02-B-VoI-Casestudy-pop-vs-pred

Sandro Gsteiger

July 11, 2020

Table of Contents

# Description

Use very simple, stylized desision model to assess the relevance of predictive uncertainty with tumour agnostic therapies.

Compare EVPI (and possibly EVPPI) for two decision rules:

1. Use what might represent single-tissue (or population?) estimates for decision making (and their uncertainty.
2. Use what might represent the predicted outcomes in a new tissue in the decision rule.

# Stylized decision model

Partitioned survival model.

Weibull model for OS and PFS, shape parameter assumed the same. Shape parameter estimated from Larotrectinib OS data. Scale paramters (with uncertainty) estimated from Larotrectinib 12-months survival data (OS and PFS).

Assessment of impact of predictive uncertainty (new tissue type).

Implementation by adapting functions publically available from DARTH group (<https://github.com/DARTH-git/Partitioned-Survival-Analysis>).

# Base case analysis

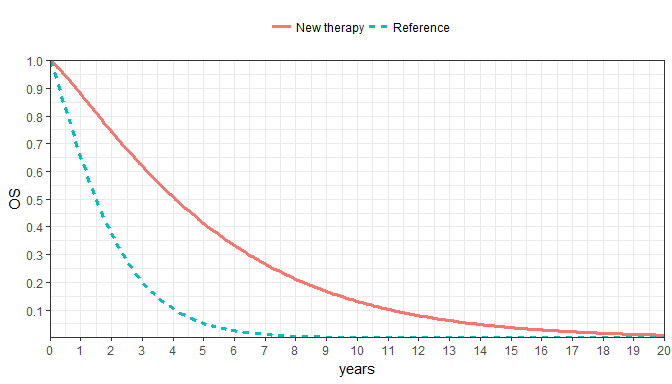
## Model inputs

* Baseline survival rates (new treatment): OS and PFS rates and their standard errors roughly taken from Laro data
* Hazard ratios to obtain comparator group: assumptions (set to obtain “workable” model for case study, i.e. new intervention reasonnable likelihood of being cost effective)
* Utilities: assumptions based on from lit
  + Pre-pog: smal diff. between treatments
  + Post-progression:
    - no difference between the two groups
    - this was the preferred assumption by the NICE committee in the Laro dossier
    - the manufacturer and some clinical experts said difference in post-prog. utility were plausible
* Costs:
  + New trt, pre-prog:
    - per pERC report about 20k CAN $ per month; -> 12k GBP
    - per NICE ACM2: list price GBP 15k per month (then confidential discount)
  + Ref trt, pre-prog: per pERC about 2k CAN $, though extreme between tissue variability
  + Post-progression state (indep. of trt); assumption
  + Make rough assumption for total costs per cycle (reflecting also drug administration, other direct and indirect medical costs such as for ecample those induced by AEs etc.)
* Discount rate: 3.5% for costs and outcomes per NHS/NICE recommendation
* Time-horizon: 30 years.

Note that *time scale* is *years* (but *cycle length* is *months*).

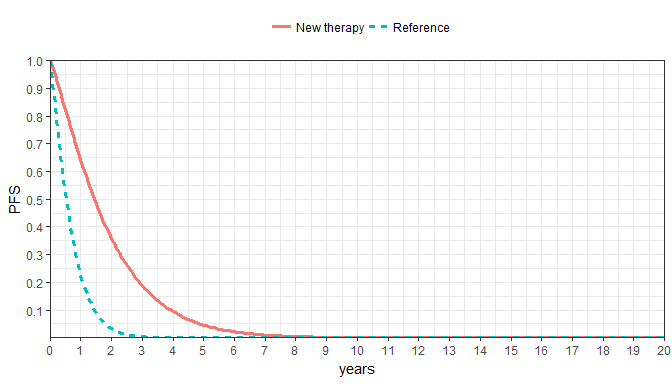
model\_pars <- list(  
 shape = 1.2,  
 OS\_scale = 0.128,   
 PFS\_scale = 0.446,  
 HR\_OS = 0.3,  
 HR\_PFS = 0.3,  
 u\_pre\_trt = 0.8, # utility in pre-progrsssion state, new treatment  
 u\_pre\_ref = 0.7, # idem, reference treatment  
 u\_post = 0.6, # utility, post-progression state  
 u\_death = 0,  
 c\_pre\_trt = 4500, # costs in pre-progrsssion state per cycle, new treatment  
 c\_pre\_ref = 3000, # idem, reference treatment  
 c\_post = 1000, # costs per cycle, post-progression state  
 c\_death = 0   
)  
  
model\_settings <- list(  
 state\_names = c("Pre-progression", "Post-progression", "Death"),  
 horizon = 30,  
 cycle\_length = 1 / 12, # cycle length 1-month (time scale: years)   
 disc\_rate = 0.035 # rate per year  
)

