« Bayesian statistical methods in ecotoxicology »

Bayesian inference in practice

Elise Billoir (ebilloir@pole-ecotox.fr)

Rovaltain Research facility for Environmental Toxicology and Ecotoxicology

Bayesian inference in pratice

Outline – Practical organization

- A few slides about software aspects
- A first dose-response example guided with slides and step-by-step explanation
- Then
  - More dose-response examples with survival, growth and reproduction data
  - If you're done with the previous, time-dose-response examples with survival data
  - Concluding remarks, questions and evaluation

Software

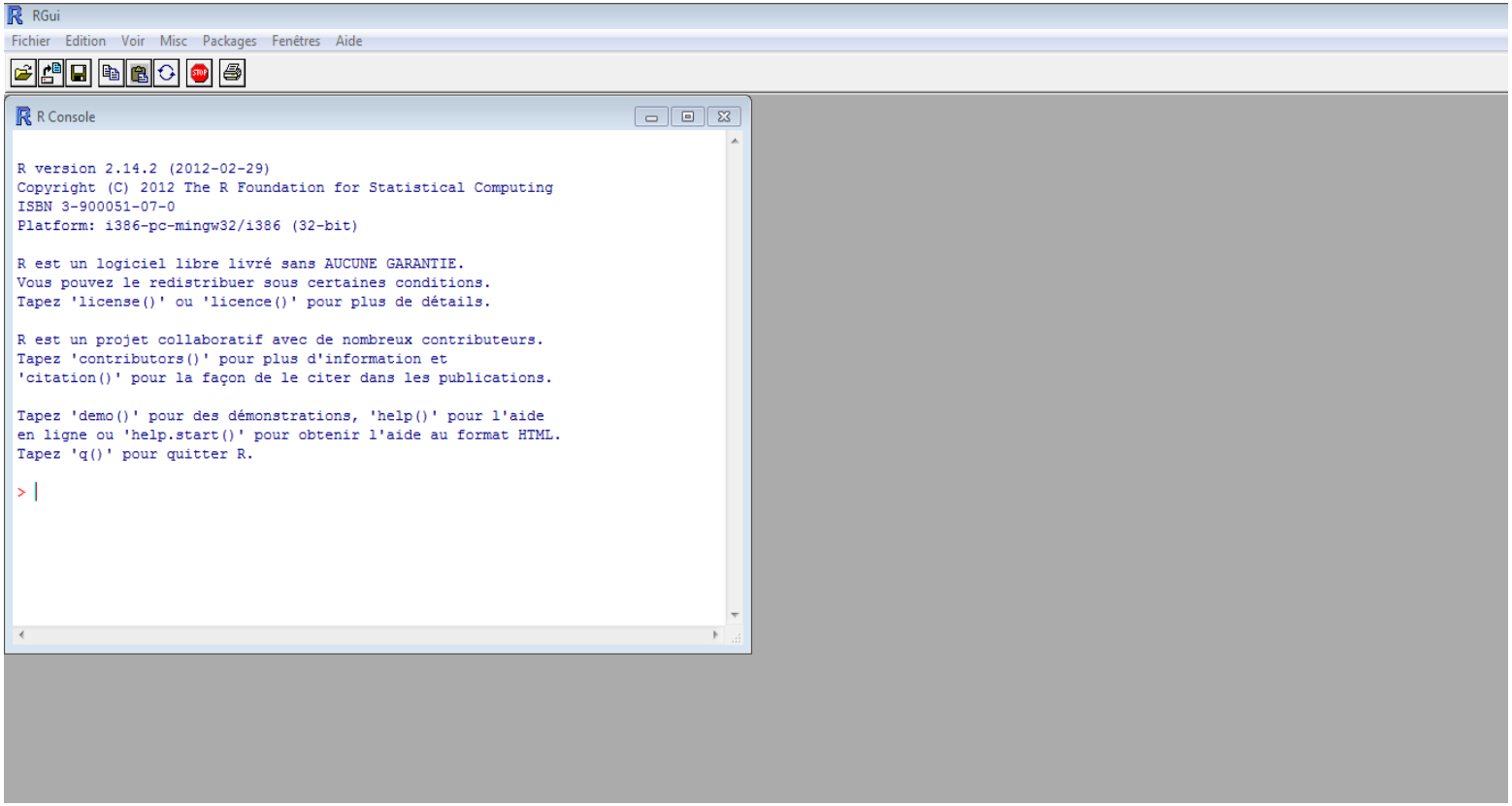
- There are several tools to carry out Bayesian inference
- We will use  and its package rjags (which requires JAGS to be installed)
- Has everyone  , JAGS and rjags installed on his computer?

Software

-  is very convenient for statistics in general, powerful but unfortunately it's actually not user-friendly
-  is a command-line software (no menus nor buttons)
- but don't worry, we will see together the few  things you need to know for today pratical exercices
- for those interested in further using  you can find a plethora of introductory manuals at http://www.r-project.org/

