Capacity Building in Seasonal Hydrological Forecasting

Introduction to R and Programming Basics

AGRHYMET, Climate Regional Center for West-Africa and Sahel

@Arsène KIEMA

2025-10-06



i Learning outcomes

By the end of this session, you will be able to:

• Describe what R is and where it shines (and where it doesn't).

i Learning outcomes

By the end of this session, you will be able to:

- Describe what R is and where it shines (and where it doesn't).
- Install and configure R, RStudio, and a Miniconda/JupyterLab environment.

i Learning outcomes

By the end of this session, you will be able to:

- Describe what R is and where it shines (and where it doesn't).
- Install and configure R, RStudio, and a Miniconda/JupyterLab environment.
- Run R code in scripts and notebooks.

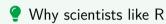
i Learning outcomes

By the end of this session, you will be able to:

- Describe what R is and where it shines (and where it doesn't).
- Install and configure R, RStudio, and a Miniconda/JupyterLab environment.
- Run R code in scripts and notebooks.
- Prepare your workstation for the rest of the training.

R is a free, open-source programming language (mid-1990s, lhaka & Gentleman) widely used for statistics, data analysis, data visualization, and scientific computing.





• World-class statistical modeling and graphics.

- Why scientists like R
 - World-class statistical modeling and graphics.
 - Huge ecosystem: CRAN (20k+ packages), Bioconductor, rOpenSci.

- Why scientists like R
 - World-class statistical modeling and graphics.
 - Huge ecosystem: CRAN (20k+ packages), Bioconductor, rOpenSci.
 - Strong culture of **reproducible research** (Quarto/R Markdown).

• Open-source, cross-platform.

- Open-source, cross-platform.
- Rich packages for time series and spatial / spatio-temporal analysis (sf, stars, terra).

- Open-source, cross-platform.
- Rich packages for time series and spatial / spatio-temporal analysis (sf, stars, terra).
- Elegant data manipulation with **tidyverse** (dplyr, tidyr).

- Open-source, cross-platform.
- Rich packages for time series and spatial / spatio-temporal analysis (sf, stars, terra).
- Elegant data manipulation with **tidyverse** (dplyr, tidyr).
- Publication-quality graphics (ggplot2) and interactive maps (leaflet, tmap).

• Mature tooling for hydrological time series and spatial hydrology.

Key takeaway

- Mature tooling for hydrological time series and spatial hydrology.
- Packages you'll see in this training:

Key takeaway

- Mature tooling for hydrological time series and spatial hydrology.
- Packages you'll see in this training:
 - tidyverse, data.table (data wrangling)

Key takeaway

- Mature tooling for hydrological time series and spatial hydrology.
- Packages you'll see in this training:
 - tidyverse, data.table (data wrangling)
 - sf, terra, stars (GIS & rasters)

Key takeaway

- Mature tooling for hydrological time series and spatial hydrology.
- Packages you'll see in this training:
 - tidyverse, data.table (data wrangling)
 - sf, terra, stars (GIS & rasters)
- tidymodels, airGR, HYPEtools, and WASS2SHydroR (our focus)

Key takeaway

We support two workflows: **RStudio-centric** and **JupyterLab via Miniconda**. Choose one (you can have both).

Option A — R + RStudio (simple desktop setup)

• Install **R** from CRAN: https://cran.r-project.org/

First run checklist

We support two workflows: **RStudio-centric** and **JupyterLab via Miniconda**. Choose one (you can have both).

Option A — R + RStudio (simple desktop setup)

- Install R from CRAN: https://cran.r-project.org/
- Install RStudio Desktop: https://posit.co/download/rstudio-desktop/
- First run checklist

We support two workflows: **RStudio-centric** and **JupyterLab via Miniconda**. Choose one (you can have both).

Option A — R + RStudio (simple desktop setup)

- Install R from CRAN: https://cran.r-project.org/
- Install RStudio Desktop: https://posit.co/download/rstudio-desktop/
- First run checklist
 - ullet Open RStudio o Console o run install.packages("tidyverse").

We support two workflows: **RStudio-centric** and **JupyterLab via Miniconda**. Choose one (you can have both).

Option A — R + RStudio (simple desktop setup)

- Install R from CRAN: https://cran.r-project.org/
- Install RStudio Desktop: https://posit.co/download/rstudio-desktop/
- First run checklist
 - ullet Open RStudio o Console o run install.packages("tidyverse").
 - Set a project folder: $File \rightarrow New \ Project$.

Option B — Miniconda + JupyterLab (R & Python together)

Install Miniconda: https://docs.conda.io/en/latest/miniconda.html

```
conda create -n wass2s_hydro -c conda-forge -y \
   r-base=4.3 r-irkernel \
   r-tidyverse r-sf r-stars r-terra r-data.table \
   jupyterlab nodejs
conda activate wass2s_hydro
jupyter lab
```

Option B — Miniconda + JupyterLab (R & Python together)

- Install Miniconda: https://docs.conda.io/en/latest/miniconda.html
- Oreate an environment with R + JupyterLab:

```
conda create -n wass2s_hydro -c conda-forge -y \
  r-base=4.3 r-irkernel \
  r-tidyverse r-sf r-stars r-terra r-data.table \
  jupyterlab nodejs
conda activate wass2s_hydro
jupyter lab
```

Verifying Your Setup

Run these in a fresh R session (RStudio Console or Jupyter cell):

```
R. version. string
sessionInfo()
pkgs <- c("tidyverse", "tidymodels", "sf", "terra", "stars",</pre>
          "data.table", "ecmwfr")
installed <- sapply(pkgs, requireNamespace, quietly = TRUE)
data.frame(package = pkgs, installed = installed)
```

Verifying Your Setup



If something is missing

Install missing packages:

install.packages(c("tidyverse", "data.table"))

Working Modes

i Best practice

Use scripts for production code and notebooks for exploration and reporting. Keep data in data/, outputs in outputs/, and code in R/ or notebooks/.

THANK YOU FOR YOUR ATTENTATION

