Appendix: R Resources to Implement Network Analysis and Meta-analysis Methods

This appendix presents the computer resources used in this book to implement network analysis and meta-analysis methods. It details the content of the KenSyn R package (R Core Team 2019), developed specifically for this book. This package contains both the data sets used in the examples and the R functions used in the statistical analyses. Some of the functions presented here are essential to implement the mixed model. Other functions facilitate the extraction of information from adjusted models, the analysis of uncertainties associated with estimates or the formatting of results according to conventional standards.

KenSyn Package: R Code and Data Sets of the Examples Presented in the Different Chapters

All the examples presented in this book have been gathered together into a KenSyn R package for "Knowledge Synthesis in Agriculture".

Installation

You can install the KenSyn package from the CRAN repository (http://cran.r-project.org/web/packages/KenSyn), if you have an Internet connection, by following these instructions:

- Step 1: click on "Packages" in the R menu and select "Install the package(s).... ".
- Step 2: Choose the "0-Cloud" location or a location near you from the list of mirror download sites.
- Step 3: Choose "KenSyn" from the list of available packages.

D. Makowski et al., *From Experimental Network to Meta-analysis*,
https://doi.org/10.1007/978-94-024-1696-1

- Step 4: Check the message in the console window indicating that this installation has been successful. Any dependencies are installed.

Another way to install the **KenSyn** package is to execute the following instruction in the R console.

```
install.packages ("KenSyn", rest = "http://cran.rstudio.com")
```

You can also use an R editor like Rstudio (https://www.rstudio.com), which makes it easier to write code, explore results and install R packages.

The current version of **KenSyn** is version 0.3 (April 2019). It is fully functional with the current version of R (version 3.5.3). As **nlme** and **lmer** packages have evolved very regularly, it is recommended to have a recent version of R and it is possible that slightly different results may be obtained depending on the versions of R and the different packages.

In addition to **KenSyn**, depending on the examples, you may need to install other packages used in the different example of this book. They are then loaded at the beginning of the various scripts by the function **library(package_name)**. To do this, proceed in the same way as for **KenSyn** package.

Content and Use

The **KenSyn** package is an R package (a library) that brings together data sets, examples, and functions in a shareable and documented form. This package includes the code in the form of a demo for all the examples that are provided in this book (thus you can run them yourself) and the necessary data sets.

Examples in the Form of Demos

To see all the demo scripts, you can run the following lines

```
library(KenSyn)
demo(package="KenSyn")
```

A table similar to the one presented in Table 1 will be displayed. The beginning of the name (chxxx_) allows you to match the chapters of this book.

Then, to execute a demo script, you must specify the name and execute the following instruction.

```
demo(ch01_mixedmodel_nlme, package="KenSyn")
```

Table 1 List of scripts containing the examples as R demo files in the KenSyn package

Name of the demo	Description
ch01_mixedmodel_nlme	ch01. Simple mixed-effects model example on wheat yield (with nlme and lme4)
ch03_network_compblock	ch03. Network of experiments: complete blocks. Wheat varieties on a single year
ch04a_network_mean_data	Ch04a. Network of experiments: Wheat varieties on a single year. Analysis of means of data
ch04b_network_hetero_var	ch04b. Network of experiments: Wheat varieties on a single year. Heterogeneous variances
ch04c_network_pluriannual	ch04c. Network of experiments: Wheat varieties on several years
ch06a_metaanalysis_organic	ch06. Meta-analysis comparing organic vs conventional cropping system with (nlme and metafor)
ch06b_metaanalysis_metareg	ch06. Meta-analysis: code for illustrating the main principles of meta-regression (with nlme)
ch06c_metaanalysis_metareg_m	ch06. Meta-analysis: code for illustrating the main principles of meta-regression (with metafor)
ch07a_metaanalysis_citrus	ch07. Meta-analysis to estimate the effectiveness of a fungicidal treatment to control *Phyllosticta citricarpa*, a citrus disease (with lme4)
ch07b_metaanalysis_N2O	ch07. Meta-analysis to estimate the response of N2O emissions to the applied N fertilizer rate (with nlme)
ch07c_metaanalysis_bayesian	ch07. Meta-analysis with Bayesian approach: comparing organic vs conventional cropping system (with MCMCglmm)
others_network_machines	Others. Network of experiments: complete blocks. Test of worker performance on machine
others_network_var_itk	Others. Network of experiments: Wheat varieties with different crop managements
others_network_var_soil	Others. Network of experiments: Wheat varieties with 2 different soils

The script is then executed step by step in an interactive way. Simply press the *Enter* key on the keyboard in the console area to execute the following instructions and see the results appear as they are computed, either in the console area or in the graphics area.