## Visualize the resulting survival models

**Figure** Resulting OS Weibull models 

**Figure** Resulting PFS Weibull models

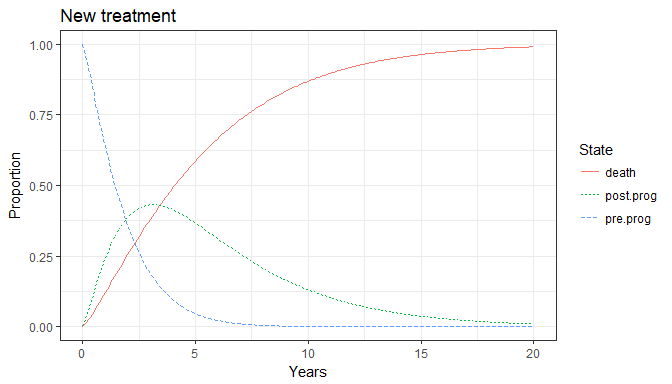
## Warning: Removed 240 rows containing missing values (geom\_path).



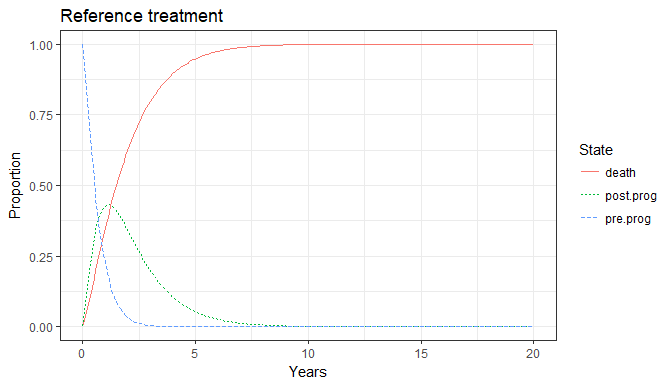
## Economic evaluation

**Figure** Markov traces (base case analysis)

## Warning: Removed 360 rows containing missing values (geom\_path).



## Warning: Removed 360 rows containing missing values (geom\_path).



Calculate total discounted costs and QALYs for the new and the reference treatments.

\*\* Model outputs (base case)\*\*

## new treatment  
ee\_trt

## total.costs total.QALYs   
## 1.299364e+05 3.094613e+00

## ref treatment  
ee\_ref

## total.costs total.QALYs   
## 39151.669666 1.182232

## incremental costs and QALYs  
delta <- ee\_trt - ee\_ref  
names(delta) <- paste("delta.", names(delta), sep = "")  
delta

## delta.total.costs delta.total.QALYs   
## 90784.728134 1.912381

## ICER  
ICER <- c(costs.per.QALY = delta[[1]] / delta[[2]])  
ICER

## costs.per.QALY   
## 47472.1

# PSA: population model

## Distributions used for parameter uncertainty

# OS\_scale parameter, new trt:  
OS\_scale\_lsd <- (log(0.211) - log(0.0513)) / 3.92  
OS\_scale\_lsd

## [1] 0.360757

dscale <- data.frame(x = seq(0.01, 0.3, 0.01)) %>%  
 mutate(y = dlnorm(x, meanlog = log(model\_pars$OS\_scale), sdlog = OS\_scale\_lsd))  
qplot(x, y, data = dscale, geom = "line") +   
 labs(title = "OS", x = "scale", y = "")



qlnorm(meanlog = log(model\_pars$OS\_scale), sdlog = OS\_scale\_lsd, p = c(0.025, 0.975))