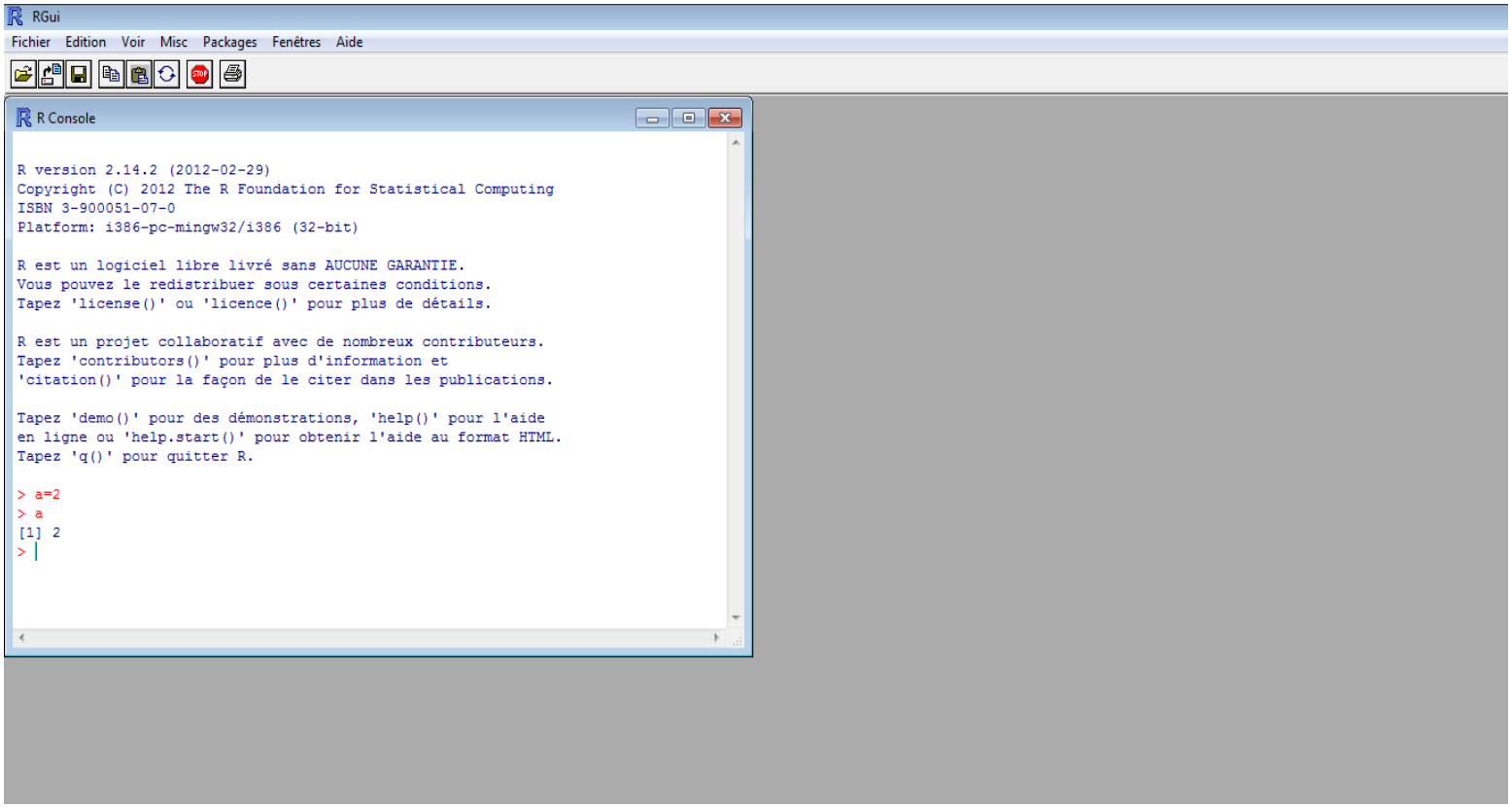
Software

- What does  look like ?



