University of Glasgow

MRC Hubs for Trials Methodology Research
ConDuCT-II Hub

University of BRISTOL

An overview of a suite of functions for CEA in R using continuous-time multi-state modelling

Dr Claire Williams

R for CEA Workshop, UCL

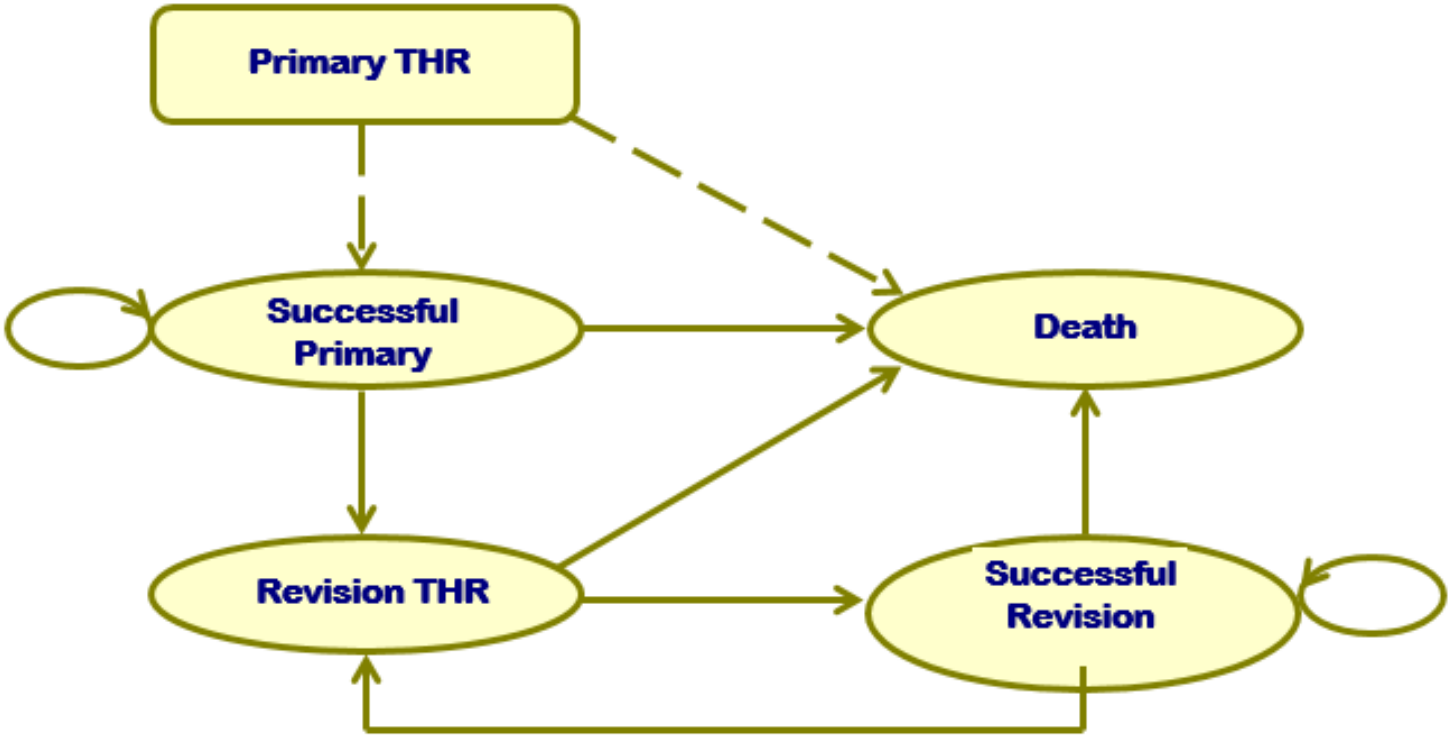
11th July 2018

Motivation

- Excel-based Markov modelling course gave the incentive to use `R` as an alternative