## [1] 0.06311507 0.25958935

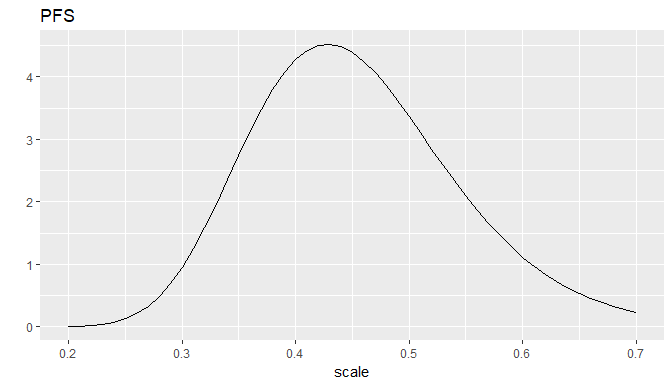
# PFS\_scale parameter, new trt:  
qnorm(mean = 0.446, sd = (0.635 - 0.288)/6, p = c(0.025, 0.975))

## [1] 0.3326487 0.5593513

PFS\_scale\_lsd <- (log(0.635) - log(0.288)) / 3.92  
PFS\_scale\_lsd

## [1] 0.2017001

dscale <- data.frame(x = seq(0.2, 0.7, 0.01)) %>%  
 mutate(y = dlnorm(x, meanlog = log(model\_pars$PFS\_scale), sdlog = PFS\_scale\_lsd))  
qplot(x, y, data = dscale, geom = "line") +   
 labs(title = "PFS", x = "scale", y = "")



qlnorm(meanlog = log(model\_pars$PFS\_scale), sdlog = PFS\_scale\_lsd, p = c(0.025, 0.975))

## [1] 0.3003637 0.6622505

# HR OS, PFS ~ Beta(30, 70) => Expectation: 3/10  
qbeta(30, 70, p = c(0.025, 0.975))

## [1] 0.2146680 0.3929087

# costs pre-progression state new treatment  
# median:   
model\_pars$c\_pre\_trt

## [1] 4500

qlnorm(meanlog = log(model\_pars$c\_pre\_trt), sdlog = 0.05, p = c(0.025, 0.975))

## [1] 4079.927 4963.324

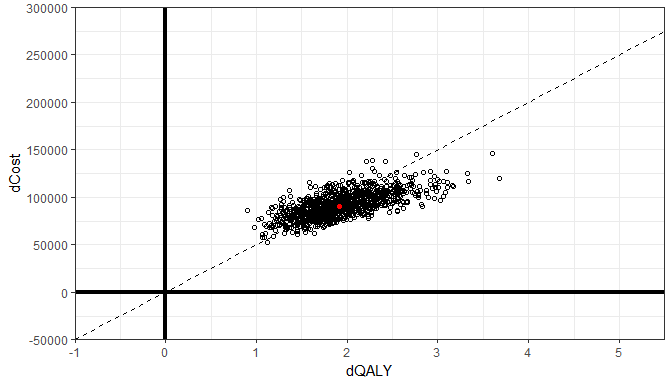
# costs pre-progression state ref treatment  
# median:   
model\_pars$c\_pre\_ref

## [1] 3000

qlnorm(meanlog = log(model\_pars$c\_pre\_ref), sdlog = 0.05, p = c(0.025, 0.975))

## [1] 2719.952 3308.882

## Run the PSA and plot the cost-effectiveness plane



## Summarize PSA outputs and (re)analyze the PSA outputs with the BCEA package

# PSA outputs  
e <- model\_psa[, c("QALYs.trt", "QALYs.ref")] # ref in bcea terminology is our trt!!  
c <- model\_psa[, c("costs.trt", "costs.ref")] # caution to this !!  
me <- apply(e, MAR = 2, FUN = mean)  
me

## QALYs.trt QALYs.ref   
## 3.133103 1.201379

mc <- apply(c, MAR = 2, FUN = mean)  
mc

## costs.trt costs.ref   
## 131710.90 39773.12

diff(mc) / diff(me)

