Accessing R from Fortran

Mark Thyer
Michael Leonard
Dmitri Kavetski
Stephen Need

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

THE UNIVERSITY OF ADELAIDE AUSTRALIA
SUB CRUCE LUMEN

Introduction

- Fortran 90/95 (F95) is a well-established language
  - many existing programs written in F95
  - ongoing code development
  - users attracted to speed and memory advantages
- R is open-source graphical/statistical package
  - Provides many new statistical algorithms
  - Good graphical support
  - Fast growing list of packages

RFortran is open source software for exploiting R functionality from Fortran

www.rfortran.org

Motivation:
Exploiting R Functions directly from Fortran

- Fortran lacks graphics functionality. Frustrating for
  - debugging or interactively analysing/simulating data
  - generating numerous diagnostic plots
- RFortran enables high-quality graphics from Fortran

- Latest statistical/numerical algorithms released via R packages
- RFortran enables calling R functions from Fortran
  - enables researchers to trial latest algorithms
  - saves time recoding algorithms in Fortran
  - Expedites research results

Why use RFortran?

Current Approach

R platform

R calls Fortran routines via DLLS

Fortran Code (DLL)

- Dynamic link libraries (DLL's) can be painful
- Some programs hard/ impossible to package as DLL

RFortran Approach

Fortran Code (EXE)

Fortran calls R routines via COM

R platform

- Convenient for users with significant F95 code need R's graphics/statistical routines
- Easier to use than DLL's

RFortran Overview

- Suite of Fortran modules
- Uses Component Object Model (COM) interface developed by Microsoft ® to access R

- Core Functionality: 3 Fortran Routines
  - **`Rinit:`** initialise R, COM interface, etc.
  - **`Rput:`** put data/commands from F95 to R
  - **`Rget:`** get data from R into F95

- Numerous other functions for string handling, error logging, missing values, etc.

Example: Graph using R Scripts

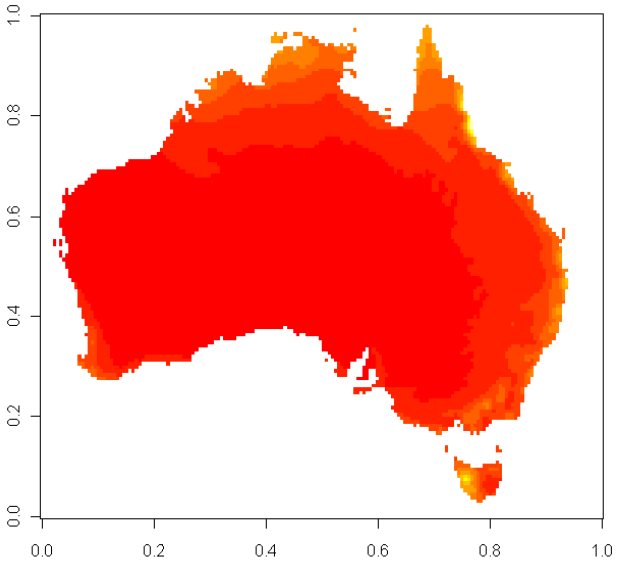
- Scenario: Produce high quality graph using R scripts

```
program AustraliaRainfall_ScriptedGraph
use RFortran                                    ! Access RFortran
! Declare variables and read data - full code at www.rfortran.org

ok = Rinit(Rfortran_ScriptPathIn='/Rscripts/tutorial_examples/')

ok = Rput("lat",lat) ! Rainfall grid
ok = Rput("lon",lon) ! lat/lon values
ok = Rput("coast",coast,mv=(coast<=-9999)) ! Missing vals due to
ok = Rput("x",x,mv=(x<0))                  ! islands/ocean pixels

ok = Rput("AustraliaRainfall_ScriptedGraph(lat,lon,x,coast)")
                                  !Call R script to do plotting
```

