Appendix 2.1A: baytrends–Quick Start Instructions

This document contains brief instructions for getting started with baytrends. Specifically, the following issues are covered.

- How to access the help files for specific baytrends functions.
- How to open and run example scripts provided with the help files.
- Instructions on how to open and navigate the GAM trend scripts.
- Instructions on where to make changes to default options and how to run the scripts.
- Example MS Word and csv output.
- An explanation of what is in the csv output files.
- How to access the help files for specific functions.

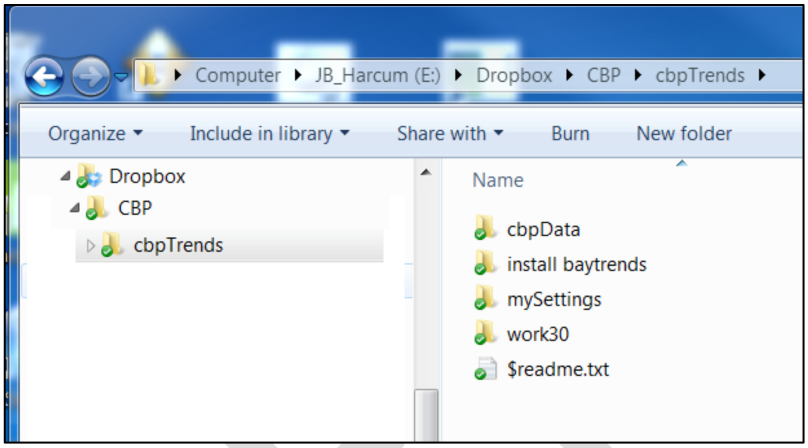
A note to potential users of the attached material. Some of the figures in this document are too small too effectively read in the printed form. It is strongly recommend that users open the files as described in this document and organize the windows on their monitor to emulate the figures included in this document. Then use the included figures to help navigate what information to review in detail from the user's monitor.

1.0 Installing baytrends

These instructions assume that you have already installed R and R Studio on your workstation.

<table>
<tr><th colspan="2">Uncompress distributed zip file</th></tr>
<tr><td>Acquire the baytrend start up files and organize them in a location that you have regular access.

In the example to the right, we located the files in the folder E:\Dropbox\CBP.

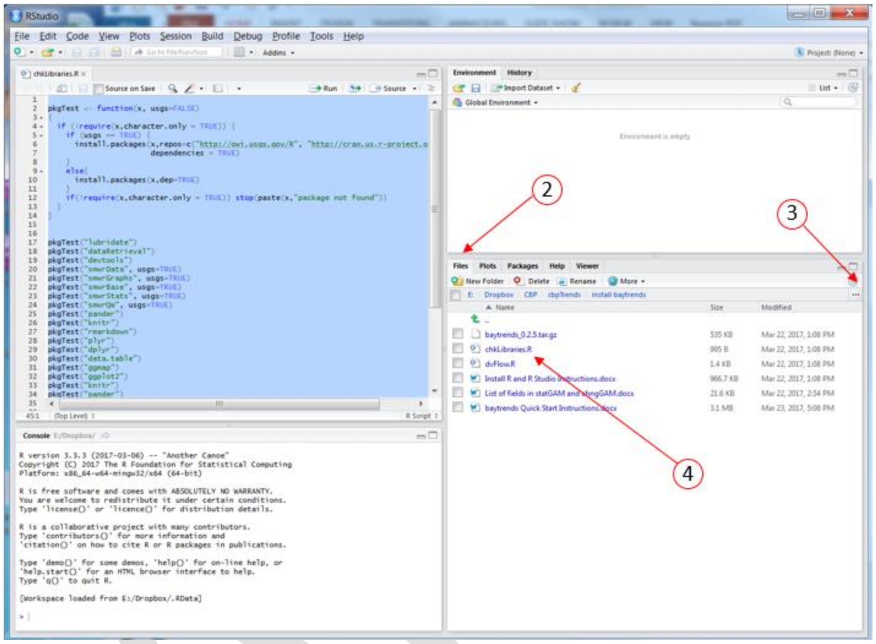
The primary folder is called ‘cbpTrends’ and contains four additional subfolders.</td><td>
</td></tr>
<tr><td colspan="2">Examine the $readme.txt file for a detailed description of the folders.

*For now, it is important to know that the **install baytrends** folder contains files that you will need to complete Section 1.0, related to installing baytrends. The **Work30** folder contains files that we’ll use in Sections 3.0 of these instructions.*</td></tr>
</table>

Install libraries that support baytrends

1. Open R Studio.
2. Click on the ‘Files’ tab in the bottom right panel.
3. Navigate to the folder with ‘chkLibraries.R’. (e.g., E:\Dropbox\CBP\cbpTrends \install baytrends)
4. Click on chkLibraries.R (the file should open in the upper left panel).
5. Using your mouse, in the upper left panel, click and drag to highlight all rows in chkLibraries.R.
6. Click the Run button.

   You will see the R code that you just highlighted being executed in the Console window. This process downloads and installs a series of R packages that are required for running the baytrends package. This process will take several minutes.



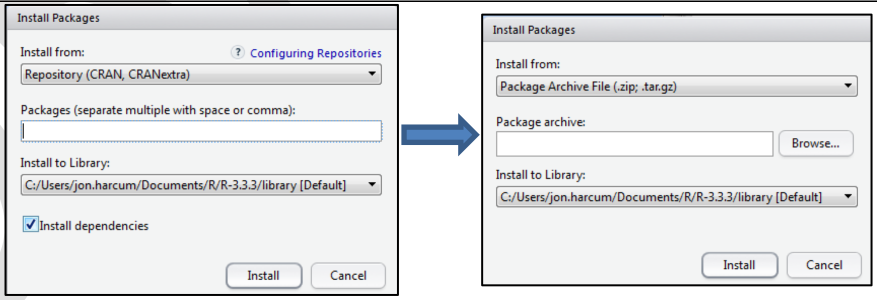
```r

pkgTest <- function(x, usgs=FALSE)
{
  if (!require(x,character.only = TRUE)) {
    if (usgs == TRUE) {
      install.packages(x,repos=c("http://owi.usgs.gov/R", "http://cran.us.r-project.o
                       dependencies = TRUE)
    }
    else{
      install.packages(x,dep=TRUE)
    }
    if(!require(x,character.only = TRUE)) stop(paste(x,"package not found"))
  }
}


pkgTest("lubridate")
pkgTest("dataRetrieval")
pkgTest("devtools")
pkgTest("smwrData", usgs=TRUE)
pkgTest("smwrGraphs", usgs=TRUE)
pkgTest("smwrBase", usgs=TRUE)
pkgTest("smwrStats", usgs=TRUE)
pkgTest("smwrQW", usgs=TRUE)
pkgTest("pander")
pkgTest("knitr")
pkgTest("rmarkdown")
pkgTest("plyr")
pkgTest("dplyr")
pkgTest("data.table")
pkgTest("ggmap")
pkgTest("ggplot2")
pkgTest("knitr")
pkgTest("pander")
```

```
R version 3.3.3 (2017-03-06) -- "Another Canoe"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-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.

[Workspace loaded from E:/Dropbox/.RData]

> |
```

Name	Size	Modified
..		
baytrends_0.2.5.tar.gz	535 KB	Mar 22, 2017, 1:08 PM
chkLibraries.R	995 B	Mar 22, 2017, 1:08 PM
dvFlow.R	1.4 KB	Mar 22, 2017, 1:08 PM
Install R and R Studio Instructions.docx	966.7 KB	Mar 22, 2017, 1:08 PM
List of fields in statGAM and [illegible]GAM.docx	21.6 KB	Mar 22, 2017, 2:54 PM
baytrends Quick Start Instructions.docx	3.1 MB	Mar 23, 2017, 5:08 PM

Install baytrends

1. From within R Studio, click Tools > Install Packages.
2. Select ‘Package Archive File …” from the ‘Install from’ drop down list.



Install from: Repository (CRAN, CRANextra)

Packages (separate multiple with space or comma):

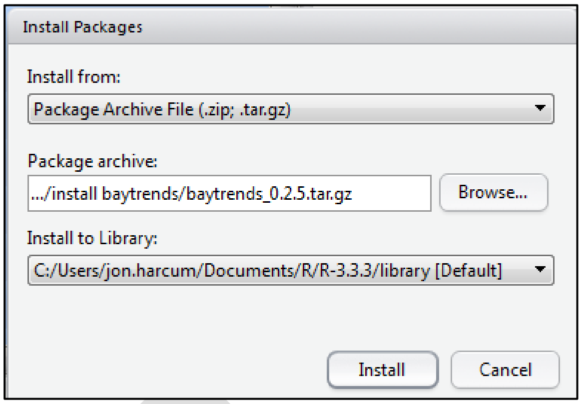
Install to Library: C:/Users/jon.harcum/Documents/R/R-3.3.3/library [Default]

☑ Install dependencies

Install from: Package Archive File (.zip; .tar.gz)

Package archive:

Install to Library: C:/Users/jon.harcum/Documents/R/R-3.3.3/library [Default]

3. Click the Browse button and navigate to the location of the baytrends_x.x.x.tar.gz file and click on the file. (In our case, the file is located in …\cbpTrends\install baytrends) 4. Click Install. Once this process has finished executing, the baytrends package is successfully installed.	 Install from: Package Archive File (.zip; .tar.gz) Package archive: .../install baytrends/baytrends_0.2.5.tar.gz Install to Library: C:/Users/jon.harcum/Documents/R/R-3.3.3/library [Default]

2.0 baytrends Primer

baytrends is similar to other R packages, Part 1—you must load any library that you want to use <u>every time</u> you start R Studio.

