tom.aldenberg@outlook.com

What's New in bcmfR

0.4-18.docx 2019-07-18

What's New in *bcmfR* Version 0.4-18?

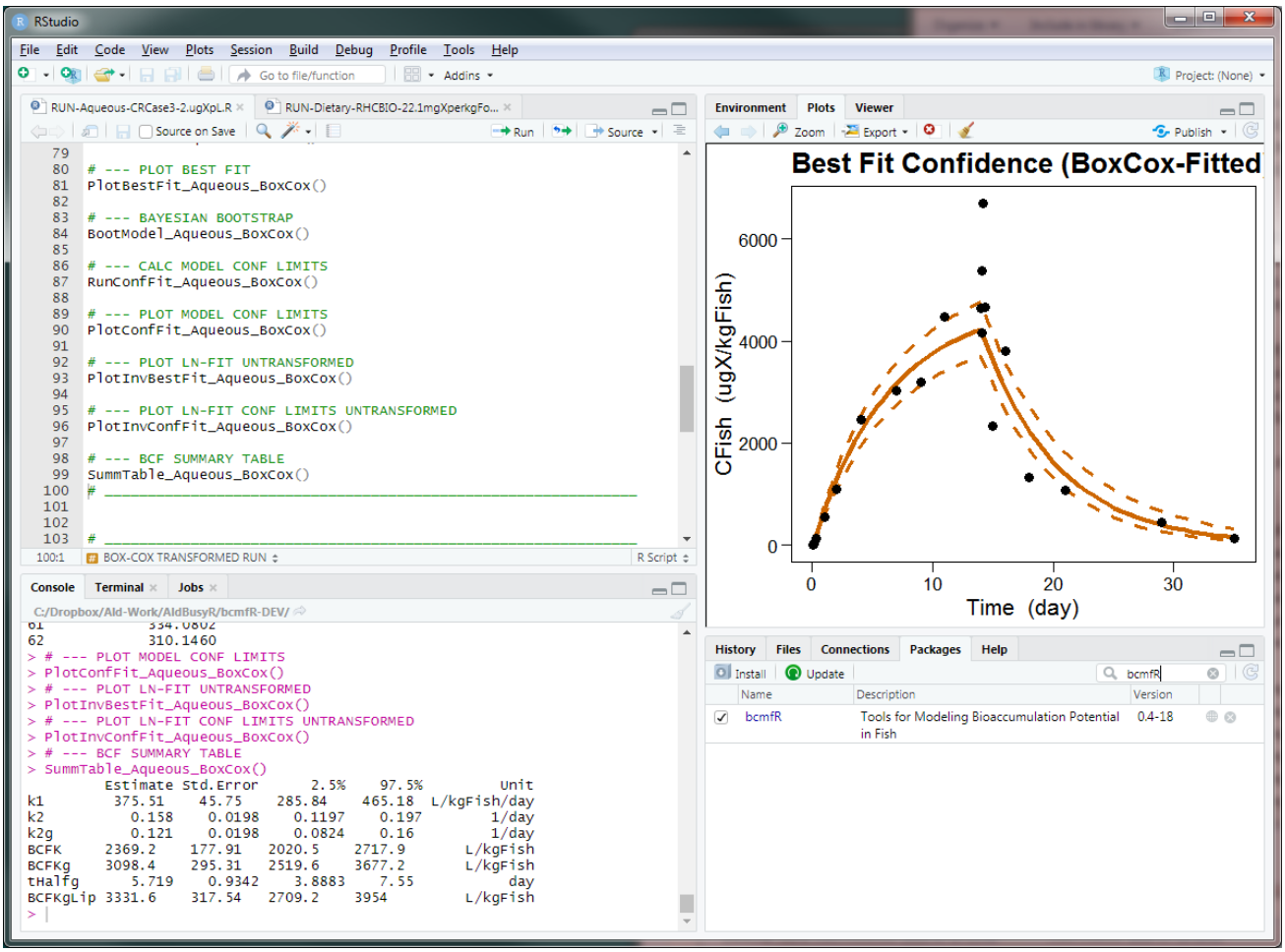
Introduction

Appreciation of the possibilities of the *bcmfR* R-package for modeling bioaccumulation and bioconcentration according to *OECD Test Guideline 305 (Bioaccumulation in Fish: Aqueous and Dietary Exposure)*, and its related *Guidance Document to OECD TG 305*, has been primarily hampered by R-upgrades and R-package incompatibilities (e.g. package rio).

Secondly, for a variety of personal reasons the maintainer was not able to provide much support, ever since version 0.3-2 appeared in late 2016. I sincerely apologize for not responding.

But, *bcmfR* is up and running again: we have solved the inconsistencies, as well as tweaked the user interface, adapted the numerical procedures, in order to make it more reliable and, hopefully, fun to apply to your datasets.

Here it is!



As you can see, the primary way of running *bcmfR* is still through the user interface provided by RStudio©. If you are not familiar with that, either learn the bare essentials to use RStudio yourself: this *is* possible, as there is no programming involved (well, you've learned to use MS-Excel© also, didn't you?), or have someone running it for you.

The road ahead may involve a CRAN R-package release, with (much) more documentation features, CRAN-checks, and possibilities to extend to Mac and Linux PCs. A web-based user interface may surface from using RStudio's Shiny©.

Model extensions could be considered also. With Martin Wiech (Ref.) writing his PhD in Norway on Cd-accumulation in North Sea Crab, I have prototyped two extensions: (1) the need to consider $k_2 = 0$, when depuration is nil, and (2), when data are noisy, to improve on the numerical convergence behind the scenes. The latter improvement landed in this *bcmfR* version 0.4-18. Now, one can fiddle in the numerical options more easily, if trouble arises.

A relatively easy future improvement might be to use robust (outlier resistant) nonlinear fitting procedures, to aid to the ubiquitous Gaussian non-robust nonlinear least squares algorithm, employed throughout the program.

Dr. Wiech also had dietary accumulation data involving measurements *during* the build-up phase, i.e. before depuration. One work-around was to use the aquatic runs, while interpreting k_1 as k_f, with appropriate unit changes. A more polished dietary version, with a genuine build-up phase, did not surface in this version.

Required Packages

As of *bcmfR* version `0.4-18`, certain functions from other R-packages are used. The following packages need to be installed before using *bcmfR*. These are:

```
car 3.0-3
csvy 0.3.0
ggplot2 3.2.0
ggthemes 4.2.0
lucid 1.7
nlstools 1.0-2
```

It may very well be that older versions also work. Note that we have abandoned package `rio`.

My current R-version is `3.6.1`. RStudio, used here, is version `1.2.1555`, which is a preview version of the `1.2` branch.

Input File Format (.csvy)

The input file formats, both aqueous and dietary types, are still `.csvy`, which is a concatenation of a YAML-part, and a CSV-part, without change so far. See Appendices.

These files are easily composed into a Notepad look-alike, or directly into the RStudio editor. Note that the indentation (no tabs!) is strict in the YAML part, e.g. 2 or 4 spaces, throughout. The CSV-part is more free-format.

Note the three dashes (---) at the beginning and end of the YAML-part. They are essential.

The tremendous advantage for all of us: risk assessors, experimentalists, risk managers, compliance testers, and so on, is that test design data, i.e. the experimental conditions, as well as some additional info to your liking, as well as the measurements *are available in one simple text file*.

We have not explained in the previous *bcmfR* version, which fields are obligatory in the YAML-part, and which ones are optional.

In the aqueous input files, everything before (here at line 23, but not obligatory put there):

```
# aqueous exposure
cwater:
    value: 2.00
    unit:  ugX/Lis optional.
```

is optional. Hence, the lines above that, e.g. species info, chemical info, pre-preparations, web-addresses, etc., is optional, as long as it conforms to YAML. This information is not read by *bcmfR*.

Similarly, in the dietary input files, everything before (again but not necessarily at line 23):

```
# food concentration and ingestion rate
cfood:
    value: 22.10
    unit: mgX/kgFood
```

is optional.

Thus one may develop a lab-style, or personal style, to annotate these data files.

The input data files include the units of test conditions, measurement variables, which carry through to the BCF/BMF derivatives. Yes, let's face it: units are a problem, as we might get confused all too easily: in group-related work, as well as in personal work.

Little provision is made to couple the units in the data files to the units in the program output. The example file for Aqueous is in ugX/L (u = micrograms, X is the chemical substance evaluated), and so is the output. But if the user would change to mg (milligrams) in the input, we risk inconsistency. The unit handling by *bcmfR* needs further review, soonest.

RUN Files (Scripts)

The *bcmfR* RUN Files initialize the program variables, read the input, fit the models, plot output, and summarize the model related parameter estimates and their BCF and BMF derivatives.

The supplied RUN Files are reproduced in the Appendices.

The individual commands are calls to functions, with, by default, *no arguments* (variables and values) between parentheses. Their structure is further simplified in this version of *bcmfR*.

One immediately observes, that quite useful calculations can be done, without any complex function invocations with loads of function arguments.

Here is the RUN sequence of a BCF model fit for the aqueous experiment, without transformation of the dependent *y*-axis (CFish over time):

```
# --- ATTACH bcmfR PACKAGE
library(bcmfR)

# --- INITIALIZE PROGRAM
InitProgram()

# --- READ AQUEOUS TEST DATA
ReadTest_Aqueous("TEST-Aqueous-CRCase3-2.0ugXperLiter.csvy")

# --- FIT BCF MODEL UNTRANSFORMED
FitModel_Aqueous()

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Aqueous()

# --- CALC BEST FIT
RunBestFit_Aqueous()

# --- PLOT BEST FIT
PlotBestFit_Aqueous()

# --- BAYESIAN BOOTSTRAP
BootModel_Aqueous()

# --- CALC MODEL CONF LIMITS
RunConfFit_Aqueous()
```

```
# --- PLOT CONF FIT
PlotConfFit_Aqueous()

# --- BCF SUMMARY TABLE
SummTable_Aqueous()
```

Lines with a hash character are comments, here explaining the line with the function call that immediately follows it.

The Appendix has an example with three such segments: Untransformed model fit, Box-Cox transformed, and Ln-transformed. It is by no means necessary to do all three.

Each time, the ModelDiagnostics plot and tests will reveal, whether a particular fit is acceptable.

The Box-Cox model fit, when successful, may indicate a useful transformation exponent, called Lambda. If it turns out be 1, approximately, then untransformed fitting may suffice. If it is near 0 (zero), a log-transform may be implied. Other powers, e.g. 0.5 (square root), or –1 (inverse values), can be helpful to arrive at a more satisfactory, and *therefore more informative*, model fit.

The *bcmfR* package will plot data and model in the transformed unit, but will also try to back-transform to the original unit, which can then be compared to the (possibly unsatisfactory) untransformed fit.

Our experience is that careless log-transformed fitting, as often done for the depuration phase data, may display unwanted deviations of model fits, caused by very low observation points, either in the uptake phase, or in the depuration phase, or both. *This may completely ruin your model!*

There are two ways to evaluate a RUN file.

Running Interactively

One is evaluating functions interactively in RStudio.

Load the .R file and evaluate line by line, or the whole script at once. It will search for the data file specified in the Working Directory.

Use RStudio menu item:

> Session > Set Working Directory > Choose Directory

When you run interactively, you may inspect the global variables (data, fit results, bootstrap samples, confidence lines, etc.) that are generated step by step.

The

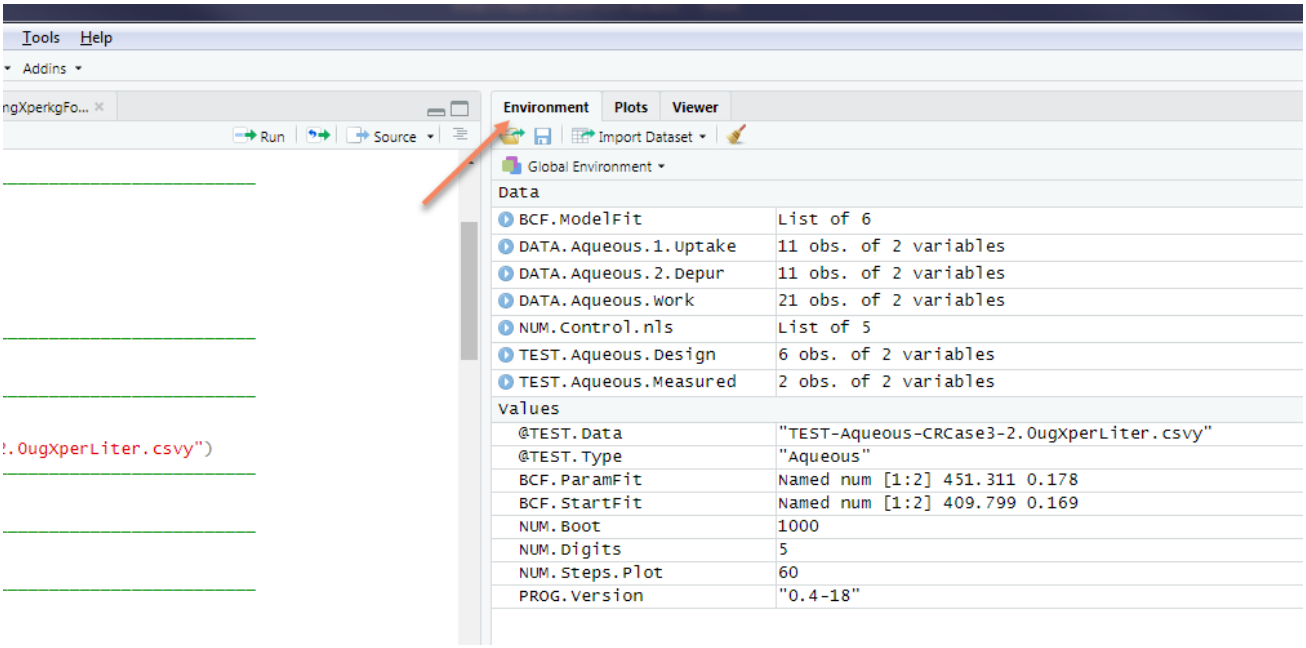
```
ReadTest_Aqueous("TEST-Aqueous-CRCase3-2.0ugXperLiter.csvy")
```

function call will pull in the `.csvy` data file. It must be present in the Session Working Directory.

The following function calls all have empty parentheses as default: `FitModel_Aqueous()`, `ModelDiagnostics_Aqueous()`, and so on.

This indicates that we have employed a lot of default inputs to these functions. They create and use, for example, global variables behind the scenes, such as working dataset: `DATA.Aqueous.Work`, `BCF.ModelFit`, among others. Note the typeface conventions used, and semantical similarities to assist the user in guessing their meaning.

You can see many of these global *user-oriented variables* under the 'Environment' tab (orange arrow) in RStudio.



These global variables contain both input data and output results, which can in turn be used as input in other procedures.

Although using global variables is considered evil by some, careful usage is very convenient and makes the analysis quite structured, and uncluttered by endless lists of function inputs.

If one wants to hack any of the global variables, e.g. by slipping in a 10-fold outlier, one may, but that is obviously very risky in the middle of a computation, as earlier results may be inconsistent.

A more potentially useful use of interactive (one by one) calling function procedures, could be to indeed provide additional arguments.

The plot function call for the inverse (back-transformed) best fit in the aqueous case reads:

```
PlotInvBestFit_Aqueous_Ln()
```

silently calling six global variables:

```
PlotInvBestFit_Aqueous_Ln <- function(testdata    = DATA.Aqueous.Work,
                                      testmeas    = TEST.Aqueous.Measured,
                                      bestfitdata = BCF.BestFit.Ln,
                                      plotstyle   = PLOT.StyleFit.Ln,
                                      plottheme   = PLOT.Theme,
                                      plottitle   = PLOT.Title.BestFit.Ln,
                                      plotxlab    = NULL,
                                      plotylab    = NULL)
```

When the plot x and y-label variable are NULL, the internal routine tries to derive them from the input file units, such as: “CFish (ugX/kgFish)”.

If these do not suffice, one may supply one’s own labels. Either by editing the ggplot2 outputs in an interactive ggplot2 editor, or, one may want to supply `plotxlab`, `plotylab` arguments in the RUN file Plot functions.

Thus, as a work-around one may insert one’s own axis labels:

```
PlotInvBestFit_Aqueous_Ln (plotxlab = “my xlabel”, plotylab = “my ylabel”)
```

to tweak the plot labeling.

When updating the package, we might rethink this, when necessary.

Saving a Session for Future Use

When done, interactively, one way to save results is by saving the whole session.

Use RStudio menu item:

> Session > Save Workspace As

which will save the whole session as an `.RData` file.

