"R in Hydrological Modelling: Why we should try it ?

Mauricio Zambrano Bigiarini
PhD candidate, 3rd year

Dep. of Civil and Env. Engineering
University of Trento, Italy

mauricio.zambrano@ing.unitn.it

July 08th, 2009

Overview

Objective

To present some features and packages that make of **R** a powerful **environment** for **pre-processing** and **analysing** input data of **hydrological models** and **post-processing** its results. In particular, examples are taken from using **R** to analyse data of a large river basin (85000 km2).

Overview

Objective

To present some features and packages that make of **R** a powerful **environment** for **pre-processing** and **analysing** input data of **hydrological models** and **post-processing** its results. In particular, examples are taken from using **R** to analyse data of a large river basin (85000 km2).

Some areas that take advantage of R's features:

- Batch reading of input files

Overview

Objective

To present some features and packages that make of **R** a powerful **environment** for **pre-processing** and **analysing** input data of **hydrological models** and **post-processing** its results. In particular, examples are taken from using **R** to analyse data of a large river basin (85000 km2).

Some areas that take advantage of R's features:

- Batch reading of input files
- Exploratory data analysis

Overview

Objective

To present some features and packages that make of **R** a powerful **environment** for **pre-processing** and **analysing** input data of **hydrological models** and **post-processing** its results. In particular, examples are taken from using **R** to analyse data of a large river basin (85000 km2).

Some areas that take advantage of R's features:

- Batch reading of input files
- Exploratory data analysis
- Time series management and analysis

Overview

Objective

To present some features and packages that make of **R** a powerful **environment** for **pre-processing** and **analysing** input data of **hydrological models** and **post-processing** its results. In particular, examples are taken from using **R** to analyse data of a large river basin (85000 km2).

Some areas that take advantage of R's features:

- Batch reading of input files
- Exploratory data analysis
- Time series management and analysis
- Geostatistics and spatial analysis

Overview

Objective

To present some features and packages that make of **R** a powerful **environment** for **pre-processing** and **analysing** input data of **hydrological models** and **post-processing** its results. In particular, examples are taken from using **R** to analyse data of a large river basin (85000 km2).

Some areas that take advantage of R's features:

- Batch reading of input files
- Exploratory data analysis
- Time series management and analysis
- Geostatistics and spatial analysis
- GIS & RDBMS linkage

Overview

Objective

To present some features and packages that make of **R** a powerful **environment** for **pre-processing** and **analysing** input data of **hydrological models** and **post-processing** its results. In particular, examples are taken from using **R** to analyse data of a large river basin (85000 km2).

Some areas that take advantage of R's features:

- Batch reading of input files
- Exploratory data analysis
- Time series management and analysis
- Geostatistics and spatial analysis
- GIS & RDBMS linkage
- Goodness-of-fit between observed and simulated values

Overview

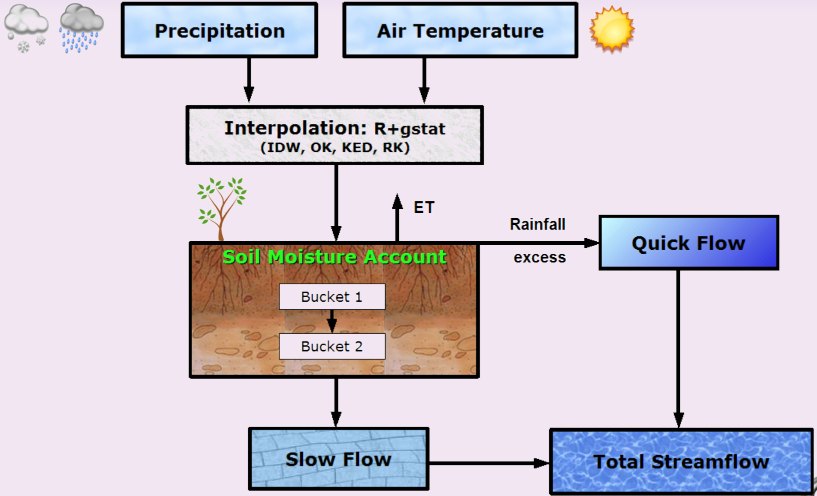
Objective

To present some features and packages that make of **R** a powerful **environment** for **pre-processing** and **analysing** input data of **hydrological models** and **post-processing** its results. In particular, examples are taken from using **R** to analyse data of a large river basin (85000 km2).

