A dip into R for decision modelling

Howard Thom [a,b]

[a] Surgical Innovation theme of the National Institute for Health Research (NIHR) Bristol Biomedical Research Centre (BRC), Bristol Medical School, University of Bristol, Bristol, UK

[b] MRC ConDuCT-II Hub for Trials Methodology Research (Collaboration and Innovation for Difficult or Complex Randomised Controlled Trials in Invasive Procedures), Bristol Medical School, University of Bristol, Bristol, UK

Live Content Slide

When playing as a slideshow, this slide will display live content

Poll: What software do you mostly use for cost-effectiveness analysis?

Live Content Slide

When playing as a slideshow, this slide will display live content

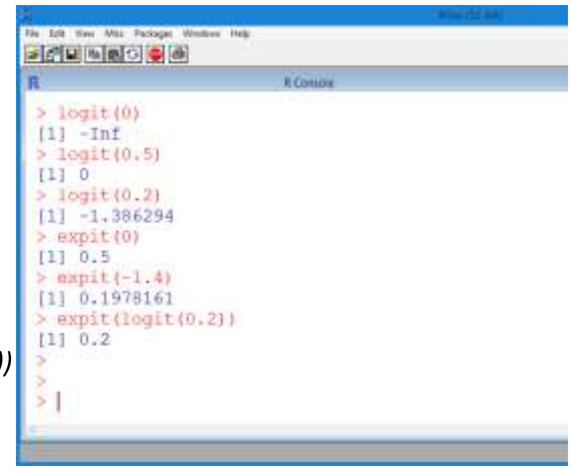
Poll: Do you think R is better for cost-effectiveness analysis and modelling than Excel?

Overview

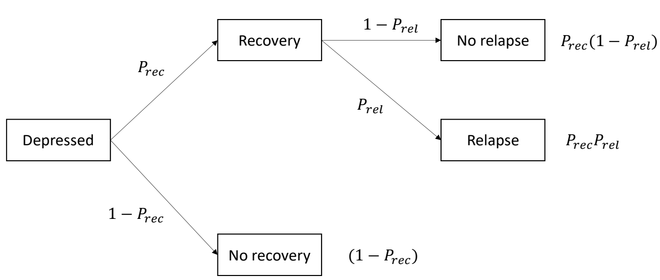
- Heard the wonderful things you can do in R.
- Itching to get your hands dirty?
- We'll now talk through the specifics of programming a (probabilistic) decision tree in R.
- The most boring presentation of ISPOR? We'll see…

Simple functions – logit and its inverse

```
# Logistic link function
logit<-function(x)
{
        return(log(x/(1-x)))
}
# Inverse of logit
expit<-function(x)
{
        return(1/(1+exp(-x)))
}
```

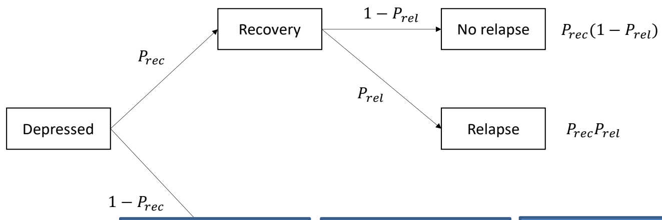


Simple decision tree in R



- Consider this simple decision tree with artificial input parameters.
- Probabilities of recovery and relapse for no treatment (option 1), cognitive behavioural therapy (option 2), and antidepressants (option 3).
- This toy model is available on GitHub: https://github.com/Bogdasayen/Depression-toy-decision-tree-in-R

Implementing a decision tree in R



effects<-p.rec(1-p.rel)*q.rec+p.rec*p.rel*q.rel+(1-p.rec)*q.norec*

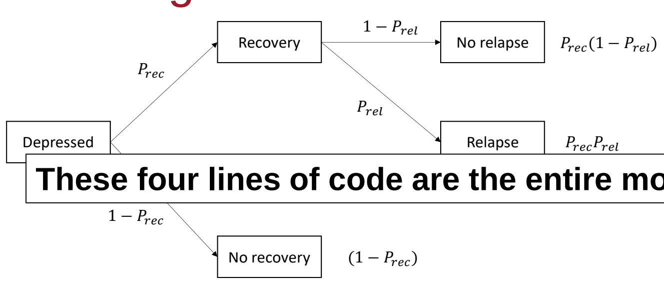
costs<-c.treat+p.rec(1-p.rel)*c.rec+p.rec*p.rel*c.rel+(1-p.rec)*c.norec*