1. To get started, click File> New File > R Script.

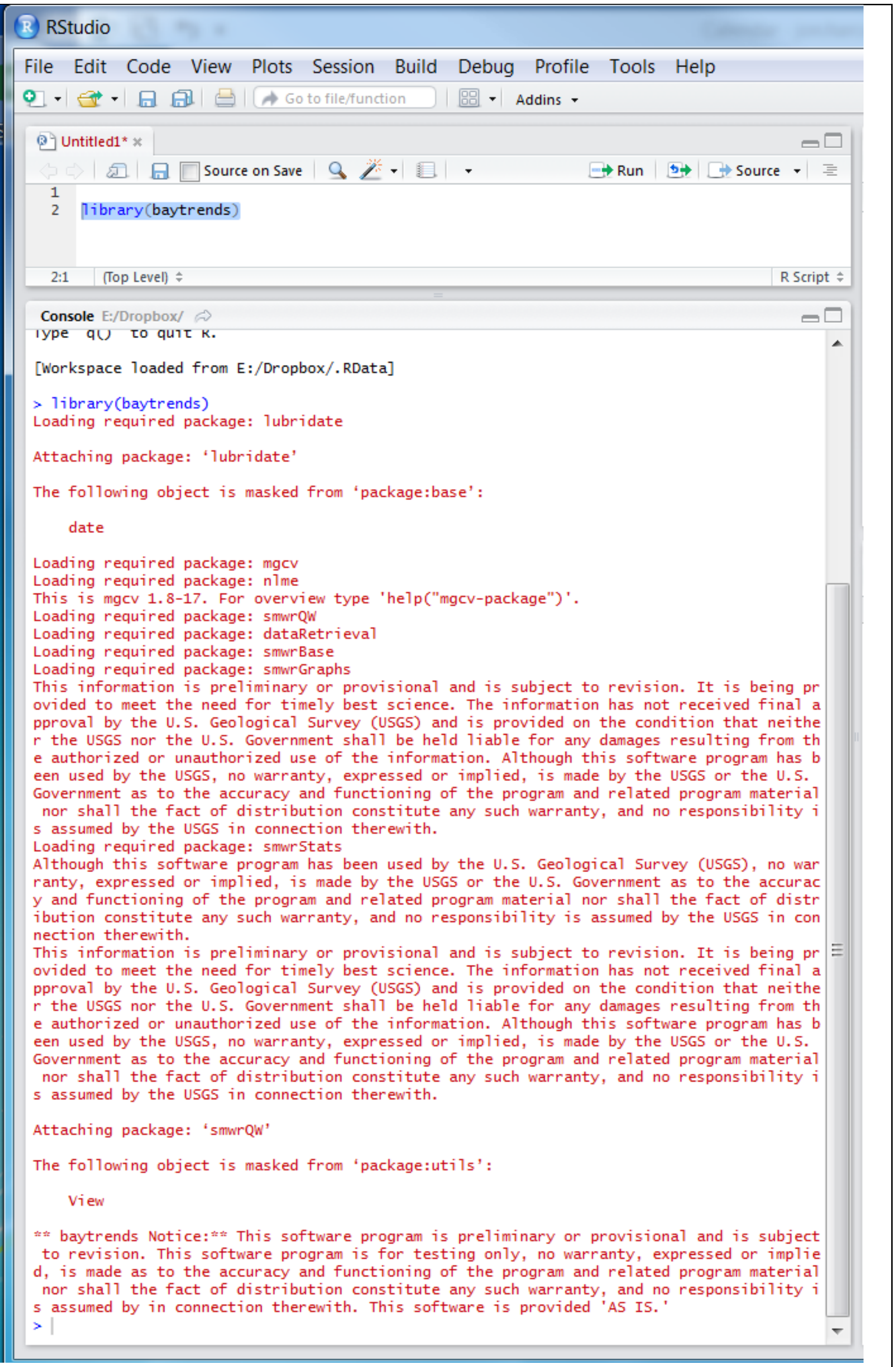
You will get an ‘Untitled1’ program file to work with.

2. Enter ‘library(baytrends)’ as it is shown to the right and click Run.

Examine the Console. This is a normal load.

Note, that you might get slightly different results depending of package versions.

You should get in the practice of scanning the loading to look for abnormal results such as a library not found.

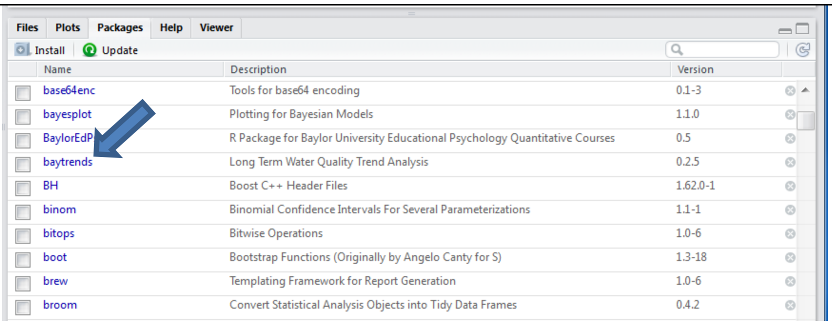


```r

library(baytrends)
```

```
Type  q()  to quit R.

[Workspace loaded from E:/Dropbox/.RData]

> library(baytrends)
Loading required package: lubridate

Attaching package: ‘lubridate’

The following object is masked from ‘package:base’:

    date

Loading required package: mgcv
Loading required package: nlme
This is mgcv 1.8-17. For overview type 'help("mgcv-package")'.
Loading required package: smwrQW
Loading required package: dataRetrieval
Loading required package: smwrBase
Loading required package: smwrGraphs
This information is preliminary or provisional and is subject to revision. It is being pr
ovided to meet the need for timely best science. The information has not received final a
pproval by the U.S. Geological Survey (USGS) and is provided on the condition that neithe
r the USGS nor the U.S. Government shall be held liable for any damages resulting from th
e authorized or unauthorized use of the information. Although this software program has b
een used by the USGS, no warranty, expressed or implied, is made by the USGS or the U.S.
Government as to the accuracy and functioning of the program and related program material
 nor shall the fact of distribution constitute any such warranty, and no responsibility i
s assumed by the USGS in connection therewith.
Loading required package: smwrStats
Although this software program has been used by the U.S. Geological Survey (USGS), no war
ranty, expressed or implied, is made by the USGS or the U.S. Government as to the accurac
y and functioning of the program and related program material nor shall the fact of distr
ibution constitute any such warranty, and no responsibility is assumed by the USGS in con
nection therewith.
This information is preliminary or provisional and is subject to revision. It is being pr
ovided to meet the need for timely best science. The information has not received final a
pproval by the U.S. Geological Survey (USGS) and is provided on the condition that neithe
r the USGS nor the U.S. Government shall be held liable for any damages resulting from th
e authorized or unauthorized use of the information. Although this software program has b
een used by the USGS, no warranty, expressed or implied, is made by the USGS or the U.S.
Government as to the accuracy and functioning of the program and related program material
 nor shall the fact of distribution constitute any such warranty, and no responsibility i
s assumed by the USGS in connection therewith.

Attaching package: ‘smwrQW’

The following object is masked from ‘package:utils’:

    View

** baytrends Notice:** This software program is preliminary or provisional and is subject
 to revision. This software program is for testing only, no warranty, expressed or implie
d, is made as to the accuracy and functioning of the program and related program material
 nor shall the fact of distribution constitute any such warranty, and no responsibility i
s assumed by in connection therewith. This software is provided 'AS IS.'
> 
```

baytrends is similar to other R packages, Part 2—help files are built-in with the software.

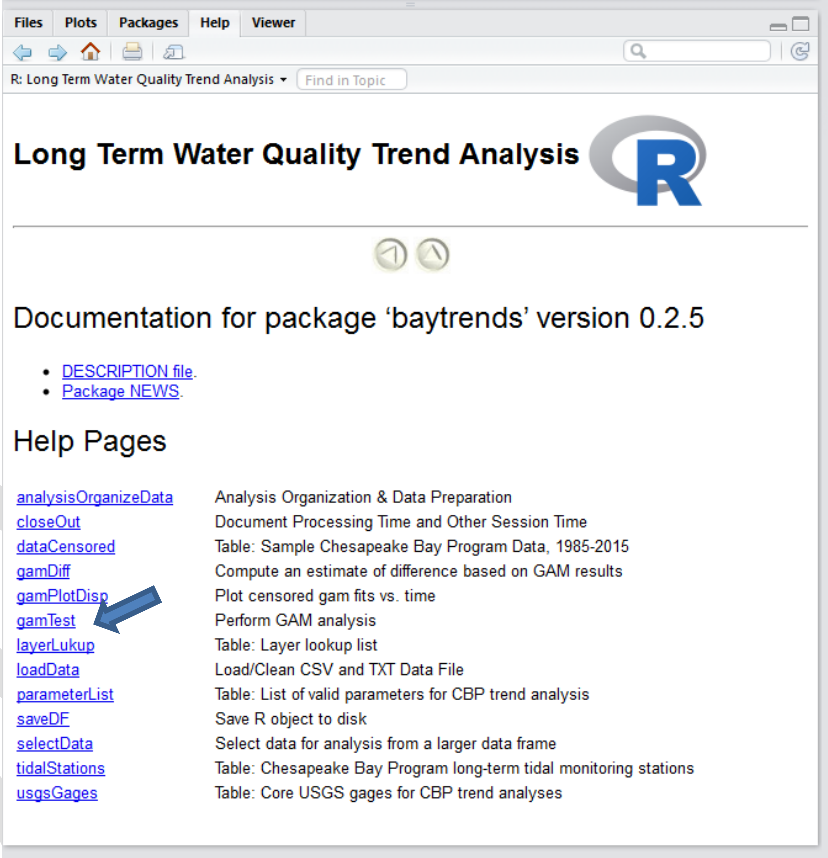
Click on baytrends from the Package tab to see the help.