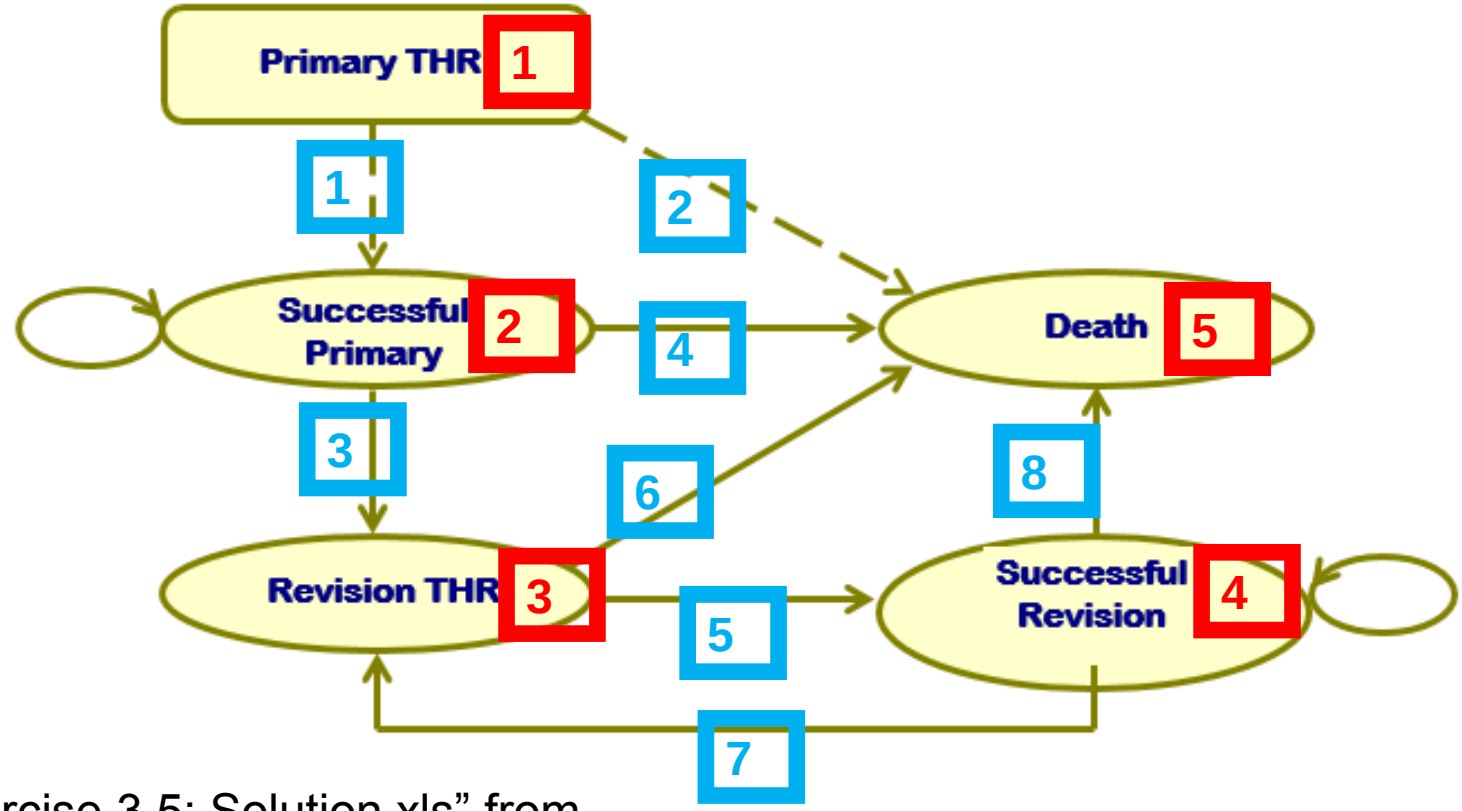
- Particular approach is based on `mstate` package in `R`
  - `mstate` uses IPD to build Cox models for each transition
  - `mstate` was adapted to use parametric distributions for the hazards: exponential, Weibull, Gompertz, log normal, log-logistic, generalised gamma
  - original motivation: extrapolation
  - another advantage: don't necessarily need IPD – just need to supply a cumulative hazard

Hip fracture Markov model example



Source: “Exercise 3.5: Solution.xls” from https://www.gla.ac.uk/hehta/continuingprofessionaldevelopment/advancedmodellingmethods/#/downloads

Hip fracture Markov model example



Source: “Exercise 3.5: Solution.xls” from https://www.gla.ac.uk/hehta/continuingprofessionaldevelopment/advancedmodellingmethods/#/downloads

State transition diagram R code

```
tmat <- transMat(x = list(c(2, 5), c(3,5),c(4,5),
c(3,5), c()), names=c("Primary", "PriSuc",
                      "Revision", "RevSuc","Death"))
```

```
tmat
           to
from        Primary PriSuc Revision RevSuc Death
  Primary        NA      1       NA     NA     2
  PriSuc         NA     NA        3     NA     4
  Revision       NA     NA       NA      5     6
  RevSuc         NA     NA        7     NA     8
  Death          NA     NA       NA     NA    NA
```