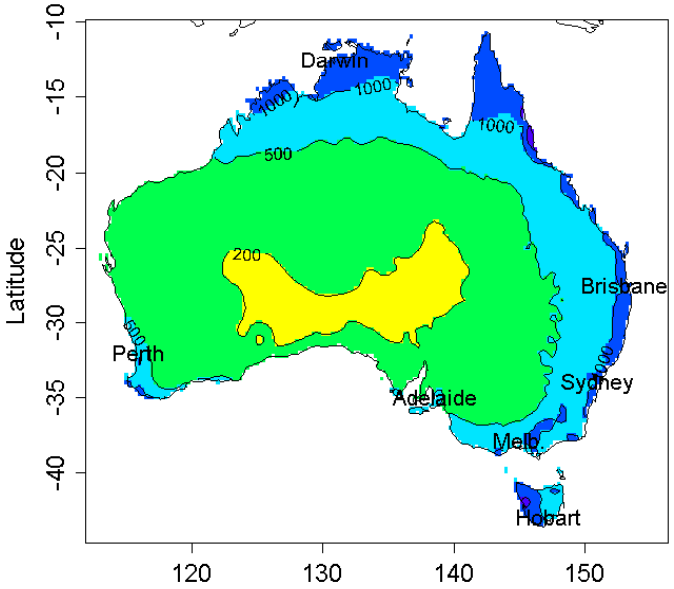
Example: Graph using R Scripts

- Scripting allows detailed command sequences

```
AustraliaRainfall_ScriptedGraph<-function(lat,lon,x,coast){
  image(lon,lat,x, xlab="Longitude",ylab="Latitude",cex.lab=1.5,cex.axis=1.5,
   breaks=c(100,200,500,1000,2000,5000),col=topo.colors(5)[5:1])
   contour(lon,lat,x,levels=c(100,200,500,1000,2000,5000),add=TRUE,labcex= 1)
  lines(coast)
     site.x<-c(115.9,138.6,147.3,145.0,151.0,153.1,130.9)
     site.y<-c(-31.95,-34.92,-42.87,-37.78,-34.0,-27.48,-12.47)
     names<-c("Perth","Adelaide","Hobart","Melb.","Sydney","Brisbane","Darwin")
  text(site.x,site.y,names,cex=1.2)
 }
```



Simple Graph



HQ Graph using R Scripts

Feature List

Core Functionality

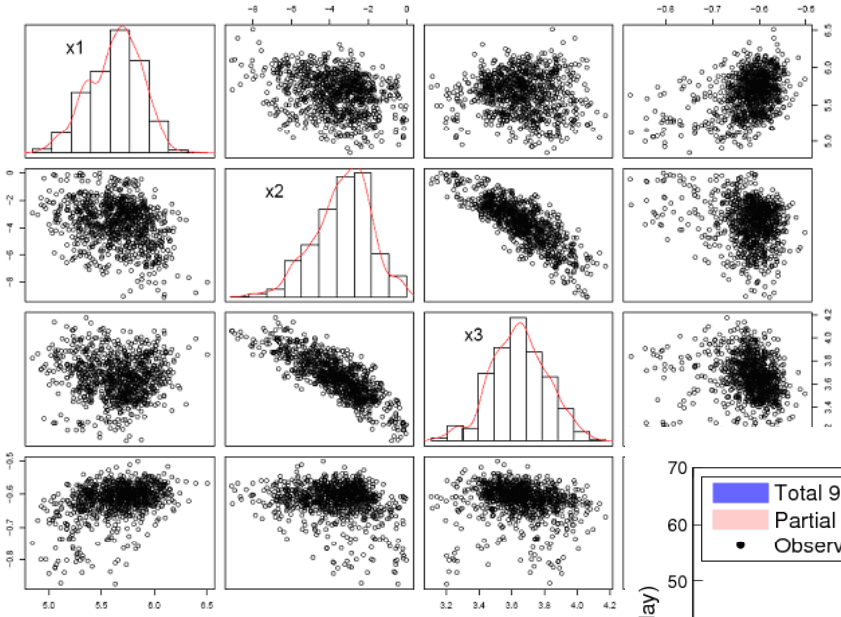
- Initialisation (**`Rinit`**)
  - Auto-opens R and initialises R-Fortran interfaces
- Transfer R commands to R From Fortran
  - Any valid R command can be transferred
- Transfer variables between R and Fortran
  - `Integer, real, logical, character` types supported
  - 1D, 2D and 3D arrays (easily increased)
- Errors – all error messages output to a log file
  - Notifies for syntax errors
  - When transfer variables from R to Fortran notifies for
    - Type, dimension and size mismatches
    - Invalid values (NA, NaN, Inf)