	Name	Description	Version
☐	base64enc	Tools for base64 encoding	0.1-3
☐	bayesplot	Plotting for Bayesian Models	1.1.0
☐	BaylorEdP	R Package for Baylor University Educational Psychology Quantitative Courses	0.5
☐	baytrends	Long Term Water Quality Trend Analysis	0.2.5
☐	BH	Boost C++ Header Files	1.62.0-1
☐	binom	Binomial Confidence Intervals For Several Parameterizations	1.1-1
☐	bitops	Bitwise Operations	1.0-6
☐	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-18
☐	brew	Templating Framework for Report Generation	1.0-6
☐	broom	Convert Statistical Analysis Objects into Tidy Data Frames	0.4.2

From here you will see a list of functions that you can access.

Click on ‘gamTest’ to see the help for gamTest.



Long Term Water Quality Trend Analysis

Documentation for package ‘baytrends’ version 0.2.5

- DESCRIPTION file.
- Package NEWS.

Help Pages

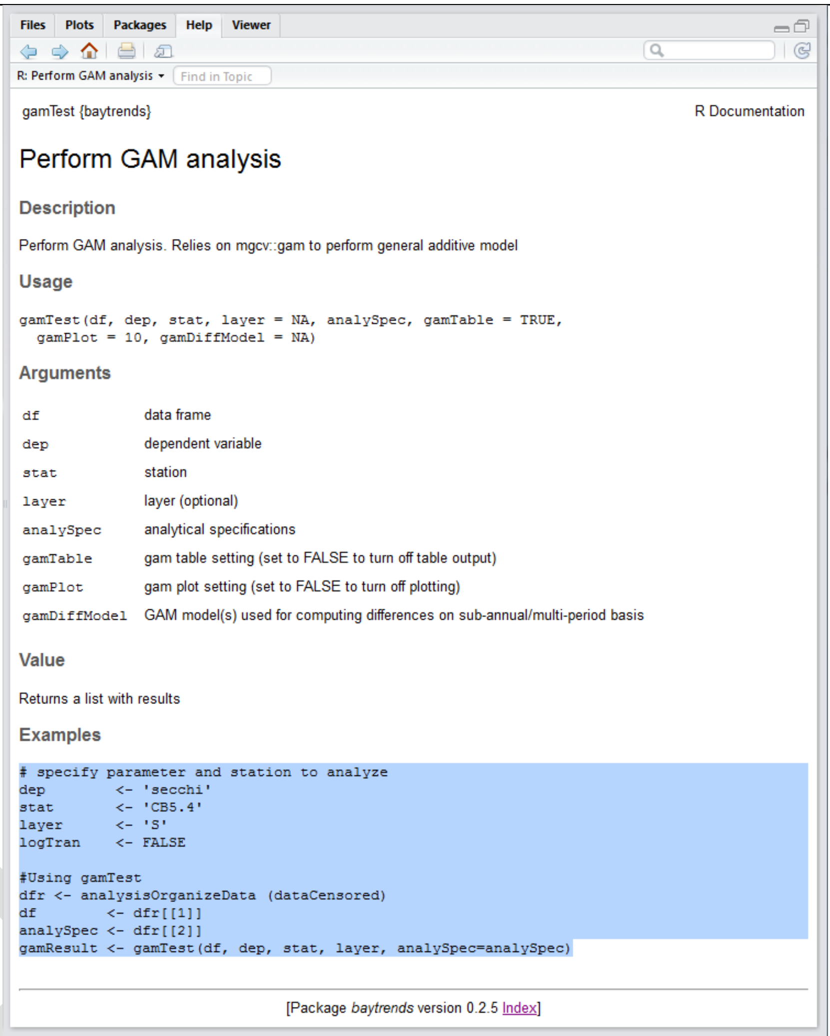
analysisOrganizeData	Analysis Organization & Data Preparation
closeOut	Document Processing Time and Other Session Time
dataCensored	Table: Sample Chesapeake Bay Program Data, 1985-2015
gamDiff	Compute an estimate of difference based on GAM results
gamPlotDisp	Plot censored gam fits vs. time
gamTest	Perform GAM analysis
layerLukup	Table: Layer lookup list
loadData	Load/Clean CSV and TXT Data File
parameterList	Table: List of valid parameters for CBP trend analysis
saveDF	Save R object to disk
selectData	Select data for analysis from a larger data frame
tidalStations	Table: Chesapeake Bay Program long-term tidal monitoring stations
usgsGages	Table: Core USGS gages for CBP trend analyses

baytrends is similar to other R packages, Part 3

Typical help files include a brief description of the function, the usage and arguments, etc.

Perhaps the most useful aspect of the help files are the examples.

Try copying the blue highlighted code chunk from the help panel to the 'Untitled1'



gamTest {baytrends} R Documentation

Perform GAM analysis

Description

Perform GAM analysis. Relies on mgcv::gam to perform general additive model

Usage

```
gamTest(df, dep, stat, layer = NA, analySpec, gamTable = TRUE,
  gamPlot = 10, gamDiffModel = NA)
```

Arguments

`df`	data frame
`dep`	dependent variable
`stat`	station
`layer`	layer (optional)
`analySpec`	analytical specifications
`gamTable`	gam table setting (set to FALSE to turn off table output)
`gamPlot`	gam plot setting (set to FALSE to turn off plotting)
`gamDiffModel`	GAM model(s) used for computing differences on sub-annual/multi-period basis

Value

Returns a list with results

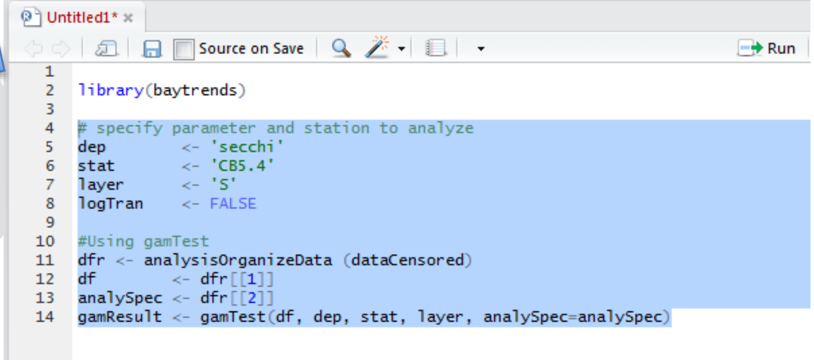
Examples

```
# specify parameter and station to analyze
dep        <- 'secchi'
stat       <- 'CB5.4'
layer      <- 'S'
logTran    <- FALSE

#Using gamTest
dfr <- analysisOrganizeData (dataCensored)
df         <- dfr[[1]]
analySpec <- dfr[[2]]
gamResult <- gamTest(df, dep, stat, layer, analySpec=analySpec)
```

[Package *baytrends* version 0.2.5 Index]

Highlight all of the rows copied from the help panel and click Run.

Explore the Console and Plots that are created.



```

library(baytrends)

# specify parameter and station to analyze
dep        <- 'secchi'
stat       <- 'CB5.4'
layer      <- 'S'
logTran    <- FALSE

#Using gamTest
dfr <- analysisOrganizeData (dataCensored)
df         <- dfr[[1]]
analySpec <- dfr[[2]]
gamResult <- gamTest(df, dep, stat, layer, analySpec=analySpec)
```

Explore the help file to see what the function analysisOrganizeData does—its explanation is a bit longer than the above function. (It is a workhorse for prepping data and doing initial housekeeping to get ready for running the GAM analysis.)

3.0 Trend Analyses with R Markdown Scripts (.Rmd)

R scripts are simple files with lines of code. You already started writing an R script by following the instructions in Section 2. It's not terribly different from other programming languages—lines of code with occasional comment lines to remind the programmer what the code is doing at key places. The usual challenge is that simple code tends to generate output tables and plots, but then there is a lot of backend work integrating tables and graphics into a report, as well as documentation.

There are a couple strategies for implementing analyses that can yield integrated reports that document the performed analyses. Our strategy is to use R markdown. In the approach used here, we rely on existing R packages including knitr and pander. There are certainly other options such as Sweave and LaTeX. Our review of tools led us to believe that R markdown and knitr were an improvement over Sweave and had a lower learning curve than LaTeX. The following website has some helpful information if you want to delve into R markdown syntax:

http://rmarkdown.rstudio.com/authoring_basics.html

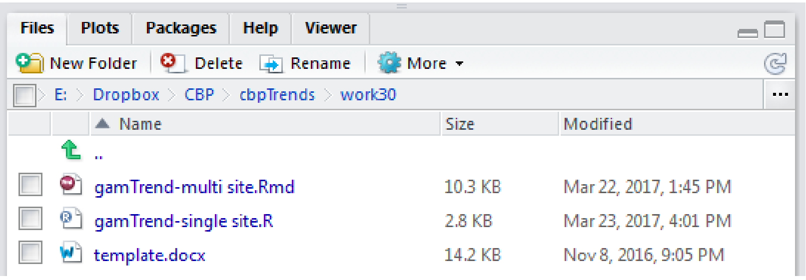
The key advantage of R markdown is that you can integrate written text, tables, graphs and even the code used to do the analysis into one report—documented and reproducible. The process of creating these integrated reports is sometimes referred to as knitting, e.g., 'knit a Word file'. While it is possible to run individual lines in an R markdown script, it is usually intended to 'knit' the entire program at one time. (See Section 4.0 for companion R script.)

The absolute best way to get started with R markdown script is to compare the script with the generated output. So that is where we will start.

3.1 GAM Analysis

To begin, open R Studio and navigate to the **cbpTrends/Work30** folder.

We generally envision that you will set up additional folders (e.g., Work31, Work32, …) as needed to store/manage different analysis tasks. That is, it will likely be a good practice as a beginning R user to store the R markdown script that you run in the same folder as you generate the output.



Name	Size	Modified
..		
gamTrend-multi site.Rmd	10.3 KB	Mar 22, 2017, 1:45 PM
gamTrend-single site.R	2.8 KB	Mar 23, 2017, 4:01 PM
template.docx	14.2 KB	Nov 8, 2016, 9:05 PM

Next, open **gamTrend-multi site.Rmd** in R Studio. (Refer to Section 1.0 to learn how to navigate among folders in R Studio and open a file.) We wrote the R markdown script to be somewhat self-explanatory and instructive.

