Pre-course material - HARMONY / SBED conference - Brussels, Belgium, 31st January 2024

Homework and software installation for the advanced training school on Bayesian methods

Matthew Denwood, Eleftherios Meletis, Giles Innocent

2024-01-13

Overview

This document will help you to prepare for the advanced training school on Bayesian methods being held in Brussels, Belgium on 31st January 2024. The workshop will be oriented around practical exercises, so it is very important that you are using a computer with the necessary software installed, and that you are sufficiently familiar with R so that you can use it for basic tasks. You will also need to download the teaching material from our GitHub site, so you might find it helpful to have created an account with GitHub before the course so that you can access the material more easily during the week (including any updates we make). Finally, the advanced training workshop will start from the assumption that you have already taken one of the basic training schools that we have provided previously. The purpose of this document is to ensure that you have the necessary software installed and access to the GitHub repository where the teaching material will be made available.

Software-Package installation

You need to install R (version 4.3.0 or later) from https://cran.r-project.org/ and we recommend that you also use Rstudio which can be downloaded separately from https://www.rstudio.com/products/rstudio/download/

Please also install the latest versions of the following R packages from CRAN:

```
packages <- c("tidyverse", "PriorGen", "rjags", "runjags",
                    "coda", "TeachingDemos", "knitr", "ggdag")
if(length(find.package(packages)) != length(packages)){
  install.packages(packages)
}
```

You will also need the standalone JAGS software (version 4.3.1 or later) for the course -

download the installer for your platform from here: https://sourceforge.net/projects/mcmc-jags/files/JAGS/4.x/

Note for Windows

It is very important that you have R version 4.2.x or 4.3.x installed along with JAGS version 4.3.1 (and NOT 4.3.0), otherwise you will not be able to run JAGS from R. If you are not able to install these versions of the software then please let us know in advance of the course.

Note for macOS

If you are using an M1 (Apple Silicon, aka arm64) mac, then you should use the official installer of JAGS 4.3.1 as this will run natively on both x86_64 and arm64 hardware. If you are using an older Intel mac then you can use either JAGS 4.3.0 or JAGS 4.3.1 (there will not be a difference between the two for your hardware). This is the same irrespective of which version of R you use.

Note for Linux

It is very important that the compiler toolchain matches between R and JAGS - if you are installing from an official repository then this should be taken care of for you, but if you are compiling yourself then you do need to ensure that the same compilers (C++ and Fortran) are used for both.

Checking installation

To check that you have installed the software correctly please run the following code within R (or RStudio) and make sure that no errors are produced:

```
stopifnot(getRversion() >= "4.1.0")
stopifnot(require('tidyverse'))
stopifnot(require('PriorGen'))
stopifnot(require('rjags'))
stopifnot(require('runjags'))
stopifnot(require('coda'))
stopifnot(require('TeachingDemos'))
stopifnot(packageVersion("runjags") >= numeric_version("2.2.1-7"))
stopifnot(testjags()$JAGS.available)
stopifnot(numeric_version(testjags()$JAGS.version) >= "4.3.1")
stopifnot(testjags()$rjags.found)
stopifnot(numeric_version(testjags()$rjags.version) >= "4-13")
```

If you have any difficulties installing the software or get errors from the code above, please let us know immediately so that we can resolve these before the start of the course.

GitHub

Basics

GitHub is an online code repository that in it's most basic form stores the version history of any code project you are working on, and allows you to go back in time to a previous version before someone (perhaps you!) introduced a bug. It is particularly useful when collaborating with others because it allows you to see who made a change, when it was made, and what the code looked like before that. It also allows changes from different people to be merged into the same central repository to ensure that nobody gets out of sync with everybody else's code.

We will primarily be using GitHub as a way to disseminate the lecture notes and R code for the exercises on course, so you only need to use the most basic features of GitHub (but it is a good thing to learn).

Simple Web Usage

We have created a public repository containing the teaching material for the training workshop. This means that anyone can view the teaching material at any time via the following website: https://github.com/ku-awdc/harmony-sbed-advanced

You should see a number of folders listed with names like "S0_Preparation", "S1_Material" etc, and within each of these folders you will find material for each of the training school sessions. For each session there are two files: a .pdf file showing just the presentation that we will give for the didactic session, and a .html file that also includes the exercises and solutions for these exercises. You can click on any of these files to view them, although of the different file formats the .pdf version is probably easiest to download directly from the website. There are two problems with this:

1 - You will likely encounter problems when copy/pasting R code from the PDF files, whereas copy/pasting R code from html files is much more reliable.

2 - If/when any of the files are updated, you will not get an automatic update of the new version.

We therefore recommend that you follow the instructions below to clone the repository directly to your computer.

Creating an Account

If you have never used GitHub before, you should create an account via http://github.com - it is free and easy. Remember to make a note of the username and password that you just created!

Using Git

There are two main ways to use Git on your machine: GitHub Desktop and RStudio.

1. GitHub Desktop is a free standalone application provided by GitHub that is designed to integrate seamlessly with your GitHub account by simply installing the application and signing in using your GitHub account credentials. However, this does require you to have administrative rights on your computer, so may not be possible if you e.g. must install software via an IT portal at your institution.
2. You may be most used to using RStudio to edit R code, but it can also be used to sync changes with Git. The advantage of using RStudio is that you will already have it installed, but the disadvantage is that configuration/setup can be trickier than with GitHub Desktop.

Each of these seems to work best for different people, so we will give instructions for both here - but you only need to have one of the two methods working for the workshop.

Method 1: GitHub Desktop

Go to https://desktop.github.com and download/install the software. Then open it and sign in using your GitHub account (the username and password that you just created). This should become the primary way in which you interact with GitHub, rather than via your browser.