State transition diagram R code

```
tmat <- transMat(x = list(c(2, 5), c(3,5),c(4,5),
c(3,5), c()), names=c("Primary", "PriSuc",
                      "Revision", "RevSuc","Death"))
```

```
tmat
           to
from        Primary PriSuc Revision RevSuc Death
  Primary        NA       1       NA     NA     2
  PriSuc         NA      NA        3     NA     4
  Revision       NA      NA       NA      5     6
  RevSuc         NA      NA        7     NA     8
  Death          NA      NA       NA     NA    NA
```

State transition diagram R code

```
tmat <- transMat(x = list(c(2, 5), c(3,5),c(4,5),
c(3,5), c()), names=c("Primary", "PriSuc",
                      "Revision", "RevSuc","Death"))
```

```
tmat
           to
from        Primary PriSuc Revision RevSuc Death
  Primary        NA      1       NA     NA     2
  PriSuc         NA     NA        3     NA     4
  Revision       NA     NA       NA      5     6
  RevSuc         NA     NA        7     NA     8
  Death          NA     NA       NA     NA    NA
```

Alternative approach to relaxing the Markov property

Markov property - future states depend only on the present state and not on how it arrived in the present state

- Markov property can be formally tested by including in the model a covariate that represents history e.g. time in previous state
- If covariate is found to have a statistically significant effect there is evidence that the Markov property doesn't hold

Modelling progression → death
time in previous state included as a covariate

```
Call:
coxph(formula = Surv(Tstart, Tstop, status) ~ treat +
   prog_ty,
data = msmcancer3, method = "breslow")
n= 254, number of events= 50


             coef exp(coef) se(coef)      z Pr(>|z|)
treat      0.4416    1.5552   0.2938  1.503  0.13280
prog_ty   -0.8832    0.4135   0.3332 -2.651  0.00804
```

Markov and semi-Markov models

Markov models

- time measured from initial state regardless of state
- inherent Markov property
- predictions use Markov exact prediction formulae

Semi-Markov models

- time set back to zero every time patient enters a new state (clock-reset model)
- time depends on history so not Markov
- predictions use simulation sampling through available paths in the model

IPD – based functions: Markov or semi-Markov models

- `modelparam`
- `Markov` `semiMarkov`
- `visualMarkov` `visualsemiMarkov`
- `meanLY`
- `PSAprob`
- `PSAmeanLY` `PSAQALY`
- `CEplane`
- `CEAC`

`modelparam` function

```
modelparam(Markov=FALSE,covs="treat", trans
num=3, dist="wei", data=msmcancer)

Covariate           W.mean      Coef Exp(Coef)  se(Coef)     Wald p
treat                0.389     0.237     1.267     0.300      0.430

log(scale)                     1.368                0.218      0.000
log(shape)                     0.143                0.116      0.218

Events                    45
Total time at risk        192.69
Max. log. likelihood      -109.46
LR test statistic         0.62
Degrees of freedom        1
Overall p-value           0.432542
```

`Markov` and `semiMarkov` functions

```
Markov(ntrans=3, ncovs=c(1,1,2),
    covs=rbind("covariate1", "covariate1",
    c("covariate1", "covariate2")) ,
    coveval=rbind(0,0,c(0,1)),
    dist=cbind("wei", "wei", "wei"),
    dist2=cbind(NA, NA, NA),
    timeseq=seq(0,4,1/12),
    timeseq_ext=seq(49/12,15,1/12),
    data=msmcancer, trans=tmat)
```

`visualMarkov` and `visualsemiMarkov` functions

```
visualMarkov(nobjects=6,
  objects=rbind(weiexactRFC, expexactRFC, gomexactRFC,
  loglexactRFC, lognexactRFC, gamexactRFC),
  objects2=rbind(weiexactRFC[[2]], expexactRFC[[2]],
  gomexactRFC[[2]], loglexactRFC[[2]], lognexactRFC[[2]],
  gamexactRFC[[2]]) ,state=1, instate=TRUE, absorb=1,
  initialstate=TRUE,predfrom=2,
  tteach=c(181,181, 181, 181, 181, 181),
  ylab = "Probability of being progression-free",
  xlab="Years since start of study",ylim=c(0,1),xlim=c(0,15),
  lwd=2, col=rep("black",6),
  lty=c("51","dotted","21", "longdash","twodash", "16"),

continued on next slide
```

`visualMarkov` and `visualsemiMarkov` functions

```
obsdata=RFCdata,  CI=FALSE, observed="KM",
KMtime=RFCdata$progdeath_ty, KMstatus=RFCdata$progdeath,
ncr=2, legendpos="topright",legendncol=1,legendcex=1,
  legendcurves=c("Kaplan-Meier", "Exponential", "Lognormal",
  "Loglogistic", "Weibull", "Generalised gamma", "Gompertz"),
 legendlty=c("solid","dotted","twodash","longdash","51","16",
"21" ),
  legendbty="n" ,legendlwd=2,legendcol="black",
  main="Probability of being progression-free",
cex.main=1.25)
```