It is quite convenient to get the r script directly. This script is stored in the installation directory of the package on your computer, you can know the path with the following instruction, remembering to add the extension .r after the name of the example.

```
system.file("demo", "ch01_mixedmodel_nlme.r", package = "KenSyn")
[1] "D:/XXXXXX/Documents/R/win-library/3.5/KenSyn/demo/
ch01_mixedmodel_nlme.r"
```

Table 2 List of data sets in the KenSyn package

Data set	Description
N2O	Meta-analysis dataset to estimate the response of N2O emissions to the applied N fertilizer rate
Citrus	Meta-analysis dataset: estimation the effectiveness of a fungicidal treatment to control *Phyllosticta citricarpa*, a citrus fungus
machines	Workers testing a new machine.
Organic	Meta-analysis dataset on comparison of organic to conventional crop systems.
wheat_var	Network of experiment to evaluate Wheat varieties on one single year
wheat_var_itk	Network of experiment to evaluate Wheat varieties on one single year with different cropping systems
wheat_var_soil	Network of experiment to evaluate Wheatvarieties on two contrasted soils
wheat_var_years	Network of experiment to evaluate Wheat varieties on several years (2005–2010)
wheatyield	Wheat Yield at a regional scale (fake data)

You can then open it in your code editor and save it in your working directory.

Another way is to access the example code via the R help interface. To do this, you execute the instruction.

```
help(KenSyn)
```

The package help page appears in your browser. You must then go to the very bottom of this page, to access the *index*. Then a "*Code Demo*" link allows you to access the entire list of examples as a demo. This script can then be copied to work on it. Note that this access does not work under current R studio version.

Data Sets

To see all the data sets used in the different examples and mobilized in the demo scripts, you can execute the following instruction.

```
library(KenSyn)
data(package="KenSyn")
```

A table similar to the one presented in Table 2 is then displayed.

Then by launching a specific dataset help, you have a brief description of the origin and structure of the data.

```
help(wheat_var)
```

The data set is accessible as soon as the package is loaded and can be used, for example, as follows:

```
# see the structure
str(wheat_var)
# see the first lines of the data set
head(wheat_var)
# or a histogram with the values of all the yields
hist(wheat_var$rendement)
```

Implement the Mixed-Effects Model Under R

Adjust a Mixed-Effects Model

Two main R packages are available to implement the mixed-effects model under R. These are packages that are widely used and well maintained by the community. Nevertheless, the evolution dynamics of these packages has remained strong in recent years and in a number of cases we regularly face code problems related to incompatibilities between versions.

To fit a mixed-effect model, there are two packages that are widely used in the R community.

nlme Package

The package **nlme** (Nonlinear Mixed-Effects Models, Pinheiro et al. 2017) is dedicated to the adjustment and comparison of linear and non-linear Gaussian mixed-effects models.

- The generic **lme** function (Linear Mixed-Effects Models) allows non-linear mixed-effects models (fixed and random) to be adjusted according to the formulation described by Laird and Ware (1982) by allowing nested random effects. Intra-group errors can be correlated and/or have unequal variances. It returns an object containing the fitted model.

In the example above, we model the variable *Rdt* by an intercept with a random effect on this intercept by *Site*. The data set containing the variables is *wheatyield*.

```
lme(Rdt~1, random=~1|Site, data= wheatyield)
```

- The **groupedData** function allows you to structure a dataset according to a precise formula and the object created is then directly usable by modeling functions as **lme**.

  In the example above, we have the same example as before, but with the preliminary step of structuring the data.

```
gData= groupedData(Rdt~1|Site, data= wheatyield)
lme(gData)
```

lme4 Package

The **lme4** package (Linear, generalized linear, and nonlinear mixed models, Bates et al. 2015) is dedicated to the adjustment and analysis of mixed-effect models, with different functions for each type of model.

- The **lmer** function is used to adjust linear mixed-effect models.

  The example shows you the same simple model with the lmer function.

```
lmer(Rdt~1+(1|Site), data= wheatyield)
```

- The **glmer** function is used to adjust generalized linear models with mixed effects.
- The **nlmer** function is used to adjust non-linear mixed-effect models.