When you open **gamTrend-multi site.Rmd**, you should also notice a new button (Knit Word) appear in the top left pane in R Studio. After we modify the code to account for where you unzipped the provided files, this will be the button to click and 'knit' a Word document. (You can also Knit HTML files. An option for Knit PDF also exists, but requires additional software that we didn't include in the instructions.)

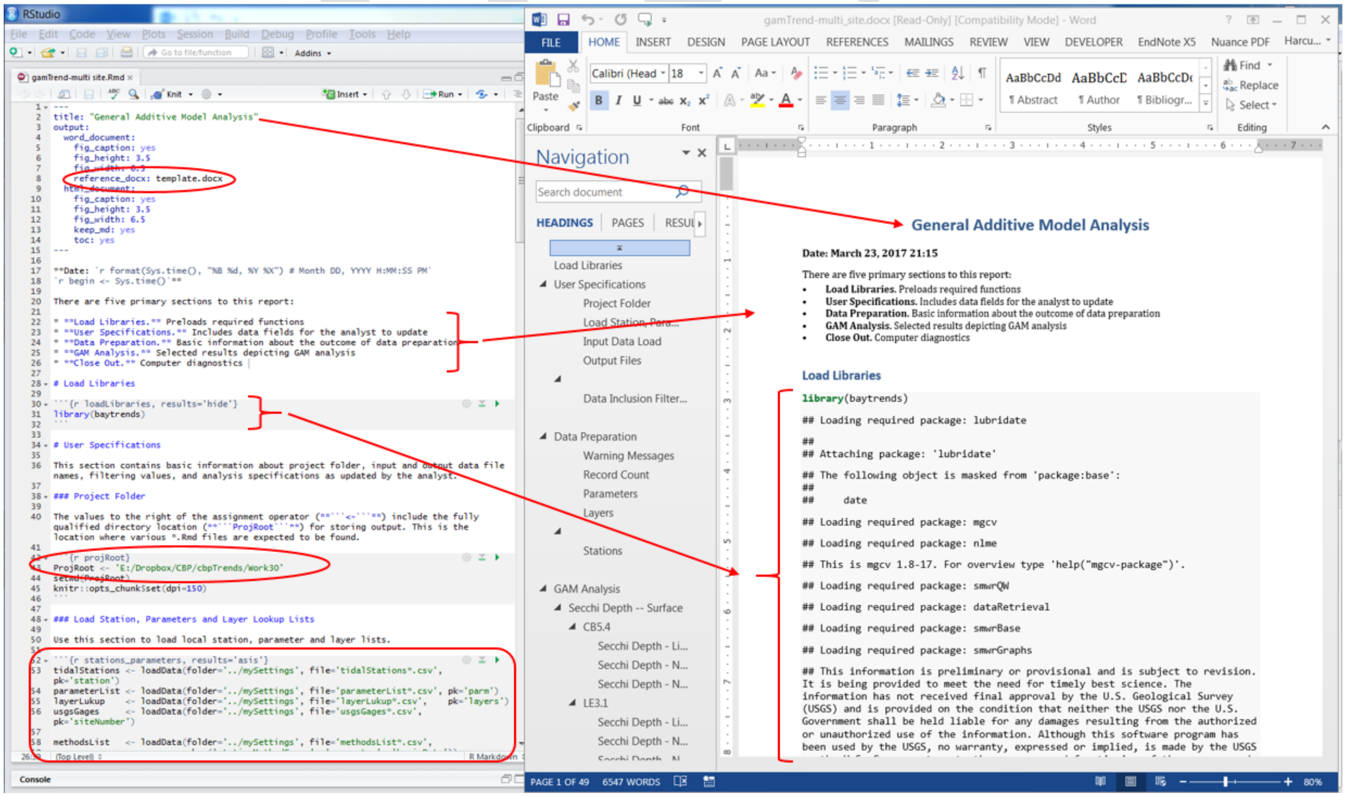
For now, however, simply click on **gamTrend-multi site.docx**. This Word document was created using **gamTrend-multi site.Rmd**.

In the below graphic, we have overlaid the output MS Word file alongside the Rmd script. It is instructive to compare the Rmd script on the left to the MS Word output. (For ease of reading, you will probably find it easier to do this on your computer monitor than try to read the below graphic in detail.)

Some things to note:

reference_docx: template.docx – **The Rmd script is expecting the word file 'template.docx' to be in the project folder**. The file, **template.docx**, includes some customization in the styles for headers, font style, and spacing. You can change it if you like, but the included template.docx works pretty well for the scripts that were included with baytrends.

ProjRoot <- 'E:/Dropbox/CBP/cbpTrends/Work30' – This is the location of the script and the above template.docx. **You'll need to modify this line of code depending on where you installed the files under Section 1.0.** (Note, the angle of the slashes ["/"] rather than the customary back slash ["\"] used



~~~
---
title: "General Additive Model Analysis"
output:
  word_document:
    fig_caption: yes
    fig_height: 3.5
    fig_width: 6.5
    reference_docx: template.docx
  html_document:
    fig_caption: yes
    fig_height: 3.5
    fig_width: 6.5
    keep_md: yes
    toc: yes
---

**Date: `r format(Sys.time(), "%B %d, %Y %X") # Month DD, YYYY H:MM:SS PM`
`r begin <- Sys.time()`**

There are five primary sections to this report:

* **Load Libraries.** Preloads required functions
* **User Specifications.** Includes data fields for the analyst to update
* **Data Preparation.** Basic information about the outcome of data preparation
* **GAM Analysis.** Selected results depicting GAM analysis
* **Close Out.** Computer diagnostics

# Load Libraries

```{r loadLibraries, results='hide'}
library(baytrends)
~~~

User Specifications

This section contains basic information about project folder, input and output data file
names, filtering values, and analysis specifications as updated by the analyst.

Project Folder

The values to the right of the assignment operator (**```<-```**) include the fully
qualified directory location (**```ProjRoot```**) for storing output. This is the
location where various *.Rmd files are expected to be found.

```{r projRoot}
ProjRoot <- 'E:/Dropbox/CBP/cbpTrends/Work30'
setwd(ProjRoot)
knitr::opts_chunk$set(dpi=150)
```

Load Station, Parameters and Layer Lookup Lists

Use this section to load local station, parameter and layer lists.

```{r stations_parameters, results='asis'}
tidalStations <- loadData(folder='../mySettings', file='tidalStations*.csv',
pk='station')
parameterList <- loadData(folder='../mySettings', file='parameterList*.csv', pk='parm')
layerLukup    <- loadData(folder='../mySettings', file='layerLukup*.csv',    pk='layers')
usgsGages     <- loadData(folder='../mySettings', file='usgsGages*.csv',
pk='siteNumber')

methodsList   <- loadData(folder='../mySettings', file='methodsList*.csv',
```

General Additive Model Analysis

Date: March 23, 2017 21:15

There are five primary sections to this report:

- **Load Libraries.** Preloads required functions
- **User Specifications.** Includes data fields for the analyst to update
- **Data Preparation.** Basic information about the outcome of data preparation
- **GAM Analysis.** Selected results depicting GAM analysis
- **Close Out.** Computer diagnostics

Load Libraries

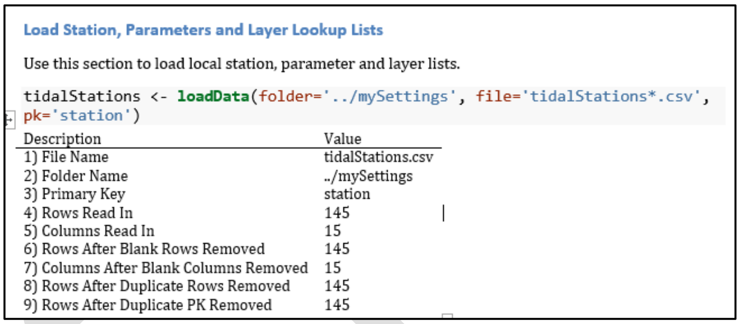
```
library(baytrends)

## Loading required package: lubridate

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-17. For overview type 'help("mgcv-package")'.

## Loading required package: smwrQW

## Loading required package: dataRetrieval

## Loading required package: smwrBase

## Loading required package: smwrGraphs

## This information is preliminary or provisional and is subject to revision.
It is being provided to meet the need for timely best science. The
information has not received final approval by the U.S. Geological Survey
(USGS) and is provided on the condition that neither the USGS nor the U.S.
Government shall be held liable for any damages resulting from the authorized
or unauthorized use of the information. Although this software program has
been used by the USGS, no warranty, expressed or implied, is made by the USGS
```

in Windows.) Once you modify the project folder location, you should be able to “knit” the document (i.e., click on Knit Word ▾).

tidalStations, parameterList, layerLukup, usgsGages – As used here, the loadData function loads user defined tables. baytrends is distributed with default tables that work with the built in data set described latter. (The **methodsList** table is optional and there is no built in table.)

The graphic shows the output associated with loading the **tidalStations** csv file.

Once you’ve taken some time to scan the language in the R markdown, you’ll probably see that we’ve added some aides to facilitate understanding what each step is doing. It serves as documentation and instruction.

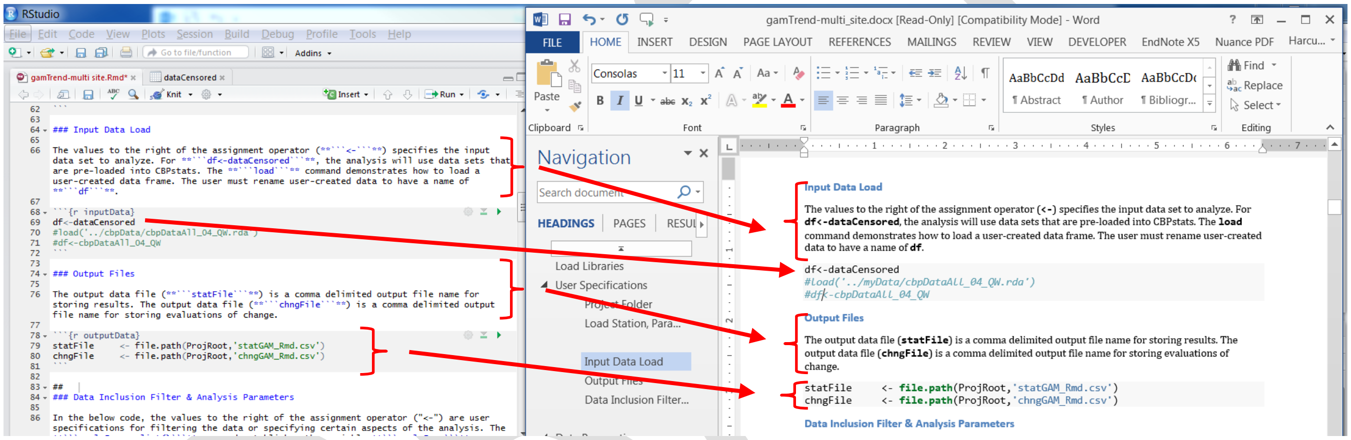


Load Station, Parameters and Layer Lookup Lists

Use this section to load local station, parameter and layer lists.