Some areas that take advantage of R's features:

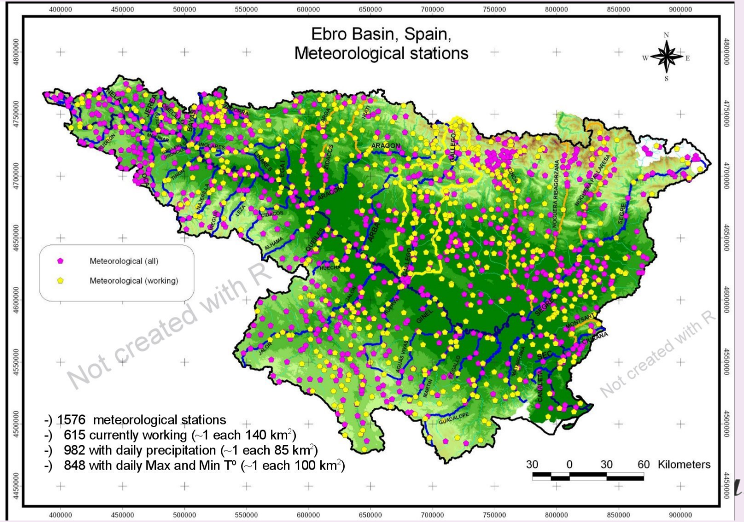
- Batch reading of input files
- Exploratory data analysis
- Time series management and analysis
- Geostatistics and spatial analysis
- GIS & RDBMS linkage
- Goodness-of-fit between observed and simulated values
- Easy re-use of already developed functions/procedures (scripts/packages)

Hydrological Modelling



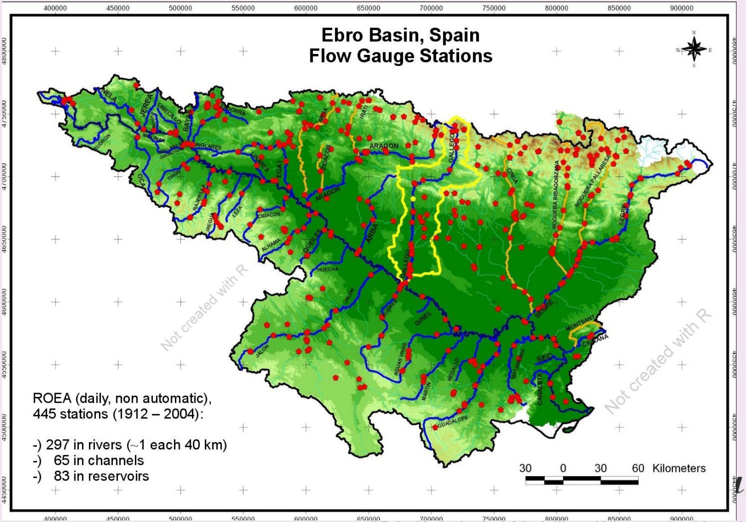
The problem

1576 **meteorological stations** with daily data from 1912-2004



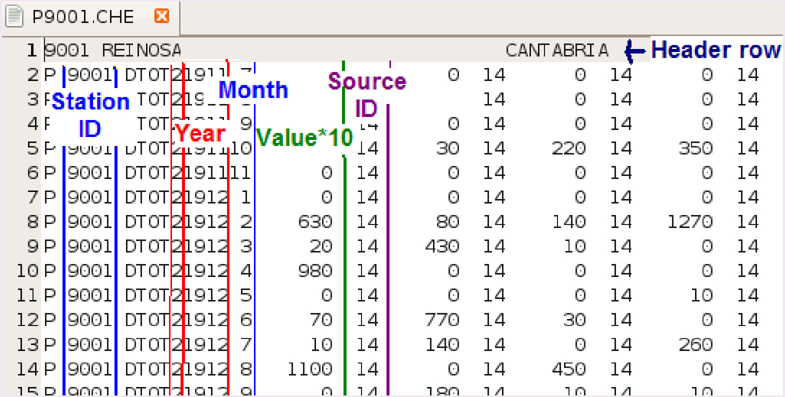
The problem (cont.)