Loading a New Data File

Presently, invoking a new

> `ReadTest(`*`filename.csvy`*`)`

after a previous analysis, be it Aqueous or Dietary, will automatically save the session to a file named

> **`@TEST.LAST-Aqueous.RData`**, or
>
> **`@TEST.LAST-Dietary.RData`**, respectively,

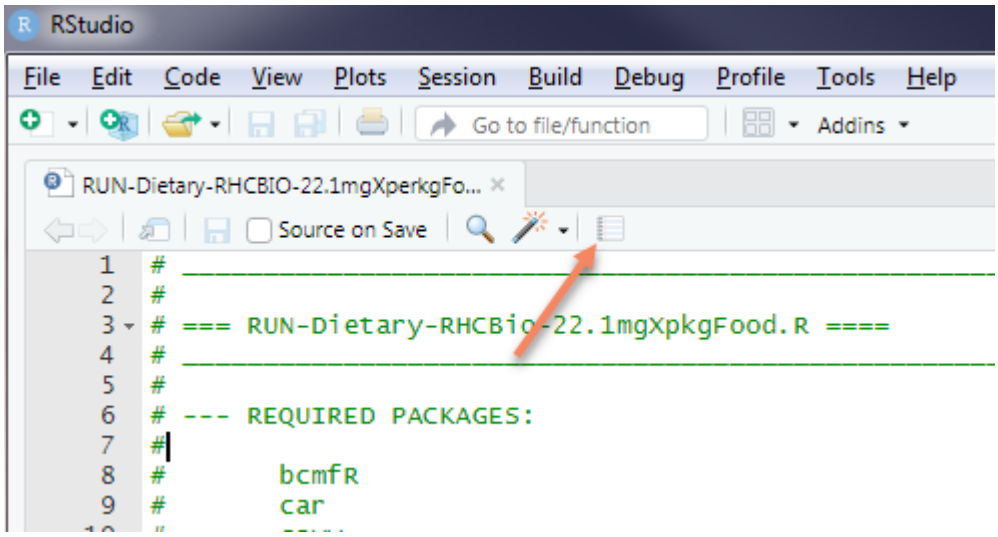
depending on the current analysis.

Previous **`.RData`** files of the same type are overwritten.

We are thinking of something more sophisticated.

Running as a Notebook (Reproducible Computing)

However, if things seem to work well, one may try, no I would say: *must try* the Notebook button (orange arrow below) to compile a complete input-output report with the whole analysis.



This revolutionary procedure is called *reproducible computing*. We have tried to include as much useful and informative output: numerical, graphical, so as to completely document the data and the model(s) applied.

There are three possible output formats for the report: HTML, PDF, or MS Word. The MS Word Notebook outputs, for both aqueous and dietary cases, is included in the Appendices here.

A small inconvenience in the Notebook output is that the four-panel ModelDiagnostics graphs may get transported, so as to not exactly follow their invocations (function calls).

Note the accompanying statistics: the Shapiro-Wilk test (important) and Runs test (somewhat less important).

Aqueous Data file format (.csvy)

```
---
# Aqueous Exposure Fish Test
filename: Test-Aqueous-CRCase3-2.0ugXperLiter.csvy

# test experiment
type: aqueous exposure fish test
guideline: OECD 305

# where data came from
data source:
    name: CR
    origin:
    comments:

# test chemical and biological species
chemical:
    name: CRCase3
    chemical info:
species:
    name: Danio
    biological info:

# aqueous exposure
cwater:
    value: 2.00
    unit:  ugX/L

# time events
tstart:
    value: 0.00
    unit:  day
tdepur:
    value: 14.00
    unit:  day
tend:
    value: 35.00
    unit:  day

# fish parameters (both assumed)
kgrowth:
    value: 0.0373
    unit:  1/day
lipidfish:
    value: 4.65
    unit:  percent

# measured timeseries
time:
    name: Time
    unit: day
cfish:
    name: CFish
    unit: ugX/kgFish
---
Time,CFish
0.04,10.50
0.08,7.73
0.17,24.12
0.33,128.80
1.00,553.70
2.00,1105.47
4.00,2464.88
7.00,3025.53
9.00,3195.05
11.00,4485.04
14.00,4652.28
14.04,4167.07
14.08,5385.64
14.17,6692.33
14.33,4674.34
15.00,2329.99
16.00,3797.43
18.00,1328.29
21.00,1080.29
29.00,438.57
35.00,128.83
```

Aqueous RUN File Example (.R)

```
# ____________________________________________________________________
#
# === RUN-Aqueous-CRCase3-2.ugXperLiter.R ====
# ____________________________________________________________________
#
# --- REQUIRED PACKAGES
#
#       car
#       csvy
#       ggplot2
#       ggthemes
#       lucid
#       nlstools
# ____________________________________________________________________
#
# --- ATTACH bcmfR PACKAGE
library(bcmfR)
#
#
# --- INITIALIZE PROGRAM
InitProgram()
# ____________________________________________________________________
#
#
# ____________________________________________________________________
#
# --- READ AQUEOUS TEST DATA
ReadTest_Aqueous("TEST-Aqueous-CRCase3-2.0ugXperLiter.csvy")
# ____________________________________________________________________


# ____________________________________________________________________
#
# === UNTRANSFORMED RUN ====
# ____________________________________________________________________

# --- FIT BCF MODEL UNTRANSFORMED
FitModel_Aqueous()

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Aqueous()

# --- CALC BEST FIT
RunBestFit_Aqueous()

# --- PLOT BEST FIT
PlotBestFit_Aqueous()

# --- BAYESIAN BOOTSTRAP
BootModel_Aqueous()

# --- CALC MODEL CONF LIMITS
RunConfFit_Aqueous()

# --- PLOT CONF FIT
PlotConfFit_Aqueous()

# --- BCF SUMMARY TABLE
SummTable_Aqueous()
# ____________________________________________________________________


# ____________________________________________________________________
#
# === BOX-COX TRANSFORMED RUN ====
# ____________________________________________________________________

# --- BOX-COX PLOT AND LAMBDA ESTIMATE
ModelTrans_BoxCox_Aqueous()

# --- FIT BCF MODEL BoxCox-TRANSFORMED
FitModel_Aqueous_BoxCox()

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Aqueous_BoxCox()

# --- CALC BEST FIT
RunBestFit_Aqueous_BoxCox()

# --- PLOT BEST FIT
PlotBestFit_Aqueous_BoxCox()
```

```
# --- BAYESIAN BOOTSTRAP
BootModel_Aqueous_BoxCox()

# --- CALC MODEL CONF LIMITS
RunConfFit_Aqueous_BoxCox()

# --- PLOT MODEL CONF LIMITS
PlotConfFit_Aqueous_BoxCox()

# --- PLOT LN-FIT UNTRANSFORMED
PlotInvBestFit_Aqueous_BoxCox()

# --- PLOT LN-FIT CONF LIMITS UNTRANSFORMED
PlotInvConfFit_Aqueous_BoxCox()

# --- BCF SUMMARY TABLE
SummTable_Aqueous_BoxCox()
# ____________________________________________________________


# ____________________________________________________________
#
# === LN-TRANSFORMED RUN ====
# ____________________________________________________________

# --- FIT BCF MODEL LN-TRANSFORMED
FitModel_Aqueous_Ln()

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Aqueous_Ln()

# --- CALC BEST FIT
RunBestFit_Aqueous_Ln()

# --- PLOT BEST FIT
PlotBestFit_Aqueous_Ln()

# --- BAYESIAN BOOTSTRAP
BootModel_Aqueous_Ln()

# --- CALC MODEL CONF LIMITS
RunConfFit_Aqueous_Ln()

# --- PLOT MODEL CONF LIMITS
PlotConfFit_Aqueous_Ln()

# --- PLOT LN-FIT UNTRANSFORMED
PlotInvBestFit_Aqueous_Ln()

# --- PLOT LN-FIT CONF LIMITS UNTRANSFORMED
PlotInvConfFit_Aqueous_Ln()

# --- BCF SUMMARY TABLE
SummTable_Aqueous_Ln()
# ____________________________________________________________
#
# === END of R-Script ====
# ____________________________________________________________.
```

RUN-Aqueous-CRCase3-2.0ugXperLiter.R

aldenbet

2019-07-17

```
# ____________________________________________________________________
#
# === RUN-Aqueous-CRCase3-2.0ugXperLiter.R ====
# ____________________________________________________________________
#
# --- REQUIRED PACKAGES
#
#       car
#       csvy
#       ggplot2
#       ggthemes
#       lucid
#       nlstools
# ____________________________________________________________________
#
# --- ATTACH bcmfR PACKAGE
library(bcmfR)
#
#
# --- INITIALIZE PROGRAM
InitProgram()

## 0.4-18

# ____________________________________________________________________
#
#
# ____________________________________________________________________
#
# --- READ AQUEOUS TEST DATA
ReadTest_Aqueous("TEST-Aqueous-CRCase3-2.0ugXperLiter.csvy")

## TEST.Aqueous.Design:
##             value    unit
## cwater     2.0000   ugX/L
## tstart     0.0000     day
## tdepur    14.0000     day
## tend      35.0000     day
## kgrowth    0.0373   1/day
## lipidfish  4.6500 percent
##
## TEST.Aqueous.Measured:
##        name       unit
## time   Time        day
## cfish CFish ugX/kgFish
##
## DATA.Aqueous.Work:
##    time.data cfish.data
```

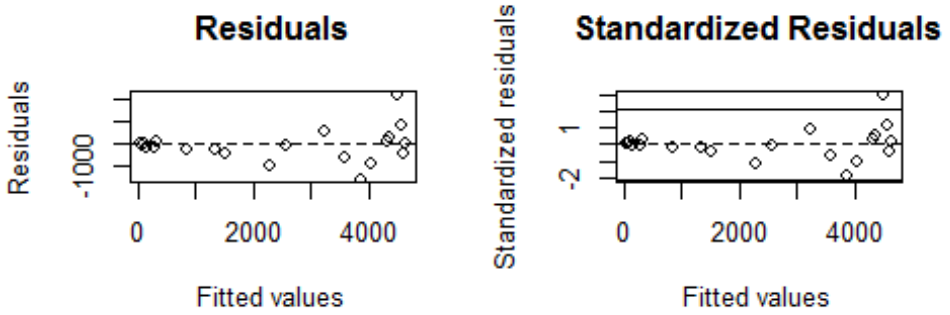
```
## 1       0.04      10.50
## 2       0.08       7.73
## 3       0.17      24.12
## 4       0.33     128.80
## 5       1.00     553.70
## 6       2.00    1105.47
## 7       4.00    2464.88
## 8       7.00    3025.53
## 9       9.00    3195.05
## 10     11.00    4485.04
## 11     14.00    4652.28
## 12     14.04    4167.07
## 13     14.08    5385.64
## 14     14.17    6692.33
## 15     14.33    4674.34
## 16     15.00    2329.99
## 17     16.00    3797.43
## 18     18.00    1328.29
## 19     21.00    1080.29
## 20     29.00     438.57
## 21     35.00     128.83

# _____________________________________________________________________


# _____________________________________________________________________
#
# === UNTRANSFORMED RUN ====
# _____________________________________________________________________

# --- FIT BCF MODEL UNTRANSFORMED
FitModel_Aqueous()

##
## Formula: cfish.data ~ RunModel_Aqueous(time.data, cwater, tdepur, fitk1,
##     fitk2)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## fitk1 451.31053   79.04969   5.709 1.67e-05 ***
## fitk2   0.17842    0.03793   4.704 0.000154 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 753.8 on 19 degrees of freedom
##
## Number of iterations to convergence: 3
## Achieved convergence tolerance: 2.88e-06

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Aqueous()

##
## ------
##  Shapiro-Wilk normality test
```

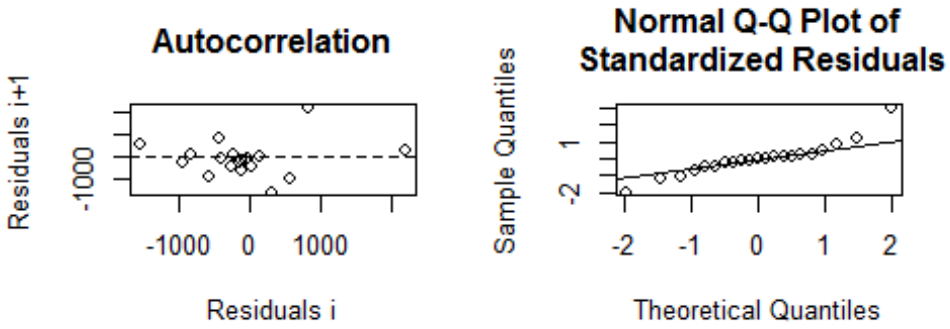
```
##
## data:  stdres
## W = 0.89165, p-value = 0.02419
##
##
## ------

##
##  Runs Test
##
## data:  as.factor(run)
## Standard Normal = -1.1114, p-value = 0.2664
## alternative hypothesis: two.sided
```

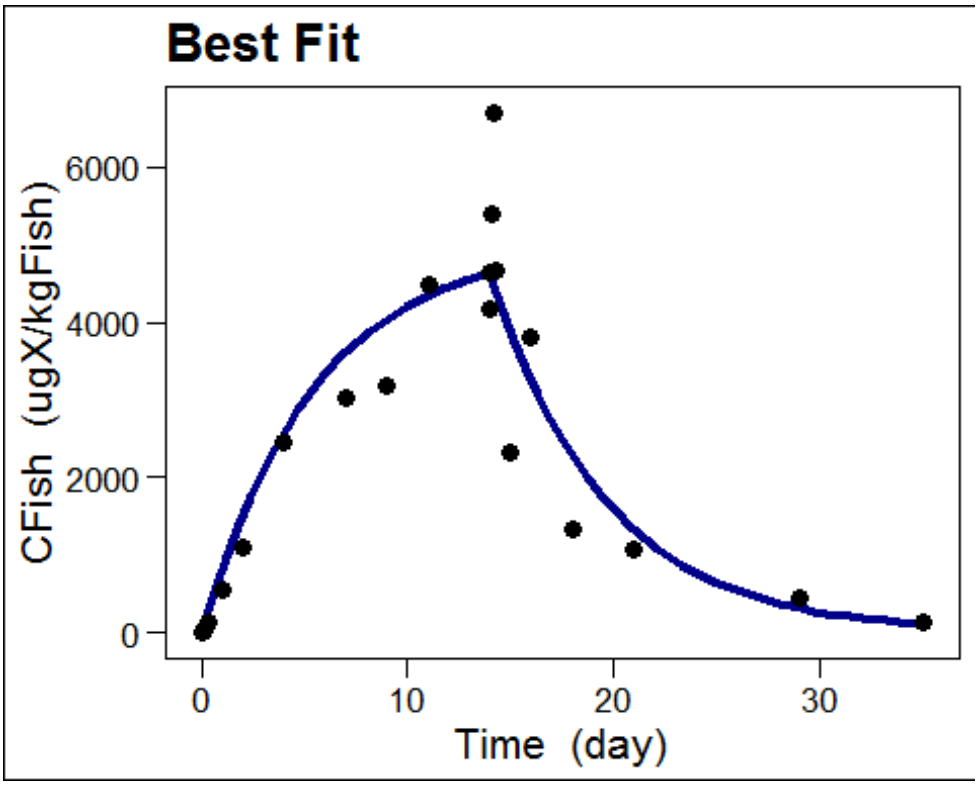
```
# --- CALC BEST FIT
RunBestFit_Aqueous()
```

```
##    time.best cfish.best
## 1  0.0400000   35.97631
## 2  0.6226667  531.93661
## 3  1.2053333  978.92617
## 4  1.7880000 1381.78033
## 5  2.3706667 1744.85697
## 6  2.9533333 2072.08371
## 7  3.5360000 2367.00033
## 8  4.1186667 2632.79710
## 9  4.7013333 2872.34931
## 10 5.2840000 3088.24832
##    ...
##    time.best cfish.best
## 53  29.75600   279.1777
## 54  30.33867   251.6119
## 55  30.92133   226.7680
## 56  31.50400   204.3771
## 57  32.08667   184.1970
## 58  32.66933   166.0096
## 59  33.25200   149.6179
## 60  33.83467   134.8447
## 61  34.41733   121.5303
## 62  35.00000   109.5305
```

```
# --- PLOT BEST FIT
PlotBestFit_Aqueous()
```