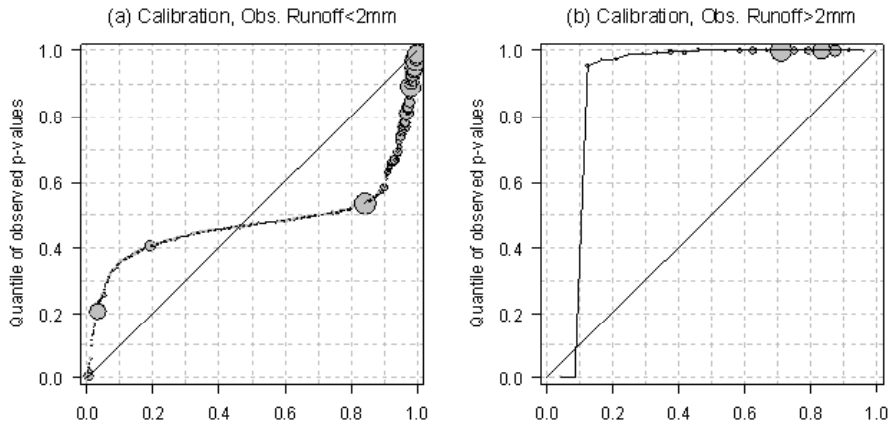
Enhanced Functionality

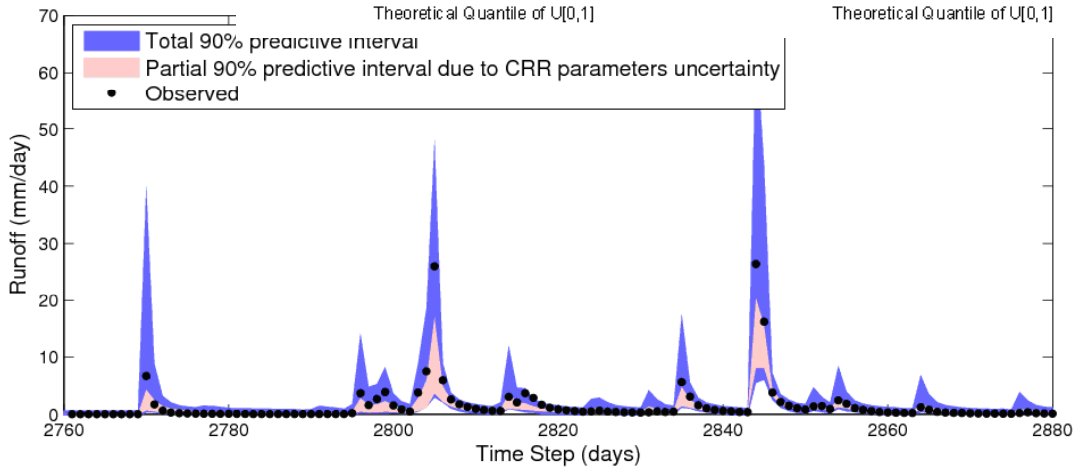
- specialized Fortran functions that perform several tasks (e.g. plot data, add axis labels, legend etc)
- full list available on www.rfortran.org

Case Study 1 – Bayesian Total Error Analysis (BATEA) for rainfall-runoff models

- Used to automatically generate large numbers of diagnostic plots stored in a .pdf file