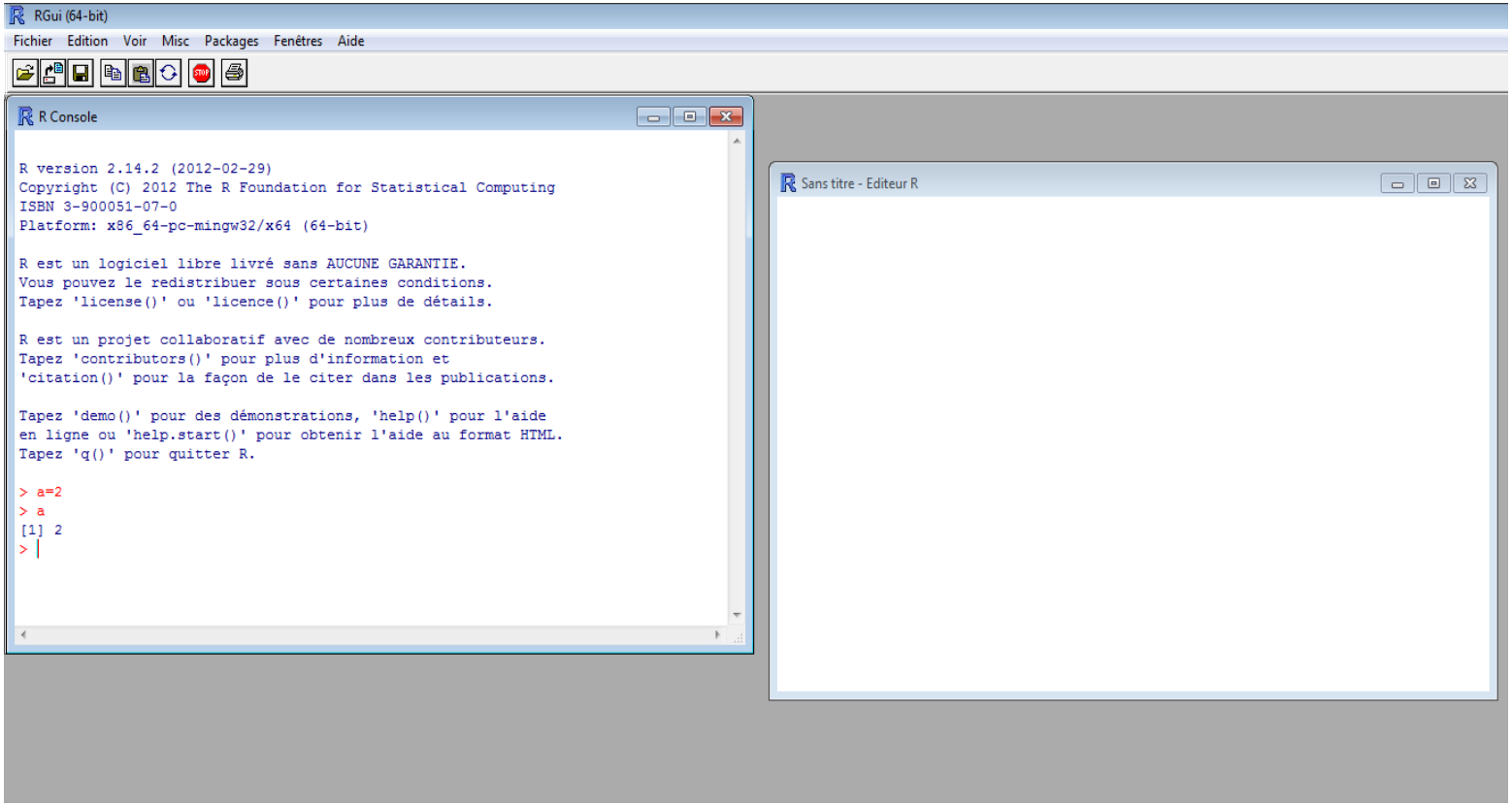
Software

- The console mode



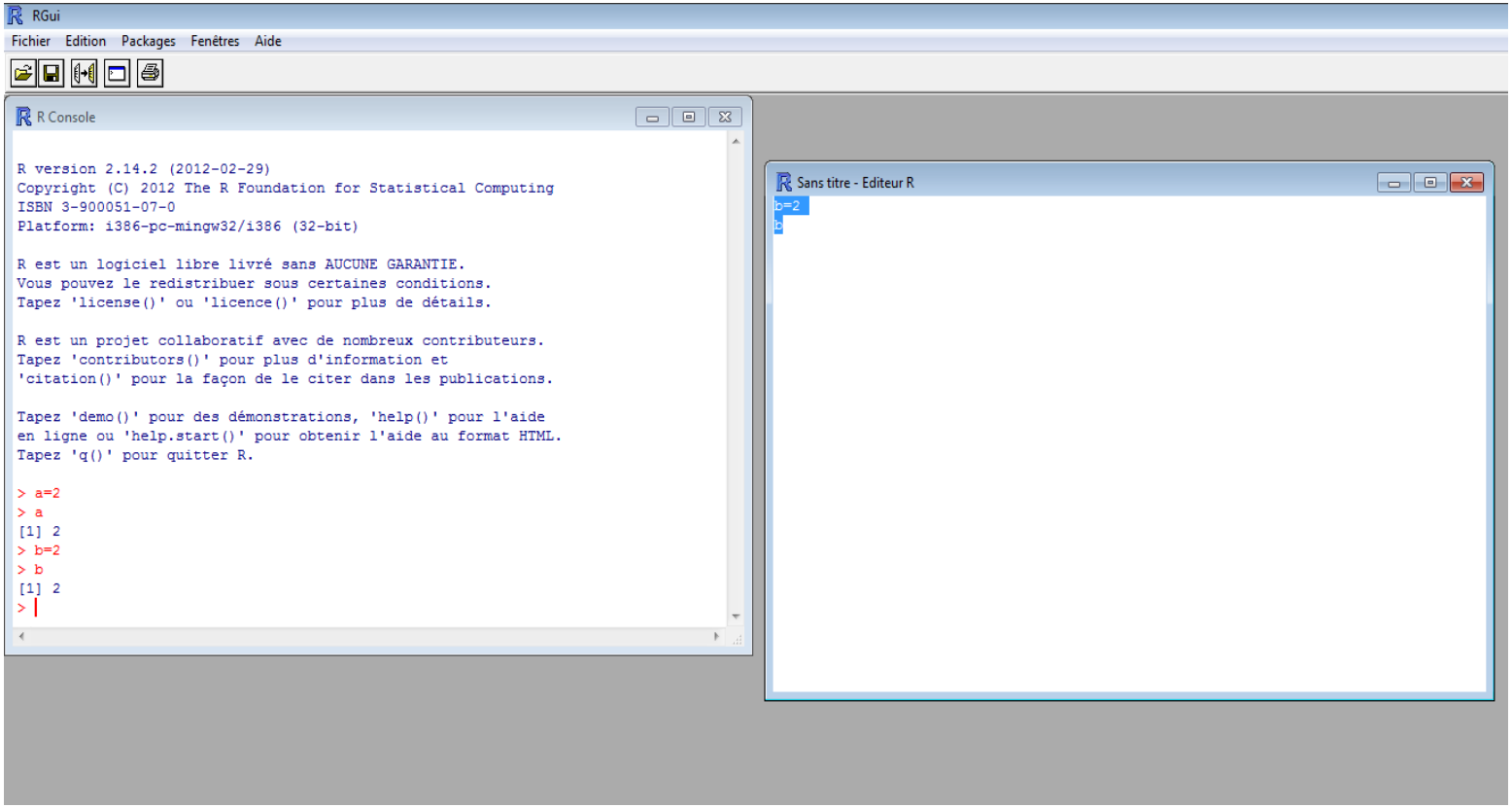
Software

- The script mode: **File / New script**



Software

- The script mode: write the commands in the script – **Ctrl-R** to run the highlighted line(s)

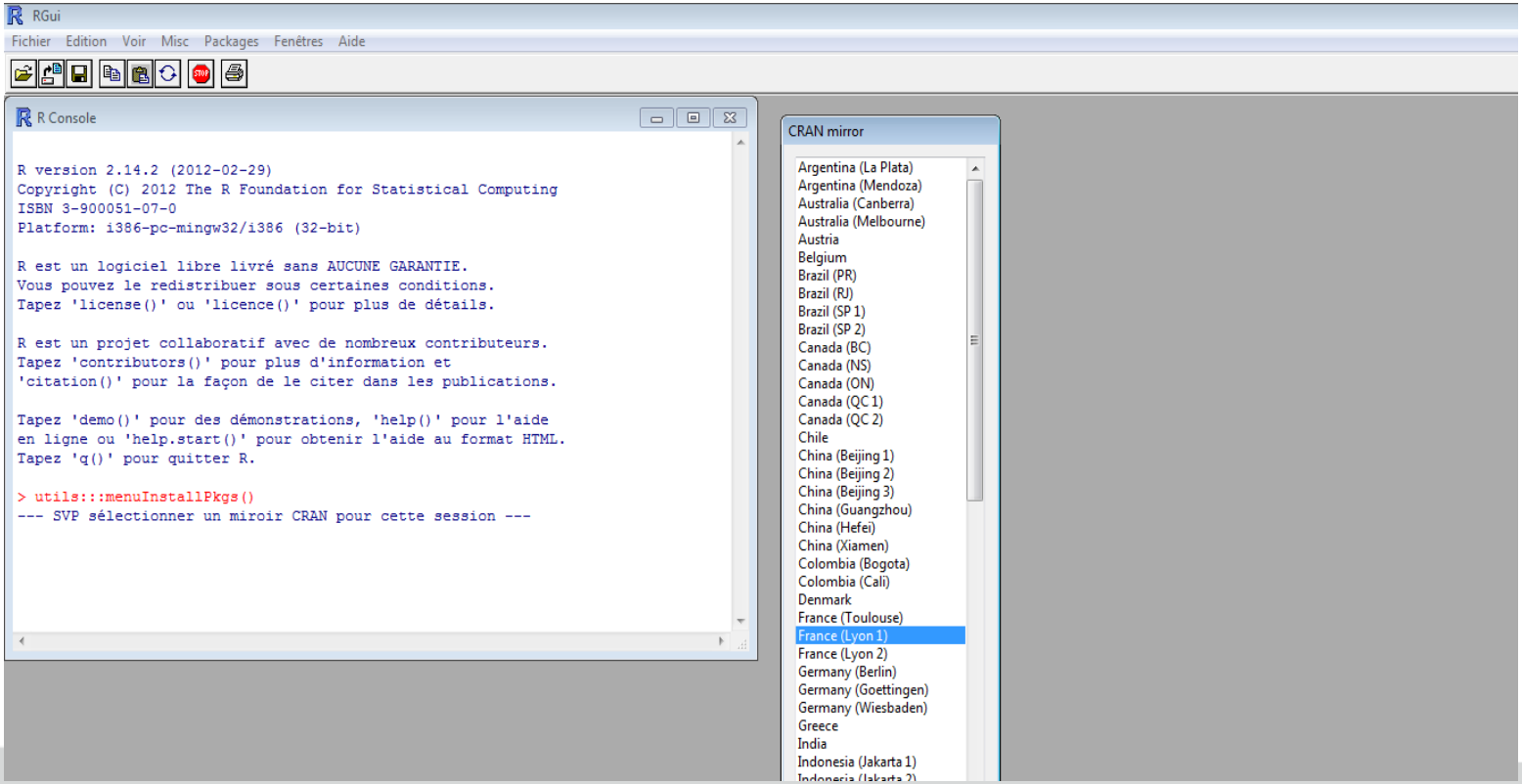


Software

- To install a package (we will need the ***rjags*** package)

  Packages / Install package(s)