*net.benefit<-lambda*effects-costs*

incremental.nb<-net.benefit-net.benefit[,1]

Treatment cost Recovery cost Relapse cost No recovery cost

Implementing a decision tree in R



These four lines of code are the entire model!

effects<-p.rec(1-p.rel)*q.rec+p.rec*p.rel*q.rel+(1-p.rec)*q.norec*

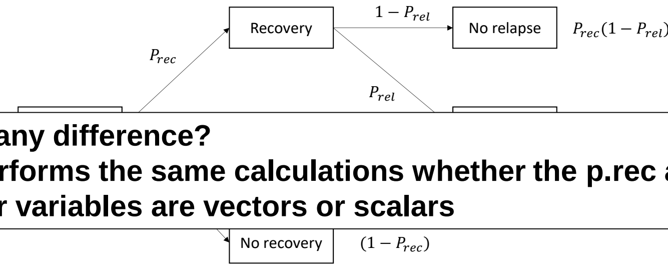
costs<-c.treat+p.rec(1-p.rel)*c.rec+p.rec*p.rel*c.rel+(1-p.rec)*c.norec*

*net.benefit<-lambda*effects-costs*

incremental.nb<-net.benefit-net.benefit[,1]

And now we make it probabilistic…

Making it probabilistic (model code)



See any difference?
R performs the same calculations whether the p.rec and other variables are vectors or scalars

effects<-p.rec(1-p.rel)*q.rec+p.rec*p.rel*q.rel+(1-p.rec)*q.norec*

costs<-c.treat+p.rec(1-p.rel)*c.rec+p.rec*p.rel*c.rel+(1-p.rec)*c.norec*

*net.benefit<-lambda*effects-costs*

incremental.nb<-net.benefit-net.benefit[,1]

Making it probabilistic (Costs, Utilities)

Outcome	Costs	QALYS
Recovery, no relapse	$C_{rec} = N(\mu = 1000, \sigma = 50)$	$Q_{rec} = N(\mu = 26, \sigma = 2)$
Recovery, relapse	$C_{rel} = N(\mu = 2000, \sigma = 100)$	$Q_{rel} = N(\mu = 23, \sigma = 3)$
No recovery	$C_{no\ rec} = N(\mu = 2500, \sigma = 125)$	$Q_{no\ rec} = N(\mu = 20, \sigma = 4)$

```
# Costs for recovery, relapse, and no recovery
c.rec<-rnorm(n=n.samples, mean=1000, sd=50)
c.rel<-rnorm(n=n.samples, mean=2000, sd=100)
c.norec<-rnorm(n=n.samples, mean=2500, sd=125)

# QALYs for recovery, relapse, and no recovery
q.rec<-rnorm(n=n.samples, mean=26, sd=2)
q.rel<-rnorm(n=n.samples, mean=23, sd=3)
q.norec<-rnorm(n=n.samples, mean=20, sd=4)
```

Making it probabilistic (Treatment effects)

- Log odds ratios follow multivariate normal

Recovery: $\begin{pmatrix} lor_{2,rec} \\ lor_{3,rec} \end{pmatrix} \sim MVN\left(\begin{pmatrix} 0.99 \\ 1.33 \end{pmatrix}, \begin{pmatrix} 0.22 & 0.15 \\ 0.15 & 0.20 \end{pmatrix}\right)$

Relapse: $\begin{pmatrix} lor_{2,rel} \\ lor_{3,rel} \end{pmatrix} \sim MVN\left(\begin{pmatrix} -1.48 \\ -0.40 \end{pmatrix}, \begin{pmatrix} 0.14 & 0.05 \\ 0.05 & 0.11 \end{pmatrix}\right)$