`meanLY` function

```
meanLY(Markov=FALSE,object, state=1, instate=TRUE,
       discounted=TRUE, dis1yronwards=TRUE, rate=0.035)
```

	RFC	FC	Incremental
Mean Life Years	5.82	5.60	0.21
Mean Life Years Progression-free	3.30	2.56	0.74
Mean Life Years in Progression	2.52	3.04	-0.53
Mean QALYs	4.15	3.87	0.28
Mean QALYs Progression-free	2.64	2.05	0.59
Mean QALYs in Progression	1.51	1.83	-0.32

incorporating costs

	RFC	FC	Incremental
Mean Life Years	5.82	5.60	0.21
Mean Life Years Progression-free	3.30	2.56	0.74
Mean Life Years in Progression	2.52	3.04	-0.53
Mean QALYs	4.15	3.87	0.28
Mean QALYs Progression-free	2.64	2.05	0.59
Mean QALYs in Progression	1.51	1.83	-0.32
Mean Total Cost	£25,917	£15,508	**£10,408**
Cost per Life Year Gained			**£48,772**
Cost per QALY Gained			**£37,665**

PSAprob function

```
PSAprob(ntrans=3, ncovs=c(1,1,2),
 covs=rbind("treat", "treat", c("treat", "lemedian")) ,
                coveval=rbind(0,0,c(0,0)),
                dist=cbind("wei", "wei", "wei"),
                dist2=cbind(NA, NA, NA),
                timeseq=seq(0,4,1/12),
                timeseq_ext=seq(49/12,15,1/12),
                data=msmcancer, trans=tmat,
                Markov=FALSE,
                nruns=1000,seedno=12345, M=100)
```

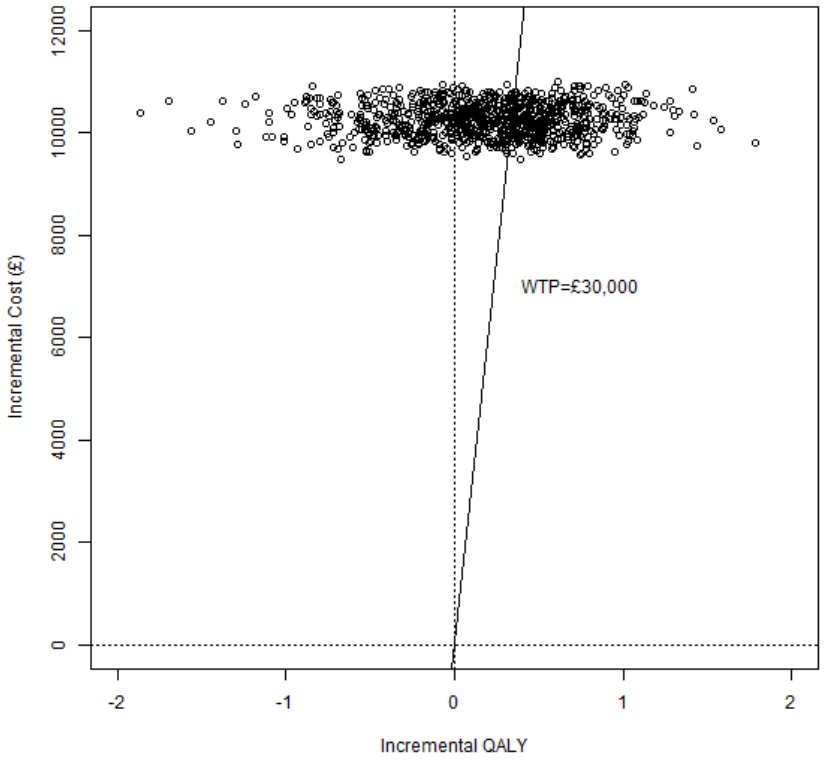
PSAmeanLY and PSAQALY functions

```
PSAmeanLY(object=PSASMSARFC, state=1,instate=TRUE,
discounted=TRUE, rate=0.035))

set.seed(12345)
utility1=rbeta(1000,800,200)

PSARFCQALYPFSdis<- PSAQALY(object=PSASMSARFC,
utility=utility1, state=1,discounted=TRUE,
dis1yronwards=TRUE, rate=0.035)

mean(PSARFCQALYPFSdis)
```