445 **streamflow stations** with daily data from 1912-2004



Batch reading and data organization

Thousands of raw data → 1 *data.frame* (`base::list.files`, `utils::read.fwf`)



Batch reading and data organization (cont.)

Thousands of raw data → 1 *data.frame*
(`base::list.files`, `utils::read.fwf`)

```
# Creating a list with all the FILES with an extension equal to 'file.ext'
# in the directory specified by 'drty'
files <- list.files(path= drty, pattern=paste(".", file.ext, "$", sep=""), ignore.case =TRUE )
            .
            .
# Reading the raw data file
# c(11,1,4,2) = 11 positions for descriptive string; 1 for number of digits in the variable; 4 for
# rep(c(7,4),31) = repeat 31 times (maximum number of days/month) the "7 4" values;
#                7 positions (including the leading space) for the Value of the variable and 4
# skip=1 for skipping the first comment line
pp <- read.fwf(p_che_filename, widths=c( c(11,1,4,2), rep(c(7,4),31) ), skip=1)
```

Batch reading and data organization (cont.)

Matrix notation for subsetting data (numeric, dates, factors...)

```
Ini <- "1961-01-01" #01-Jan-1961
End <- "1961-02-03" #03-Feb-1961
Window <- seq(from=as.Date(Ini), to= as.Date(End), by="days")

x.ts[x.ts$Date %in% Window, c(1:3, 5, 10:12)]
         Date P9001 P9008X P9015 P9041 P9044 P9048
 1961-01-01   0.2     10    NA   0.0    NA    NA
 1961-01-02   7.7     19    NA  20.3    NA    NA
 1961-01-03   2.1      0    NA   0.0    NA    NA
 1961-01-04   7.2     15    NA  22.0    NA    NA
 1961-01-05   0.2      0    NA   0.0    NA    NA
 1961-01-06   5.3     19    NA  10.0    NA    NA
 1961-01-07   1.4      1    NA   1.1    NA    NA
 1961-01-08   0.8      0    NA   0.0    NA    NA
 1961-01-09   0.2      5    NA   5.6    NA    NA
 1961-01-10  17.3     18    NA   0.0    NA    NA
 1961-01-11  19.4     12    NA  26.0    NA    NA
 1961-01-12   0.0      0    NA   0.0    NA    NA
 1961-01-13   0.0      0    NA   0.0    NA    NA
 1961-01-14   0.0      0    NA   0.0    NA    NA
```

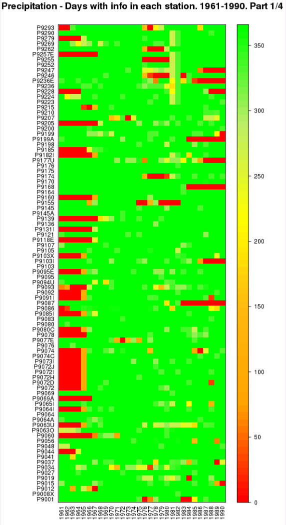
Batch reading and data organization (cont.)

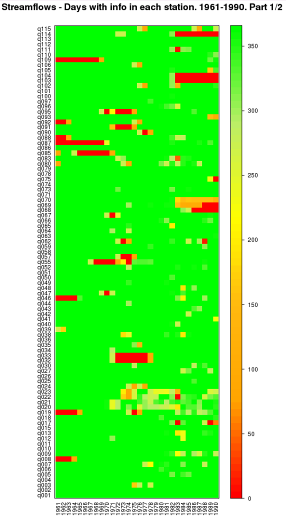
Easy **summary** of the time series stored in each station, within a target period (`base::summary`)

```
> a <- x.ts[x.ts$Date %in% Window, c(1:3, 5, 10:12)]
> summary(a)
      Date                P9001             P9008X             P9015          P9041
 Min.   :1961-01-01   Min.   : 0.00    Min.   : 0.00    Min.   : NA    Min.   : 0.00
 1st Qu.:1961-01-10   1st Qu.: 0.00    1st Qu.: 0.00    1st Qu.: NA    1st Qu.: 0.00
 Median :1961-01-18   Median : 0.40    Median : 0.00    Median : NA    Median : 0.00
 Mean   :1961-01-18   Mean   : 2.81    Mean   : 4.41    Mean   :NaN    Mean   : 3.59
 3rd Qu.:1961-01-26   3rd Qu.: 3.88    3rd Qu.: 9.50    3rd Qu.: NA    3rd Qu.: 4.47
 Max.   :1961-02-03   Max.   :19.40    Max.   :19.00    Max.   : NA    Max.   :26.00
                                                        NA's   : 34
```