```
tidalStations <- loadData(folder='../mySettings', file='tidalStations*.csv',
pk='station')
```

Description	Value
1) File Name	tidalStations.csv
2) Folder Name	../mySettings
3) Primary Key	station
4) Rows Read In	145
5) Columns Read In	15
6) Rows After Blank Rows Removed	145
7) Columns After Blank Columns Removed	15
8) Rows After Duplicate Rows Removed	145
9) Rows After Duplicate PK Removed	145



```
```

Input Data Load

The values to the right of the assignment operator (**```<-```**) specifies the input
data set to analyze. For **```df<-dataCensored```**, the analysis will use data sets that
are pre-loaded into CBPstats. The **```load```** command demonstrates how to load a
user-created data frame. The user must rename user-created data to have a name of
```df```.

```{r inputData}
df<-dataCensored
#load('../cbpData/cbpDataAll_04_QW.rda')
#df<-cbpDataAll_04_QW
```

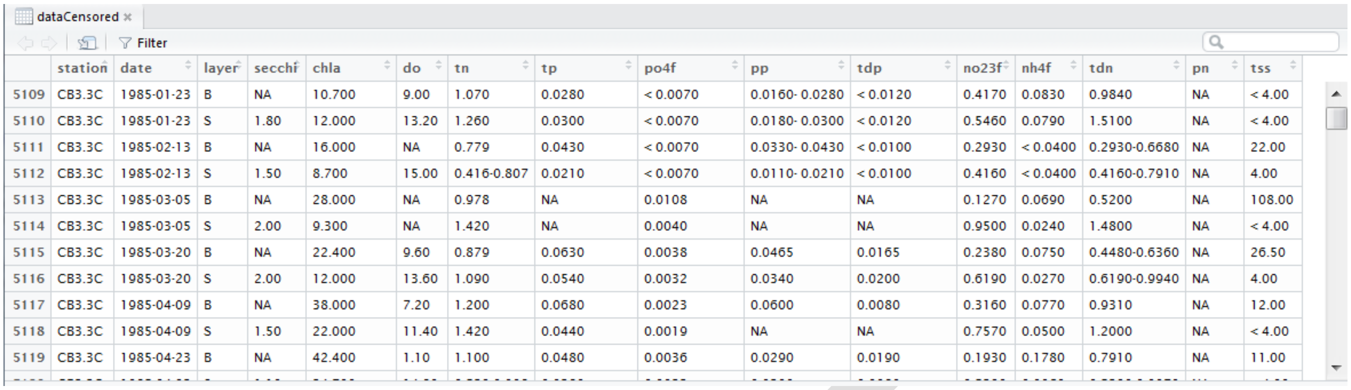
Output Files

The output data file (**```statFile```**) is a comma delimited output file name for
storing results. The output data file (**```chngFile```**) is a comma delimited output
file name for storing evaluations of change.

```{r outputData}
statFile     <- file.path(ProjRoot,'statGAM_Rmd.csv')
chngFile     <- file.path(ProjRoot,'chngGAM_Rmd.csv')
```

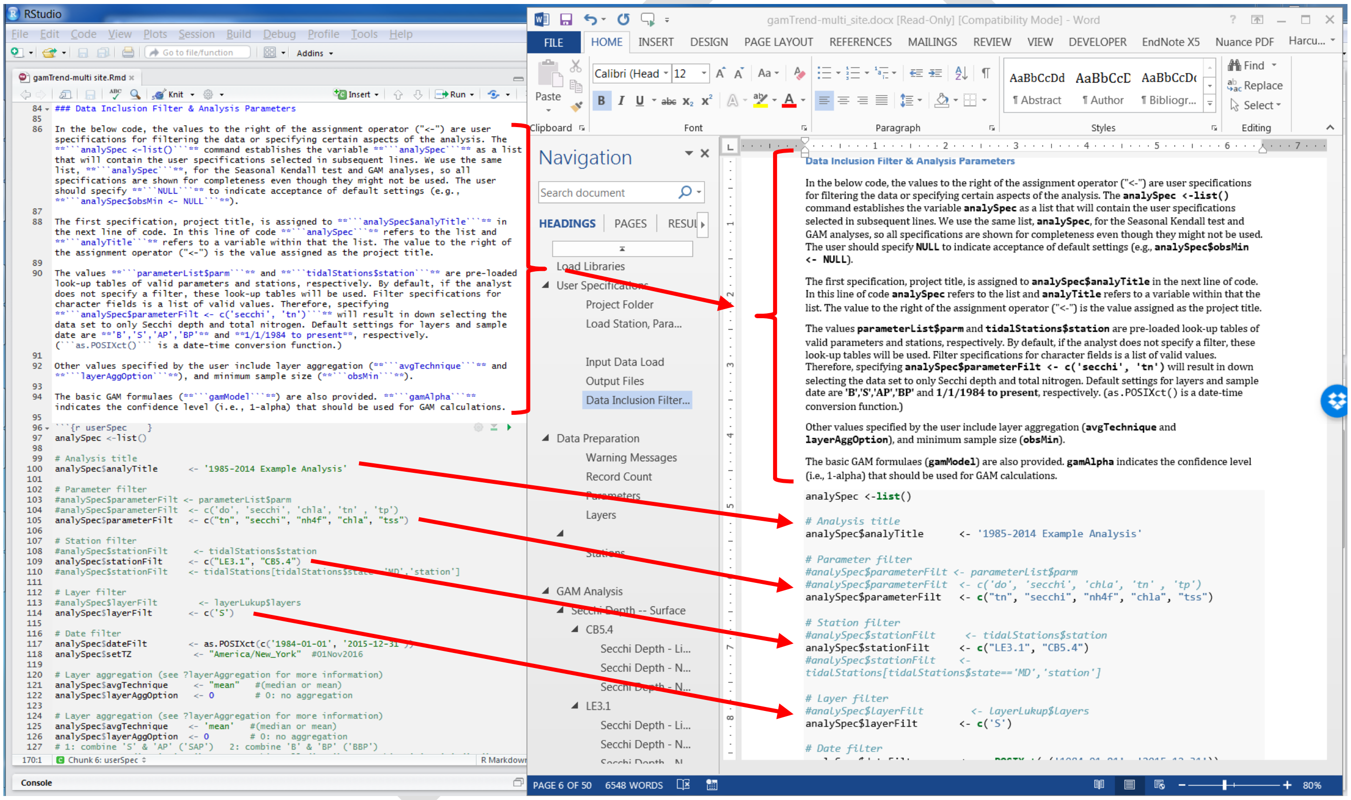
##
Data Inclusion Filter & Analysis Parameters

In the below code, the values to the right of the assignment operator ("<-") are user
specifications for filtering the data or specifying certain aspects of the analysis. The
```

Input Data Load

The values to the right of the assignment operator (`<-`) specifies the input data set to analyze. For **`df<-dataCensored`**, the analysis will use data sets that are pre-loaded into CBPstats. The **`load`** command demonstrates how to load a user-created data frame. The user must rename user-created data to have a name of **`df`**.

```
df<-dataCensored
#load('../myData/cbpDataALL_04_QW.rda')
#df<-cbpDataALL_04_QW
```

Output Files

The output data file (**`statFile`**) is a comma delimited output file name for storing results. The output data file (**`chngFile`**) is a comma delimited output file name for storing evaluations of change.