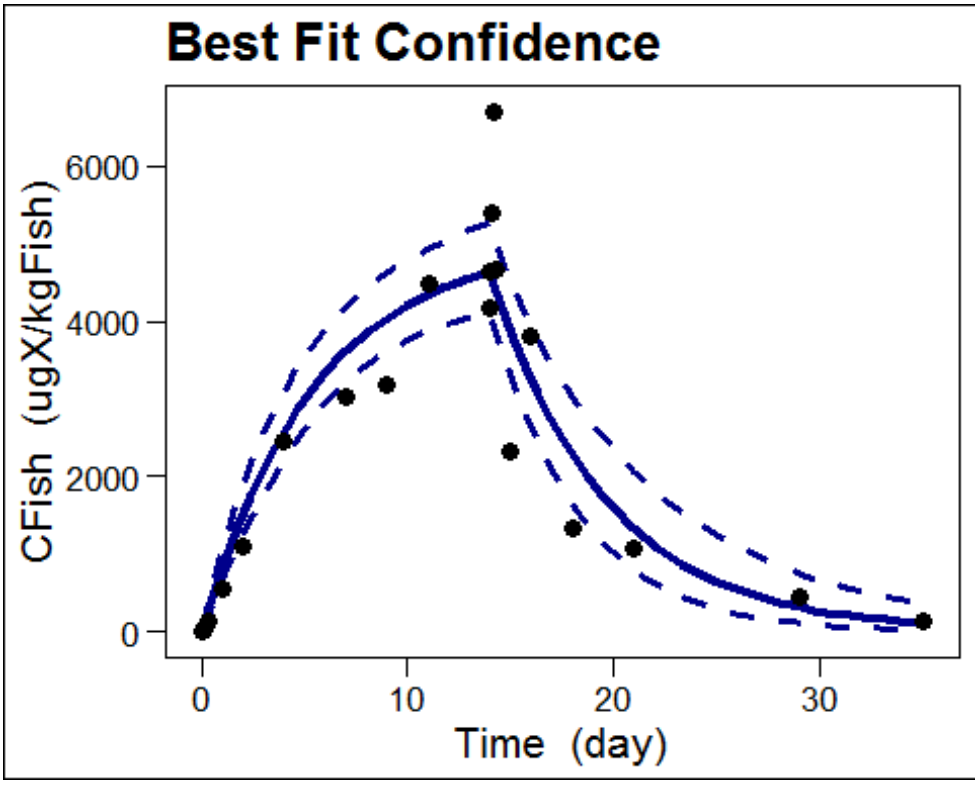







```
# --- BAYESIAN BOOTSTRAP
BootModel_Aqueous()

##      k1.boot   k2.boot
## 1  451.3105 0.1784222
## 2  415.6563 0.1656369
## 3  406.9074 0.1550974
## 4  487.9328 0.2214305
```

```
## 5  455.9869 0.1920577
## 6  493.0304 0.1876778
## 7  533.9926 0.1901586
## 8  449.5715 0.1851126
## 9  432.4899 0.1834000
## 10 656.0436 0.2955020
##    ...
##       k1.boot   k2.boot
## 991  405.2125 0.1451861
## 992  463.4106 0.1956812
## 993  353.0592 0.1352558
## 994  379.8802 0.1431932
## 995  529.7077 0.2112232
## 996  530.8682 0.2143236
## 997  426.1981 0.1724907
## 998  497.8224 0.1874416
## 999  474.7211 0.2020617
## 1000 407.6598 0.1474865

# --- CALC MODEL CONF LIMITS
RunConfFit_Aqueous()

##    time.conf cfish.conflo cfish.confup
## 1  0.0400000     28.77678      47.7433
## 2  0.6226667    429.97907     692.6163
## 3  1.2053333    801.09744    1252.6793
## 4  1.7880000   1143.81004    1746.8215
## 5  2.3706667   1461.47378    2170.9418
## 6  2.9533333   1752.81932    2550.3880
## 7  3.5360000   2018.64075    2873.3269
## 8  4.1186667   2261.09251    3151.1166
## 9  4.7013333   2486.62487    3412.3708
## 10 5.2840000   2693.96898    3635.7870
##    ...
##    time.conf cfish.conflo cfish.confup
## 53  29.75600     92.49266     687.1148
## 54  30.33867     80.10543     638.7077
## 55  30.92133     69.37718     593.9835
## 56  31.50400     60.08574     552.9552
## 57  32.08667     52.02310     515.0171
## 58  32.66933     44.99249     479.5157
## 59  33.25200     38.91213     445.3080
## 60  33.83467     33.65358     414.0497
## 61  34.41733     29.10574     385.6745
## 62  35.00000     25.15527     359.0002

# --- PLOT CONF FIT
PlotConfFit_Aqueous()
```

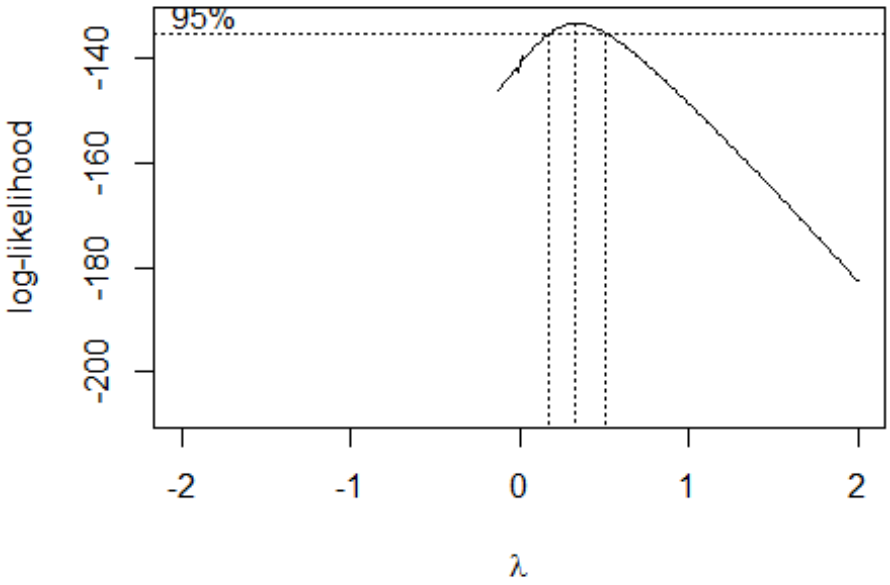



```
# --- BCF SUMMARY TABLE
SummTable_Aqueous()
```

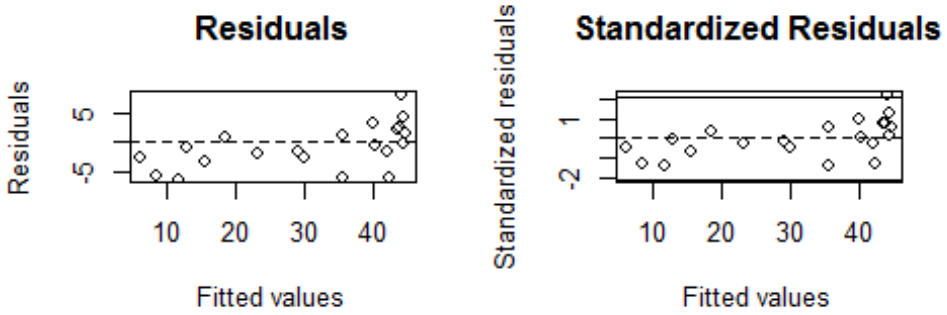
```
##          Estimate Std.Error      2.5%    97.5%         Unit
## k1        451.31    79.05    296.37    606.25  L/kgFish/day
## k2          0.178    0.0379    0.1041    0.253        1/day
## k2g         0.141    0.0379    0.0668    0.215        1/day
## BCFK     2529.5    179.68   2177.3    2881.6       L/kgFish
## BCFKg    3198      369.8    2473.2    3922.8       L/kgFish
## tHalfg      4.912    1.3201    2.3243    7.499          day
## BCFKgLip 3438.7    397.64   2659.4    4218.1       L/kgFish
```

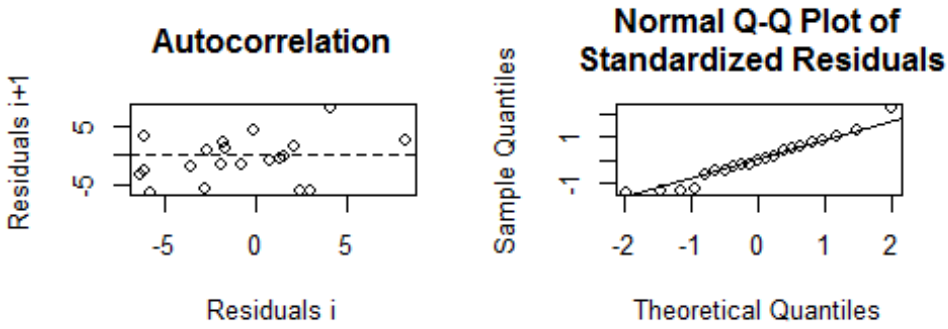
```
# ____________________________________________________________
```

```
# ____________________________________________________________
#
# === BOX-COX TRANSFORMED RUN ====
# ____________________________________________________________

# --- BOX-COX PLOT AND LAMBDA ESTIMATE
ModelTrans_BoxCox_Aqueous()
```




```
##       fit    conflo    confup
## 0.3300000 0.1743490 0.5114643

# --- FIT BCF MODEL BoxCox-TRANSFORMED
FitModel_Aqueous_BoxCox()

##
## Formula: bc.cfish.data ~ car::bcPower(RunModel_Aqueous(time.data, cwater,
##     tdepur, fitk1, fitk2) + lnstarter, lambda)
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## fitk1 375.5078    45.7500   8.208 1.14e-07 ***
## fitk2   0.1585     0.0198   8.007 1.66e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.931 on 19 degrees of freedom
##
## Number of iterations to convergence: 3
## Achieved convergence tolerance: 1.7e-07

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Aqueous_BoxCox()
```





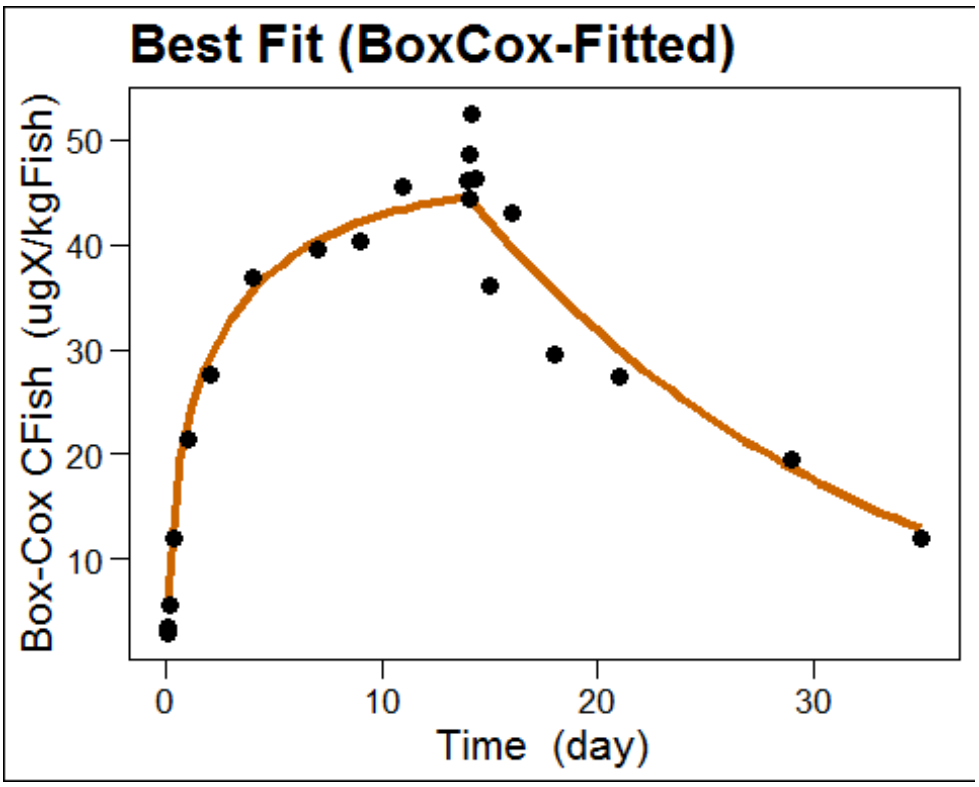

```
##
## ------
##  Shapiro-Wilk normality test
##
## data:  stdres
## W = 0.96146, p-value = 0.546
##
##
## ------

##
##  Runs Test
##
## data:  as.factor(run)
## Standard Normal = -1.1114, p-value = 0.2664
## alternative hypothesis: two.sided
```

```
# --- CALC BEST FIT
RunBestFit_Aqueous_BoxCox()
```

```
##    time.best bc.cfish.best inv.bc.cfish.best
## 1  0.0400000      6.273831           29.9456
## 2  0.6226667     19.644357          445.2979
## 3  1.2053333     24.750440          824.0105
## 4  1.7880000     28.151663         1169.3153
## 5  2.3706667     30.704226         1484.1595
## 6  2.9533333     32.731286         1771.2301
## 7  3.5360000     34.395384         2032.9772
## 8  4.1186667     35.791683         2271.6345
## 9  4.7013333     36.981492         2489.2390
## 10 5.2840000     38.007011         2687.6477
```

```
##    ...
##    time.best bc.cfish.best inv.bc.cfish.best
## 53  29.75600      17.86509          347.6199
## 54  30.33867      17.23790          316.9550
## 55  30.92133      16.62954          288.9952
## 56  31.50400      16.03943          263.5018
## 57  32.08667      15.46704          240.2573
## 58  32.66933      14.91183          219.0633
## 59  33.25200      14.37328          199.7389
## 60  33.83467      13.85090          182.1192
## 61  34.41733      13.34420          166.0538
## 62  35.00000      12.85271          151.4055
```

```
# --- PLOT BEST FIT
PlotBestFit_Aqueous_BoxCox()
```



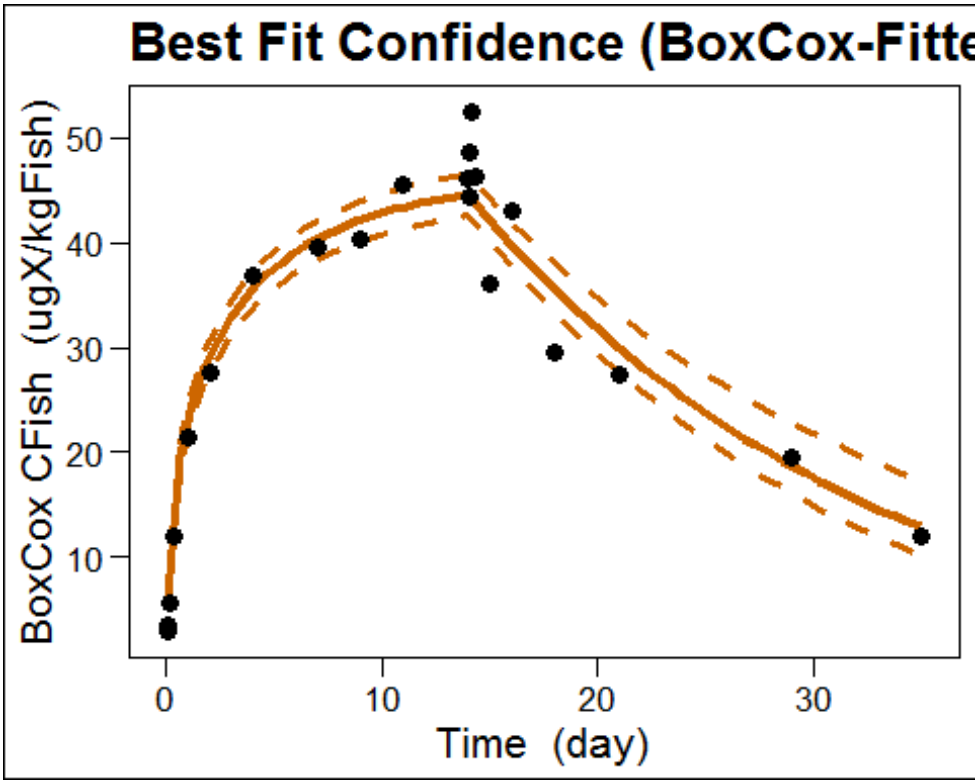

```
# --- BAYESIAN BOOTSTRAP
BootModel_Aqueous_BoxCox()
```

```
##    k1.boot.bc k2.boot.bc
## 1    375.5078  0.1584950
## 2    363.5124  0.1536890
## 3    375.5609  0.1490377
## 4    364.6923  0.1690230
## 5    395.2395  0.1769365
## 6    411.5733  0.1671984
## 7    435.2820  0.1702213
## 8    367.6380  0.1538827
## 9    337.1836  0.1579921
## 10   350.4597  0.1590535
##    ...
```

