Using R in hydrology

Programme

8.30	Identification of hydrologic events	Conrad Wasko + Danlu Guo
8.50	Flood forecast verification in R	Andrea Ficchi
9.10	Machine learning for spatio-temporal modelling	Razi Sheikholeslami
9.30	The (mis)use of colours in scientific visualizations	Michael Stoelzle
9.50	Using R in education	Claudia Brauer

Today's short course is co-organised by the Young Hydrologic Society. Do consider joining the group!
www.younghs.com
Twitter: @YoungHydrology

Using R in education

Claudia Brauer

Hydrology and Quantitative Water Management Group,
Wageningen University, The Netherlands

Instructor's problem

"I want to use R in my course,

but I don't want to sacrifice precious practical time

to teach students basic programming skills."

Our famous R intro

A (very) short introduction to R

Paul Torfs & Claudia Brauer
Hydrology and Quantitative Water Management Group,
Wageningen University, The Netherlands

31 January 2018

1 Introduction

R is a powerful language and environment for statistical computing and graphics. It is a public domain (a so called "GNU") project which is similar to the commercial S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S, and is widely used as an educational language and research tool.

The main advantages of R are the fact that R is freeware and that there is a lot of help available on-

`www.r-project.org`

and do the following (assuming you work on a windows computer):

1. click `download CRAN` in the left bar
2. choose a download site
3. choose `Windows` as target operation system
4. click `base`
5. choose `Download R 3.4.3 for Windows`[3] and choose default answers for all questions

It is also possible to run R and RStudio from a USB

The basics in only 10 pages

8 self-study modules

1. A (very) short introduction to R
2. R Programming MOOC first part
3. R Programming MOOC second part
4. Basic plotting
5. Pretty plotting
6. Reading data files
7. Matrix operations
8. Spatial data

1–3 hours each (depending on background)

Self-study modules downloadable from GitHub

github.com/ClaudiaBrauer/A-very-short-introduction-to-R

1. A (very) short introduction to R

The base document, with 10 pages of background and exercises and 2 pages listing useful functions (to use as a reference). Working through this document takes 1 to 2 hours (depending on your background). An old version of this document can also be downloaded from the R website (as contributed document), but the newest version can always be found here.

2. Doing "A (very) short introduction to R" in the interactive swirl environment

Instead of reading the pdf and doing the ToDo exercises, you can also go through the text and exercises in an interactive environment called swirl (developed by swirlstats.com). This short manual gets you started with the (very) short introduction to R. It also points you to some nice follow-up classes created by others.

3. Writing your own R scripts

After learning the basics, you have to gain experience in building R scripts. In this document you learn to set up a script step by step. The examples are from hydrology, but the exercises are useful for everyone.

4. Portable versions of R and RStudio

In case you want to take R everywhere you go (you may want to install the programs on a USB stick in case of administrator rights issues).

5. Examples of R scripts for hydrology

Here we collect scripts for hydrological data analysis, which you can adapt for your own application.

Module 1: the basics

A (very) short introduction to R

Paul Torfs & Claudia Brauer
Hydrology and Quantitative Water Management Group,
Wageningen University, The Netherlands

31 January 2018

1 Introduction

R is a powerful language and environment for statistical computing and graphics. It is a public domain (a so called "GNU") project which is similar to the commercial S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S, and is widely used as an educational language and research tool.

The main advantages of R are the fact that R is freeware and that there is a lot of help available on-

`www.r-project.org`

and do the following (assuming you work on a windows computer):

1. click `download CRAN` in the left bar
2. choose a download site
3. choose `Windows` as target operation system
4. click `base`
5. choose `Download R 3.4.3 for Windows`[3] and choose default answers for all questions

It is also possible to run R and RStudio from a USB

Designed for people who never progammed before

Module 1 – swirl version

Doing

A (very) short introduction to R

in the interactive swirl environment

2.3 Starting swirl

On the swirl website www.swirlstats.com, on the tab "Learn", you'll find the steps you'll have to take to get swirl. You have already installed R and RStudio, so you can skip the first steps and:

1. Open RStudio.
2. Install the swirl package by typing `install.packages("swirl")`
3. Load the package by typing `library(swirl)`
4. Start swirl by typing `swirl()`