- As it's a statistical language, the multivariate normal is implemented simply in R:

```
lor.rec<-mvrnorm(n=n.samples,mu=c(0.99,1.33),
sigma=matrix(c(0.22,0.15,0.15,0.20),nrow=2))
lor.rel<-mvrnorm(n=n.samples,mu=c(-1.48,-0.40),
sigma=matrix(c(0.14,0.05,0.05,0.11),nrow=2))
```

Instead use MCMC via R2OpenBUGS

- Link directly with network meta-analysis code in OpenBUGS (or JAGS/STAN etc.)

library(R2OpenBUGS)

- Load some BUGS model file

source("fixed.effects.binary.R")

- Set simulation parameters

*n.chains<-2; num.sims<-10000*n.chains; burn.in<-50000*n.chains*

- Call R2OpenBUGS key function

bugs.object<-
bugs(data=bugs.data.recovery,inits=NA,model=fixed.effects.binary…)

- Then get parameter samples from bugs.object$sims.array

Instead MCMC via R2OpenBUGS

- Or load precalculated log odds ratios for recovery (similarly for relapse)

mcmc.recovery<-read.csv(file="lor.recovery.bugs.csv")

- Can use just the first n.samples of the matrix

lor.rec<-mcmc.recovery[1:n.samples,]

Making it probabilistic (Reference probabilities)

Parameter	No Treatment (Option 1)
P_{rec}	$P_{1,rec} = Beta(\alpha = 6, \beta = 200)$
P_{rel}	$P_{1,rel} = Beta(\alpha = 2, \beta = 100)$

- The beta distribution is another of many implemented in base R.
- Note however the idiosyncratic naming convention of the parameters.
- α is *shape1* and β is *shape2*.

p.rec[,1]<-rbeta(n=n.samples, shape1=6, shape2=200)
p.rel[,1]<-rbeta(n=n.samples, shape1=2, shape2=100)

University of BRISTOL | MRC Hubs for Trials Methodology Research | NHS University Hospitals Bristol NHS Foundation Trust

Making it probabilistic (Comparator probabilities)

Parameter	CBT (Option 2)	Antidepressant (Option 3)
P_{rec}	$P_{2,rec} = expit(\text{logit}(P_{1,rec}) + lor_{2,rec})$	$P_{3,rec} = expit(\text{logit}(P_{1,rec}) + lor_{3,rec})$
P_{rel}	$P_{2,rel} = expit(\text{logit}(P_{1,rel}) + lor_{2,rel})$	$P_{3,rel} = expit(\text{logit}(P_{1,rel}) + lor_{3,rel})$

- We can use a loop over the number of treatments *n.treat*

```
for(i in 2:n.treat){
        p.rec[,i]<-expit(logit(p.rec[,1])+lor.rec[,i-1])
        p.rel[,i]<-expit(logit(p.rel[,1])+lor.rel[,i-1])
}
```

Funded by NIHR

University of BRISTOL | MRC Hubs for Trials Methodology Research | NHS University Hospitals Bristol NHS Foundation Trust

Making it probabilistic - vectorise

Parameter	CBT (Option 2)	Antidepressant (Option 3)
P_{rec}	$P_{2,rec} = expit(\text{logit}(P_{1,rec}) + lor_{2,rec})$	$P_{3,rec} = expit(\text{logit}(P_{1,rec}) + lor_{3,rec})$
P_{rel}	$P_{2,rel} = expit(\text{logit}(P_{1,rel}) + lor_{2,rel})$	$P_{3,rel} = expit(\text{logit}(P_{1,rel}) + lor_{3,rel})$

- Or we can vectorise, which is much faster than a loop

```
p.rec[,c(2:n.treat)]<-expit(logit(p.rec[,1])+lor.rec[,c(2:n.treat)-1])
p.rel[,c(2:n.treat)]<-expit(logit(p.rel[,1])+lor.rel[,c(2:n.treat)-1])
```

- The expit and logit functions work on vectors and matrices.
- Can set n.treat to any number without having to duplicate code.