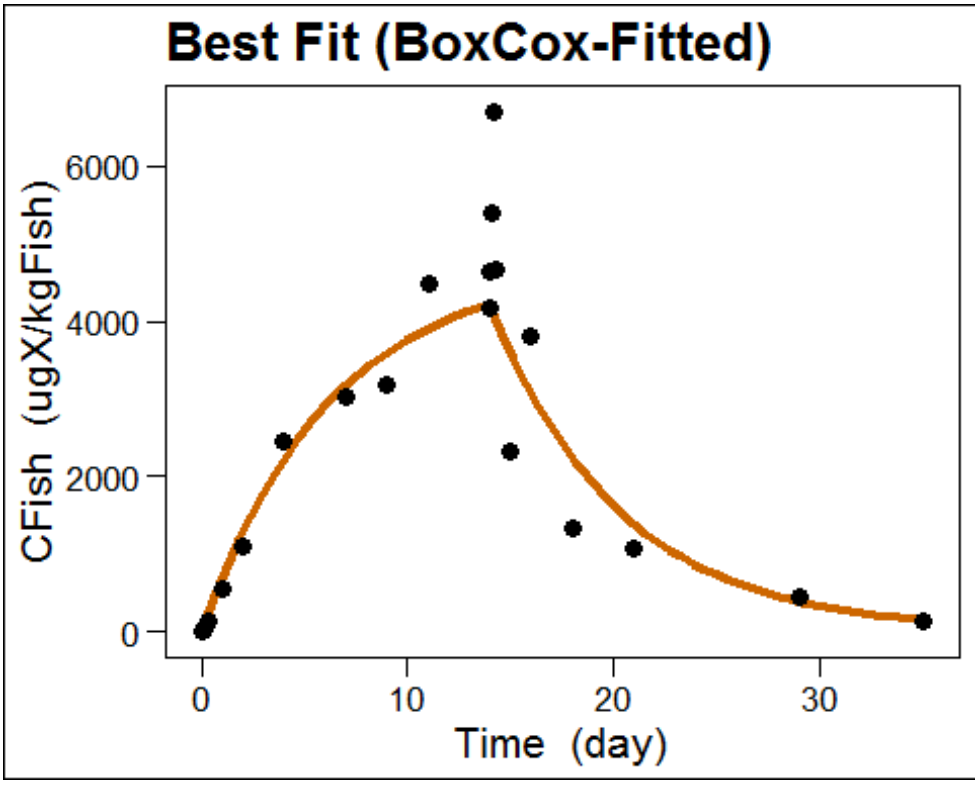
```
##      k1.boot.bc k2.boot.bc
## 991    386.1810  0.1536926
## 992    361.9360  0.1681749
## 993    356.8187  0.1513213
## 994    361.0099  0.1550951
## 995    432.3012  0.1865760
## 996    368.0308  0.1574488
## 997    340.3920  0.1500873
## 998    418.2864  0.1625546
## 999    414.1444  0.1959689
## 1000   369.0786  0.1497850

# --- CALC MODEL CONF LIMITS
RunConfFit_Aqueous_BoxCox()

##    time.conf bc.cfish.conflo bc.cfish.confup inv.bc.cfish.conflo
## 1  0.0400000        5.745852        6.756161            25.08704
## 2  0.6226667       18.382029       20.792928           374.34043
## 3  1.2053333       23.280633       26.125216           698.86717
## 4  1.7880000       26.557772       29.665626           997.42657
## 5  2.3706667       29.019045       32.263307          1270.69065
## 6  2.9533333       30.988463       34.341048          1522.37870
## 7  3.5360000       32.625229       36.039592          1755.36053
## 8  4.1186667       33.995457       37.443358          1967.85777
## 9  4.7013333       35.112361       38.659131          2153.30704
## 10 5.2840000       36.115227       39.703259          2329.49053
##    inv.bc.cfish.confup
## 1             34.90179
## 2            517.22662
## 3            953.89188
## 4           1349.97637
## 5           1701.92125
## 6           2024.04613
## 7           2315.87821
## 8           2577.33583
## 9           2819.16881
## 10          3038.61655
##    ...
##    time.conf bc.cfish.conflo bc.cfish.confup inv.bc.cfish.conflo
## 53  29.75600        15.18600        22.01702            229.36591
## 54  30.33867        14.54694        21.43114            205.84096
## 55  30.92133        13.93030        20.85897            184.72927
## 56  31.50400        13.33530        20.28643            165.78324
## 57  32.08667        12.77093        19.72034            149.05823
## 58  32.66933        12.22703        19.16800            134.04728
## 59  33.25200        11.70185        18.62907            120.54801
## 60  33.83467        11.19475        18.10343            108.40820
## 61  34.41733        10.70482        17.59293             97.48475
## 62  35.00000        10.23154        17.09317             87.65704
##    inv.bc.cfish.confup
## 53            602.0358
## 54            560.3688
## 55            521.5862
## 56            484.6196
```

```
## 57            449.8365
## 58            417.5503
## 59            387.5818
## 60            359.7747
## 61            334.0802
## 62            310.1460
```

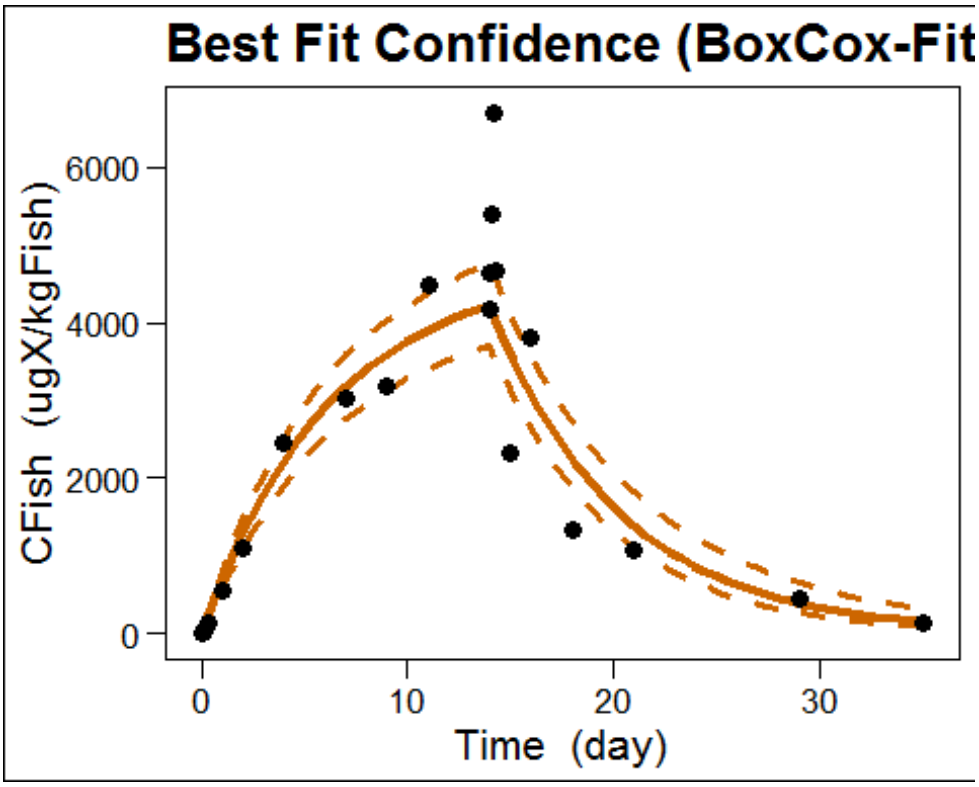
```
# --- PLOT MODEL CONF LIMITS
PlotConfFit_Aqueous_BoxCox()
```




```
# --- PLOT LN-FIT UNTRANSFORMED
PlotInvBestFit_Aqueous_BoxCox()
```



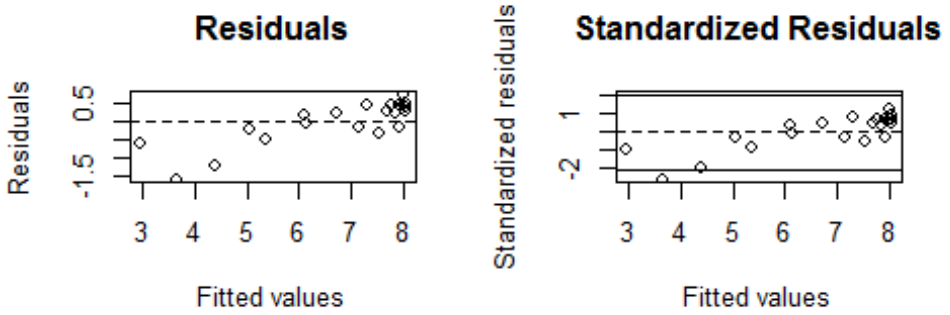

```
# --- PLOT LN-FIT CONF LIMITS UNTRANSFORMED
PlotInvConfFit_Aqueous_BoxCox()
```

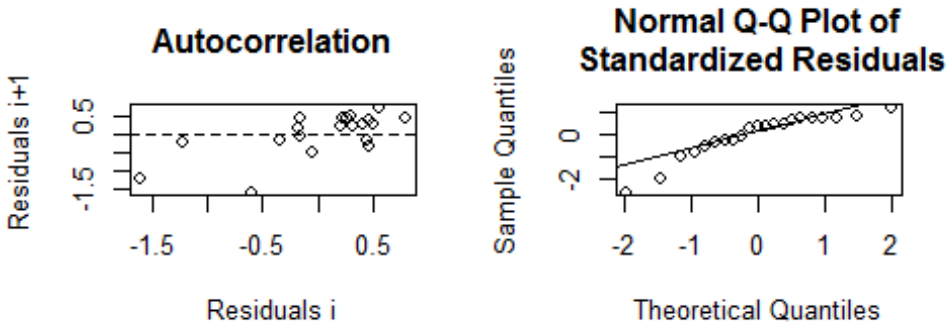



```
# --- BCF SUMMARY TABLE
SummTable_Aqueous_BoxCox()
```

```
##          Estimate Std.Error      2.5%    97.5%         Unit
## k1         375.51    45.75    285.84    465.18  L/kgFish/day
```

```
## k2          0.158    0.0198    0.1197    0.197        1/day
## k2g         0.121    0.0198    0.0824    0.16         1/day
## BCFK     2369.2    177.91   2020.5    2717.9       L/kgFish
## BCFKg    3098.4    295.31   2519.6    3677.2       L/kgFish
## tHalfg      5.719    0.9342    3.8883    7.55           day
## BCFKgLip 3331.6    317.54   2709.2    3954         L/kgFish

# ____________________________________________________________


# ____________________________________________________________
#
# === LN-TRANSFORMED RUN ====
# ____________________________________________________________

# --- FIT BCF MODEL LN-TRANSFORMED
FitModel_Aqueous_Ln()

##
## Formula: ln.cfish.data ~ log(RunModel_Aqueous(time.data, cwater, tdepur,
##     fitk1, fitk2) + lnstarter)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## fitk1 240.37455   42.88993   5.604 2.10e-05 ***
## fitk2   0.12771    0.02066   6.180 6.13e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6119 on 19 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 4.556e-07

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Aqueous_Ln()
```





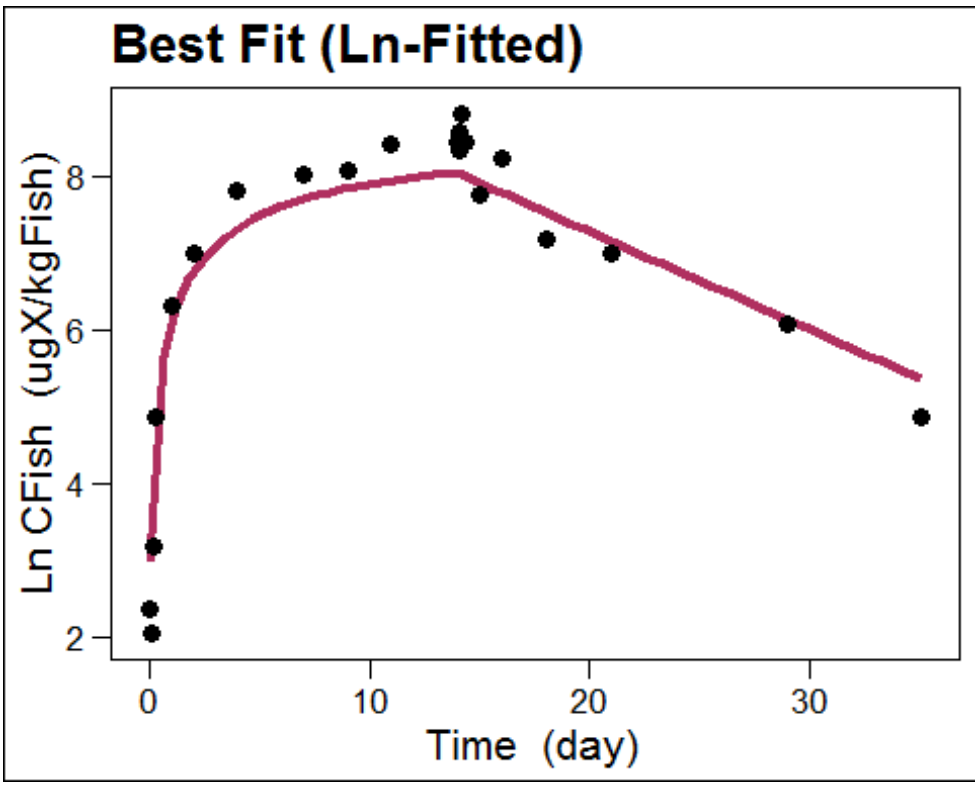

```
##
## ------
##  Shapiro-Wilk normality test
##
## data:  stdres
## W = 0.8784, p-value = 0.01364
##
##
## ------

##
##  Runs Test
##
## data:  as.factor(run)
## Standard Normal = -2.8764, p-value = 0.004023
## alternative hypothesis: two.sided
```

```
# --- CALC BEST FIT
RunBestFit_Aqueous_Ln()
```

```
##    time.best ln.cfish.best inv.ln.cfish.best
## 1  0.0400000      2.953917          19.18093
## 2  0.6226667      5.662106         287.75406
## 3  1.2053333      6.286125         537.06820
## 4  1.7880000      6.644446         768.50440
## 5  2.3706667      6.890960         983.34465
## 6  2.9533333      7.075622        1182.77902
## 7  3.5360000      7.221041        1367.91224
## 8  4.1186667      7.339388        1539.76984
## 9  4.7013333      7.437974        1699.30378
## 10 5.2840000      7.521533        1847.39778
```

```
##    ...
##    time.best ln.cfish.best inv.ln.cfish.best
## 53  29.75600      6.038155          419.1189
## 54  30.33867      5.963745          389.0645
## 55  30.92133      5.889336          361.1652
## 56  31.50400      5.814926          335.2666
## 57  32.08667      5.740517          311.2251
## 58  32.66933      5.666107          288.9076
## 59  33.25200      5.591698          268.1905
## 60  33.83467      5.517288          248.9590
## 61  34.41733      5.442879          231.1065
## 62  35.00000      5.368469          214.5342
```

```
# --- PLOT BEST FIT
PlotBestFit_Aqueous_Ln()
```



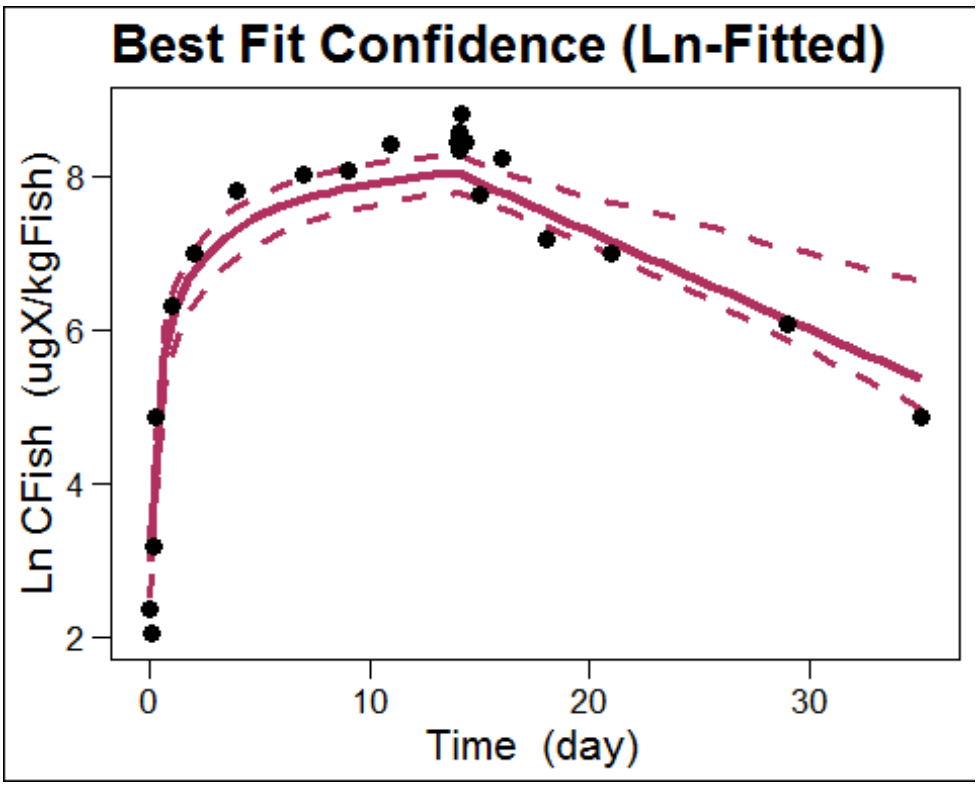

```
# --- BAYESIAN BOOTSTRAP
BootModel_Aqueous_Ln()
```

```
##    k1.boot.ln k2.boot.ln
## 1    240.3745  0.1277051
## 2    245.8037  0.1270976
## 3    294.5253  0.1296240
## 4    251.2393  0.1273047
## 5    274.7444  0.1446049
## 6    260.5584  0.1195576
## 7    260.0626  0.1308963
## 8    237.5308  0.1077665
## 9    198.4913  0.1158478
## 10   195.6149  0.1020064
##    ...
```