2.4 Installing the course

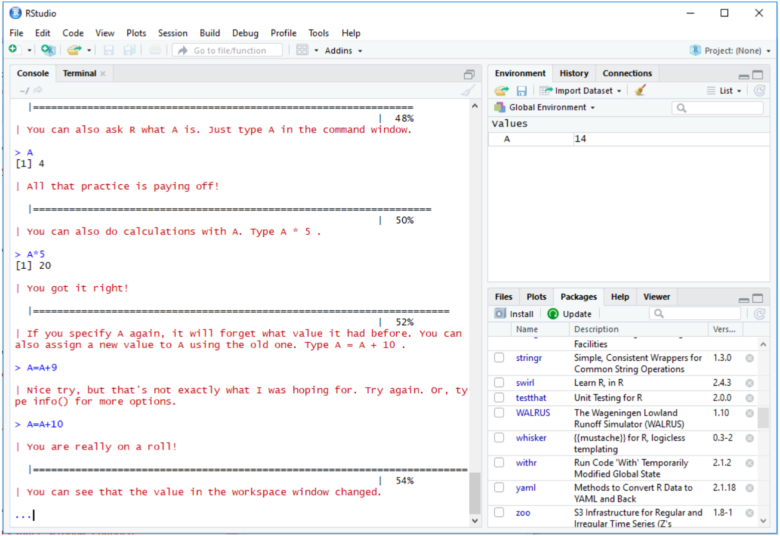
After entering `swirl()`, the program will ask you (interactively) which course you want. If you want a new course, you have to install it first from internet . Type:

```
install_course("A-(very)-short-
introduction-to-R")
```

"A (very) short introduction" is divided into 3 swirl modules, corresponding to Sections 2–3, 5–7 and 8–11 in the pdf version

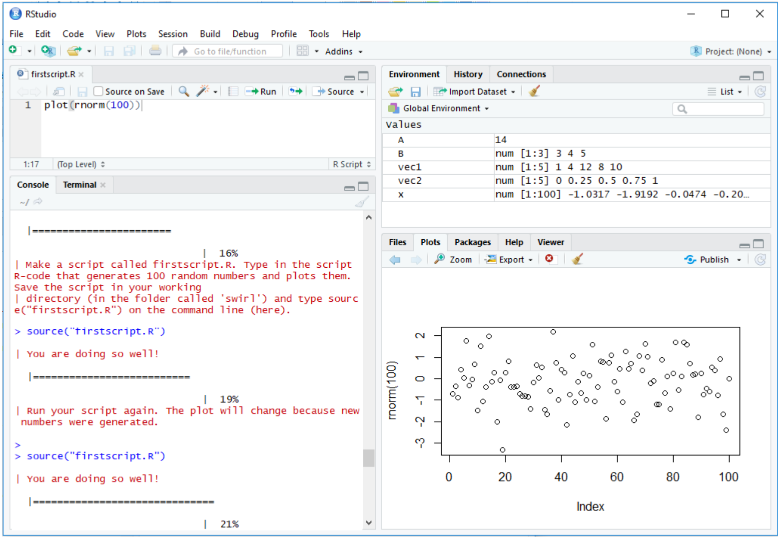
Interactive format: "Learn R, in R", www.swirlstats.com

Swirl examples



Red = instructions, blue = response, black = output.

Swirl examples



Also script building assignments.

WAGENINGEN
UNIVERSITY & RESEARCH

www.github.com/ClaudiaBrauer

Modules 2+3: from R programming MOOC

Swirl exercises made by Johns Hopkins University
for the Coursera MOOC "R programming".
`www.swirlstats.com`

Module 2

1. Basic Building Blocks
2. Workspace and Files
3. Sequences of Numbers
4. Vectors
5. Missing Values
6. Subsetting Vectors
7. Matrices and Data Frames
8. Logic

Module 3

9. Functions
10. lapply and sapply
11. vapply and tapply
12. Looking at Data
13. Simulation
14. Dates and Times
15. Base Graphics

Example from class: BSc year 2

Getting started

Make a folder on your computer or OneDrive. Take a moment to think about how you want to organize all files for this course so you can easily find everything later. Save the `zip` file from Brightspace for this practical here and unzip it. Open the script `timeseries.R` in RStudio and take a look at it.

▶ Read Sections 2.3 *RStudio Layout* and 5 *Scripts* in "A (very) short introduction to R" (provided in the `zip` file). Make sure you understand the use of the separate windows.