The first step is to clone the `harmony-sbed-advanced` repository on to your computer. From inside GitHub Desktop, go to the File menu then Clone Respository, then click on the URL tab. Then type/paste `ku-awdc/harmony-sbed-advanced` into the box where it says `Filter your reposotories` or `Repository URL` or `GitHub username and repository` (depending on your version of GitHub Desktop). A suggested local path folder will be created automatically at the bottom but feel free to change this. Then click on Clone and wait for the files to download.

Creating the clone copies all of the files currently on the GitHub repository to your computer at the local path specified above (if you can't remember what this was choose 'Show in Windows/Finder' or similar, depending on your OS, under the Respository menu). This is where all of the course material will be located. For example, look inside the `'S0_Preparation'` folder and open up the `'Session_Preparation.html'` file - that is a local copy of the same document you are currently reading! But now you also have the PDF version - use this if you want to print the document for some reason, but it is a good idea to stick to using the HTML version if you want to copy/paste R code (you will probably encounter problems with quotation marks if you copy/paste R code from PDF files). The two versions should be identical.

Modifying Files Once you have set up the local copy of the repository, you can then add, delete and modify any of the files within your local copy of the repository. Try doing this by deleting the `'Session_Preparation.html'` file. Now go back to GitHub Desktop where you should see a line appear on the left with a red minus symbol in a box to the right hand side - this is telling you that a file has been deleted locally (if you had modified the file rather than deleting it, the box would be orange and contain a dot). However, you don't want to delete or modify any of the files we are providing in case we update them later. If you do this by

mistake, just right-click the relevant line in GitHub desktop and choose "Discard changes" - the file should then be restored. Do this now for the `'Session_Preparation.html'` file and check that it has reappeared. If you do want to modify a file for some reason, we suggest that you copy it and modify the copied version. If you keep the copy inside the same folder then GitHub desktop will show the new file (green plus in the box) but you can safely ignore these, or move the copied file outside of the repository folder if you want to keep things simple.

Fetching Updates An important advantage of GitHub is that we can update a file online (for example to fix a typo or add new material discussed during the workshop), and that update will be immediately available to all of you. But in order to receive the updated files you need to tell GitHub desktop to look for updates. Open GitHub desktop, then click on 'Fetch origin' and then 'Pull' to make sure that any recent changes are synced to your computer. As long as you remember to pull changes regularly then you will always have an up-to-date copy - so do this regularly. Forgetting to do this is the only real potential downside of Git.

Method 2: RStudio

Unless you already have Git installed on your system, you will need to install it in order to be able to use Git within RStudio. There are a number of ways of doing this, depending on your system and access privileges, but the easiest method is to install GitHub Desktop from https://desktop.github.com (this will also install Git as a command line tool). If you are unable to install GitHub desktop, then see the following links for some alternative options:

- https://github.com/git-guides/install-git
- https://happygitwithr.com/install-git.html

Cloning a repository inside RStudio The easiest way to clone the `harmony-sbed-advanced` repository on to your computer is via RStudio. Go to the `File` menu and select `New project`. It will open and wizard menu, giving you three alternative ways to create a project: select `Version Control` and then `GIT`. Then type/paste `https://github.com/ku-awdc/harmony-sbed-advanced.git` into the box where it says `Repository URL`. A suggested local path folder will be created automatically at the bottom but feel free to change this. Then click on `Clone` and wait for the files to download.

Creating the clone copies all of the files currently on the GitHub repository to your computer at the local path specified above. This is where all of the course material will (eventually) be located. You can browse the content using Windows File Explorer or going to the `Files` browser tab in RStudio (usually located in the lower right window). For example, look inside the `'S0_Preparation'` folder and open up the `'Session_Preparation.html'` file - that is a local copy of the same document you are currently reading! But now you also have the PDF version - use this if you want to print the document for some reason, but it is a good idea to stick to using the HTML version if you want to copy/paste R code (you will probably encounter problems with quotation marks if you copy/paste R code from PDF files).

Remember to select the option `View in web browser` when opening an html file from inside RStudio.

The next time you want to open the cloned repository, go to the `File` menu inside RStudio and select `Open project`. Locate your project and click `open`.

Modifying Files Once you have set up the local copy of the repository, you can then add, delete and modify any of the files within your local copy of the repository. Try doing this by deleting the `'Session_Preparation.html'` file. Now go to the `Git` tab in RStudio (usually located in the top right window) where you should see a line appear with a *D* letter in a box to the left hand side - this is telling you that a file has been deleted locally (if you had modified the file rather than deleting it, the box would be blue and contain a *M*). However, you don't want to delete or modify any of the files we are providing in case we update them later. If you do this by mistake, just toggle the box corresponding to the relevant line (the D will move one step to the left), right click and select `Revert` - the file should then be restored. Do this now for the `'Session_Preparation.html'` file and check that it has reappeared. If you do want to modify a file for some reason, we suggest that you copy it and modify the copied version. If you keep the copy inside the same folder then it will appear as a modification (blue M in the box) but you can safely ignore these, or move the copied file outside of the repository folder if you want to keep things simple.

Fetching Updates An important advantage of GitHub is that we can update a file online (for example to fix a typo or add new material discussed during the workshop), and that update will be immediately available to all of you. But in order to receive the updated files you need to tell RStudio to look for updates. In the `Git` tab in RStudio click on 'Pull' (blue arrow facing downwards) to make sure that any recent changes are synced to your computer. As long as you remember to pull changes regularly then you will always have an up-to-date copy - so do this regularly. Forgetting to do this is the only real potential downside of Git.

More information on Git

That is pretty much all you need to know for the training workshop but there are some good tutorials available, for example: https://happygitwithr.com.