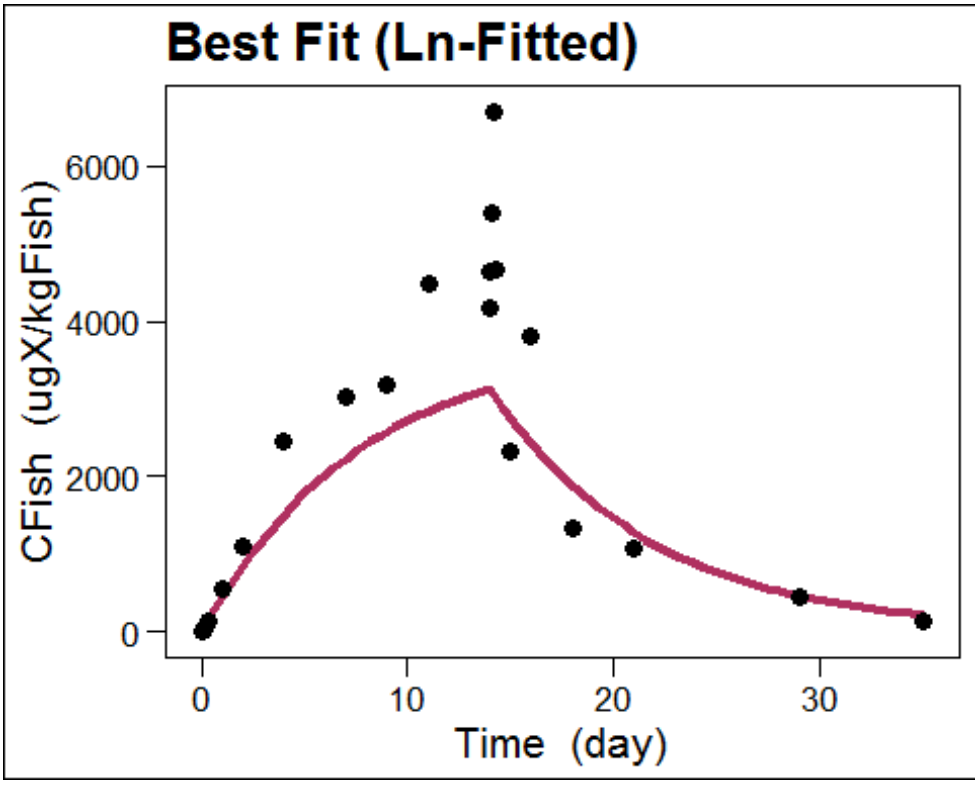
```
##      k1.boot.ln k2.boot.ln
## 991    274.6328  0.1334648
## 992    250.0139  0.1351720
## 993    292.9057  0.1452482
## 994    210.8005  0.1309656
## 995    343.1155  0.1606965
## 996    214.6261  0.1016987
## 997    233.2451  0.1304742
## 998    300.1001  0.1341678
## 999    311.4329  0.1596355
## 1000   236.6842  0.1303278

# --- CALC MODEL CONF LIMITS
RunConfFit_Aqueous_Ln()

##    time.conf ln.cfish.conflo ln.cfish.confup inv.ln.cfish.conflo
## 1  0.0400000        2.490836        3.280553            12.07137
## 2  0.6226667        5.212190        5.981075           183.49739
## 3  1.2053333        5.844183        6.600727           345.22102
## 4  1.7880000        6.213269        6.952417           499.33114
## 5  2.3706667        6.474384        7.192349           648.32146
## 6  2.9533333        6.670338        7.372192           788.66188
## 7  3.5360000        6.831160        7.511447           926.26686
## 8  4.1186667        6.963503        7.623534          1057.33865
## 9  4.7013333        7.072641        7.719095          1179.25804
## 10 5.2840000        7.168662        7.796957          1298.10688
##    inv.ln.cfish.confup
## 1             26.59047
## 2            395.86566
## 3            735.63146
## 4           1045.67622
## 5           1329.22451
## 6           1591.11701
## 7           1828.85758
## 8           2045.78018
## 9           2250.92574
## 10          2433.18980
##    ...
##    time.conf ln.cfish.conflo ln.cfish.confup inv.ln.cfish.conflo
## 53  29.75600        5.756580        7.016865            316.2649
## 54  30.33867        5.672977        6.974222            290.8998
## 55  30.92133        5.585579        6.926941            266.5547
## 56  31.50400        5.498598        6.879661            244.3504
## 57  32.08667        5.412889        6.835420            224.2787
## 58  32.66933        5.329271        6.791590            206.2877
## 59  33.25200        5.240958        6.749497            188.8509
## 60  33.83467        5.155799        6.708877            173.4343
## 61  34.41733        5.068739        6.668256            158.9738
## 62  35.00000        4.980587        6.627006            145.5606
##    inv.ln.cfish.confup
## 53           1115.3205
## 54           1068.7936
## 55           1019.4359
## 56            972.3577
```

```
## 57            930.2418
## 58            890.3303
## 59            853.6319
## 60            819.6717
## 61            787.0806
## 62            755.3207
```

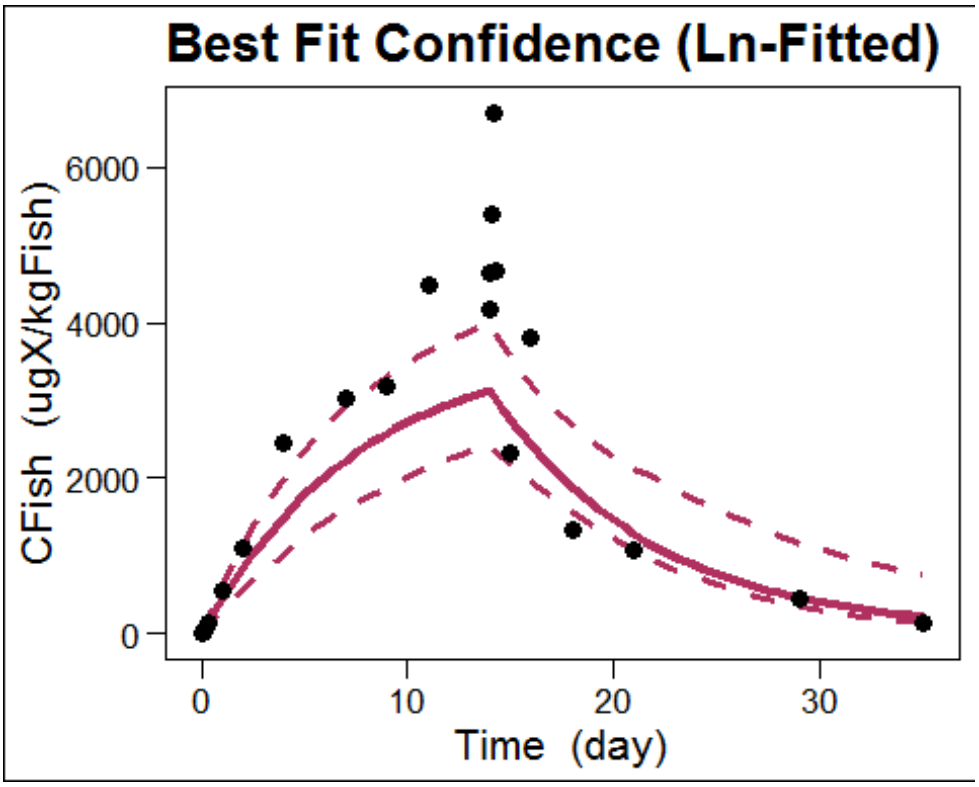
```
# --- PLOT MODEL CONF LIMITS
PlotConfFit_Aqueous_Ln()
```




```
# --- PLOT LN-FIT UNTRANSFORMED
PlotInvBestFit_Aqueous_Ln()
```




```
# --- PLOT LN-FIT CONF LIMITS UNTRANSFORMED
PlotInvConfFit_Aqueous_Ln()
```




```
# --- BCF SUMMARY TABLE
SummTable_Aqueous_Ln()
```

```
##           Estimate Std.Error      2.5%    97.5%         Unit
## k1        240.37     42.89    156.31    324.44  L/kgFish/day
```

```
## k2          0.1277    0.0207    0.0872    0.168        1/day
## k2g         0.0904    0.0207    0.0499    0.131        1/day
## BCFK      1882.3     264.29   1364.3    2400.3       L/kgFish
## BCFKg     2658.9     460.31   1756.7    3561.1       L/kgFish
## tHalfg       7.6671    1.7524    4.2325   11.102          day
## BCFKgLip 2859       494.96   1888.9    3829.1       L/kgFish

# ____________________________________________________________
#
# === END of R-Script ====
# ____________________________________________________________.
```

Dietary Data file format (.csvy)

```
---
# Dietary Exposure Fish Test
filename: Test-Dietary-RHCB10-22.1mgXperkgFood.csvy

# type of experiment
type: Dietary Exposure Fish Test
guideline: OECD 305

# where data came from
data source:
    name: DM
    origin: OECD Ring
    comments:

# test chemical and biological species
chemical:
    name: RHCB10
    chemical info:
species:
    name: Oncorhynchus?
    biological info:

# food concentration and ingestion rate
cfood:
    value: 22.10
    unit: mgX/kgFood
ingestion:
    value: 0.03
    unit: kgFood/kgFish/day

# feeding period
tfeed:
    value: 13.00
    unit: day

# lipid fractions
lipidfood:
    value: 15.00
    unit: percent
lipidfish:
    value: 4.65
    unit: percent

# time span of depuration
tdepur:
    value: 0.00
    unit: day
tend:
    value: 28.00
    unit: day

# fish growth
kgrowth:
    value: 0.0373
    unit: 1/day

# measured timeseries
time:
    name: Time
    unit: day
cfish:
    name: CFish
    unit: mgX/kgFish
---
Time,CFish
1.0,5.930
1.0,5.050
1.0,4.790
3.0,3.790
3.0,4.530
3.0,2.250
3.0,2.800
3.0,4.900
7.0,2.690
7.0,2.300
7.0,3.250
7.0,2.410
7.0,2.700
14.0,1.450
14.0,2.000
14.0,2.980
14.0,1.890
14.0,1.800
21.0,1.440
```

```
21.0,1.330
21.0,1.010
21.0,1.110
21.0,1.020
28.0,0.611
28.0,1.040
28.0,0.573
28.0,0.852
28.0,1.150
```

```
# ____________________________________________________________________
#
# === RUN-Dietary-RHCBio-22.1mgXpkgFood.R ====
# ____________________________________________________________________
#
# --- REQUIRED PACKAGES:
#
#       bcmfR
#       car
#       csvy
#       ggplot2
#       ggthemes
#       lucid
#       nlstools
# ____________________________________________________________________
#
# --- ATTACH bcmfR PACKAGE
library(bcmfR)
#
#
# --- INITIALIZE PROGRAM
InitProgram()
# ____________________________________________________________________

# ____________________________________________________________________
#
# --- READ DIETARY TEST DATA
ReadTest_Dietary("TEST-Dietary-RHCB10-22.1mgXperkgFood.csvy")

# ____________________________________________________________________
#
# === UNTRANSFORMED RUN ====
# ____________________________________________________________________

# --- FIT BCF MODEL
FitModel_Dietary()

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Dietary()

# --- CALC BEST FIT
RunBestFit_Dietary()

# --- PLOT BEST FIT
PlotBestFit_Dietary()

# --- BAYESIAN BOOTSTRAP
BootModel_Dietary()

# --- CALC MODEL CONF LIMITS
RunConfFit_Dietary()

# --- PLOT MODEL FIT AND CONF LIMITS
PlotConfFit_Dietary()

# --- BCF SUMMARY TABLE
SummTable_Dietary()
# ____________________________________________________________________


# ____________________________________________________________________
#
# === BOX-COX TRANSFORMED RUN ====
# ____________________________________________________________________

# --- BOX-COX PLOT AND LAMBDA ESTIMATE
ModelTrans_BoxCox_Dietary()

# --- FIT BCF MODEL
FitModel_Dietary_BoxCox()

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Dietary_BoxCox()

# --- CALC BEST FIT
RunBestFit_Dietary_BoxCox()

# --- PLOT BEST FIT
PlotBestFit_Dietary_BoxCox()

# --- BAYESIAN BOOTSTRAP
BootModel_Dietary_BoxCox()

# --- CALC MODEL CONF LIMITS
RunConfFit_Dietary_BoxCox()

# --- PLOT MODEL CONF LIMITS
```

```
PlotConfFit_Dietary_BoxCox()

# --- PLOT BEST FIT ON ORIGINAL SCALE
PlotInvBestFit_Dietary_BoxCox()

# --- PLOT CONF FIT ON ORIGINAL SCALE
PlotInvConfFit_Dietary_BoxCox()

# --- BCF SUMMARY TABLE
SummTable_Dietary_BoxCox()
# ____________________________________________________________________


# ____________________________________________________________________
#
# === LN-TRANSFORMED RUN ====

# --- FIT BCF MODEL
FitModel_Dietary_Ln()

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Dietary_Ln()

# --- CALC BEST FIT
RunBestFit_Dietary_Ln()

# --- PLOT BEST FIT
PlotBestFit_Dietary_Ln()

# --- BAYESIAN BOOTSTRAP
BootModel_Dietary_Ln()

# --- CALC MODEL CONF LIMITS
RunConfFit_Dietary_Ln()

# --- PLOT MODEL CONF LIMITS
PlotConfFit_Dietary_Ln()

# --- PLOT BEST FIT ON ORIGINAL SCALE
PlotInvBestFit_Dietary_Ln()

# --- PLOT CONF FIT ON ORIGINAL SCALE
PlotInvConfFit_Dietary_Ln()

# --- BCF SUMMARY TABLE
SummTable_Dietary_Ln()
# ____________________________________________________________________
#
# === END of R-Script ====
# ____________________________________________________________________.
```

RUN-Dietary-RHCBIO-22.1mgXperkgFood.R

aldenbet

2019-07-17

```
# ____________________________________________________________________
#
# === RUN-Dietary-RHCBio-22.1mgXpkgFood.R ====
# ____________________________________________________________________
#
# --- REQUIRED PACKAGES:
#
#       bcmfR
#       car
#       csvy
#       ggplot2
#       ggthemes
#       lucid
#       nlstools
# ____________________________________________________________________
#
# --- ATTACH bcmfR PACKAGE
library(bcmfR)
#
#
# --- INITIALIZE PROGRAM
InitProgram()

## 0.4-18

# ____________________________________________________________________

# ____________________________________________________________________
#
# --- READ DIETARY TEST DATA
ReadTest_Dietary("TEST-Dietary-RHCB10-22.1mgXperkgFood.csvy")

## TEST.Dietary.Design:
##             value              unit
## cfood     22.1000        mgX/kgFood
## ingestion  0.0300 kgFood/kgFish/day
## tfeed     13.0000               day
## lipidfood 15.0000           percent
## lipidfish  4.6500           percent
## tdepur     0.0000               day
## tend      28.0000               day
## kgrowth    0.0373             1/day
##
## TEST.Dietary.Measured:
##        name       unit
## time   Time        day
## cfish CFish mgX/kgFish
##
```

```
## DATA.Dietary.Work:
##    time.data cfish.data
## 1          1      5.930
## 2          1      5.050
## 3          1      4.790
## 4          3      3.790
## 5          3      4.530
## 6          3      2.250
## 7          3      2.800
## 8          3      4.900
## 9          7      2.690
## 10         7      2.300
## 11         7      3.250
## 12         7      2.410
## 13         7      2.700
## 14        14      1.450
## 15        14      2.000
## 16        14      2.980
## 17        14      1.890
## 18        14      1.800
## 19        21      1.440
## 20        21      1.330
## 21        21      1.010
## 22        21      1.110
## 23        21      1.020
## 24        28      0.611
## 25        28      1.040
## 26        28      0.573
## 27        28      0.852
## 28        28      1.150

# ____________________________________________________________________
#
# === UNTRANSFORMED RUN ====
# ____________________________________________________________________

# --- FIT BCF MODEL
FitModel_Dietary()

##
## Formula: cfish.data ~ RunModel_Dietary(time.data, fitc0d, fitk2)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## fitc0d 4.927242   0.299926  16.428 3.01e-15 ***
## fitk2  0.070471   0.008513   8.278 9.27e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6444 on 26 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 2.219e-06
```