For the the analyses in this practical, some parts are preprogrammed in R. You have to program other things yourself. Whenever you encounter "..." in the script, you have to type something. Replace "..." with the appropriate commands (or add lines with commands at those locations – you don't have to add code in other places) and run them (don't run the whole script at once).

▶ Read 2.4 *Working directory*.

```
# Aggregate the vectors P, ETpot and Q by date so you get one vector with
# means from January to December.
P_mean     = aggregate(x=data$P,      by=list(month), FUN=mean, na.rm=TRUE)        [,2]
ETpot_mean = aggregate(x=data$ETpot,  by=list(month), FUN=mean, na.rm=TRUE)        [,2]
Q_mean     = ...


# Do the same to get 10% and 90% quantiles for each month.
P_10       = aggregate(x=data$P,      by=list(month), quantile, probs=0.1, na.rm=TRUE) [,2]
ETpot_10   = ...
Q_10       = ...
P_90       = aggregate(x=data$P,      by=list(month), quantile, probs=0.9, na.rm=TRUE) [,2]
ETpot_90   = ...
Q_90       = ...
```

Example from class: BSc year 2

```r
# Load some helper functions
source("helper_functions/functions_check.R")


# Check if you did it right.
check_gauge_series()
```

Environment | History | Connections | Tutorial

Functions

check_gauge_series	function ()
check_mm5min	function ()
check_per_pixel	function ()
check_per_timestep	function ()
hint_cumulative	function ()
hint_Hupsel_pixel	function ()
hint_per_pixel	function ()
hint_per_timestep	function ()

Console | Jobs

```
>
> # Check if you did it right.
> check_gauge_series()
[1] "You correctly extracted the data and converted
 them to mm/h."
>
```

Example from class: MSc year 1

6.4 For-loop

Next, make a function called `run_model` with arguments `forc` (a data frame with forcing data P and ET), `a` and `b`. This function consists of three parts.

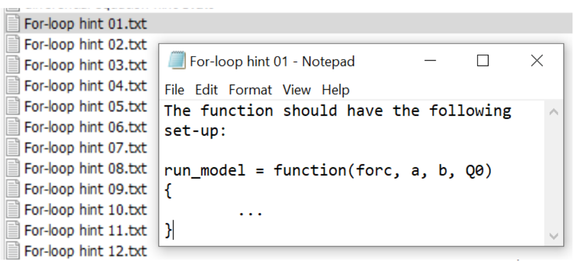
1. Make an empty vector called `x`. Fill the first element of `x` using the observed Q at $t = 0$ (this is the starting value of x), which you can extract automatically from the `forc` data frame.
2. Run a for-loop over all timesteps, starting with timestep number 2 (as you already have the first one). For each step, call the function `runge_kutta` and write the result in the vector `x`. In this way, all elements in the vector `x` will be filled step by step.
3. Compute Q from x and return this modelled discharge.

```
##############
# 6.4 FOR-LOOP
##############

# Write a function with name run_model and arguments forc, a and b.
# The function should:
# - Make an empty vector for x and fill the first element using the observed discharge at t=0.
# - Run a for-loop over all time steps (starting at 2).
# - Return a vector with computed discharges.

...
...
...
```

Example from class: MSc year 1



Modules 4–8: writing scripts

Writing your own R scripts

A follow-up to “A (very) short introduction to R”

Introduction

With the exercises in this document you learn how to set up an R script from scratch. The exercises explain step by step how to build your script. The scripts in these exercises always follow the same structure with the following Sections (see Fig. 1):

Initializing Clear memory, set working directory and load packages.

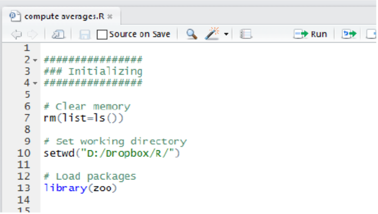
Data Read data from file and data preprocessing (such as extracting and renaming one column of the total data set).

Processing Do computations, run a model, etc.

Output Make a dataframe and/or figure and save these to file.

(different languages). You can also Google your problem (type “R” + keyword / problem) and get solutions (functions, packages, code).

- Save the scripts you made as inspiration for your future (thesis) work.



Examples from hydrology

Example of script-writing module

Basic plotting

Compare rainfall measurements

The aim of this exercise is to reproduce Figure 2. You will build it in steps. Don't forget to run (part of) the code every time you add something, to check if everything is still error-free.