  first choose a mirror

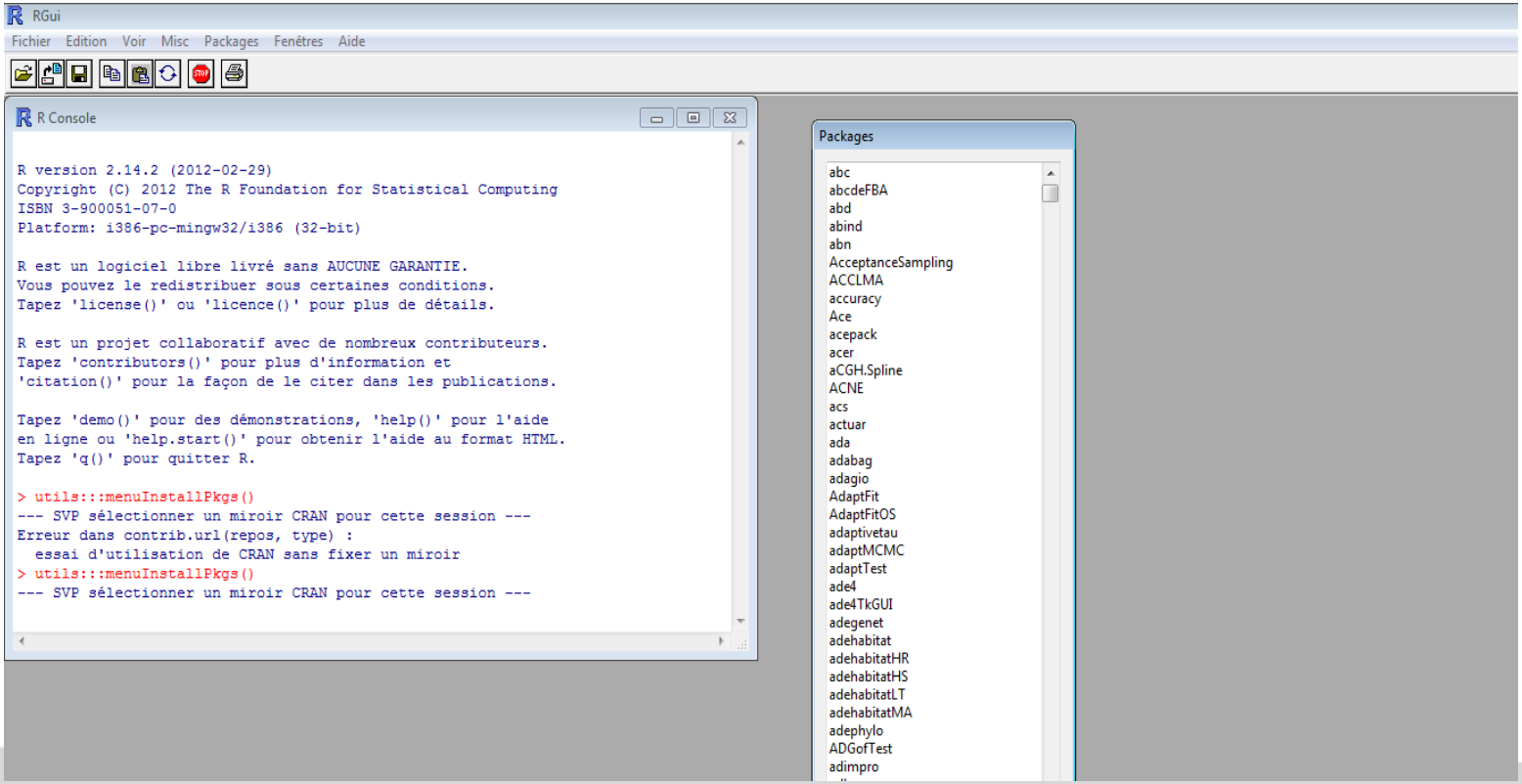


Software

- To install a package (we will need the ***rjags*** package)

  Packages / Install package(s)

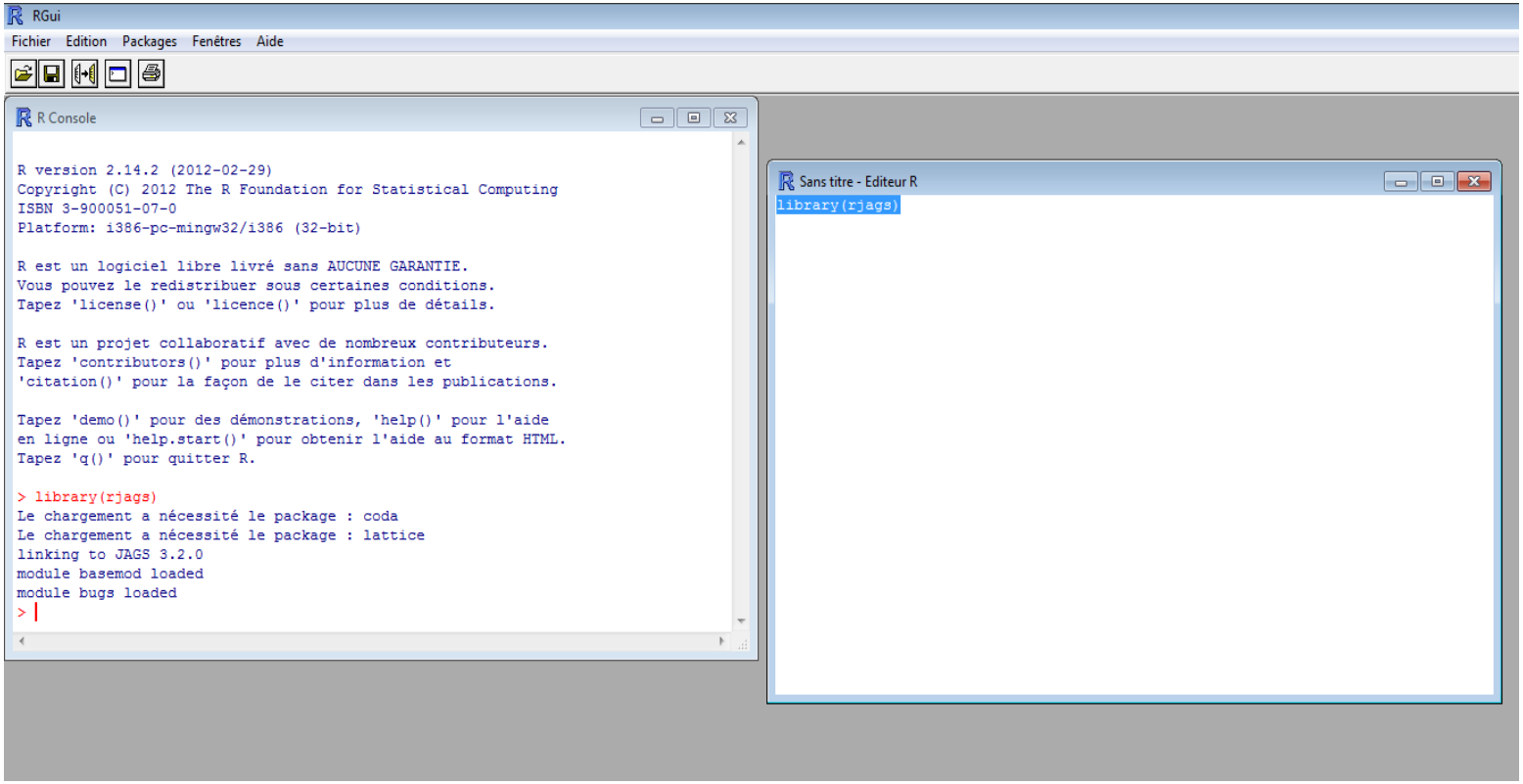
  first choose a mirror, **then select the wanted package (*rjags*)**