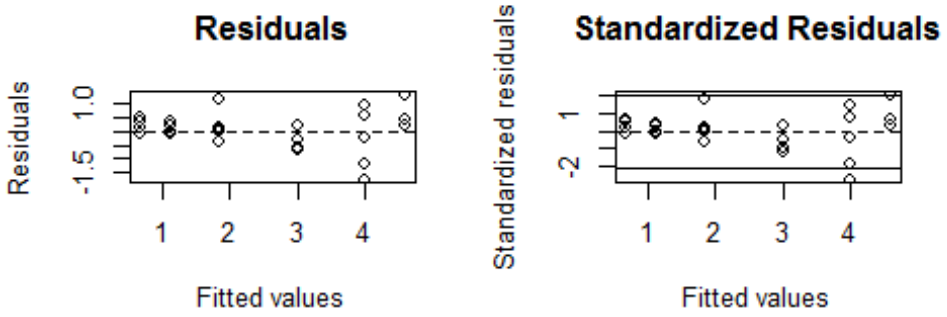
```r
# --- MODEL DIAGNOSTICS
ModelDiagnostics_Dietary()
```

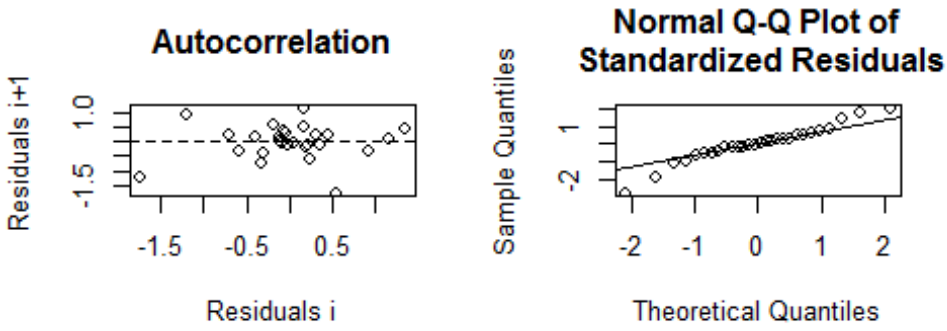
```
##
## ------
##  Shapiro-Wilk normality test
##
## data:  stdres
## W = 0.95334, p-value = 0.24
##
##
## ------

##
##  Runs Test
##
## data:  as.factor(run)
## Standard Normal = 0, p-value = 1
## alternative hypothesis: two.sided
```

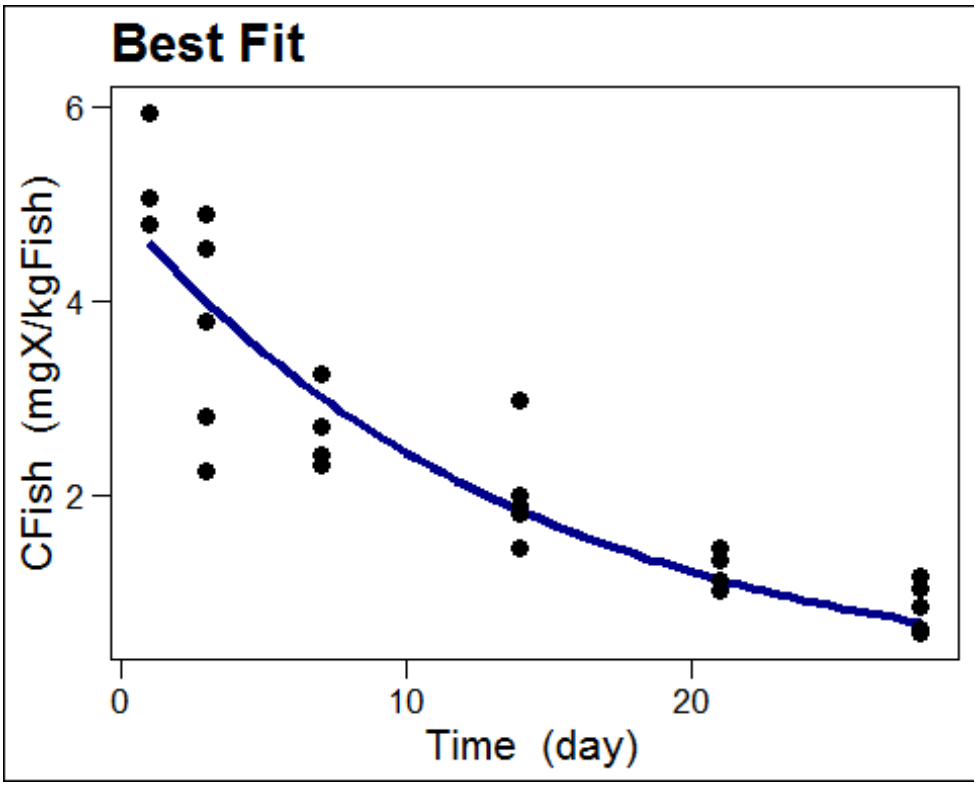
```r
# --- CALC BEST FIT
RunBestFit_Dietary()
```

```
##    time.best cfish.best
## 1       1.00   4.591967
## 2       1.45   4.448631
## 3       1.90   4.309769
## 4       2.35   4.175242
## 5       2.80   4.044914
## 6       3.25   3.918655
## 7       3.70   3.796336
## 8       4.15   3.677836
## 9       4.60   3.563034
## 10      5.05   3.451816
##    ...
##    time.best cfish.best
## 52     23.95  0.9111903
## 53     24.40  0.8827480
## 54     24.85  0.8551935
## 55     25.30  0.8284991
## 56     25.75  0.8026380
## 57     26.20  0.7775841
## 58     26.65  0.7533122
## 59     27.10  0.7297980
## 60     27.55  0.7070178
## 61     28.00  0.6849486
```

```r
# --- PLOT BEST FIT
PlotBestFit_Dietary()
```

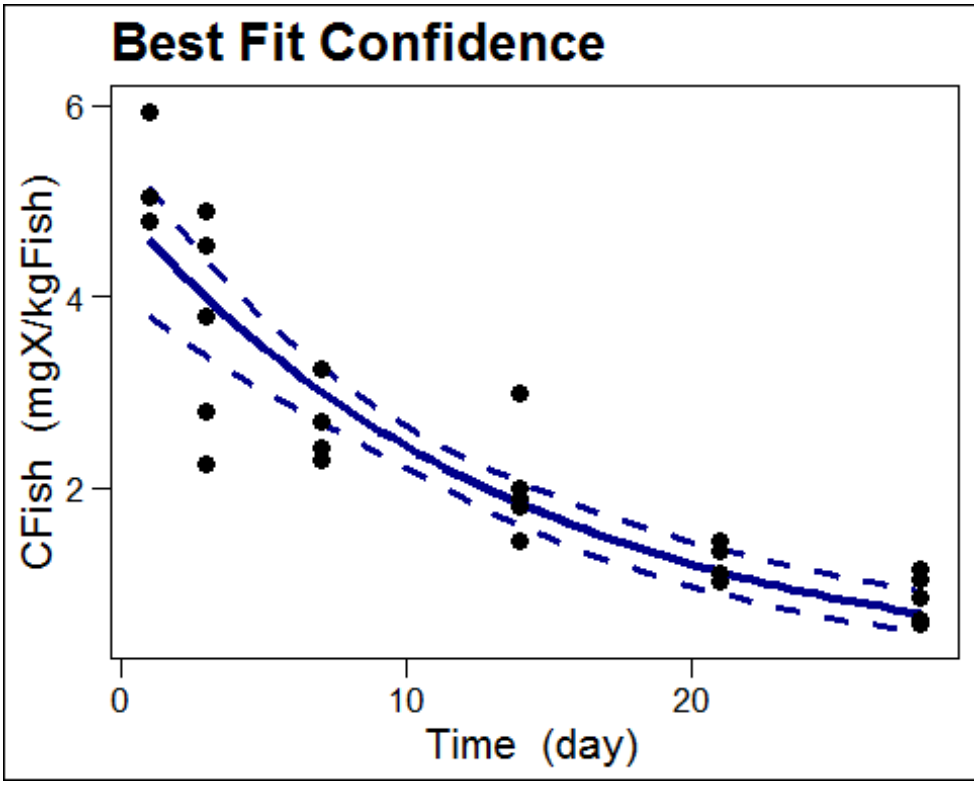







```
# --- BAYESIAN BOOTSTRAP
BootModel_Dietary()

##    c0d.boot    k2.boot
## 1  4.927242 0.07047111
## 2  5.036959 0.07592888
## 3  3.965896 0.05736467
## 4  4.442990 0.06617196
```

```
## 5  4.194194 0.05771707
## 6  5.169117 0.07410142
## 7  4.562299 0.06499439
## 8  5.478833 0.07451372
## 9  4.987564 0.07380673
## 10 5.016642 0.07047222
##    ...
##      c0d.boot    k2.boot
## 991  4.988946 0.07086736
## 992  5.298651 0.07466419
## 993  4.616114 0.06311129
## 994  4.677466 0.07101027
## 995  4.821155 0.07469125
## 996  5.301245 0.07587802
## 997  4.534909 0.05819521
## 998  4.683798 0.06826420
## 999  3.951422 0.05276979
## 1000 4.821085 0.06472509

# --- CALC MODEL CONF LIMITS
RunConfFit_Dietary()

##    time.conf cfish.conflo cfish.confup
## 1       1.00     3.785182     5.143470
## 2       1.45     3.688392     4.959009
## 3       1.90     3.594946     4.769554
## 4       2.35     3.506031     4.612757
## 5       2.80     3.415941     4.452529
## 6       3.25     3.327681     4.300325
## 7       3.70     3.245135     4.157582
## 8       4.15     3.161722     4.020726
## 9       4.60     3.079825     3.888404
## 10      5.05     3.002393     3.756848
##    ...
##    time.conf cfish.conflo cfish.confup
## 52     23.95    0.6914670    1.1510184
## 53     24.40    0.6654891    1.1215120
## 54     24.85    0.6407670    1.0924397
## 55     25.30    0.6165629    1.0644459
## 56     25.75    0.5932340    1.0394955
## 57     26.20    0.5707878    1.0134869
## 58     26.65    0.5491910    0.9879441
## 59     27.10    0.5284114    0.9624230
## 60     27.55    0.5084181    0.9377535
## 61     28.00    0.4891813    0.9141769

# --- PLOT MODEL FIT AND CONF LIMITS
PlotConfFit_Dietary()
```

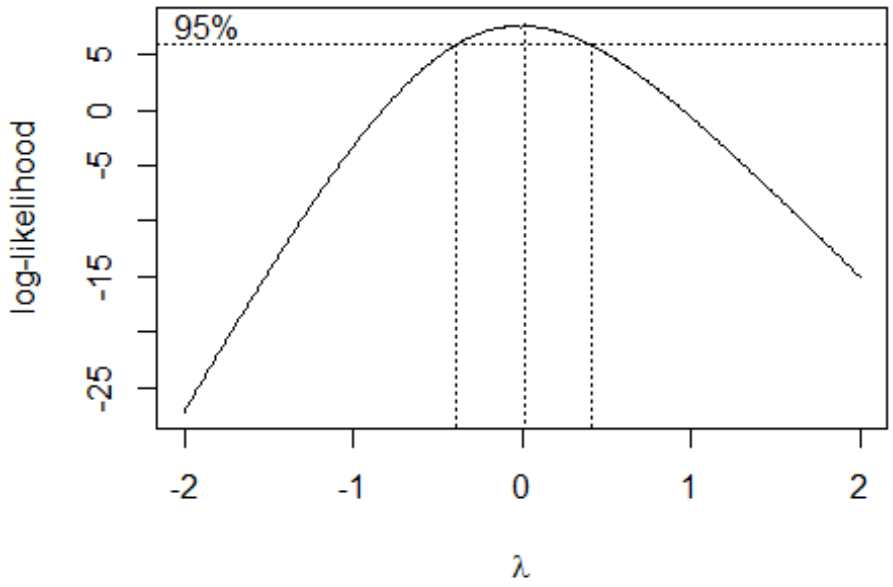



```
# --- BCF SUMMARY TABLE
SummTable_Dietary()
```

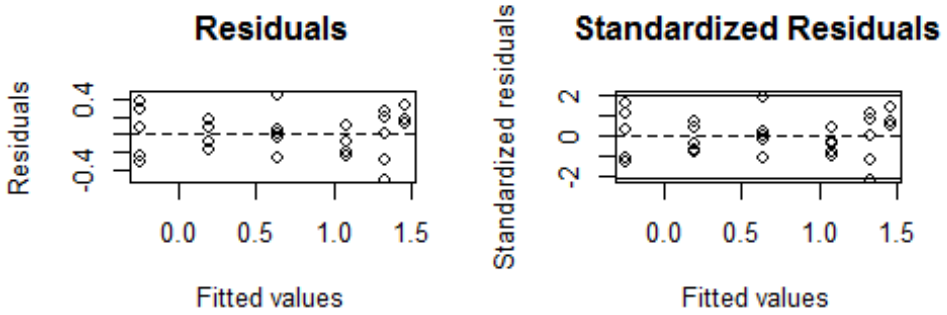
```
##              Estimate Std.Error      2.5%     97.5%              unit
## C0d          4.9272    0.29993   4.3394    5.5151          mgX/kgFish
## k2           0.070471  0.008513  0.053785  0.087157             1/day
## k2g          0.033171  0.008513  0.016485  0.049857             1/day
## kf           0.026189  0.0026    0.021092  0.031286 kgFood/kgFish/day
## alpha        0.87297   0.086682  0.70307   1.0429                   -
## BMFK         0.37163   0.020211  0.33201   0.41124      kgFood/kgFish
## BMFKg        0.78951   0.13689   0.52121   1.0578       kgFood/kgFish
## tHalfg      20.892     5.3619   10.382    31.401                 day
## BMFKgLipid   2.5468    0.44158   1.6813    3.4123       kgFood/kgFish
```

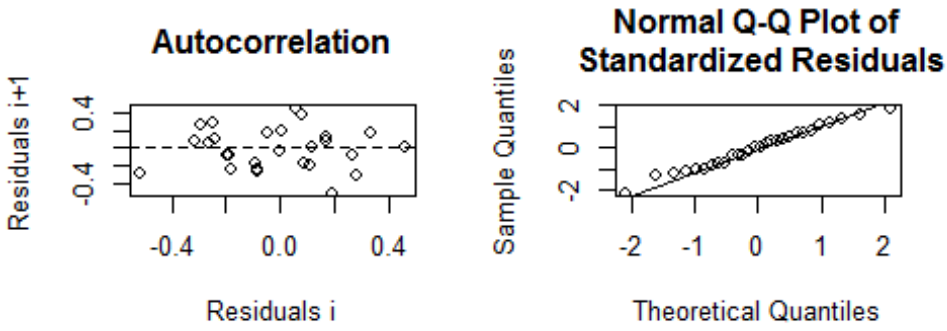
```
# ____________________________________________________________
```

```
# ____________________________________________________________
#
# === BOX-COX TRANSFORMED RUN ====
# ____________________________________________________________

# --- BOX-COX PLOT AND LAMBDA ESTIMATE
ModelTrans_BoxCox_Dietary()
```




```
##         fit     conflo     confup
##  0.0100000 -0.3951205  0.4054187

# --- FIT BCF MODEL
FitModel_Dietary_BoxCox()

##
## Formula: bc.cfish.data ~ car::bcPower(RunModel_Dietary(time.data, fitc0d,
##     fitk2) + lnstarter, lambda)
##
## Parameters:
##         Estimate Std. Error t value Pr(>|t|)
## fitc0d 4.525346   0.349292   12.96 7.53e-13 ***
## fitk2  0.062546   0.004779   13.09 5.97e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2431 on 26 degrees of freedom
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 1.314e-06

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Dietary_BoxCox()
```





