SATVI Computational Course

Session Guide

SATVI Computational Group

05 MAR 2024

Table of contents

Preface

This is a session guide book for the SATVI Computational Course.

This is a version-controlled living document that will be updated as needed as the course progresses. All changes are tracked using git.

1 Introduction

Welcome to the SATVI Computational Course! This course is designed to strengthen fundamental coding skills for SATVI trainees and staff. The curriculum will take you through the basics of R, using the terminal, creating and using git controlled projects, as well as more advanced data analysis methods commonly used at SATVI.

All lessons will be stored on the SATVI GitHub under the repository SATVI_ComputationalCourse. Navigate to the course using this link: https://github.com/SATVILab/SATVI_ComputationalCourse

Your instructors will be SATVI members with experience in each topic. For session-specific questions, please contact the relevant instructor:

```
Carly Young-Baile: carly.young-bailie@uct.ac.za

Simon Mendelsohn: simon.mendelsohn@uct.ac.za

Monika Looney: monika.looney@uct.ac.za

Anele Gela: anele.gela@uct.ac.za
```

The full curriculum can be found on the "Syllabus" page.

Happy coding!

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

2 Installations

This document provides installation guides for basic programming tools.

2.1 R

R is a commonly used coding language for computational biologists and immunologists. Many software packages and analysis pipelines depend on R. R is also a computational environment used for computing and generating graphics.

To install R for Windows or Mac, follow the instructions provided by The Comprehensive R Archive Network (CRAN) found here: https://cran.r-project.org/

It is recommended to download the precompiled binary distribution appropriate for your machine.

To learn more about R, read the following introduction provided by CRAN: https://www.r-project.org/about.html

2.2 R Studio

RStudio is an integrated development environment (IDE) based on R. It provides a user-friendly option for building code and can incorporate multiple languages including python, which is also commonly used by computational immunologists.

To donwload and install RStudio Desktop, follow this link and the provided instructions: https://posit.co/download/rstudio-desktop/#download

3 swirl

swirl (https://swirlstats.com/) is an interactive R package that helps you self-teach the basics of R. It is run from directly from the R console.

This session guide follows the instructions provided by swirl. Visit the following link to access the full tutorial: https://swirlstats.com/students.html

You can also find the full swirl course tutorial on GitHub at https://github.com/swirldev/swirl_courses

3.1 Install swirl

swirl requires R 3.1.0 or later installed on your computer. It is also recommended that you have RStudio installed which will provide a user-friendly environment to work with.

For instructions on how to install R and RStudio, visit the Installations session guide page.

Once you have downloaded R and RStudio, perform the following steps:

```
1. Open RStudio.
2. In the RStudio console, type the following where you see the command prompt > :

    install.packages("swirl")
```

3.2 Initialize swirl

Whenever you want to run swirl, you must load and initialize the package.

```
1. In the console, type the following:

    library("swirl")
    swirl()

2. Follow any prompts that come up in the console. i.e. if swirl asks "What shall I call you
```

3.3 Install an interactive course

The first time you initialize swirl, you will need to install a course.

For the SATVI Computational Course, we recommend that those who are new to coding start with "R Programming". This course will cover the basics of programming in R.

There are many courses to choose from, so those who are more advanced may opt for an intermediate or advanced course to work through in their own time. A repository with all available swirl courses can be found here: https://github.com/swirldev/swirl_courses#swirl-courses.

There is also an expansive swirl Network that expands further on open sourse interactive R lessons. You can access the Network and associated courses or become a swirl course author here: https://swirlstats.com/scn/

To install a course that is not part of the swirl course repository, type the following into the console:

```
?InstallCourses
```

3.4 Run swirl

For now, we will assume that we are starting with the basics and have chosen to install the "R Programming" course.

To run the interactive lessons:

Select a new lesson. The R Programming course offers 14 different short interactive lessons. Go through each one in order as the information from earlier lessons is required in later lessons.

3.5 Exit swirl

If at any time you need to exit a swirl lesson before it is complete, simply press the Esc key.

If you need to exit from a prompt, exit and save your work by typing: bye()

3.6 Interactive commands

While you are working in swirl, you may find that you want to skip a section that you are already comfortable with, or to work more on the current topic outside of an interactive session.

Below are some helpful commands for getting the most out of your swirl sessions:

From the R prompt (>):

```
To skip the current question: skip()

To experiment with R on your own without swirl interaction: play()

To re-initiate swirl interaction after playing: nxt()

To exit and save: bye()

To return to swirl's main menu: main()

To display these command options: info()
```

If you see a swirl output followed by … press Enter to continue.

3.7 Homework

As beginners, regular practice is critical! It is recommended that you go through one or two lessons daily to improve and retain these fundamentals.

Over the next week, in your own time, complete the 14 short interactive lessons from the "R Programming" swirl course.

3.8 FAQ

Q1: Can funcitons learned in swirl be applied when writing my own R scripts?

```
A: Absolutely! The functions that you use in swirl are all base R functions that can be used
```

Q2: If I need to use an R package, do I need to install the package each time I start a new session?

```
A: Nope! Once a package is installed, you do not have to re-install when you open a new R se
```

4 MaRcus R Training

The Marcus R Training program was developed by Hasse Walum of Emory University. The program will cover the following:

1. Importing data
2. Basic data visualization
3. Exporting and saving plots
4. Data transformation
5. R Markdown basics
6. Summarizing data
7. String manipulation and data joining

Rather than reinventing what is covered in the Marcus R Training program, we have been granted permission to use the materials for our SATVI Computational Course.

Over the next 6 weeks, we will refer to the Marcus R Training materials for our sessions.

4.1 Content access

The course and all associated resources are available at:

https://haswal.github.io/MaRcus/index.html

4.2 Homework

Please refer to the MaRcus R Training program session guides to access your homework assignments.

4.3 FAQ

Q1: What are the best ways to set your working directory?

```
A: There are a few ways to do this:

    1. If you are using Mac, you can navigate to the directory you would like to work in usi

    2. You can also set the working directory using point and click in RStudio. To do so, na

    3. A note about setting working directories in scripts. It is good practice to avoid usi
```

Q2: When generating a plot using ggplot2, does the name used in the script for the row or column we want to plot have to match the col or rowname of the associated dataframe exactly?

```
A: Yes. The names must match exactly because R searches the dataframe for col or rownames as
```

Q3: What is the difference between facet_wrap() and facet_grid()?

```
A: Both are options that can be applied to ggplot2. facet_wrap() wraps a 1d sequence of pane
```

Q4: When should I specify aes globally vs. locally?

```
A: In general, specify aes in mapping (global) so that the specifications are applied to all
```

Q5: What are HEX codes?

```
A: HEX codes are unique alphanumeric codes assigned to specific colors. They can be used to
```

Q6: What are your recommendations for using Chat GPT for help with coding?

```
A: Chat GPT is a quickly growing tool used by coders. It can be very helpful for designing /
```

5 Summary

In summary, this book has no content whatsoever.

References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.