

Introduction to the TAF package

2 Running a TAF analysis

Arni Magnusson and Colin Millar

Overview

- 1 **Background** *objectives, design*
- 2 **Running a TAF analysis** *linear regression, boot and run, structured scripts*
- 3 **TAF features** *boot procedure, data flow, new analysis, overview of functions*
- 4 **The TAF community** *browsing an existing analysis, related R packages*
- 5 **Discussion** *contents of a TAF analysis, benefits of TAF*
- 6 **Online examples** *ICES, FAO, SPC, various*






Linear Regression Example

To demonstrate how a TAF analysis works, consider a simple linear regression where the x and y coordinates come from a text file. The `linreg` example comes with the TAF package and can be copied to a convenient place to test and run:

```
library(TAF)
taf.example("linreg")
setwd("linreg")
```

Boot and Run

Before running the analysis, the workflow consists of a boot folder and four TAF scripts:

-  boot
-  data.R
-  model.R
-  output.R
-  report.R

Boot and Run

To run a TAF analysis, the first step is to start R and make sure that the **current working directory** is set to the location of the TAF scripts `data.R`, `model.R`, etc.

Some R editors do this automatically when opening an R script and in RStudio there is a menu command:

Session - Set Working Directory - To Source File Location

[Alt-s w s]

Boot and Run

All TAF analyses can be run using the following commands in R:

```
library(TAF)
taf.boot()
source.all()
```

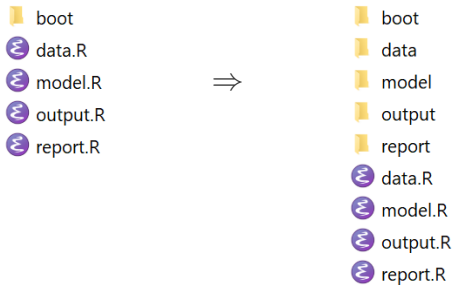
The `taf.boot()` function looks for an existing boot folder to set up the data and software required for the analysis.

Then `source.all()` runs the `data.R`, `model.R` `output.R`, and `report.R` scripts in that order.

The individual scripts can also be run using `source()` or line by line in an R editor.

Boot and Run

After running the analysis, each script has created a corresponding folder, data.R creating a data folder, etc.



Structured scripts

The `data.R` script has populated the data folder with comma-separated values (CSV) files representing the **data that are used in the analysis** and the `model.R` script produced model results in a machine-readable format.

The purpose of the `output.R` script is to **read the results** from the model folder and write out CSV files representing the results that are of primary interest.

Finally, `report.R` reads in the CSV output and produces **plots and formatted tables**, often with rounded numbers, which can be incorporated into a report.

The `report.R` script can also produce a **dynamic document** in various formats if the scientist writes the script in that way.

It is important to note that TAF does not do any of this by itself. All the work is performed by the R scripts that the scientist writes.

Structured scripts

Typically, a TAF script has the following general structure:

```
# What the script does

# Before: file1.csv, file2.rds, file3.spec (previous)
# After: file4.csv, file5.csv, file6.png (folder)

library(TAF)
library(SomePackage)
source("utilities.R")

mkdir("folder")
```

Structured scripts

```
# Read files from previous step
dat1 <- read.taf("previous/file1.csv")
dat2 <- readRDS("previous/file2.rds")
dat3 <- importSpecial("previous/file3.spec")

# Some computations
# [...]

# Write out tables and plots
write.taf(dat4, dir="folder")
write.taf(dat5, dir="folder")
taf.png("file6")
SpecialPlot(dat6)
dev.off()
```

Summary

- 1 **Background** *objectives, design*
- 2 **Running a TAF analysis** *linear regression, boot and run, structured scripts*
- 3 **TAF features** *boot procedure, data flow, new analysis, overview of functions*
- 4 **The TAF community** *browsing an existing analysis, related R packages*
- 5 **Discussion** *contents of a TAF analysis, benefits of TAF*
- 6 **Online examples** *ICES, FAO, SPC, various*