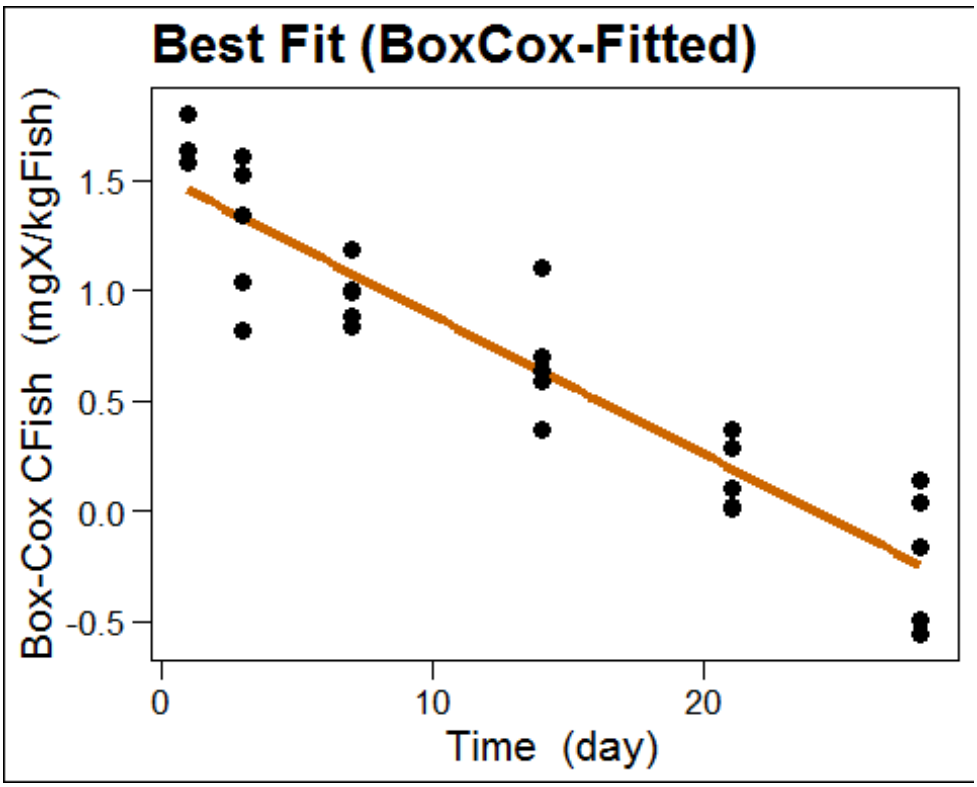

```
##
## ------
##  Shapiro-Wilk normality test
##
## data:  stdres
## W = 0.98541, p-value = 0.9543
##
##
## ------

##
##  Runs Test
##
## data:  as.factor(run)
## Standard Normal = -0.74677, p-value = 0.4552
## alternative hypothesis: two.sided
```

```
# --- CALC BEST FIT
RunBestFit_Dietary_BoxCox()
```

```
##    time.best bc.cfish.best inv.bc.cfish.best
## 1       1.00      1.457670          4.250974
## 2       1.45      1.429118          4.132996
## 3       1.90      1.400574          4.018292
## 4       2.35      1.372038          3.906771
## 5       2.80      1.343511          3.798345
## 6       3.25      1.314991          3.692929
## 7       3.70      1.286479          3.590438
## 8       4.15      1.257975          3.490792
## 9       4.60      1.229480          3.393911
## 10      5.05      1.200992          3.299719
```

```
##    ...
##    time.best bc.cfish.best inv.bc.cfish.best
## 52     23.95    0.01171983         1.0117881
## 53     24.40   -0.01642518         0.9837077
## 54     24.85   -0.04456226         0.9564065
## 55     25.30   -0.07269143         0.9298631
## 56     25.75   -0.10081268         0.9040564
## 57     26.20   -0.12892602         0.8789659
## 58     26.65   -0.15703144         0.8545717
## 59     27.10   -0.18512896         0.8308545
## 60     27.55   -0.21321857         0.8077956
## 61     28.00   -0.24130027         0.7853766
```

```
# --- PLOT BEST FIT
PlotBestFit_Dietary_BoxCox()
```



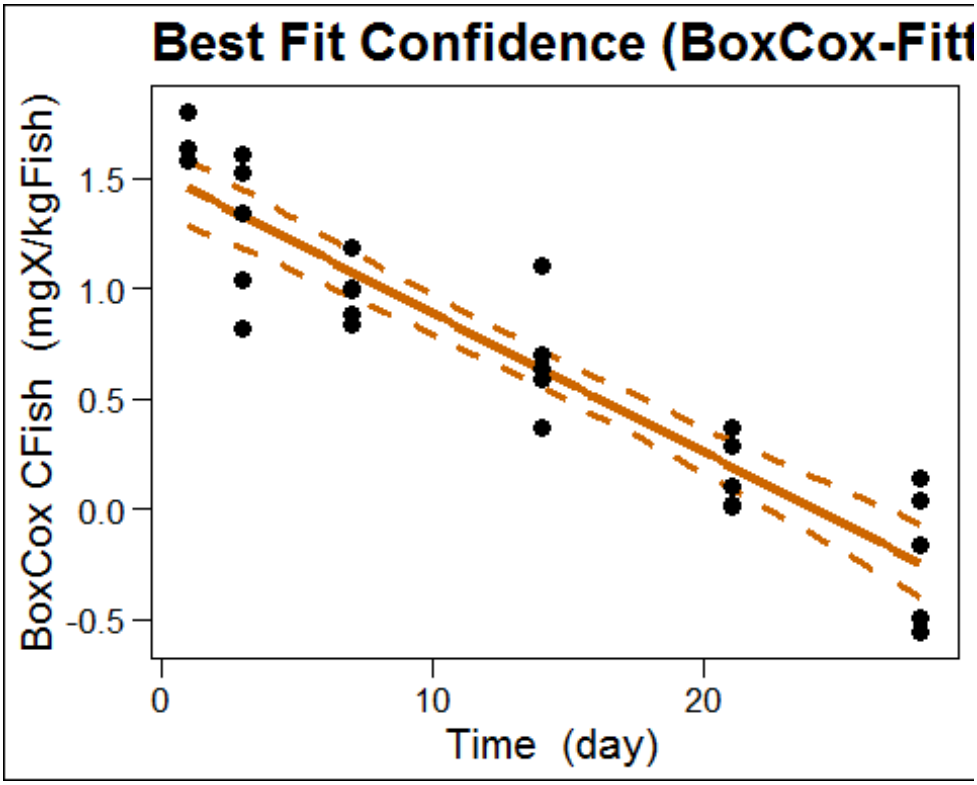

```
# --- BAYESIAN BOOTSTRAP
BootModel_Dietary_BoxCox()
```

```
##    c0d.boot.bc k2.boot.bc
## 1     4.525346 0.06254593
## 2     4.450325 0.06229939
## 3     3.841455 0.05591152
## 4     4.133812 0.05989244
## 5     4.243539 0.06016141
## 6     4.609451 0.06188936
## 7     4.461327 0.06566179
## 8     5.365550 0.07339003
## 9     4.615558 0.06740612
## 10    4.368586 0.05824859
##    ...
```

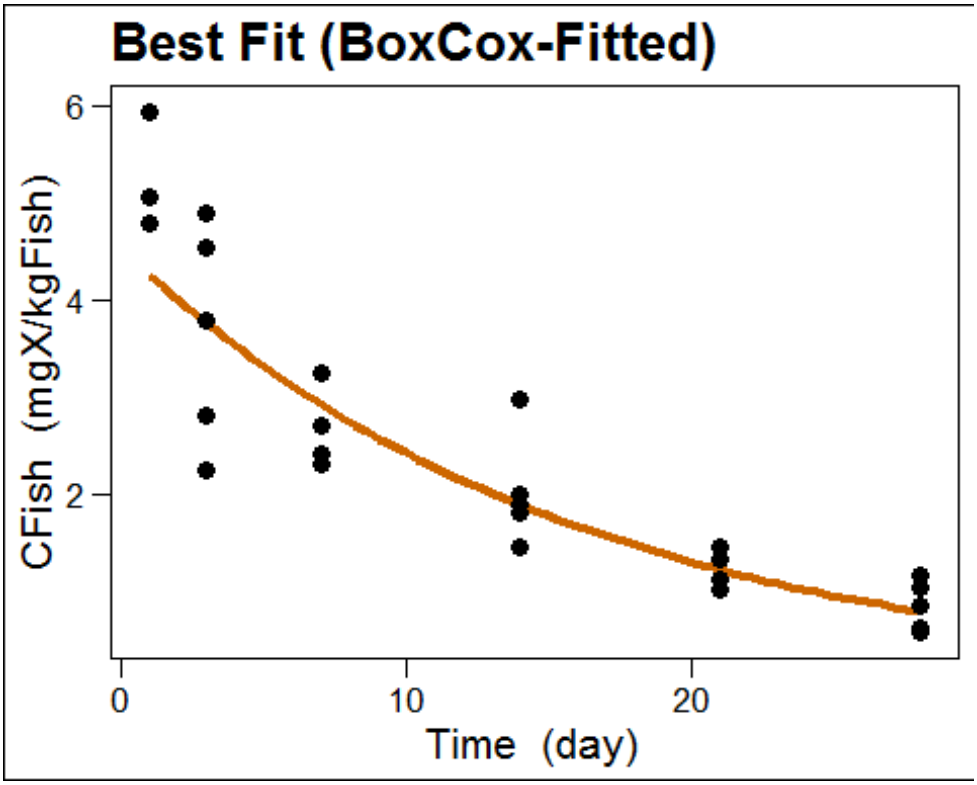
```
##      c0d.boot.bc k2.boot.bc
## 991     4.401399 0.05710446
## 992     4.694338 0.06085221
## 993     4.145406 0.05499248
## 994     3.976650 0.05868337
## 995     4.396279 0.06543243
## 996     4.781892 0.06631465
## 997     4.325198 0.05442929
## 998     4.237279 0.05743178
## 999     3.735998 0.04768364
## 1000    4.535982 0.06055540

# --- CALC MODEL CONF LIMITS
RunConfFit_Dietary_BoxCox()

##    time.conf bc.cfish.conflo bc.cfish.confup inv.bc.cfish.conflo
## 1       1.00        1.285835        1.587439            3.588158
## 2       1.45        1.261808        1.554801            3.504029
## 3       1.90        1.237423        1.523814            3.420648
## 4       2.35        1.214259        1.491501            3.343261
## 5       2.80        1.191020        1.460075            3.267364
## 6       3.25        1.168026        1.428222            3.193948
## 7       3.70        1.144918        1.396574            3.121811
## 8       4.15        1.120437        1.365358            3.047150
## 9       4.60        1.095016        1.333140            2.971493
## 10      5.05        1.069886        1.301156            2.898528
##    inv.bc.cfish.confup
## 1             4.830600
## 2             4.677846
## 3             4.537248
## 4             4.395090
## 5             4.261061
## 6             4.129344
## 7             4.002472
## 8             3.881111
## 9             3.759672
## 10            3.642839
##    ...
##    time.conf bc.cfish.conflo bc.cfish.confup inv.bc.cfish.conflo
## 52     23.95      -0.1109756     0.151299658           0.8949055
## 53     24.40      -0.1423766     0.126706982           0.8672067
## 54     24.85      -0.1745773     0.101768889           0.8396838
## 55     25.30      -0.2066310     0.077189675           0.8131459
## 56     25.75      -0.2382238     0.053269648           0.7878024
## 57     26.20      -0.2700023     0.028694852           0.7630991
## 58     26.65      -0.3022920     0.004267113           0.7387839
## 59     27.10      -0.3335605    -0.019394972           0.7159693
## 60     27.55      -0.3648192    -0.043503920           0.6938591
## 61     28.00      -0.3960681    -0.068312116           0.6724318
##    inv.bc.cfish.confup
## 52           1.1632122
## 53           1.1349934
## 54           1.1070703
## 55           1.0802148
```

```
## 56           1.0546990
## 57           1.0291065
## 58           1.0042768
## 59           0.9807902
## 60           0.9574199
## 61           0.9339483
```

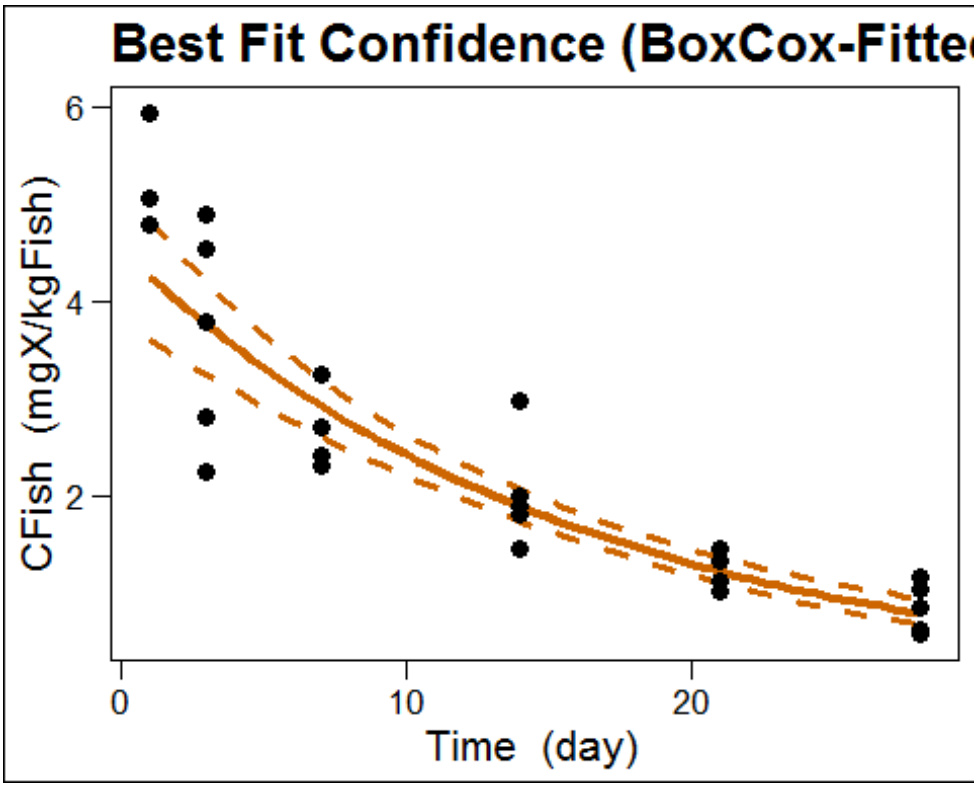
```
# --- PLOT MODEL CONF LIMITS
PlotConfFit_Dietary_BoxCox()
```




```
# --- PLOT BEST FIT ON ORIGINAL SCALE
PlotInvBestFit_Dietary_BoxCox()
```



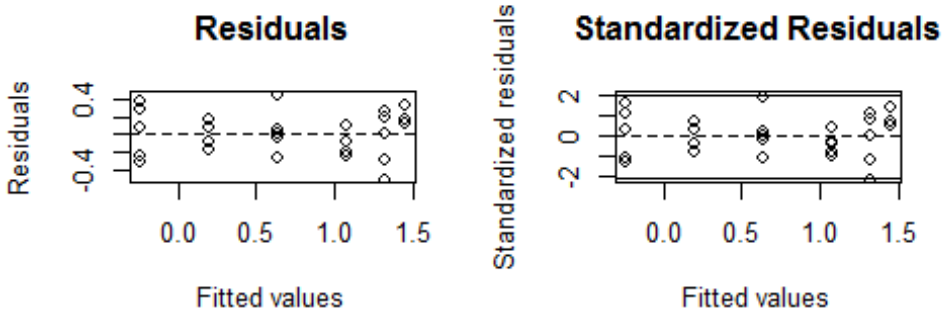

```
# --- PLOT CONF FIT ON ORIGINAL SCALE
PlotInvConfFit_Dietary_BoxCox()
```

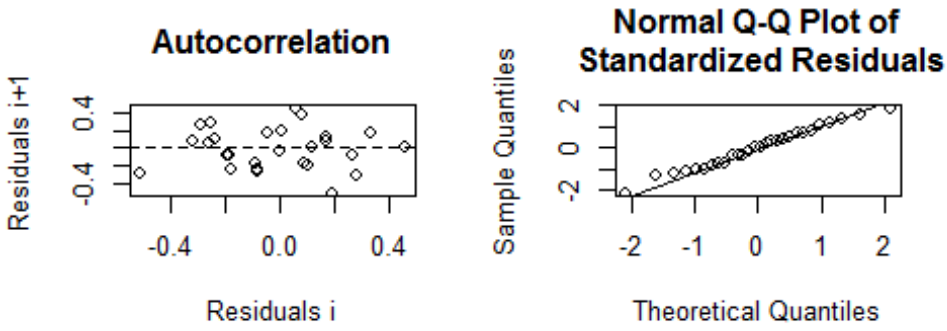



```
# --- BCF SUMMARY TABLE
SummTable_Dietary_BoxCox()
```

```
##             Estimate Std.Error      2.5%    97.5%             unit
## C0d         4.5253    0.34929    3.8407    5.21          mgX/kgFish
```

```
## k2          0.062546  0.004779  0.05318   0.07191             1/day
## k2g         0.025246  0.004779  0.01588   0.03461             1/day
## kf          0.023013  0.002305  0.018496  0.02753 kgFood/kgFish/day
## alpha       0.76711   0.076827  0.61653   0.91769                 -
## BMFK        0.36794   0.017431  0.33378   0.40211     kgFood/kgFish
## BMFKg       0.91157   0.1003    0.71498   1.1082      kgFood/kgFish
## tHalfg     27.45      5.1958   17.266    37.634                day
## BMFKgLipid  2.9405    0.32355   2.3064    3.5747      kgFood/kgFish

# ____________________________________________________________________


# ____________________________________________________________________
#
# === LN-TRANSFORMED RUN ====

# --- FIT BCF MODEL
FitModel_Dietary_Ln()

##
## Formula: ln.cfish.data ~ log(RunModel_Dietary(time.data, fitc0d, fitk2) +
##     lnstarter)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## fitc0d  4.52215    0.35060   12.90 8.33e-13 ***
## fitk2   0.06251    0.00477   13.11 5.79e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2414 on 26 degrees of freedom
##
## Number of iterations to convergence: 0
## Achieved convergence tolerance: 3.783e-09

# --- MODEL DIAGNOSTICS
ModelDiagnostics_Dietary_Ln()
```