Visual summary of available data

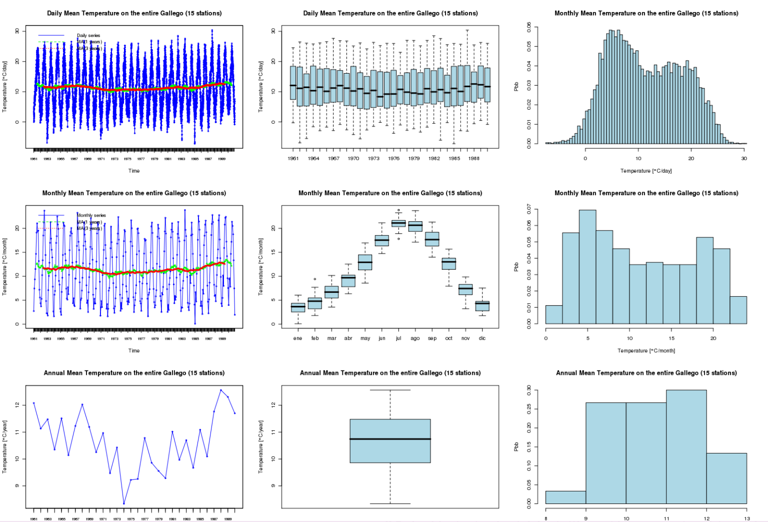
Days with information per station and year (`lattice::levelplot`)





Daily, monthly and annual plots

zoo::plot.zoo; graphics::boxplot, hist + customization



Filling in missing data on stations

Following Teegavarapu et al. (1985), a modified Inverse Distance Weighted **IDW** algorithm was used for filling in the missing daily data on each station, using the **Pearson's product-moment coefficient** instead of the spatial distance as the weight:

$$R_m = \frac{\sum_{i=1}^{N} R_i \cdot \theta_{m,i}}{\sum_{i=1}^{N} \theta_{m,i}}$$

where:

- R_m: Missing daily precipitation on station m
- $\theta_{m,i}$: CC between the time series of the target station m and the station i with a known value
- R_i: Known daily precipitation on station i
- N: Number of neighbours with the highest CC to be considered (personal contribution, unpublished)

Filling in missing data on stations (cont.)

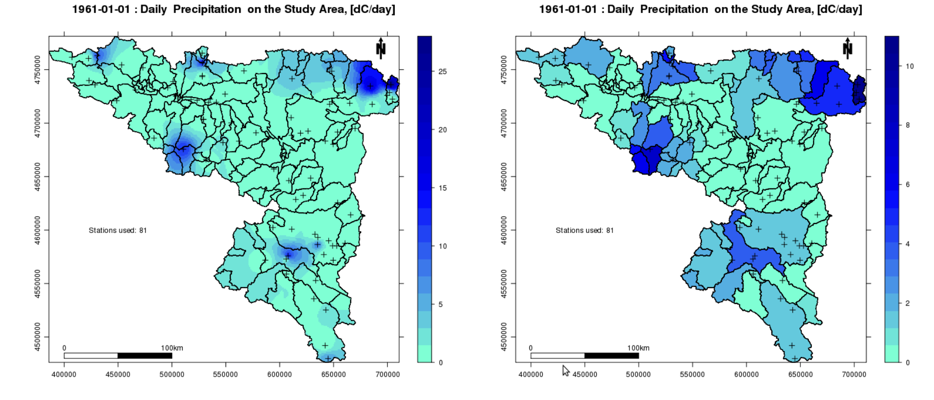
```
# The function 'interpol1' is within the 'lib_TSA_in_HydrologicalModelling.R' libr
for (s in 1:nstations) {

  starting.date <- pp.ts[1, "Date"]
  ending.date   <- pp.ts[ndates, "Date"]

  sname <- pp.gis.catch$INDICATIVO[s]

  # Printing a message that indicates the date that is being interpolated
  print( paste("station:", sname, ",", s, "of", nstations, "; Dates:",
         starting.date, "-", ending.date, ";", ndates, "days", sep=" ") )

  pp.ts.idw[1:ndates, s] <- sapply(1:ndates, function(i,x,y,z) {

  # Putting the interpolated values into the corresponding row, given by 'i',
  # of 'pp.ts.idw'
  z[i, s] <- interpol1(x, y, i, s, method="cc-neighs", n.neighs)

  }, x=pp.ts.catch, y= cc, z=pp.ts.idw)

} # FOR 's' end
```