1. Download the file `P_gauge_radar.dat` and save it on your computer. Make sure that the folders for scripts, input (data files) and output (figures and data files) are logically structured.
2. Start with an empty script. Make the main headers `Initializing`, `Data`, `Processing` and `Output`). Don't forget the #-sign to indicate that they are comments and not commands (see Sec. 5 in "A (very) short introduction to R").
3. Under `Initializing`, add commands to clear R's memory and set the working directory (see Fig. 1).
4. Add code to read the data file (Section `Data`).

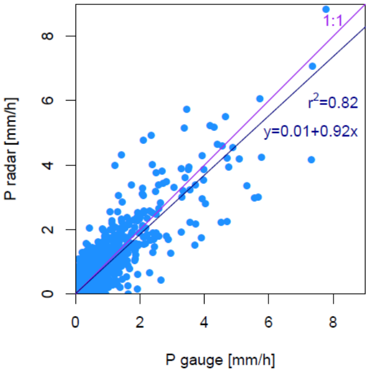


Figure 2: Comparing rainfall measured with a radar to rainfall measured with a rain gauge. Figure source: C.C. Brauer, A. Overeem, H. Leijnse, R. Uijlenhoet (2016): The effect of differences between rainfall measurement techniques on groundwater and discharge simulations in a lowland catchment, Hydrol. Process., 30, 3885–3900

Difficulty increases from module 4 to 8.

Repository for code

github.com/soil-water-atmosphere/notebooks/tree/master/R_for_hydrology

soil-water-atmosphere / **notebooks** Public

R scripts for hydrological data analysis

This folder contains some scripts that can be useful for data analysis. These are plain R scripts (not interactive Jupyter notebooks) that you can download and open in R. You can download the data necessary to run the scripts from the *data* subfolder. In the *figures* and *output* subfolders you can see what the scripts create (and try to reproduce that yourself).

The scripts are made for hydrological analyses, but most of them can be applied just as easily to examples from other fields. You can use them in courses, for theses (BSc, MSc and PhD) or any other programming project.

If you don't know how to build scripts yet, do self study modules 4-8 on the GitHub site of A (very) short introduction to R.

Note that this repository is dynamic: we will keep adding scripts. Suggestions are welcome!

ClaudiaBrauer Add files via upload

- data
- figs
- output
- Gumbel_distribution.R
- baseflow_index.R
- bootstrap.R
- combined_barplot.R
- compare_time_series.R
- discontinuous_timeseries.R
- distributions.R
- fit_surface.R
- flow_duration_curve.R

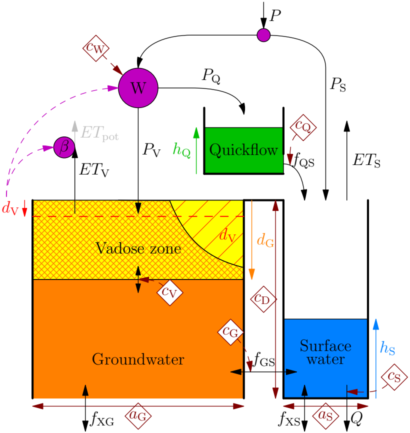
Extra: tutorial for rainfall-runoff model

WALRUS:

- Wageningen Lowland Runoff Simulator
- Simple, parametric model for catchments with shallow groundwater

Tutorial:

- Learn to understand and work with the R package
- For students, water managers, consultants and researchers
- R-package, user manual, publications and more on `www.github.com/ClaudiaBrauer/WALRUS`



Contents WALRUS tutorial

Summary

We developed:

- 8 modules of 1-3 hours:
  1. A (very) short introduction to R

  2–3 Swirl exercises from R programming MOOC (Johns Hopkins U.)

  4–8 Writing scripts from scratch
- Repository with example scripts
- Tutorial for rainfall-runoff model WALRUS
- Everything freely available via `www.github.com/ClaudiaBrauer`

Using R in hydrology

Programme

8.30	Identification of hydrologic events	Conrad Wasko + Danlu Guo
8.50	Flood forecast verification in R	Andrea Ficchi
9.10	Machine learning for spatio-temporal modelling	Razi Sheikholeslami
9.30	The (mis)use of colours in scientific visualizations	Michael Stoelzle
9.50	Using R in education	Claudia Brauer

Do you want to contribute to next year's edition?
Let us know on GitHub or email claudia.brauer@wur.nl.