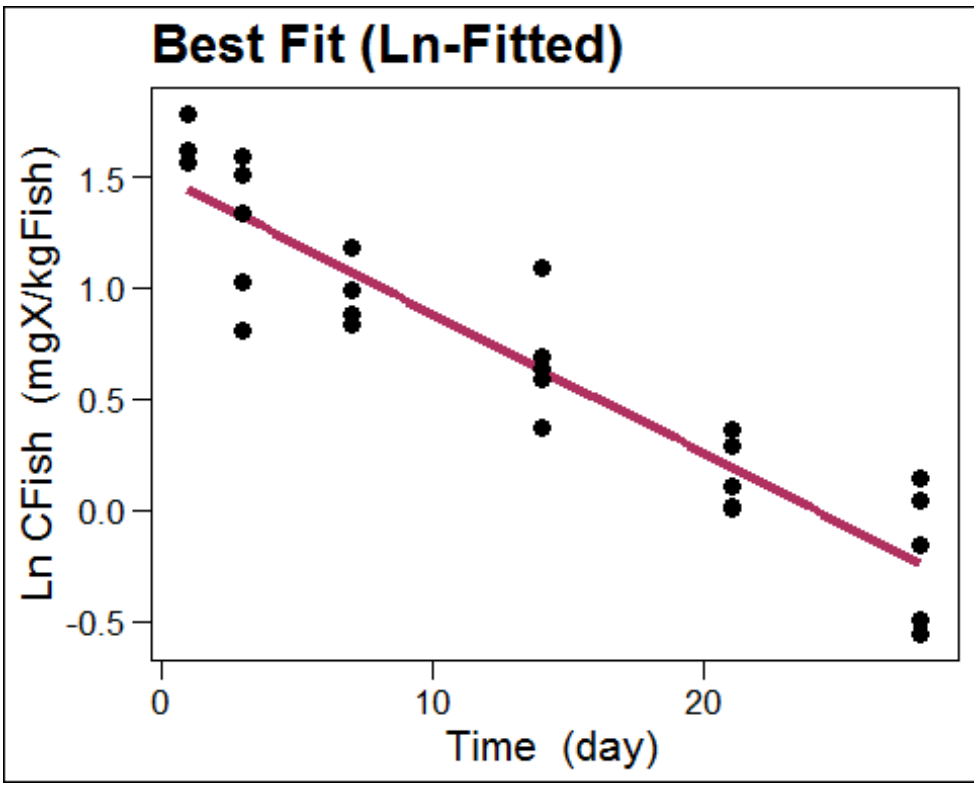





```
##
## ------
##  Shapiro-Wilk normality test
##
## data:  stdres
## W = 0.98551, p-value = 0.9557
##
##
## ------

##
##  Runs Test
##
## data:  as.factor(run)
## Standard Normal = -0.74677, p-value = 0.4552
## alternative hypothesis: two.sided

# --- CALC BEST FIT
RunBestFit_Dietary_Ln()

##    time.best ln.cfish.best inv.ln.cfish.best
## 1       1.00      1.446476          4.248118
## 2       1.45      1.418345          4.130280
## 3       1.90      1.390215          4.015712
## 4       2.35      1.362084          3.904321
## 5       2.80      1.333953          3.796020
## 6       3.25      1.305822          3.690723
## 7       3.70      1.277692          3.588347
## 8       4.15      1.249561          3.488811
## 9       4.60      1.221430          3.392036
## 10      5.05      1.193300          3.297945
```

```
##    ...
##    time.best ln.cfish.best inv.ln.cfish.best
## 52     23.95    0.01181010         1.0118801
## 53     24.40   -0.01632061         0.9838119
## 54     24.85   -0.04445131         0.9565222
## 55     25.30   -0.07258201         0.9299895
## 56     25.75   -0.10071272         0.9041928
## 57     26.20   -0.12884342         0.8791116
## 58     26.65   -0.15697412         0.8547262
## 59     27.10   -0.18510483         0.8310172
## 60     27.55   -0.21323553         0.8079658
## 61     28.00   -0.24136623         0.7855539
```

```
# --- PLOT BEST FIT
PlotBestFit_Dietary_Ln()
```



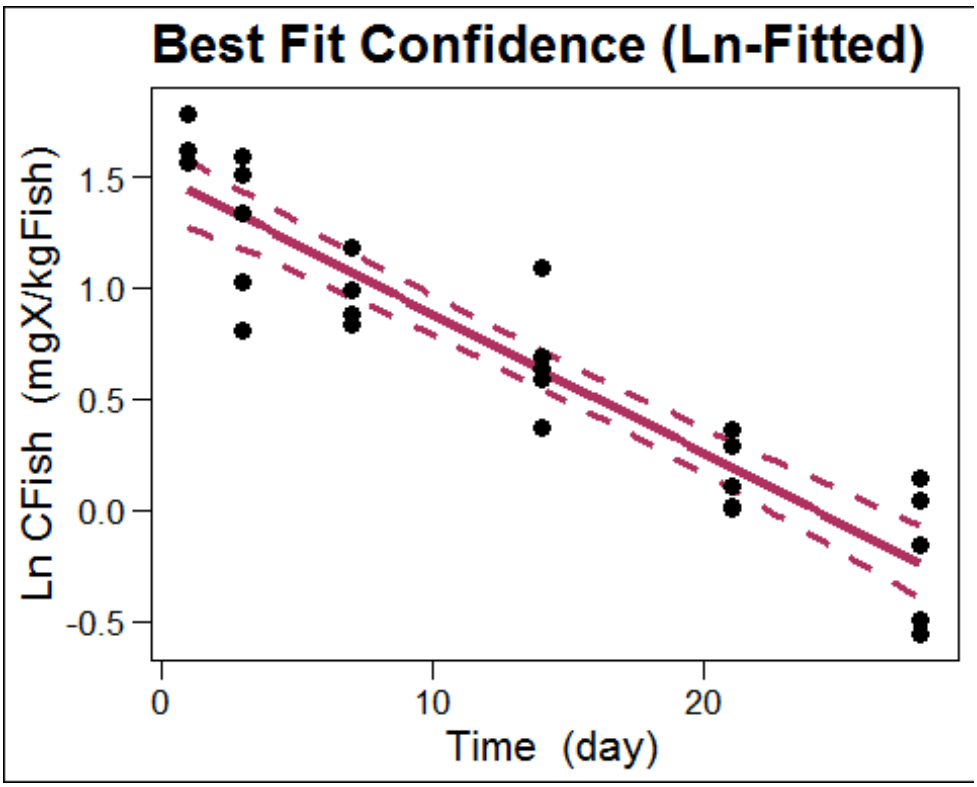

```
# --- BAYESIAN BOOTSTRAP
BootModel_Dietary_Ln()
```

```
##    c0d.boot.ln k2.boot.ln
## 1     4.522155 0.06251267
## 2     4.446436 0.06225038
## 3     3.842895 0.05595631
## 4     4.132431 0.05988845
## 5     4.245193 0.06019770
## 6     4.604656 0.06183206
## 7     4.461484 0.06568601
## 8     5.364323 0.07338472
## 9     4.612867 0.06738345
## 10    4.362250 0.05816853
##    ...
```

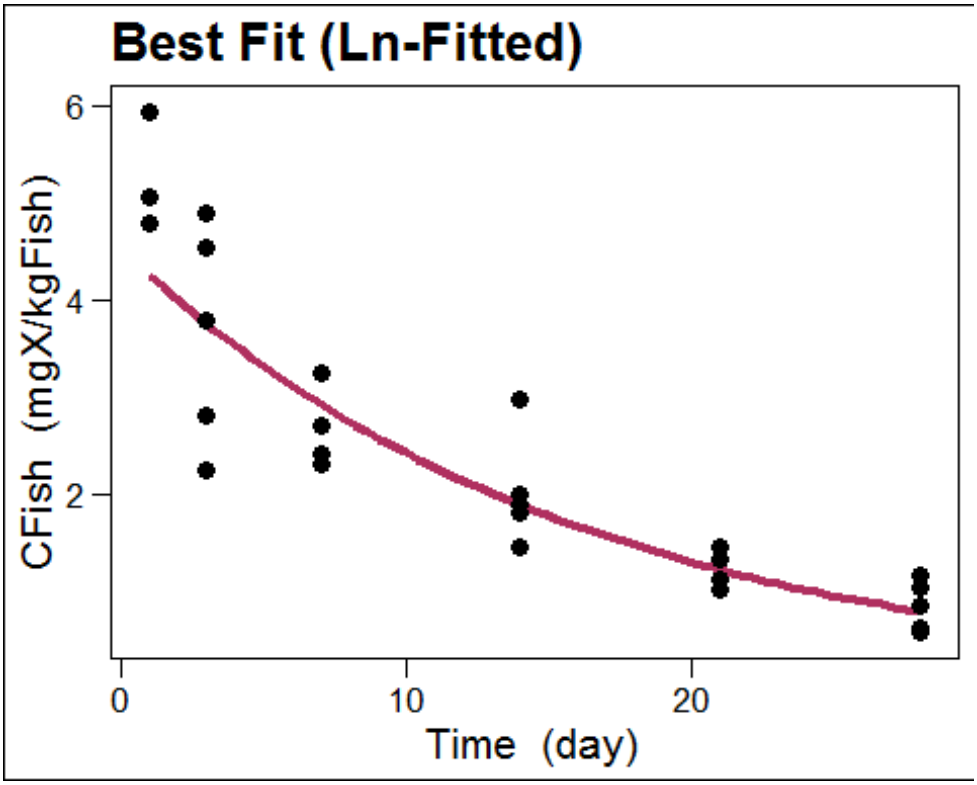
```
##      c0d.boot.ln k2.boot.ln
## 991     4.396354 0.05703356
## 992     4.688704 0.06077599
## 993     4.141513 0.05494839
## 994     3.970688 0.05860541
## 995     4.393605 0.06539896
## 996     4.777602 0.06626644
## 997     4.323905 0.05442635
## 998     4.233314 0.05737185
## 999     3.734249 0.04765822
## 1000    4.534510 0.06055582

# --- CALC MODEL CONF LIMITS
RunConfFit_Dietary_Ln()

##    time.conf ln.cfish.conflo ln.cfish.confup inv.ln.cfish.conflo
## 1       1.00        1.277018        1.574382            3.585930
## 2       1.45        1.253240        1.542572            3.501669
## 3       1.90        1.229266        1.511415            3.418719
## 4       2.35        1.206435        1.480238            3.341551
## 5       2.80        1.183551        1.449112            3.265952
## 6       3.25        1.160774        1.417843            3.192402
## 7       3.70        1.137943        1.386575            3.120343
## 8       4.15        1.114223        1.355518            3.047201
## 9       4.60        1.089068        1.324038            2.971504
## 10      5.05        1.063918        1.292461            2.897702
##    inv.ln.cfish.confup
## 1             4.827757
## 2             4.676603
## 3             4.533141
## 4             4.393992
## 5             4.259329
## 6             4.128208
## 7             4.001123
## 8             3.878771
## 9             3.758569
## 10            3.641737
##    ...
##    time.conf ln.cfish.conflo ln.cfish.confup inv.ln.cfish.conflo
## 52     23.95      -0.1112834      0.15128435           0.8946854
## 53     24.40      -0.1424570      0.12660996           0.8672248
## 54     24.85      -0.1747986      0.10196105           0.8396261
## 55     25.30      -0.2063547      0.07755878           0.8135446
## 56     25.75      -0.2383709      0.05377279           0.7879104
## 57     26.20      -0.2698513      0.02923810           0.7634930
## 58     26.65      -0.3019126      0.00482169           0.7394027
## 59     27.10      -0.3342604     -0.01881304           0.7158676
## 60     27.55      -0.3658727     -0.04337288           0.6935915
## 61     28.00      -0.3972464     -0.06769572           0.6721687
##    inv.ln.cfish.confup
## 52           1.1633274
## 53           1.1349743
## 54           1.1073404
## 55           1.0806460
```

```
## 56           1.0552449
## 57           1.0296702
## 58           1.0048345
## 59           0.9813632
## 60           0.9575545
## 61           0.9345467
```

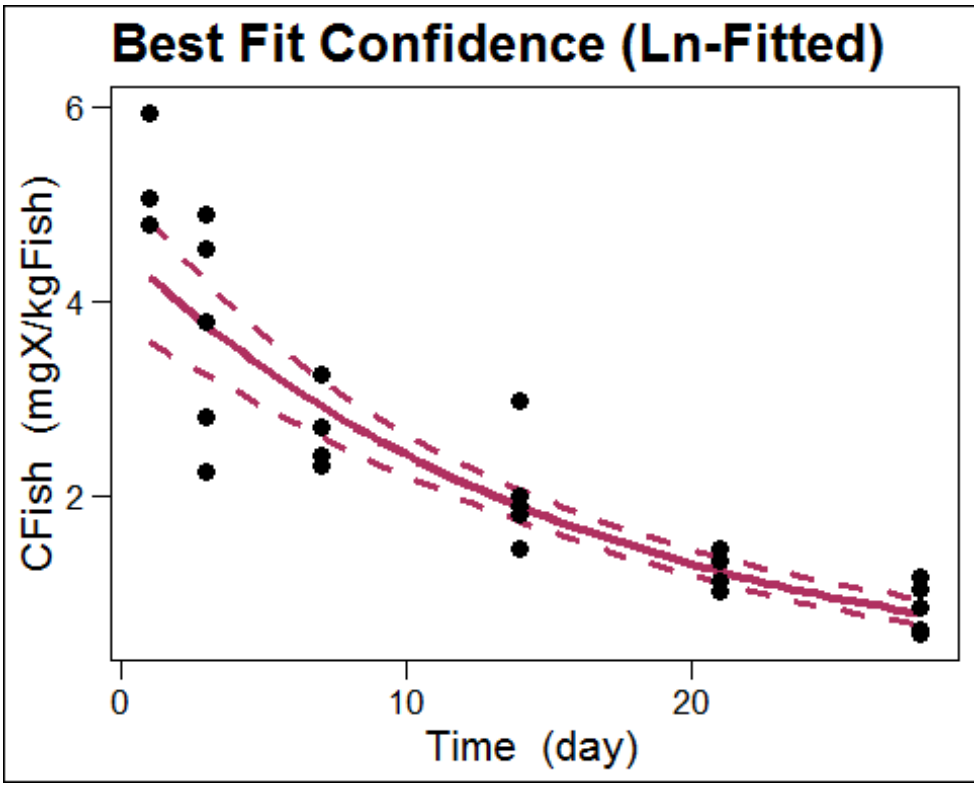
```
# --- PLOT MODEL CONF LIMITS
PlotConfFit_Dietary_Ln()
```




```
# --- PLOT BEST FIT ON ORIGINAL SCALE
PlotInvBestFit_Dietary_Ln()
```




```
# --- PLOT CONF FIT ON ORIGINAL SCALE
PlotInvConfFit_Dietary_Ln()
```




```
# --- BCF SUMMARY TABLE
SummTable_Dietary_Ln()
```

```
##             Estimate Std.Error      2.5%    97.5%             unit
## C0d           4.5222    0.3506     3.835     5.2093        mgX/kgFish
```

```
## k2           0.062513  0.00477   0.053164  0.07186             1/day
## k2g          0.025213  0.00477   0.015864  0.03456             1/day
## kf           0.022993  0.002311  0.018464  0.02752 kgFood/kgFish/day
## alpha        0.76643   0.077017  0.61547   0.91738                 -
## BMFK         0.36781   0.017471  0.33357   0.40205     kgFood/kgFish
## BMFKg        0.91195   0.099939  0.71607   1.1078      kgFood/kgFish
## tHalfg      27.486     5.1998   17.295    37.678                 day
## BMFKgLipid   2.9418    0.32238   2.3099    3.5737      kgFood/kgFish

# ____________________________________________________________________
#
# === END of R-Script ====
# ____________________________________________________________________.
```