```
statFile     <- file.path(ProjRoot,'statGAM_Rmd.csv')
chngFile     <- file.path(ProjRoot,'chngGAM_Rmd.csv')
```

Data Inclusion Filter & Analysis Parameters

The next section of the Rmd script identifies the input and output data files. As shown in the below graphic, the code loads a built-in data set called **dataCensored** and assigns it to a variable, **df**. The variable df is short for ‘data frame’ which is analogous to a sheet or table of data. The next two lines of code are commented and not executed, but show how the user code can load their own data set. If these lines were run, they would result in loading a data file (with more than 120,000 records) available in the cbpData folder and copy it to the variable df.

It’s worthwhile to examine the structure of the loaded data. You can look at the online help by entering **?dataCensored** at the Console prompt to get a list of variable names. You can also browse the data by entering **View(dataCensored)** at the console. The first few lines of data are shown below. Much of the data is similar to what you might find in a spreadsheet. What separates this data structure from a typical spreadsheet is the handling of censored data.
```



| | station | date | layer | secchi | chla | do | tn | tp | po4f | pp | tdp | no23f | nh4f | tdn | pn | tss |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5109 | CB3.3C | 1985-01-23 | B | NA | 10.700 | 9.00 | 1.070 | 0.0280 | < 0.0070 | 0.0160- 0.0280 | < 0.0120 | 0.4170 | 0.0830 | 0.9840 | NA | < 4.00 |
| 5110 | CB3.3C | 1985-01-23 | S | 1.80 | 12.000 | 13.20 | 1.260 | 0.0300 | < 0.0070 | 0.0180- 0.0300 | < 0.0120 | 0.5460 | 0.0790 | 1.5100 | NA | < 4.00 |
| 5111 | CB3.3C | 1985-02-13 | B | NA | 16.000 | NA | 0.779 | 0.0430 | < 0.0070 | 0.0330- 0.0430 | < 0.0100 | 0.2930 | < 0.0400 | 0.2930-0.6680 | NA | 22.00 |
| 5112 | CB3.3C | 1985-02-13 | S | 1.50 | 8.700 | 15.00 | 0.416-0.807 | 0.0210 | < 0.0070 | 0.0110- 0.0210 | < 0.0100 | 0.4160 | < 0.0400 | 0.4160-0.7910 | NA | 4.00 |
| 5113 | CB3.3C | 1985-03-05 | B | NA | 28.000 | NA | 0.978 | NA | 0.0108 | NA | NA | 0.1270 | 0.0690 | 0.5200 | NA | 108.00 |
| 5114 | CB3.3C | 1985-03-05 | S | 2.00 | 9.300 | NA | 1.420 | NA | 0.0040 | NA | NA | 0.9500 | 0.0240 | 1.4800 | NA | < 4.00 |
| 5115 | CB3.3C | 1985-03-20 | B | NA | 22.400 | 9.60 | 0.879 | 0.0630 | 0.0038 | 0.0465 | 0.0165 | 0.2380 | 0.0750 | 0.4480-0.6360 | NA | 26.50 |
| 5116 | CB3.3C | 1985-03-20 | S | 2.00 | 12.000 | 13.60 | 1.090 | 0.0540 | 0.0032 | 0.0340 | 0.0200 | 0.6190 | 0.0270 | 0.6190-0.9940 | NA | 4.00 |
| 5117 | CB3.3C | 1985-04-09 | B | NA | 38.000 | 7.20 | 1.200 | 0.0680 | 0.0023 | 0.0600 | 0.0080 | 0.3160 | 0.0770 | 0.9310 | NA | 12.00 |
| 5118 | CB3.3C | 1985-04-09 | S | 1.50 | 22.000 | 11.40 | 1.420 | 0.0440 | 0.0019 | NA | NA | 0.7570 | 0.0500 | 1.2000 | NA | < 4.00 |
| 5119 | CB3.3C | 1985-04-23 | B | NA | 42.400 | 1.10 | 1.100 | 0.0480 | 0.0036 | 0.0290 | 0.0190 | 0.1930 | 0.1780 | 0.7910 | NA | 11.00 |

Note, how the first row of values for po4f and pp are “<0.0070” and “0.0160-0.0280”. The po4f value is a typical less-than value, i.e., the observation lies between 0 and 0.0070. The pp value is an interval censored value and lies between 0.0160 and 0.0280. Interval censored values can arise from computing parameters (either through summation, differencing, or averaging) when some components have some censoring. For example, what is TN if no23f and tkn are <0.10 and 1.00? We treat TN as “1.00-1.10”. As shown here, each column of water quality data is essentially represented by a lower and upper bound. Less than values, like the po4f value of “<0.0070” have a lower and upper bound of 0 and 0.0070. Single values like the first record of chla of 10.7 essentially have a lower and upper bound of 10.7. More effort is needed to create data sets of this type and the user is referred to the USGS’s package smwrQW for more information.



~~~
### Data Inclusion Filter & Analysis Parameters

In the below code, the values to the right of the assignment operator ("<-") are user
specifications for filtering the data or specifying certain aspects of the analysis. The
**```analySpec <-list()```** command establishes the variable **```analySpec```** as a list
that will contain the user specifications selected in subsequent lines. We use the same
list, **```analySpec```**, for the Seasonal Kendall test and GAM analyses, so all
specifications are shown for completeness even though they might not be used. The user
should specify **```NULL```** to indicate acceptance of default settings (e.g.,
**```analySpec$obsMin <- NULL```**).

The first specification, project title, is assigned to **```analySpec$analyTitle```** in
the next line of code. In this line of code **```analySpec```** refers to the list and
**```analyTitle```** refers to a variable within that the list. The value to the right of
the assignment operator ("<-") is the value assigned as the project title.

The values **```parameterList$parm```** and **```tidalStations$station```** are pre-loaded
look-up tables of valid parameters and stations, respectively. By default, if the analyst
does not specify a filter, these look-up tables will be used. Filter specifications for
character fields is a list of valid values. Therefore, specifying
**```analySpec$parameterFilt <- c('secchi', 'tn')```** will result in down selecting the
data set to only Secchi depth and total nitrogen. Default settings for layers and sample
date are **'B','S','AP','BP'** and **1/1/1984 to present**, respectively.
(```as.POSIXct()``` is a date-time conversion function.)

Other values specified by the user include layer aggregation (**```avgTechnique```** and
**```layerAggOption```**), and minimum sample size (**```obsMin```**).

The basic GAM formulaes (**```gamModel```**) are also provided. **```gamAlpha```**
indicates the confidence level (i.e., 1-alpha) that should be used for GAM calculations.

```{r userSpec    }
analySpec <-list()

# Analysis title
analySpec$analyTitle      <- '1985-2014 Example Analysis'

# Parameter filter
#analySpec$parameterFilt <- parameterList$parm
#analySpec$parameterFilt  <- c('do', 'secchi', 'chla', 'tn' , 'tp')
analySpec$parameterFilt   <- c("tn", "secchi", "nh4f", "chla", "tss")

# Station filter
#analySpec$stationFilt     <- tidalStations$station
analySpec$stationFilt     <- c("LE3.1", "CB5.4")
#analySpec$stationFilt    <- tidalStations[tidalStations$state=='MD','station']

# Layer filter
#analySpec$layerFilt        <- layerLukup$layers
analySpec$layerFilt       <- c('S')

# Date filter
analySpec$dateFilt        <- as.POSIXct(c('1984-01-01', '2015-12-31'))
analySpec$setTZ            <- "America/New_York"  #01Nov2016

# Layer aggregation (see ?layerAggregation for more information)
analySpec$avgTechnique     <- "mean"   #(median or mean)
analySpec$layerAggOption   <- 0         # 0: no aggregation

# Layer aggregation (see ?layerAggregation for more information)
analySpec$avgTechnique    <- 'mean'   #(median or mean)
analySpec$layerAggOption  <- 0         # 0: no aggregation
# 1: combine 'S' & 'AP' ('SAP')   2: combine 'B' & 'BP' ('BBP')
~~~

**Data Inclusion Filter & Analysis Parameters**

In the below code, the values to the right of the assignment operator ("<-") are user specifications for filtering the data or specifying certain aspects of the analysis. The **analySpec <-list()** command establishes the variable **analySpec** as a list that will contain the user specifications selected in subsequent lines. We use the same list, **analySpec**, for the Seasonal Kendall test and GAM analyses, so all specifications are shown for completeness even though they might not be used. The user should specify **NULL** to indicate acceptance of default settings (e.g., **analySpec$obsMin <- NULL**).

The first specification, project title, is assigned to **analySpec$analyTitle** in the next line of code. In this line of code **analySpec** refers to the list and **analyTitle** refers to a variable within that the list. The value to the right of the assignment operator ("<-") is the value assigned as the project title.

The values **parameterList$parm** and **tidalStations$station** are pre-loaded look-up tables of valid parameters and stations, respectively. By default, if the analyst does not specify a filter, these look-up tables will be used. Filter specifications for character fields is a list of valid values. Therefore, specifying **analySpec$parameterFilt <- c('secchi', 'tn')** will result in down selecting the data set to only Secchi depth and total nitrogen. Default settings for layers and sample date are **'B','S','AP','BP'** and **1/1/1984 to present**, respectively. (as.POSIXct() is a date-time conversion function.)

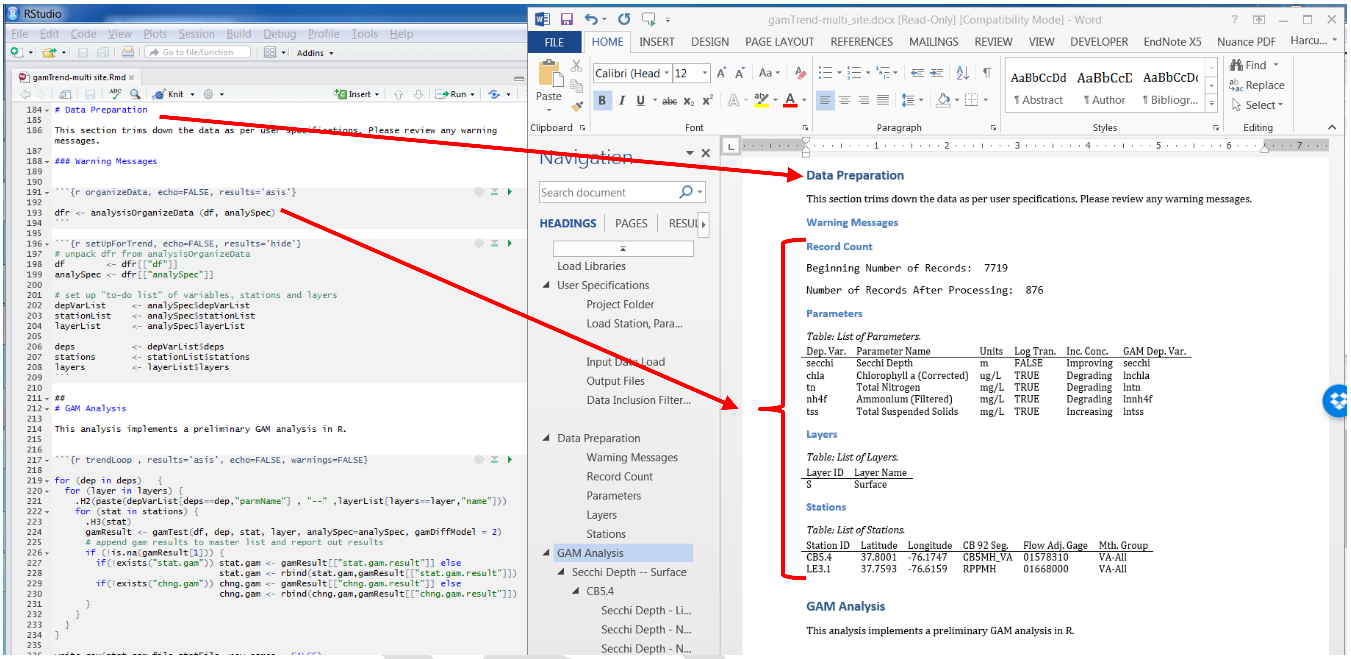
Other values specified by the user include layer aggregation (**avgTechnique** and **layerAggOption**), and minimum sample size (**obsMin**).

The basic GAM formulaes (**gamModel**) are also provided. **gamAlpha** indicates the confidence level (i.e., 1-alpha) that should be used for GAM calculations.