.org

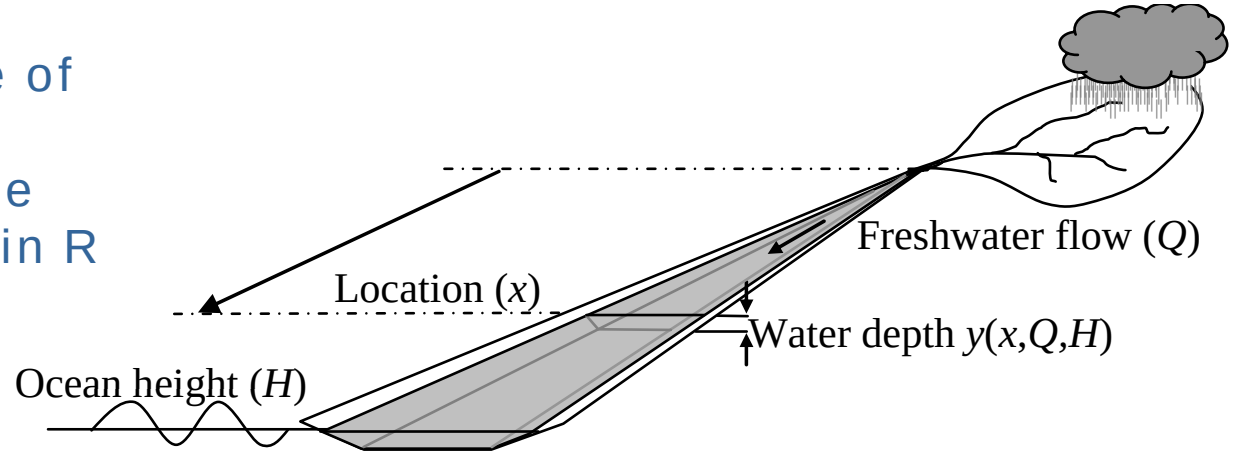
Case Study 1 – BENEFITS

- Easily produce high-quality graphs directly from computational code
- Removes tedium of graphical postprocessing of numerical results
- Single pdf file with minimal user intervention simplifies archiving, enables quick comparison
- Open-source software simplifies sharing of generating code between multiple users

Case Study 2 – Coastal flood estimation

- A hydraulic model is written in F90 with boundary values from a statistical model in R
- Exploit R functionality to expedite research output by trialling alternative algorithms and “proof-of-concept” checks without coding in Fortran

Immediate use of multivariate copula package only available in R



Case Study 2 – BENEFITS

- Exploited strength of each platform
  - Existing numerical hydraulic model in F90
  - Recently developed statistical algos implemented in R (copula)
- Avoids RE-writing
  - copula R-package in F90; or
  - porting the complicated F90 hydraulic model to R
- Able to immediately evaluate different copula models with minimal time-investment.
  - Important when testing proof-of-concept

Case Study 3 – Interactivity

- Debugging Fortran Code
  - Transfer data to R for exploratory plotting and
  - Check Fortran code using R tools

- Runtime Interaction
  - Transfer data to R and produce/update dynamic diagnostic plots
  - Monitor progress of complicated iterative algorithms in Fortran (e.g. MCMC convergence)
  - Early diagnosis of numerical problems

Limitations

- RFortran is Windows only
  - Component Object Model (COM) is a Microsoft-only technology
- Current version only supports
  - Compaq (CVF6.6) compiler or
  - Intel Visual Fortran (IVF) compiler
  - Compiler-independent version planned

Summary

- RFortran provides easy-to-use means to exploit R functionality from Fortran code
- Numerous benefits for Fortran users:
  - Quickly produce high-quality graphics
  - Access latest numerical/statistical algorithms in R
- Available online at www.rfortran.org
  - Download installer
  - Comprehensive wiki (getting started, FAQ, etc)
  - Discussion forum: fortran-users@googlegroups.com
  - Regular updates

Future Features
Forum for Fortran

- Support for more Fortran compilers
- Enhanced functionality including Interactive Graphics
- Prototype of MATFortran
  - Calling MATLAB from Fortran
  - Similar protocols as RFortran

THANKS

www.rfortran.org

RFortran Overview

- RFortran is a suite of modules that are compiled with a Fortran program
- Uses the Component Object Model (COM) interface developed by Microsoft
- RFortran is Windows only.
- Currently can only be used with Compaq (CVF6.6) or Intel Visual Fortran (IVF) compiler
  - Planned to investigate if a compiler independent version can be developed