then incorporate costs as probabilistic

`CEplane` function

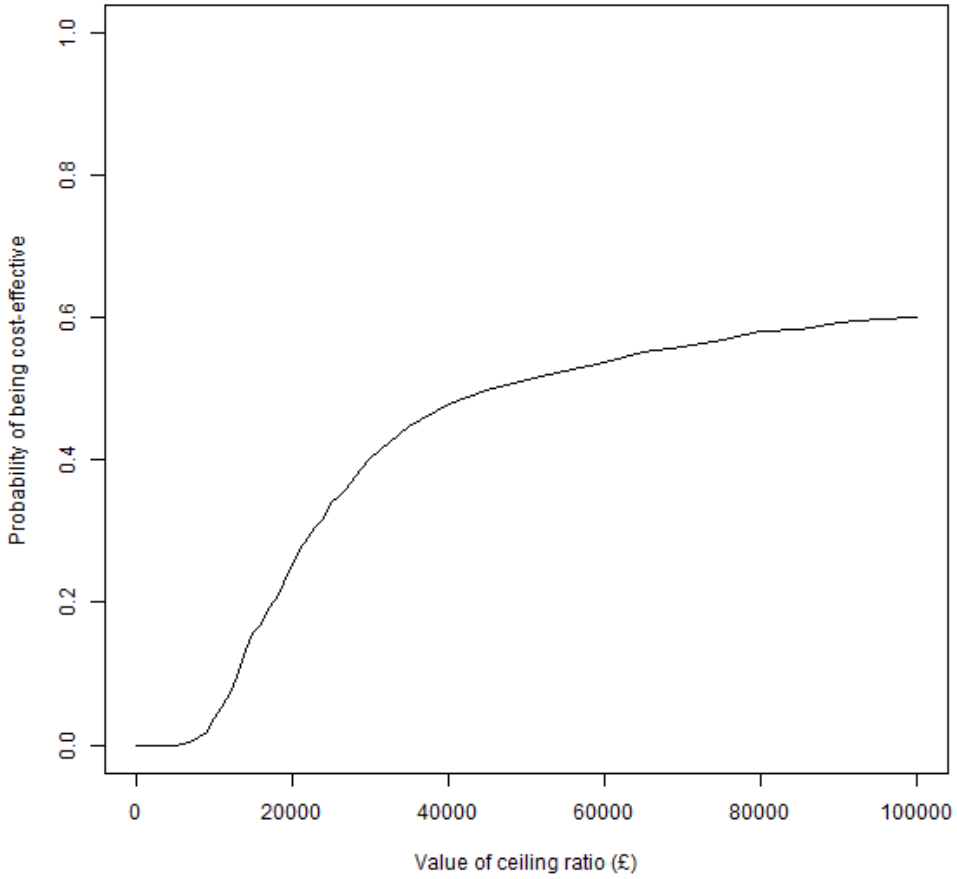
```
CEplane(x=incQALY, y=incCost, xlower=-2,
xupper=2,ylower=0, yupper=20000, ICER=30000,
text="ICER= £30,000")
```



CEAC function

```
CEAC(cRatiosim=c(seq(0,1000,100),
seq(1500,5000,500),seq(6000,30000,1000),
seq(35000,100000,5000)),nruns=1000,nruns2=1000
, xlower=0, xupper=100000, ylower=0, yupper=1,
QALY1=QALY_RFC_dis, QALY2=QALY_FC_dis,
cost1=total_cost_RFC_dis,
cost2=total_cost_FC_dis,
secondcurve=FALSE, QALY1_2=QALY_RFC_dis,
QALY2_2=QALY_FC_dis,
cost1_2=total_cost_RFC_dis,
cost2_2=total_cost_FC_dis))
```

CEAC



functions that can be used without IPD

`Markov_noipd` and `semiMarkov_noipd`

- allow previous model output and background mortality rates to be used to inform transition assumptions
- somewhat limited at the moment

Excel hip fracture model has been coded in `R`

- If interested please email me for my code:

claire.williams@bristol.ac.uk

Not included but discussed by others

- Previous speakers already discussed VoI analysis in R
- Boby and Iryna demonstrated shiny to create a more user-friendly front-end
- Looking forward: encourage more use of R for health economic modelling

Acknowledgements

Financial support for this research was provided entirely by a doctoral training grant from the Medical Research Council in the United Kingdom (MR/J50032X/1).

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

Reference

Williams, C., Lewsey, J. D., Briggs, A. H. , Mackay, D. F. (2017) Cost-effectiveness analysis in R using a multi-state modeling survival analysis framework: a tutorial. *Medical Decision Making,* 37:340-352. doi:10.1177/0272989X16651869