```
analySpec <-list()

# Analysis title
analySpec$analyTitle      <- '1985-2014 Example Analysis'

# Parameter filter
#analySpec$parameterFilt <- parameterList$parm
#analySpec$parameterFilt  <- c('do', 'secchi', 'chla', 'tn' , 'tp')
analySpec$parameterFilt   <- c("tn", "secchi", "nh4f", "chla", "tss")

# Station filter
#analySpec$stationFilt     <- tidalStations$station
analySpec$stationFilt     <- c("LE3.1", "CB5.4")
#analySpec$stationFilt    <-
tidalStations[tidalStations$state=='MD','station']

# Layer filter
#analySpec$layerFilt        <- layerLukup$layers
analySpec$layerFilt       <- c('S')

# Date filter
```

Further down in the script, you’ll uncover the “Data Inclusion Filter & Analysis Parameters” section. (You can find the section in the Word file by using the navigation pane.) This section allows you to down select your data set to the portion of data you want to analyze while maintaining a single copy of the master data file as well as making certain analysis decisions. Review the introduction to this section to learn about how to make appropriate choices.

It is very important (actually required) that all parameters that will be analyzed exist in the parameterList table. The parameterList table includes information about the full name and units of the parameter. It also includes information about whether to log transform the data for the GAM analysis. Similar to parameters, it is important (required) that all stations that will be analyzed exist in the tidalStations table. This table includes information about latitude, longitude, segments, etc.



~~~
# Data Preparation

This section trims down the data as per user specifications. Please review any warning
messages.

### Warning Messages


```{r organizeData, echo=FALSE, results='asis'}

dfr <- analysisOrganizeData (df, analySpec)
~~~

```{r setUpForTrend, echo=FALSE, results='hide'}
# unpack dfr from analysisOrganizeData
df        <- dfr[["df"]]
analySpec <- dfr[["analySpec"]]

# set up "to-do list" of variables, stations and layers
depVarList     <- analySpec$depVarList
stationList    <- analySpec$stationList
layerList      <- analySpec$layerList

deps           <- depVarList$deps
stations       <- stationList$stations
layers         <- layerList$layers
```

##
GAM Analysis

This analysis implements a preliminary GAM analysis in R.

```{r trendLoop , results='asis', echo=FALSE, warnings=FALSE}

for (dep in deps)   {
  for (layer in layers) {
    .H2(paste(depVarList[deps==dep,"parmName"] , "--" ,layerList[layers==layer,"name"]))
    for (stat in stations) {
      .H3(stat)
      gamResult <- gamTest(df, dep, stat, layer, analySpec=analySpec, gamDiffModel = 2)
      # append gam results to master list and report out results
      if (!is.na(gamResult[1])) {
        if(!exists("stat.gam")) stat.gam <- gamResult[["stat.gam.result"]] else
                                stat.gam <- rbind(stat.gam,gamResult[["stat.gam.result"]])
        if(!exists("chng.gam")) chng.gam <- gamResult[["chng.gam.result"]] else
                                chng.gam <- rbind(chng.gam,gamResult[["chng.gam.result"]])
      }
    }
  }
}

```

Data Preparation

This section trims down the data as per user specifications. Please review any warning messages.

Warning Messages

Record Count

```
Beginning Number of Records:  7719

Number of Records After Processing:  876
```

Parameters

Table: List of Parameters.

Dep. Var.	Parameter Name	Units	Log Tran.	Inc. Conc.	GAM Dep. Var.
secchi	Secchi Depth	m	FALSE	Improving	secchi
chla	Chlorophyll a (Corrected)	ug/L	TRUE	Degrading	lnchla
tn	Total Nitrogen	mg/L	TRUE	Degrading	lntn
nh4f	Ammonium (Filtered)	mg/L	TRUE	Degrading	lnnh4f
tss	Total Suspended Solids	mg/L	TRUE	Increasing	lntss

Layers

Table: List of Layers.

Layer ID	Layer Name
S	Surface

Stations

Table: List of Stations.

Station ID	Latitude	Longitude	CB 92 Seg.	Flow Adj. Gage	Mth. Group
CB5.4	37.8001	-76.1747	CB5MH_VA	01578310	VA-All
LE3.1	37.7593	-76.6159	RPPMH	01668000	VA-All

GAM Analysis

This analysis implements a preliminary GAM analysis in R.

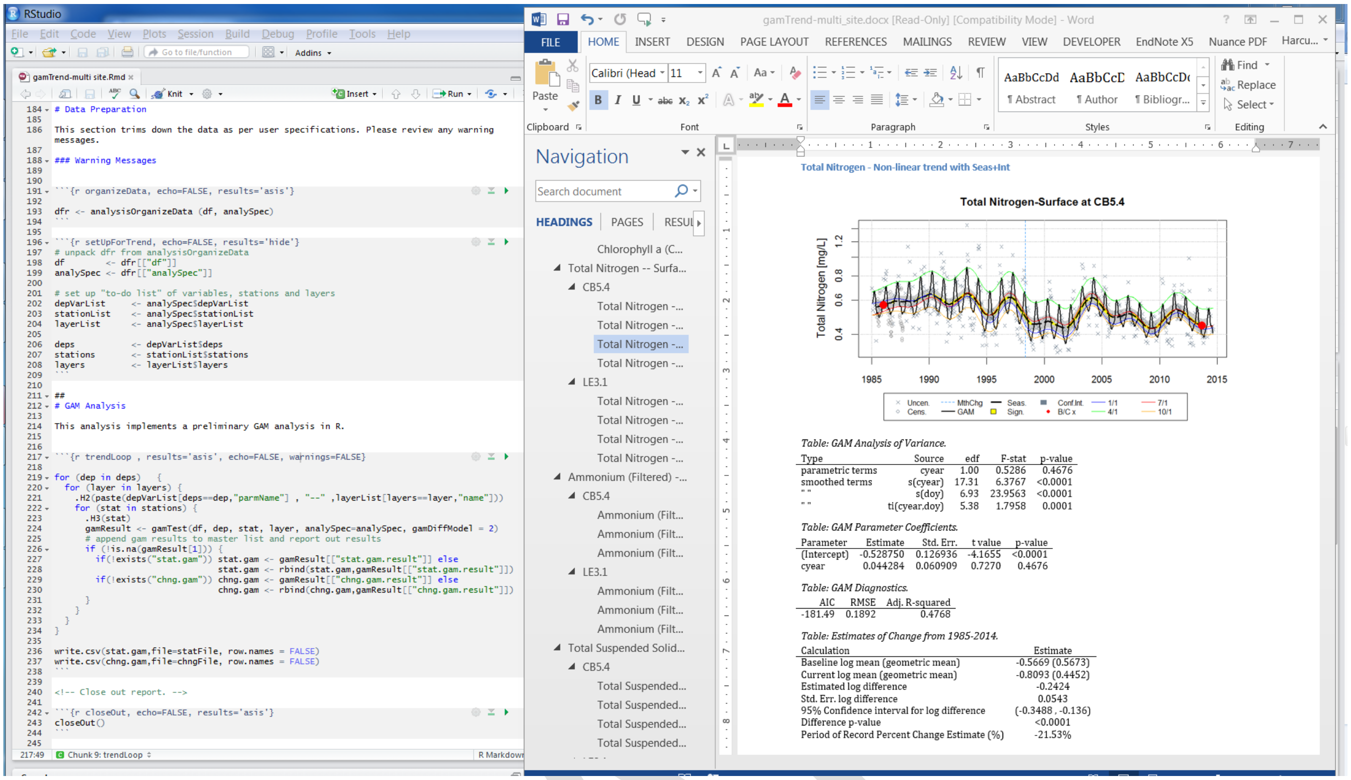
Filters can be added to down select by parameters, stations, date range, and layers. Data can be aggregated by layer. We also include options for selecting the minimum sample size, which p-levels to evaluate, and the level of detail for the output. (Review the instructions/documentation included with the script for more details.)

In the below screen capture, you'll see that we use the function analysisOrganizeData for data preparation. Besides returning a data set that has been prepared for analysis, you also get a brief report on the variables, layers and stations to be analyzed.

At this point in the script, the data are ready for analysis. We set up a simple for loop to pass through all the combinations of variables, layers and stations that were requested. The word file shows the output for total nitrogen at CB5.4.

The closeout shows information about how long it took the program to run, what versions of the R packages were running and information on who to contact with questions.