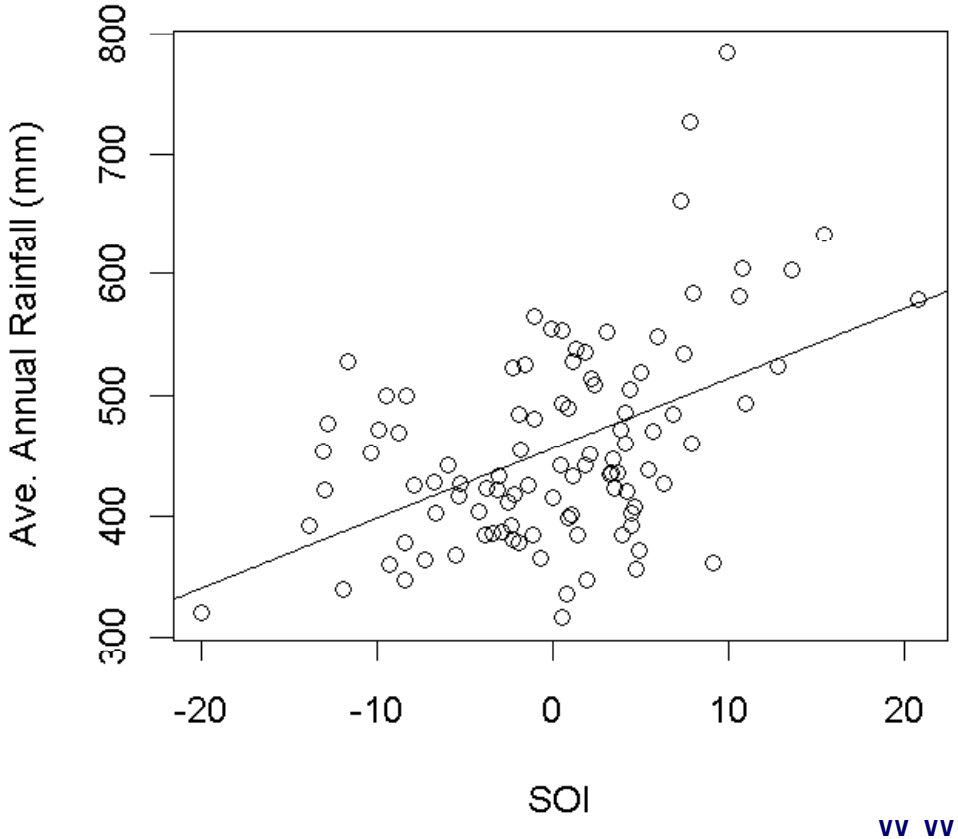
Example: Using R Functions

- Scenario: you want to exploit an R function
- Initialise R (line 11)
- Transfer some data to R (line 12)
- Run some R command (line 13)
- Get results back into F95 (line 14,15)
- Use the results in F95 computations (line 18)

```
! header lines and closing lines as per example 1
! output gives “Intercept 456.5 mm, Slope 5.76 mm, SSE 548497 mm^2”
04  integer(4) :: i,ok                          ! index counter, error flag
05  real(8)    :: yr(102),rain(102),soi(102)    ! Data matrix
06  real(8)    :: par(2),resid(102)             ! Fitted parameters,residuals
! read data here - full code provided at www.rfortran.org
11   ok = Rinit()                                ! Initialise R, no scripts
12   ok = Rput("soi",soi);ok = Rput("rain",rain) ! Transfer data to R
13   ok = Rput("ans<-lm(rain~soi)")              ! Fit a linear model
14   ok = Rget("ans$coefficients",par)           ! Get slope,intercept
15   ok = Rget("ans$residuals",resid)            ! Get slope,intercept
16   ok = Rput("plot(soi,rain,xlab='SOI',ylab='Ave. Annual Rainfall (mm)')")
17   ok = Rput("lines(soi,ans$fitted.values)")
18   write(*,*) "Intercept",par(1),"Slope",par(2),"SSE",sum(resid**2)
```

Example: Using R Functions

- The main point is getting data back to F95
- Intercept = 456.5 mm, Slope = 5.76 mm