Software

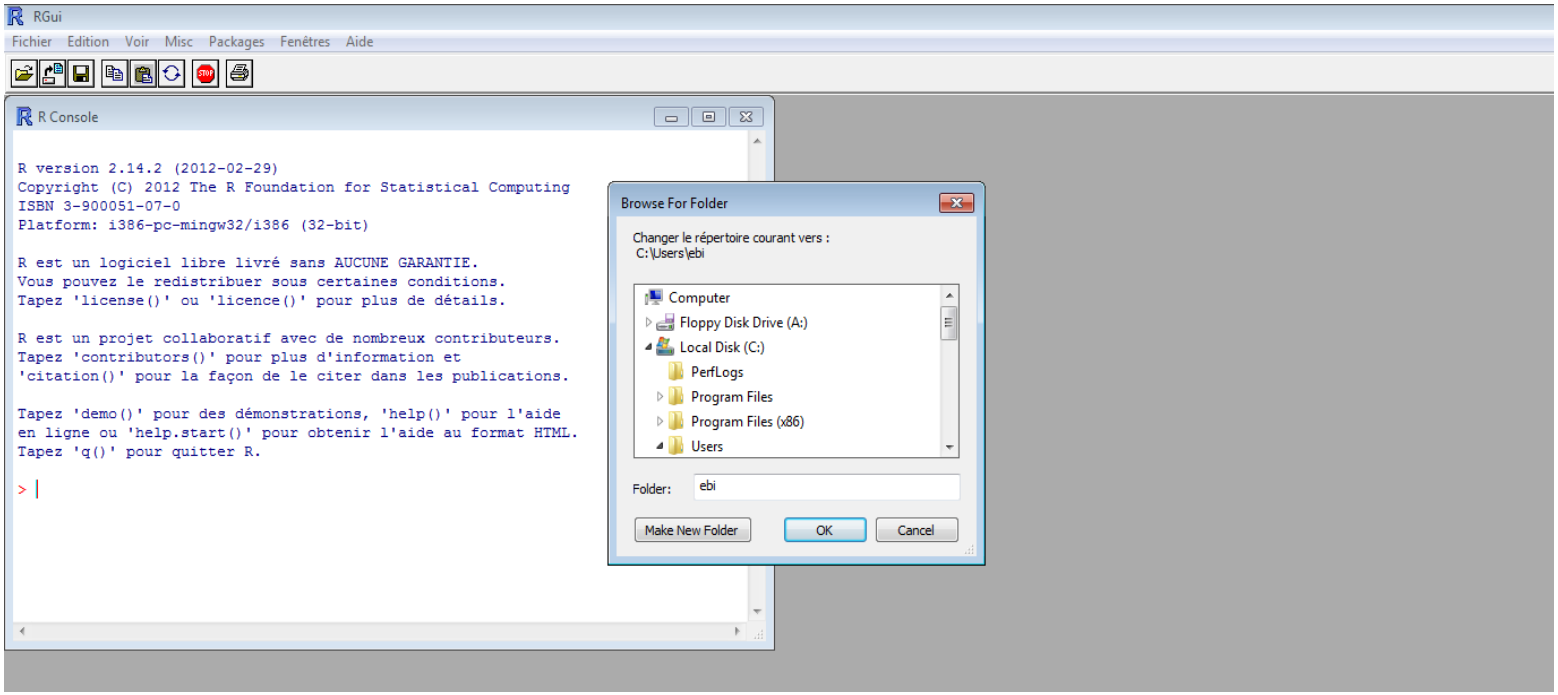
- A very first command to load the *rjags* package

  (another option is **Packages / Load packages**)



Software

- Change the working directory to one where every files (scripts, others) should be saved: **File / Change working directory**



- **All provided files (data sets) and newly created ones (model specification files, R scripts) must be in this working directory**

Software

- In  to get help on any function, **`?anyfunction`** (e.g. **`?length`**)
-  is case-sensitive (e.g. **`M1.MCMC != M1.mcmc`**)

A first example: application

- Let’s get started with a survival data set and a log-logistic dose-response curve

A first example: successive steps

- 1. Setting the **data** (including information to define priors)
- 2. Visualizing the **data**
- 3. **Specifying the model** according to BUGS/JAGS syntax
- 4. **Initializing** the model + data
- 5. **Burn-in** phase
- 6. Further **running the algorithm** + monitoring of parameters to **generate samples**
- 7. **Convergence** checking
- 8. Views of the **resulting chains**: summary statistics, sample trace and posterior distributions
- 9. View of the joint posterior distribution
- 10. Comparison of **prior and posterior** statistics
- 11. Visualizing the fitting (**model and observed data**)
- 12. Visualizing the fitting (model, observed data and **predicted data**)
- 13. Calculation of the deviance information criterion (**DIC**)

1. Setting the data

```
tableSurv21days <- read.table("chlordan_survival_21day.txt",header=TRUE)
```

"conc"	"Ninit"	"Nsurv"
0	10	10
0.18	10	9
0.73	10	9
1.82	10	9
2.9	10	3
7	10	2

Read the data frame in chlordan_survival_21day.txt

```
concentrations=tableSurv21days$conc
Ninit=tableSurv21days$Ninit
Nsurv=tableSurv21days$Nsurv
n=length(concentrations)
```

Get the length of a vector

1 (continued). Setting the information to define priors

```
# For LC50 prior
concmin <- min(sort(unique(concentrations))[-1])
concmax <- max(concentrations)
meanlog10LC50 <- (log10(concmin) + log10(concmax))/2
sdlog10LC50 <- (log10(concmax) - log10(concmin))/4
taulog10LC50 <- 1/sdlog10LC50^2
```

```
data=list(n=n, x=concentrations, Ninit=Ninit, y=Nsurv,
          meanlog10LC50 = meanlog10LC50,
          taulog10LC50 = taulog10LC50)
```

`data`

To see the content of the `data` object

Data constructed as a list (expected syntax)

2. Visualizing the data

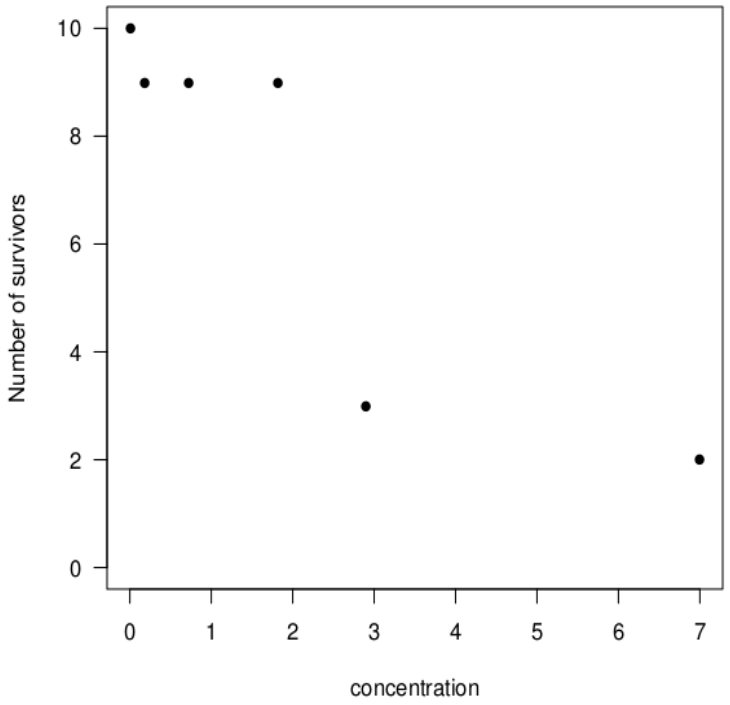
```
plot(x=tableSurv21days$conc, y=tableSurv21days$Nsurv,
     xlab="concentration",ylab="Number of survivors",
                    pch=16, ylim=c(0,10))
```

