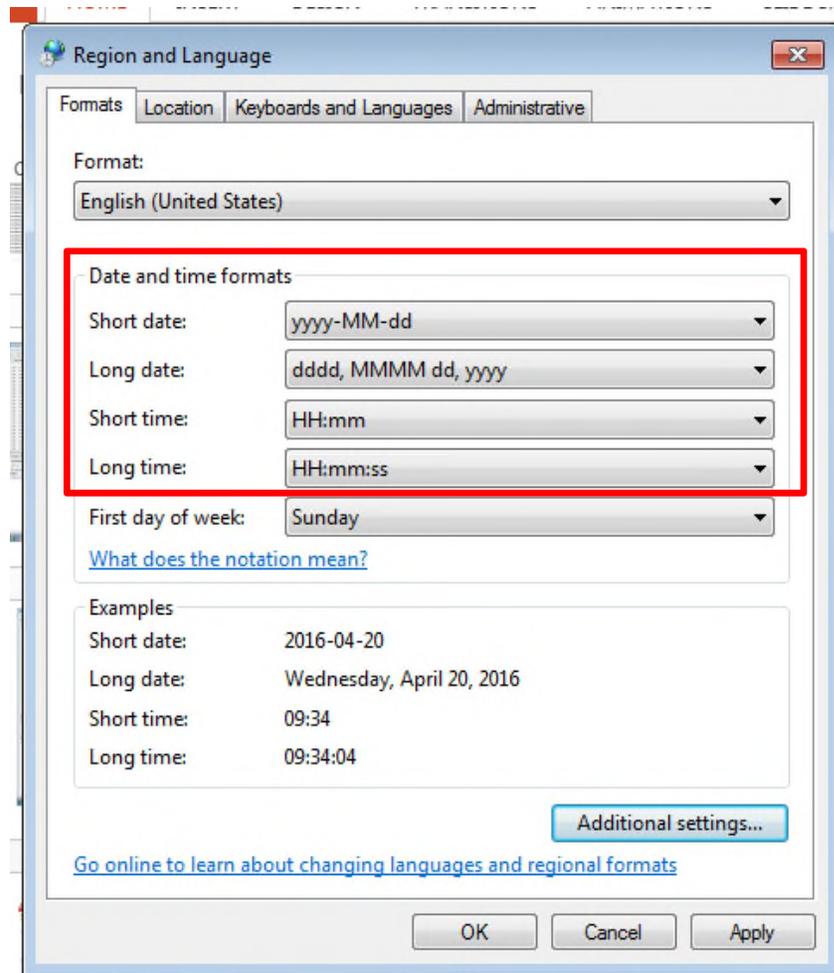
- Click on the **Time** tab
- For 'Short time' format enter **HH:mm**
- For 'Long time' format enter **HH:mm:ss**

Note: you may need administrator privileges to do this...



Then click 'OK'

Date and time formats should then look like this



And this...