Example: Graph using R Scripts

- Scenario: Produce higher quality graph using R scripts
- Load R script(s) (line 19)
- Transfer data (line20 - 22)
- Call the function in the script (line 23)

```
01 program AustraliaRainfall_ScriptedGraph ! Invoke an R script to make a plot.
02  use RFortran                            ! Access RFortan
03  implicit none
04  integer    :: i,j,ok
05  real(8)    :: x(178,139),lon(178),lat(139) ! Data,Longitude, Latitudes
06  real(8)    :: coast(61656,2) ! lat/lon coordinates of Aust. coastline
! read data here - full code provided at www.rfortran.org
17  lon = 112.0+(/(i,i=0,177)/)*0.25          ! 178 x 0.25 degree increments
18  lat= -44.5+(/(j,j=0,138)/)*0.25           ! 139 x 0.25 degree increments
19  ok = Rinit(Rfortran_ScriptPathIn='../../../Rscripts/tutorial_examples/')
20  ok = Rput("lat",lat); ok = Rput("lon",lon) !Rainfall grid lat/lon values
21  ok = Rput("coast",coast,mv=(coast<=-9999)) !Missing vals due to islands
22  ok = Rput("x",x,mv=(x<0))                  !Missing vals due to ocean pixels
23  ok = Rput("AustraliaRainfall_ScriptedGraph(lat,lon,x,coast)")
```

Example: Simple Graph

- Scenario: Plotting some data in R from F95

```
01 program AustraliaRainfall_SimpleGraph.
02 use RFortran                              ! Access RFortan
! Declare variables – full code provided at www.rfortran.org
06
07 open(10,file="../../../data/AustraliaRainfall.txt") ! Read
08 do i = 1,178; read(10,*) (x(i,j),j=1,139); end do   ! Data
09
10   ok = Rinit()                  ! Initialise R
11   ok = Rput("x",x)              ! Transfer data to R
12   ok = Rput("x[which(x<0)]<-NA") ! Convert missing values
13   ok = Rput("image(x)")         ! Create map of rainfall
14   ok = Rclose()
15
16 end program AustraliaRainfall_SimpleGraph
```

Example: Simple Graph

- Scenario: you have some data in F95 to plot
- Initialise R (line 11)
- Transfer some data to R (line 12)
- Run a R commands (line 13,14)

```
01 program AustraliaRainfall_SimpleGraph ! Read rainfall and plot in R.
02   use RFortran                        ! Access RFortan
03   implicit none
04   integer(4) :: i,j,ok                ! Index counters, error flag
05   real(8)    :: x(178,139)            ! Data matrix
06
07   open(10,file="../../../data/AustraliaRainfall.txt") ! Read file
08   do i = 1,178
09     read(10,*) (x(i,j),j=1,139)
10   end do
11   ok = Rinit()                        ! Initialise R, no scripts
12   ok = Rput("x",x)                    ! Transfer data to R
13   ok = Rput("x[which(x<0)]<-NA")      ! Convert missing values
14   ok = Rput("image(x)")               ! Create a map of the rainfall
15   ok = Rclose()
16 end program AustraliaRainfall_SimpleGraph
```

Why use RFortran?

- Why not just use Fortran?
  - you want graphics or functions in R
- Why not just use R then?
  - R is slower (scripted)
  - Large amount of F95 Legacy Code
- Why not just plug a F95 DLL into R?
  - User convenience, DLLs can be tricky at times
  - Some programs are hard to package as DLLs

- Aimed at users who have significant F95 code and wish to use some of R's graphics and packages to make life a little easier!