- Plot the data
- x-coordinates of the points
- y-coordinates of the points
- Set the limits on the y-axis

Many other plot options
`?plot`

2. Visualizing the data

```
plot(x=tableSurv21days$conc, y=tableSurv21days$Nsurv,
     xlab="concentration",ylab="Number of survivors",
                pch=16, ylim=c(0,10))
```



3. Specifying the model according to BUGS/JAGS syntax

```
model
{
for (i in 1:n)
        {
        p[i]<-1/(1+(x[i]/LC50)^b)
        y[i]~ dbin(p[i],Ninit[i])
        }

# specification of priors
log10b ~ dunif(-2,2)
log10LC50 ~ dnorm(meanlog10LC50, taulog10LC50)

b <- pow(10,log10b)
LC50 <- pow(10,log10LC50)
}
```

The model specification has to start with this keyword

Deterministic part of the model

Stochastic part of the model
Binomial distribution

Uninformative prior for b
Informative prior for LC50 according to the range of concentration tested

3. Specifying the model according to BUGS/JAGS syntax

```
model
{
for (i in 1:n)
        {
        p[i]<-1/(1+(x[i]/LC50)^b)
        y[i]~ dbin(p[i],Ninit[i])
        }

# specification of priors (may be changed if needed)
log10b ~ dunif(-2,2)
log10LC50 ~ dnorm(meanlog10LC50 , taulog10LC50)

b <- pow(10,log10b)
LC50 <- pow(10,log10LC50)
}
```

To be saved in a file `logistic_binomial.txt`

3. Specifying the model according to BUGS/JAGS syntax

- Many distributions (resp. functions) in the BUGS/JAGS syntax to specify the stochastic part (resp. deterministic part) of the wanted model

Name	Usage	Density	Lower	Upper
Beta	`dbeta(a,b)` $a > 0, b > 0$	$\frac{x^{a-1}(1-x)^{b-1}}{\beta(a,b)}$	0	1
Chi-square	`dchisqr(k)` $k > 0$	$\frac{x^{\frac{k}{2}-1}\exp(-x/2)}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})}$	0	
Double exponential	`ddexp(mu,tau)` $\tau > 0$	$\tau\exp(-\tau\lvert x-\mu\rvert)/2$		
Exponential	`dexp(lambda)` $\lambda > 0$	$\lambda\exp(-\lambda x)$	0	
F	`df(n,m)` $n > 0, m > 0$	$\frac{\Gamma(\frac{n+m}{2})}{\Gamma(\frac{n}{2})\Gamma(\frac{m}{2})}(\frac{n}{m})^{\frac{n}{2}}x^{\frac{n}{2}-1}\{1+\frac{nx}{m}\}^{-\frac{(n+m)}{2}}$	0	
Gamma	`dgamma(r, mu)` $\mu > 0, r > 0$	$\frac{\mu^r x^{r-1}\exp(-\mu x)}{\Gamma(r)}$	0	
Generalized gamma	`dgen.gamma(r,mu,beta)` $\mu > 0, \beta > 0, r > 0$	$\beta\mu^{\beta r}x^{\beta r-1}\exp\{-(\mu x)^{\beta}\}$	0	
Log-normal	`dlnorm(mu,tau)` $\tau > 0$	$\tau^{\frac{1}{2}}x^{-1}\exp\{-\tau(\log(x)-\mu)^2/2\}$	0	
Normal	`dnorm(mu,tau)` $\tau > 0$	$(\frac{\tau}{2\pi})^{\frac{1}{2}}\exp\{-(x-\mu)^2\tau\}$		

- See also the JAGS manual (Chapters 5 and 6)

4. Initializing the model + data

```
M1=jags.model(file="logistic_binomial.txt", data=data,
n.chains=3)
```

File where the model is specified

Object (list) where the data are set

We run 3 MCMC/chains/algorithms in parallel which will provide us with 3 independent samples of posterior distributions

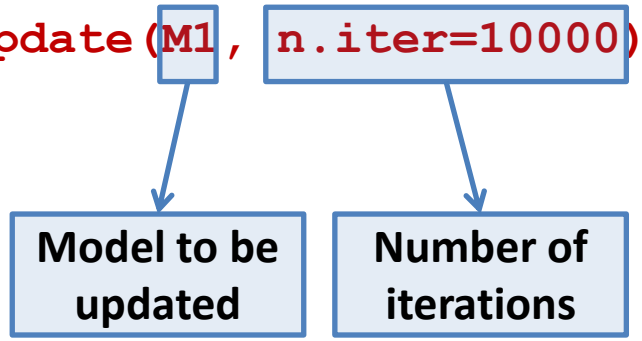
4. Initializing the model + data

```
M1=jags.model(file="logistic_binomial.txt", data=data,
n.chains=3)
```

Proposed name for this first model

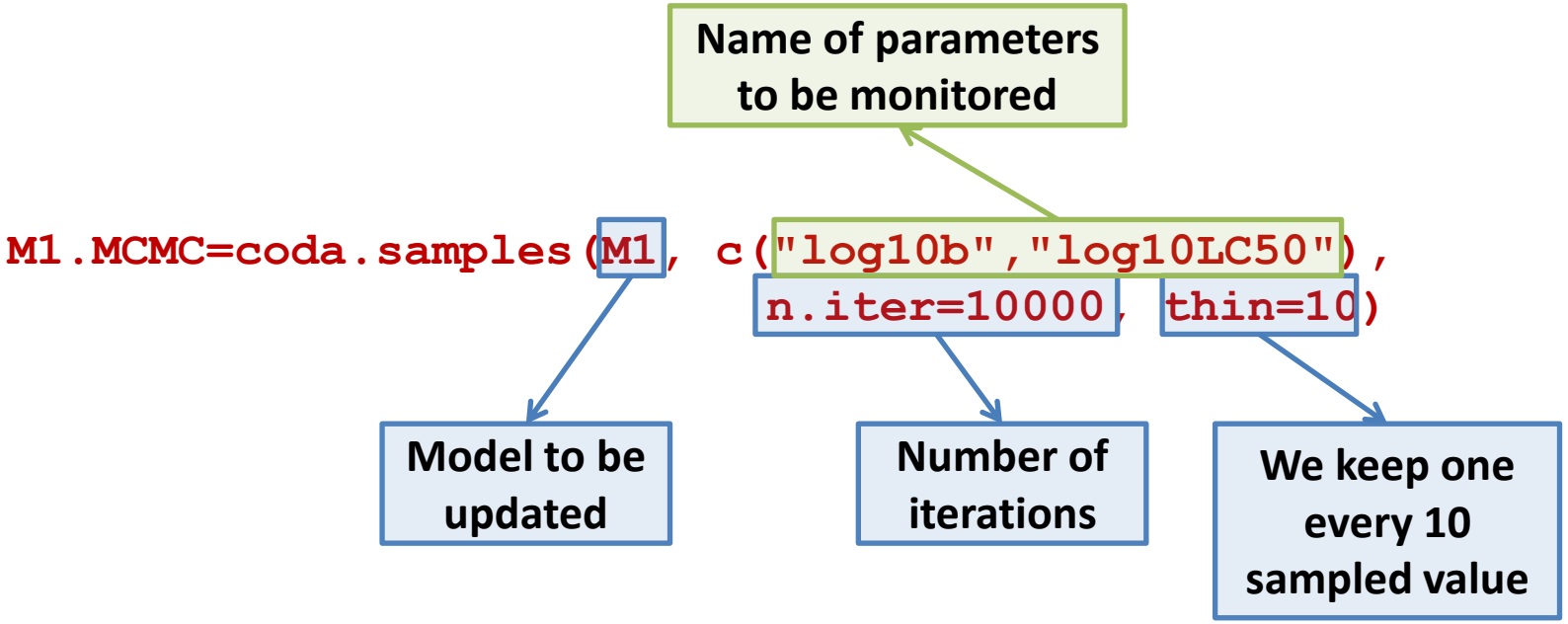
5. Burn-in phase

```
update(M1, n.iter=10000)
```