Mean Precipitation on Subcatchments

Modified Block IDW:

1. IDW over a **square grid** with cells of 1 km2 (`maptools::readShapePoly; sp::spsample`)
2. Only the 5 nearest neighbours (with data) are considered
3. For each day, the **mean value** in each one of the 120 subcatchments is computed, averaging over all the cells belonging to each sub-catchment `gstat::krige`

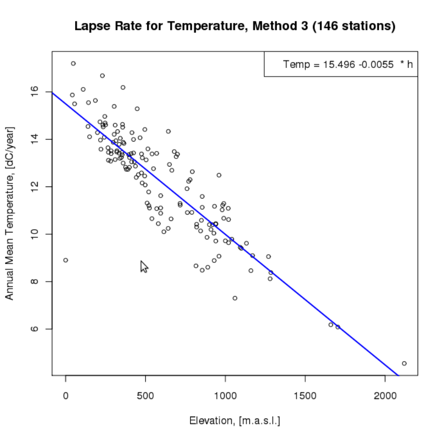
Mean Precipitation on Subcatchments



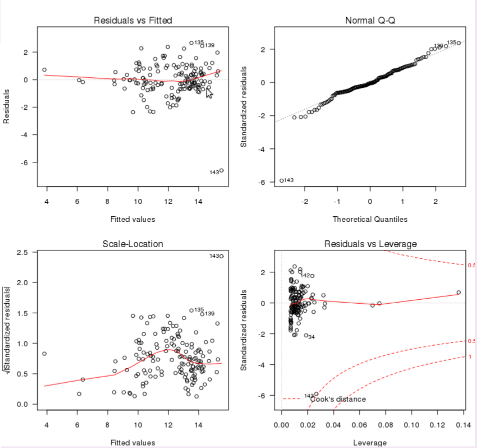

```
# Defining a sampling GRID. If grid.type="regular', then the grid is made of squared cells of 'cell.size'
# meters x 'cell.size' meters with regular spacing.
catchment.grid <- spsample(SubCatchments.shp, type=grid.type, cellsize=cell.size, offset = c(0.5, 0.5))

# Making possible that the grid can be used in the interpolations:
gridded(catchment.grid) <- TRUE

# Interpolating with the INVERSE DISTANCE WEIGHTED, , using the
# 'N.max' nearest neighbours, 'N.min' minimum number of station and  Dist.Max' maximum distance
x.idw <- krige(value~1, locations=x.work, newdata=catchment.grid, nmin=N.min, nmax=N.max, maxdist=Dist.Max)
```

Lapse rates computation

Linear model for temperature (*stats::lm*):

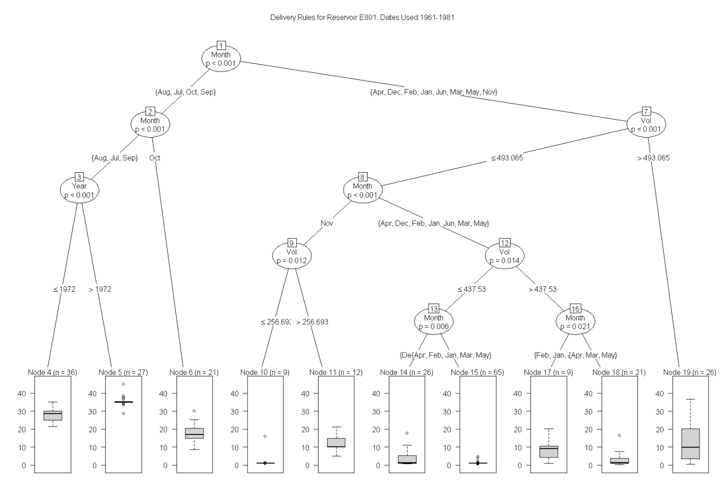


Residuals:



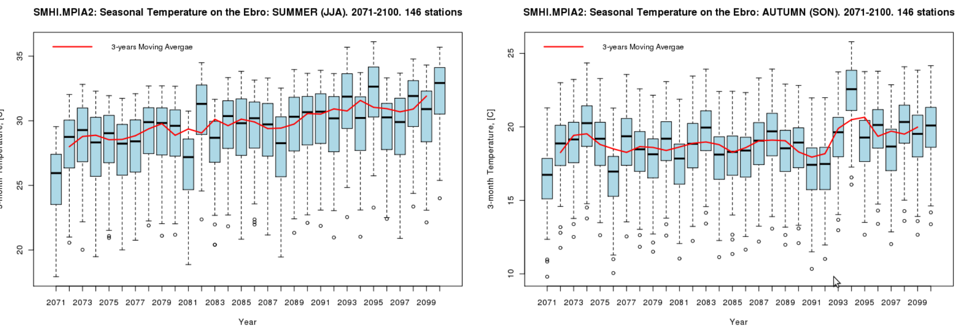
Reservoir Rules

`party::ctree` was used for getting the **monthly delivery** of the reservoir as a function of the **month** and the **stored volume**



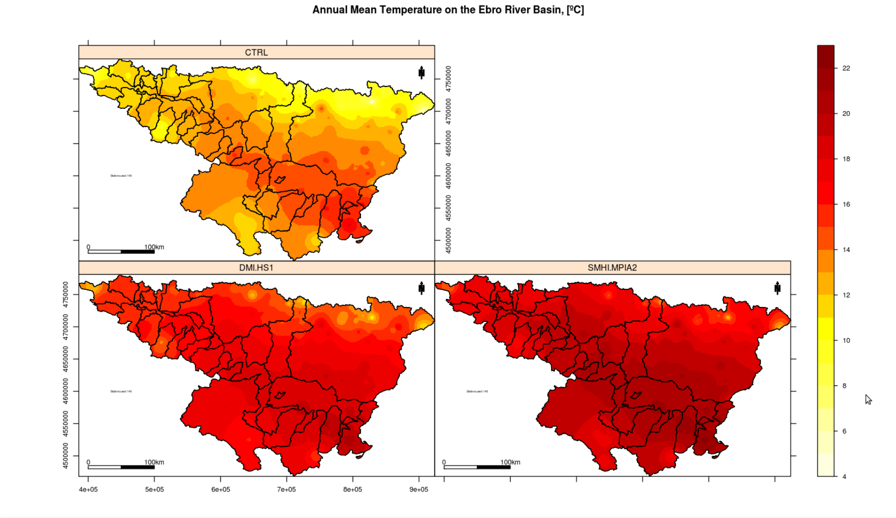
Seasonal evolution of temperature

graphics::boxplot, lines + customization:



Comparison of spatio-temporal patterns

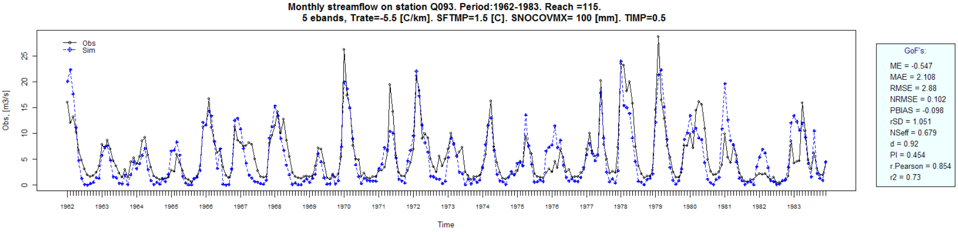
`sp::spplot` + customization:

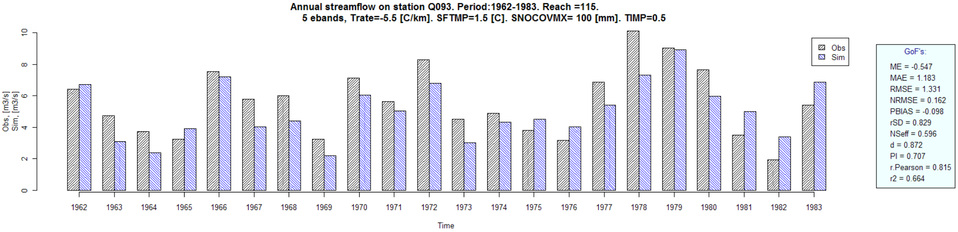


Reading output files with fixed format

```
read.out.rch <- function(drty=getwd(), sim.tstep) {

  # Checking that the user provides a 'sim.tstep' value for the 'output.rch' file
  if (missing(sim.tstep)) {
        stop("Missing argument value: 'sim.tstep' must be in c('Daily','Monthly','Annual')")
  } else  # Checking the validity of the 'unit' argument
         if ( is.na( match(sim.tstep, c("Daily", "Monthly", "Annual") ) ) ) {
            stop("Invalid argument value: 'sim.tstep' must be in c('Daily', 'Monthly', 'Annual')" ) }

  rch.names <- c("TYPE", "RCH", "GIS", "MON", "DrAREAkm2", "FLOW_INcms", "FLOW_OUTcms",
                 "EVAPcms", "TLOSScms", "SED_INtons", "SED_OUTtons", "SEDCONCmg/kg",
                 "ORGN_INkg", "ORGN_OUTkg", "ORGP_INkg", "ORGP_OUTkg", "NO3_INkg",
                 "NO3_OUTkg", "NH4_INkg", "NH4_OUTkg", "NO2_INkg", "NO2_OUTkg",
                 "MINP_INkg", "MINP_OUTkg", "Algae_INkg", "Algae_OUTkg", "CBOD_INkg",
                 "CBOD_OUTkg", "DISOX_INkg", "DISOX_OUTkg", "SOLPST_INmg",
                 "SOLPST_OUTmg", "SORPST_INmg", "SORPST_OUTmg", "REACTPSTmg",
                 "VOLPSTmg", "SETTLPSTmg", "RESUSP_PSTmg", "DIFFUSEPSTmg",
                 "REACBEDPSTmg", "BURYPSTmg", "BED_PSTmg", "BACTP_OUTct",
                 "BACTLP_OUTct", "CMETAL#1kg", "CMETAL#2kg", "CMETAL#3kg")

  rch.widths <- c(6,4,9,6,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,
                  12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,12,
                  12,12,12,12)

  # Adding the path to the filename
  fname <- paste(drty, "/", "output.rch", sep="")

  # Reading the output file of the simulation
  rch <- read.fwf(fname, widths= rch.widths, header=FALSE, skip=9, sep = "\t")

  colnames(rch) <- rch.names
```

Streamflows: Simulated v/s Observed





Why a hydrological modeller should invest time in trying R ?

1. Models, graphics and analysis can be easily tailored to particular needs
2. Many ready-to-use algorithms
3. Write once use many times
4. Huge and active user community
5. Documentation is available in several languages
6. Multi-platform (GNU/Linux, MacOS, Windows)
7. Open Source
8. Free :)

Why a hydrological modeller should invest time in trying R ? (cont.)

Other useful areas/packages (not discussed here):

1. **Geostatistics** (`automap`, `geoR`, `geoRglm`, `fields`, `spBayes`, `RandomFields`)
2. **GIS** (`spgrass6`, `RSAGA`, `RGoogleMaps`, `rgdal`, `mampproj`)
3. **Wavelet analysis** (`wavelets`)
4. **HPC** (`jit`, `NWS`, `Rmpi`, `snow`, `taskPR`, `multicore`)
5. **Programming language interfaces** (C, Fortran, Python, Perl, Java...)
6. **Optimization** (`optim`)
7. Linkage to **Spreadsheets & DB** (`RExcelInstaller`, `RPostgreSQL`, `RMySQL`, `RSQLite`)
8. Linkage to other **statistical software**, e.g: S, SAS, SPSS, Stata, Systat (`foreign`)
9. **Bayesian statistics**

Summary

R

Can be thought as an **environment** that **provides** the latest research developments in (spatio-temporal) statistics to efficiently **tackle** *most* of the **practical problems** that reality poses to the hydrological modeller

Where to start?

1. http://cran.r-project.org/manuals.html
2. http://cran.r-project.org/web/packages/
3. http://addictedtor.free.fr/graphiques/
4. http://www.statmethods.net/index.html
5. http://r-spatial.sourceforge.net/
6. http://casoilresource.lawr.ucdavis.edu/drupal/node/438
7. http://www.rseek.org/

Thanks !

Questions ?