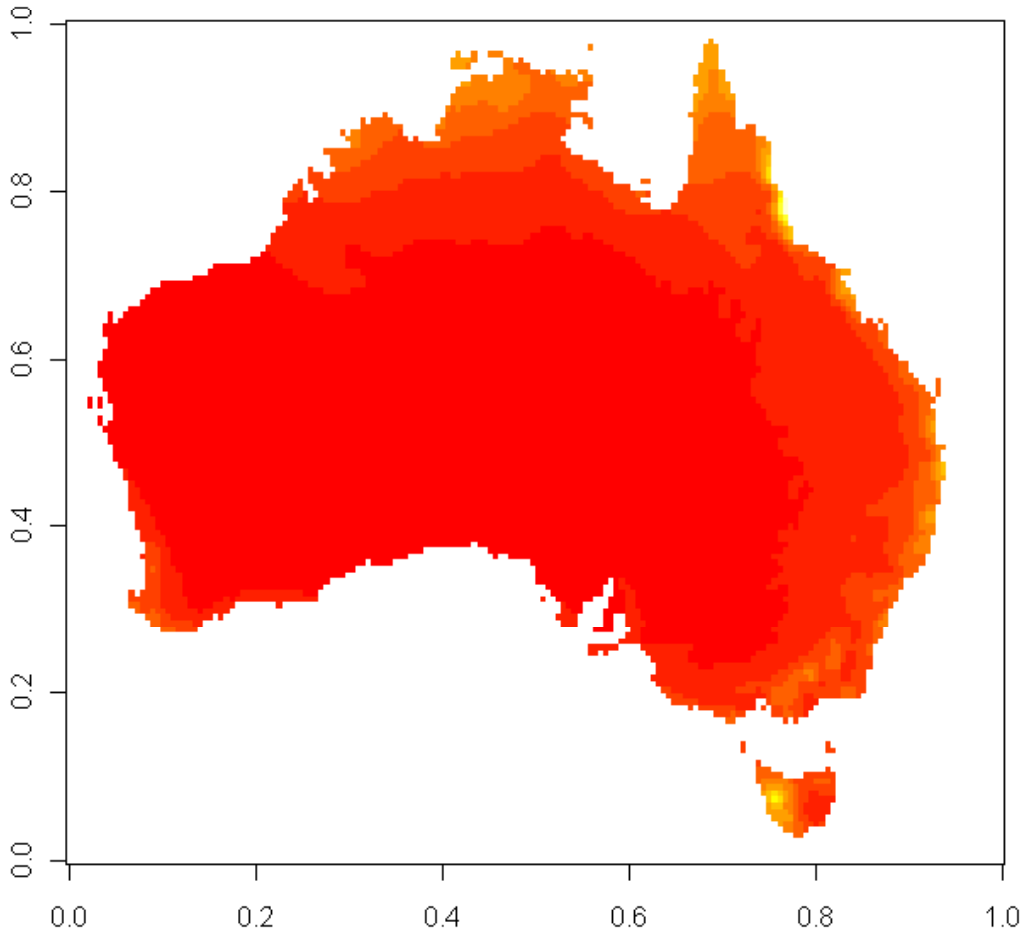
Example: Simple Graph

- Scenario: Plotting some data in R from F95

```
program AustraliaRainfall_SimpleGraph.
use RFortran                              ! Access RFortan
! Declare variables – full code at www.rfortran.org
! Read Data
open(10,file="../../../data/AustraliaRainfall.txt")
do i = 1,178; read(10,*) (x(i,j),j=1,139); end do

ok = Rinit()                    ! Initialise R
ok = Rput("x",x)                ! Transfer data to R
ok = Rput("x[which(x<0)]<-NA") ! Convert missing values
ok = Rput("image(x)")           ! Create map of rainfall
ok = Rclose()

end program AustraliaRainfall_SimpleGraph
```

Example: Simple Graph

- A basic plot



Introduction

- Fortran 90/95 (F95) is a well-established language
  - many existing programs written in F95
  - ongoing code development
  - users attracted to speed and memory advantages
- R is open-source graphical/statistical package
  - Provides many new statistical algorithms
  - Good graphical support
  - Fast growing list of packages
- ***RFortran is an open source software library that links to R from Fortran***

Motivation (1):
Producing R Graphics directly from Fortran Code

- Fortran lacks graphics functionality
- Many Fortran users dump outputs to file and plot using other software
- Frustrating/tedious/time-consuming when
  - debugging your code
  - interactively analysing/simulating data
  - generating numerous diagnostic plots
- RFortran enables producing high-quality R graphics directly from Fortran code