6. Further running the algorithm + monitoring of parameters

- To generate samples of posterior distributions



6. Further running the algorithm + monitoring of parameters

- To generate samples of posterior distribution

```
M1.MCMC=coda.samples(M1, c("log10b","log10LC50"),
                     n.iter=10000, thin=10)
```

Proposed name for the object that will contain the MCMC/chains/samples

7. Resulting chains: checking convergence

- Gelman and Rubin criterion

```
gelman.diag(M1.MCMC)
Potential scale reduction factors:

          Point est. Upper C.I.
log10b             1          1
log10LC50          1          1
```

- Looking at the resulting chains
  - Do the chains accord?
  - Are the posterior distributions unimodal?

8a. Resulting chains: summary statistics

```
M1.su=summary(M1.MCMC)
M1.su
```

To see the content of the `M1.su` object

Object containing the MCMC/chains/samples

8a. Resulting chains: summary statistics

```
M1.su=summary(M1.MCMC)
M1.su

Iterations = 13010:23000
Thinning interval = 10
Number of chains = 3
Sample size per chain = 1000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

            Mean    SD Naive SE Time-series SE
log10LC50 0.4334 0.134  0.00244        0.00255
log10b    0.0546 0.160  0.00292        0.00308

2. Quantiles for each variable:

            2.5%     25%    50%   75% 97.5%
log10LC50  0.190  0.3492 0.4295 0.515 0.711
log10b    -0.294 -0.0406 0.0661 0.166 0.324
```

8b and 8c. Resulting chains: sample trace and posterior distributions

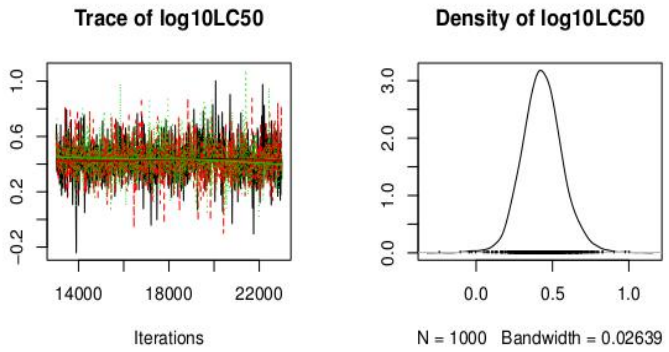
```
plot(M1.MCMC, trace=TRUE, density=TRUE)
```

- `M1.MCMC`: Object containing the MCMC/chains/samples
- `trace=TRUE`: To see the trace of the chains
- `density=TRUE`: To see the density/histogram of the samples

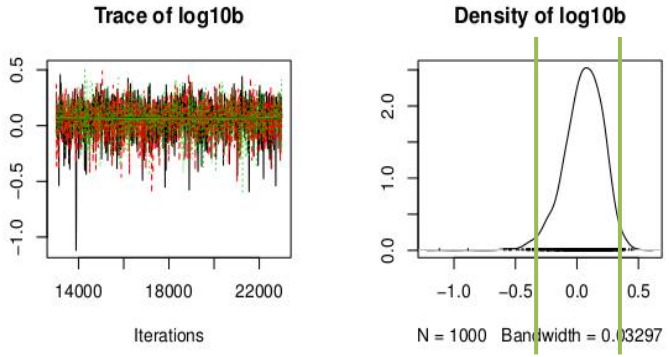
8b and 8c. Resulting chains: sample trace and posterior distributions

```
plot(M1.MCMC, trace=TRUE, density=TRUE)
```

log10LC50



log10b



Quantiles for each variable:

	2.5%	50%	97.5%
log10LC50	0.190	0.4295	0.711
log10b	-0.294	0.0661	0.324

9. Resulting chains: view of the joint posterior distribution

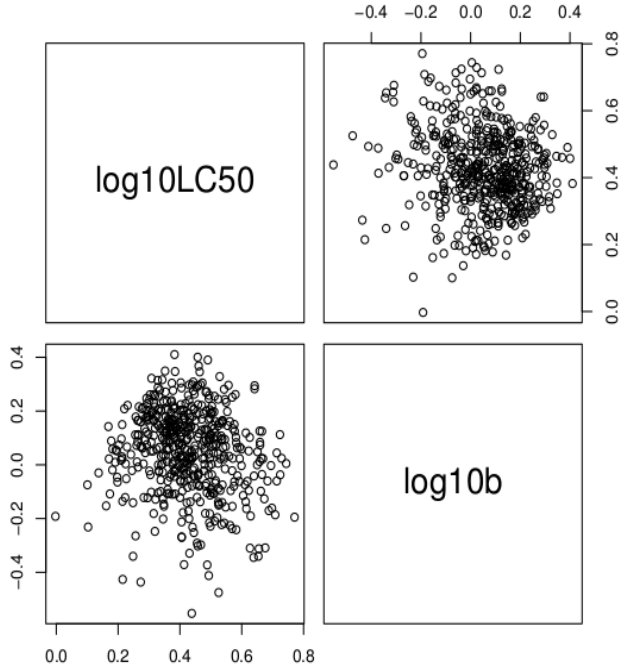
```
M1.MCMCtot=rbind(M1.MCMC[[1]],M1.MCMC[[2]],M1.MCMC[[3]])
pairs(M1.MCMCtot)
```

Pairwise plot of the parameter distributions

To bind the three parallel chains

9. Resulting chains: view of the joint posterior distribution

```
M1.MCMCtot=rbind(M1.MCMC[[1]],M1.MCMC[[2]],M1.MCMC[[3]])
pairs(M1.MCMCtot)
```



10. Comparison of prior and posterior statistics

```
data0 <- list(n=n, x=concentrations, Ninit=Ninit,
              meanlog10LC50=meanlog10LC50,
              taulog10LC50=taulog10LC50)

M10 <- jags.model(file="logistic_binomial.txt", data=data0,
n.chains=3)
update(M10,5000)
M10.MCMC <- coda.samples(M10, c("log10b","log10LC50"),
                                          n.iter=1000)
```

We run the model (steps 1 and 4 to 6) without providing the Nsurv as data

```
summary(M10.MCMC)$quantiles
summary(M1.MCMC)$quantiles
```

We compare the resulting summary statistics

10. Comparison of prior and posterior statistics

```
summary(M10.MCMC)$quantiles

              2.5%      25%      50%     75%  97.5%
log10LC50 -0.719   -0.229    0.042   0.327   0.829
log10b    -1.917   -1.034   -0.095   0.978   1.892
```

```
summary(M1.MCMC)$quantiles

              2.5%      25%      50%     75%  97.5%
log10LC50  0.190    0.3492  0.4295   0.515   0.711
log10b    -0.294   -0.0406  0.0661   0.166   0.324
```