Work with the Results of the Mixed Models Under R

Basic Features

Various generic functions allow to extract useful information in a readable format from objects resulting from mixed-effect model adjustments (but also from other models)

Let's save the result of the adjustment of a model defined with **nlme** or **lmer** into an object named *fit*.

```
fit = lme(Rdt~1, random=~1|Site, data= wheatyield)
```

- **summary** formats the main results of a fitted model in a readable form.

```
summary(fit)
```

- **fitted** retrieves the adjustments by the model.

```
fitted(fit)
```

- **residuals** retrieves the residuals.

```
residuals(fit)
```

- **ranef** display random effects.

```
ranef(fit)
```

- **predict** make predictions with the model, on the same data as used for the adjustment. In this case, this corresponds to the results of the fitted function.

```
predict(fit)
```

Or **predict** with new data, with in this case, the site effect that can be taken into account if the site was in the data set used to fit the model. Thus, in the example, the prediction for data from site 3 is computed.

```
predict(fit, newdata= data.frame(Site=3,Rdt=NA))
```

Emmeans Package

The **emmeans** package (Lenth 2017) computes the estimated marginal means (EMM) for specified factors or combinations of factors in a linear model and, possibly, comparisons or contrasts between them. EMMs are also known as "least square means". It replaces the obsolete Ismeans package.

- **emmeans** calculates adjusted means for each level of a factor.
- **pairs** performs all 2–2 comparisons between means using the "**tukey**" method to adjust the probability of tests to take into account the multiplicity of tests.
- **cld** build homogeneous groups.
- **contrast**, used with the "**eff**" method, calculates all the differences between each level and the overall mean. The "**sidak**" adjustment method adapts the probability of tests to take into account the multiplicity of tests. Used with the argument **method="trt.vs.ctrl"** and **ref=n**, contrast allows to compare all levels to level n.
- **confint** computes confidence intervals of the differences between the levels and a control level or the confidence intervals for the effects of a variable, depending on the **contrast** method.

This package requires the **pbkrtest** and **lmerTest** packages for some tests.

Car Package

The car package (Companion to Applied Regression) contains functions and example of a book (Fox and Weisberg 2011). The **Anova** function is mainly used to calculate the analysis of variance table (with a capital A, different from the basic **anova** function).

Outliers Package

The **outliers** package (Komsta 2011) contains tests commonly used to identify outliers and in our case it can be used to detect suspicious residues.

The Metafor Package, to Performing Meta-analyses Under R

The **metafor** package (A Meta-Analysis Package for R) provides a complete collection of functions for performing meta-analyses in R (Viechtbauer 2010). The package includes functions to calculate different effect sizes or results frequently used in meta-analyses. It allows the user to adjust different models with fixed, random or mixed-effects. Finally, it provides analyses and graphs traditionally used in meta-analyses.

- **escalc** is used to calculate the effect size.
- **rma.uni** adjusts linear models for meta-analyses with fixed, random or mixed-effects.
- **forest** allows you to plot the forest graph with the effect sizes very often used in meta-analysis.
- **funnel** allows you to draw a plot to evaluate possible publication biases.

Bayesian Approach with the Mixed-Effect Model

MCMCglmm Package

The **MCMCglmm** package (MCMC Generalised Linear Mixed Models, Hadfield 2010) is used to implement generalised linear mixed-effects models in a Bayesian framework using Markov Chain Monte Carlo (MCMC). The **MCMCglmm** function applies the Markov Chain Monte Carlo sampler to adjust generalized linear mixed-effects models.

Coda Package

The **coda** package (Plummer et al. 2006) provides functions to summarize and plot the chains sampled by Markov Chain Monte Carlo methods and to perform diagnostic tests of convergence to the equilibrium distribution of the Markov chain. The **gelman.plot** function plots the evolution of the Gelman and Rubin factor as a function of the number of iterations to diagnose graphically the convergence of the MCMC algorithm.

References

Bates D, Maechler M, Bolker B, Walker S (2015) Fitting linear mixed-effects models using lme4. J Stat Softw 67(1):1–48. https://doi.org/10.18637/jss.v067.i01

Fox J, Weisberg S (2011) An R companion to applied regression, 2nd edn. Sage, Los Angeles

Hadfield JD (2010) MCMC methods for multi-response generalized linear mixed models: the MCMCglmm R package. J Stat Softw 33(2):1–22. www.jstatsoft.org/v33/i02

Komsta L (2011) Outliers: Tests for outliers. R package version 0.14. https://CRAN.R-project.org/package=outliers

Laird NM, Ware JH (1982) Random-effects models for longitudinal data. Biometrics 38:963–974

Lenth R (2017) emmeans: Estimated marginal means, aka least-squares means. R package version 0.9.1. CRAN.R-project.org/package=emmeans

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2017) nlme: linear and nonlinear mixed effects models_. R package version 3.1-131, CRAN.R-project.org/package=nlme

Plummer M, Best N, Cowles K, Vines K (2006) CODA: convergence diagnosis and output analysis for MCMC. R News 6:7–11

R Core Team (2019) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. www.R-project.org

Viechtbauer W (2010) Conducting meta-analyses in R with the metafor package. J Stat Softw 36(3):1–48. www.jstatsoft.org/v36/i03