Funded by NIHR

Formatting results

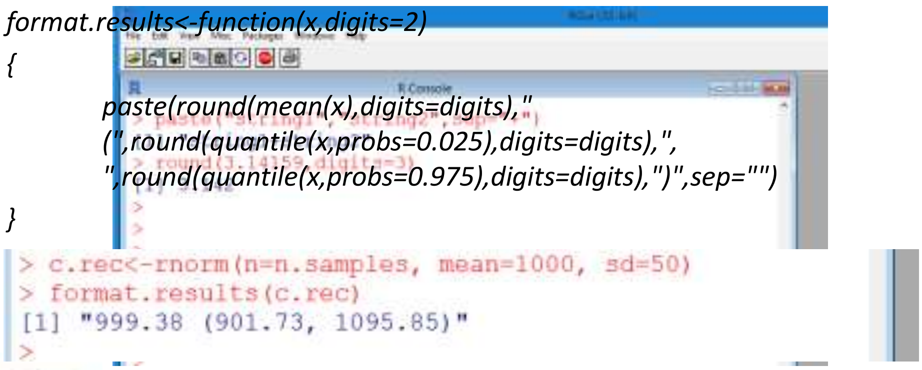
- Use *paste("string1", "string2")* function for string concatenation
- Use *round(x,digits=3)* for numeric formatting

format.results<-function(x,digits=2)

{

paste(round(mean(x),digits=digits)," (",round(quantile(x,probs=0.025),digits=digits),", ",round(quantile(x,probs=0.975),digits=digits),")",sep="")

}



17

Decision tree results

- Build a results matrix

results.matrix<-matrix(NA, nrow=4,ncol=n.treat)

- Name the rows and columns

rownames(results.matrix)<-c("Total costs","Total QALYs", "Net Benefit","Incremental NB")

colnames(results.matrix)<-t.names

- Then calculate summaries

for(i.treat in 1:n.treat)

{

results.matrix["Total costs",i.treat]<-format.results(x=costs[,i.treat])

results.matrix["Total QALYs",i.treat]<-format.results(x=effects[,i.treat])

results.matrix["Net Benefit",i.treat]<-format.results(x=net.benefit[,i.treat])

results.matrix["Incremental NB",i.treat]<-format.results(x=incremental.nb[,i.treat])

}

18

Exporting the results matrix to Excel

- Export as a csv

write.csv(results.matrix, file="depression.results.csv")

- Or as an Excel file

library(xlsx)

write.xlsx(results.matrix, file="depression.results.xlsx", sheetName="CEA results")

	No treatment	CBT	Antidepressant
Total costs	2458.08 (2216.38, 2692.91)	2678.9 (2424.37, 2937.03)	2366.58 (2087.97, 2621.49)
Total QALYs	20.09 (12.87, 27.59)	20.41 (13.54, 27.56)	20.59 (14.02, 27.52)
Net Benefit	549117.1)	548404.61)	548049.21)
Incremental NB	0 (0, 0)	6162.68 (-1978.38, 26095.58)	9996.67 (-2660.86, 36001.2)

So you can link back to Excel if you really can't resist.

And next?

- The model is available for you to try:
  https://github.com/Bogdasayen/Depression-toy-decision-tree-in-R
- A full Markov cost-effectiveness model is also available:
  https://github.com/Bogdasayen/DOACs-AF-Economic-model
- Bristol University will run a 2-day introductory course on R for Economic Evaluation.
  - Internal pilot in 2019, open externally in 2020.
- Devin will now show you what can be done once R is mastered…

Thank you!

The HTMR Collaboration and innovation in Difficult and Complex randomised controlled Trials In Invasive procedures (ConDuCT-II) hub provided support for this research.

This study was supported by the NIHR Biomedical Research Centre at the University Hospitals Bristol NHS Foundation Trust and the University of Bristol. The views expressed in this presentation are those of the author(s) and not necessarily those of the NHS, the National Institute for Health Research or the Department of Health.