11. Visualizing the fitting (model and observed data)

```
x <- seq(0,max(concentrations),length=100)
M1.su <- summary(M1.MCMC)
b <- 10^M1.su$quantiles["log10b","50%"]
LC50 <- 10^M1.su$quantiles["log10LC50","50%"]

Nsurvtheo <- 10*1/(1+(x/LC50)^b)
```

```
plot(x=tableSurv21days$conc,y=tableSurv21days$Nsurv,
     xlab="concentration",ylab="Number of survivors")
lines(x,Nsurvtheo,col="red",type="l")
```

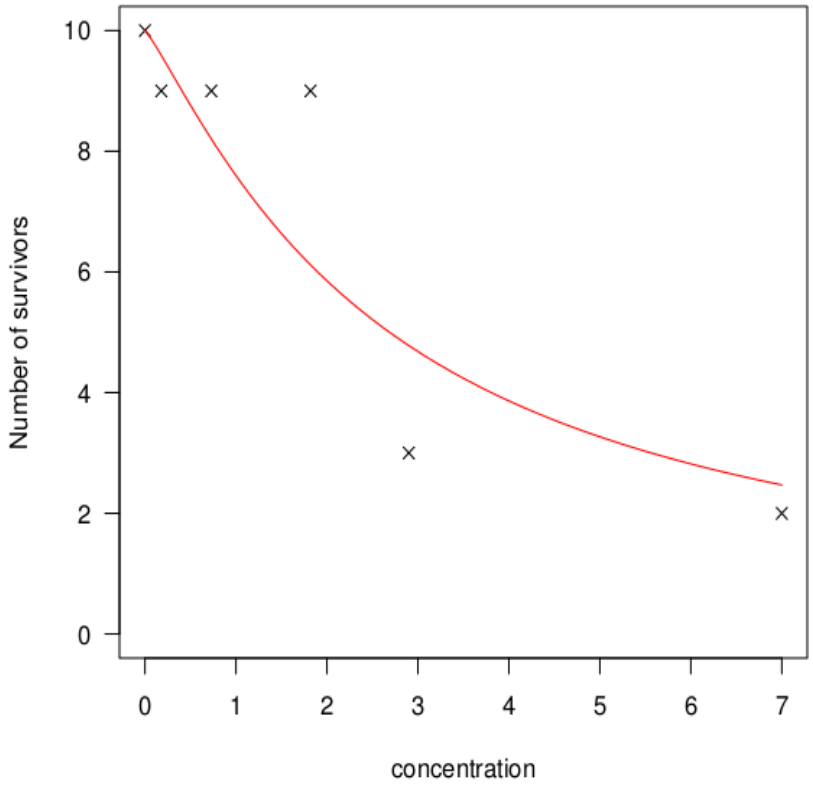
Plot the points for the observed number of survivors then the fitted curve corresponding to a given function

remind the object M1.su

```
M1.su$quantiles
```

	2.5%	25%	50%	75%	97.5%
log10LC50	0.190	0.3492	0.4295	0.515	0.711
log10b	-0.294	-0.0406	0.0661	0.166	0.324

11. Visualizing the fitting (model and observed data)



12. Visualizing the fitting (model, observed data and predicted data)

```
b <- 10^M1.MCMCtot[,"log10b"]
LC50 <- 10^M1.MCMCtot[,"log10LC50"]
```

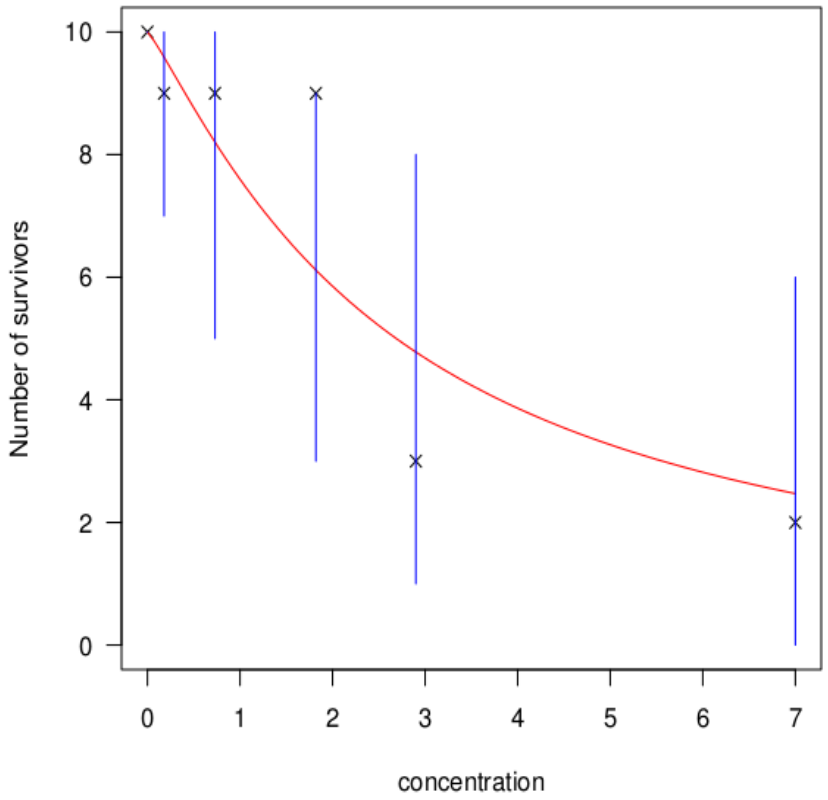
Get the k values of b and LC50

```
k=nrow(M1.MCMCtot)

plot(x=tableSurv21days$conc, y=tableSurv21days$Nsurv,
     xlab="concentration",ylab="Number of survivors")
lines(x,Nsurvtheo,col="red",type="l")

for(i in 1:length(concentrations)){
NsurvPred=rbinom(k,size=Ninit[i],
      prob=1/(1+exp(b* (log(concentrations[i]) - log(LC50)) )))
qinf95=quantile(NsurvPred,probs=0.025)
qsup95=quantile(NsurvPred,probs=0.975)
segments(x0=concentrations[i],y0=qinf95,
                    x1=concentrations[i],y1=qsup95,col="blue")
}
```

12. Visualizing the fitting (model, observed data and predicted data)



13. Calculation of the deviance information criterion (DIC)

```
dic.samples(M1,n.iter=5000)

  Mean deviance:  19.6
  penalty 2.13
  Penalized deviance: 21.7
```

Useful only if used to compare models fitted on a same data set!

Next examples

- Fitting of another dose-response curve (so-called PiresFox) to this survival data set
  - Change the deterministic part of the model specification

- Fitting of two dose-response curves (log-logistic and PiresFox) to growth data
  - Set the growth data set (using file `chlordan_growth_21day.txt`)
  - Adapt the deterministic part of the model specification
  - Adapt the stochastic part of the model specification
  - Adapt the name of parameters to be monitored
  - Etc.

- Other examples with reproduction data and time-dependent survival data

- To do that, guiding instructions are in « Bayesian inference - Practical exercices »
- The three instructors of this short course are yours, don’t remain stucked !