Appendix 2.1A



~~~
# Data Preparation

This section trims down the data as per user specifications. Please review any warning
messages.

### Warning Messages


```{r organizeData, echo=FALSE, results='asis'}

dfr <- analysisOrganizeData (df, analySpec)
~~~

```{r setUpForTrend, echo=FALSE, results='hide'}
# unpack dfr from analysisOrganizeData
df        <- dfr[["df"]]
analySpec <- dfr[["analySpec"]]

# set up "to-do list" of variables, stations and layers
depVarList     <- analySpec$depVarList
stationList    <- analySpec$stationList
layerList      <- analySpec$layerList

deps           <- depVarList$deps
stations       <- stationList$stations
layers         <- layerList$layers
```

##
GAM Analysis

This analysis implements a preliminary GAM analysis in R.

```{r trendLoop , results='asis', echo=FALSE, warnings=FALSE}

for (dep in deps)   {
  for (layer in layers) {
    .H2(paste(depVarList[deps==dep,"parmName"] , "--" ,layerList[layers==layer,"name"]))
    for (stat in stations) {
      .H3(stat)
      gamResult <- gamTest(df, dep, stat, layer, analySpec=analySpec, gamDiffModel = 2)
      # append gam results to master list and report out results
      if (!is.na(gamResult[1])) {
        if(!exists("stat.gam")) stat.gam <- gamResult[["stat.gam.result"]] else
                                stat.gam <- rbind(stat.gam,gamResult[["stat.gam.result"]])
        if(!exists("chng.gam")) chng.gam <- gamResult[["chng.gam.result"]] else
                                chng.gam <- rbind(chng.gam,gamResult[["chng.gam.result"]])
      }
    }
  }
}

write.csv(stat.gam,file=statFile, row.names = FALSE)
write.csv(chng.gam,file=chngFile, row.names = FALSE)
```

<!-- Close out report. -->

```{r closeOut, echo=FALSE, results='asis'}
closeOut()
```
```

Total Nitrogen - Non-linear trend with Seas+Int

*Table: GAM Analysis of Variance.*

| Type | Source | edf | F-stat | p-value |
|---|---|---|---|---|
| parametric terms | cyear | 1.00 | 0.5286 | 0.4676 |
| smoothed terms | s(cyear) | 17.31 | 6.3767 | <0.0001 |
| " " | s(doy) | 6.93 | 23.9563 | <0.0001 |
| " " | ti(cyear,doy) | 5.38 | 1.7958 | 0.0001 |

*Table: GAM Parameter Coefficients.*

| Parameter | Estimate | Std. Err. | t value | p-value |
|---|---|---|---|---|
| (Intercept) | -0.528750 | 0.126936 | -4.1655 | <0.0001 |
| cyear | 0.044284 | 0.060909 | 0.7270 | 0.4676 |

*Table: GAM Diagnostics.*

| AIC | RMSE | Adj. R-squared |
|---|---|---|
| -181.49 | 0.1892 | 0.4768 |

*Table: Estimates of Change from 1985-2014.*

| Calculation | Estimate |
|---|---|
| Baseline log mean (geometric mean) | -0.5669 (0.5673) |
| Current log mean (geometric mean) | -0.8093 (0.4452) |
| Estimated log difference | -0.2424 |
| Std. Err. log difference | 0.0543 |
| 95% Confidence interval for log difference | (-0.3488 , -0.136) |
| Difference p-value | <0.0001 |
| Period of Record Percent Change Estimate (%) | -21.53% |

In addition to the MS Word report, the program also creates two summary tables that are exported to comma delimited files: statGAM_Rmd.csv and chngGAM_Rmd.csv. A list of parameters is provided in Table 1.

Table 1. List of fields in statGAM.csv and chngGAM.csv.

| Category | Field | Description |
|---|---|---|
| Station/Parm meta | station | Station identifier |
| Station/Parm meta | dep | Parameter identifier |
| Station/Parm meta | layer | Sample layer |
| Station/Parm meta | latitude | Station latitude |
| Station/Parm meta | longitude | Station Longitude |
| Station/Parm meta | cbSeg92 | CBP segment |
| Station/Parm meta | state | State |
| Station/Parm meta | stationGrpName | Station Group |
| Station/Parm meta | parmName | Full parameter name |
| Station/Parm meta | numObservations | Number of observations evaluated |
| Station/Parm meta | yearRng | Year range of data |
| Station/Parm meta | yearBegin | Beginning year of data |
| Station/Parm meta | yearEnd | Ending year of data |
| Station/Parm meta | numYrs | Number of years of data |
| Censoring | yearRangeDropped | Year range of data removed because of too high a level of censoring |
| Censoring | fracLT | fraction observations evaluated that are “<” censored |
| Censoring | fracUnc | fraction observations evaluated that are uncensored |
| Censoring | fracInt | fraction observations evaluated that are interval censored |
```

Appendix 2.1A

Category	**Field**	**Description**
Censoring	fracRecen	fraction observations evaluated that are a negative number and were recensored
Censoring	recensor	value the negative values were recensored to
GAM model	depGAM	Parameter identifier for GAM
GAM model	logTrans	Were the data log transformed
GAM model	gamOption	GAM formula number (analySpec$gamModels)
GAM model	gamName	GAM formula name (analySpec$gamModels)
GAM model	gamSelect	Setting for select argument in mgcv::gam function (either TRUE or FALSE)
GAM model	gamK	Setting use in s(cyear, k=gamK)
GAM Coeff.	cyear.coeff	GAM parameter cyear coefficient
GAM Coeff.	cyear.pv	GAM parameter cyear p-value
GAM Coeff.	interB.label	GAM parameter intervention B coefficient
GAM Coeff.	interB.coeff	GAM parameter intervention B coefficient
GAM Coeff.	interB.pv	GAM parameter intervention B p-value
GAM Coeff.	interC.label	GAM parameter intervention C coefficient
GAM Coeff.	interC.coeff	GAM parameter intervention C coefficient
GAM Coeff.	interC.pv	GAM parameter intervention C p-value
GAM Coeff.	interD.label	GAM parameter intervention D coefficient
GAM Coeff.	interD.coeff	GAM parameter intervention D coefficient
GAM Coeff.	interD.pv	GAM parameter intervention D p-value
GAM Coeff.	interE.label	GAM parameter intervention E coefficient
GAM Coeff.	interE.coeff	GAM parameter intervention E coefficient
GAM Coeff.	interE.pv	GAM parameter intervention E p-value
GAM ANOVA	p.cyear.pv	GAM parametric cyear p-value
GAM ANOVA	s.cyear.pv	GAM smoothed cyear p-value
GAM ANOVA	s.doy.pv	GAM smoothed doy p-value
GAM ANOVA	ti.pv	GAM smoothed trend interaction [ti(cyear,doy)] p-value
GAM ANOVA	p.inter.pv	GAM parametric intervention p-value
GAM ANOVA	ti.interA.pv	GAM smoothed trend interaction, intervention A p-value
GAM ANOVA	ti.interB.pv	GAM smoothed trend interaction, intervention B p-value
GAM ANOVA	ti.interC.pv	GAM smoothed trend interaction, intervention C p-value
GAM ANOVA	ti.interD.pv	GAM smoothed trend interaction, intervention D p-value
GAM ANOVA	ti.interE.pv	GAM smoothed trend interaction, intervention E p-value
GAM ANOVA	edfMin	Minimum edf value from ANOVA table
GAM ANOVA	edfMinSource	Source of minimum edf value from ANOVA table
GAM ANOVA	FstatFlag	Indication of unreliable F-stat statistic in from ANOVA table
POR change	mn.doy	**obsolete** previously: Day of year used to construct seasonally adjusted model
POR change	sa.sig.inc	Periods of significant increases
POR change	sa.sig.dec	Periods of significant decreases
POR change	por.diffType	POR comparison using regular or adjusted mean
POR change	por.bl.mn	POR baseline mean (expressed as log value if logTrans=TRUE)
POR change	por.cr.mn	POR current mean (expressed as log value if logTrans=TRUE)
POR change	por.bl.mn.obs	POR baseline mean (observed units, geo. mean if logTrans=TRUE)
POR change	por.cr.mn.obs	POR current mean (observed units, geo. mean if logTrans=TRUE)
POR change	por.abs.chg	POR absolute change (diff. of log values if logTrans=TRUE
POR change	por.abs.chg.obs	POR absolute change (observed units)
POR change	por.pct.chg	POR percent change estimate (%)
POR change	por.chg.pv	POR change p-value
GAM fit diagnostics	aic	Akaike information criterion
GAM fit diagnostics	rmse	Root mean squared error
GAM fit diagnostics	adjR2	Adjusted R squared

Category	Field	Description
Analysis Spec.	periodName*	User-supplied period name (see analySpec$gamDiffPeriods)
Analysis Spec.	seasonName*	User-supplied season name (see analySpec$gamDiffSeasons)
Analysis Spec.	periodStart*	Start years used to compute difference (see analySpec$gamDiffPeriods)
Analysis Spec.	periodEnd*	End years used to compute difference (see analySpec$gamDiffPeriods)
Analysis Spec.	seasonMonths*	Months used to compute difference (see analySpec$gamDiffSeasons)
Customized Change	gamDiff.diffType	Comparison using regular or adjusted mean
Customized Change	gamDiff.bl.mn*	Baseline mean (expressed as log value if logTrans=TRUE)
Customized Change	gamDiff.cr.mn*	Current mean (expressed as log value if logTrans=TRUE)
Customized Change	gamDiff.bl.mn.obs*	Baseline mean (observed units, geo. mean if logTrans=TRUE)
Customized Change	gamDiff.cr.mn.obs*	Current mean (observed units, geo. mean if logTrans=TRUE)
Customized Change	gamDiff.abs.chg*	Absolute change (diff. of log values if logTrans=TRUE)
Customized Change	gamDiff.abs.chg.obs*	Absolute change (observed units)
Customized Change	gamDiff.pct.chg*	Percent change estimate (%)
Customized Change	gamDiff.chg.pval*	P value associated with absolute change

*available in chngGAM.csv output