## costs.ref   
## 47593.64

# Compare with basecase analysis  
ee\_trt

## total.costs total.QALYs   
## 1.299364e+05 3.094613e+00

ee\_ref

## total.costs total.QALYs   
## 39151.669666 1.182232

delta

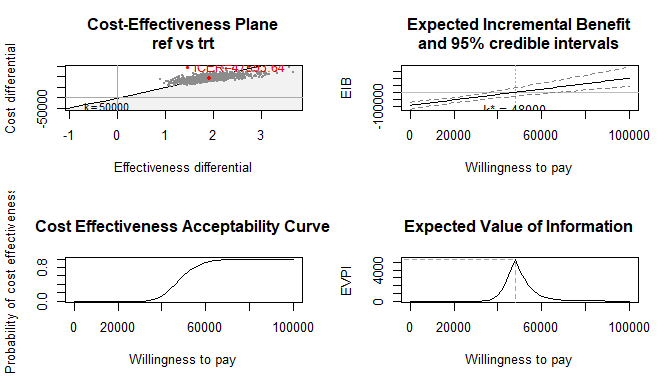
## delta.total.costs delta.total.QALYs   
## 90784.728134 1.912381

ICER

## costs.per.QALY   
## 47472.1

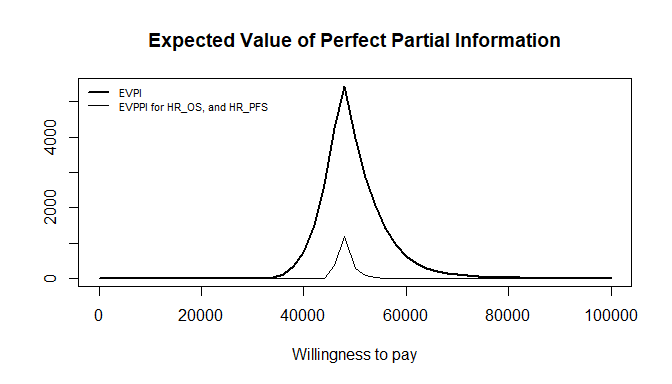
# re-analyze with bcea  
res <- bcea(e, c, interventions = c("ref", "trt"),   
 Kmax = 100000, wtp = seq(0, 100000, 2000))  
  
summary.bcea(res, wtp = 50000)

##   
## Cost-effectiveness analysis summary   
##   
## Reference intervention: ref  
## Comparator intervention: trt  
##   
##   
##   
## Analysis for willingness to pay parameter k = 50000  
##   
## Expected utility  
## ref 24944  
## trt 20296  
##   
## EIB CEAC ICER  
## ref vs trt 4648.4 0.616 47594  
##   
## Optimal intervention (max expected utility) for k=50000: ref  
##   
## EVPI 3958.1

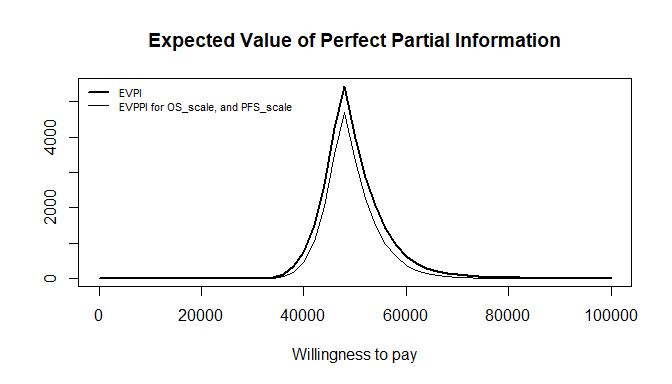


## Calculate EVPPI

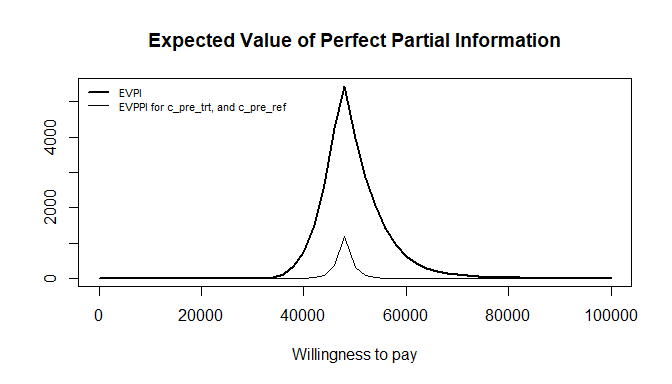
##   
## Calculating fitted values for the GAM regression   
##   
## Calculating fitted values for the GAM regression   
## Calculating EVPPI



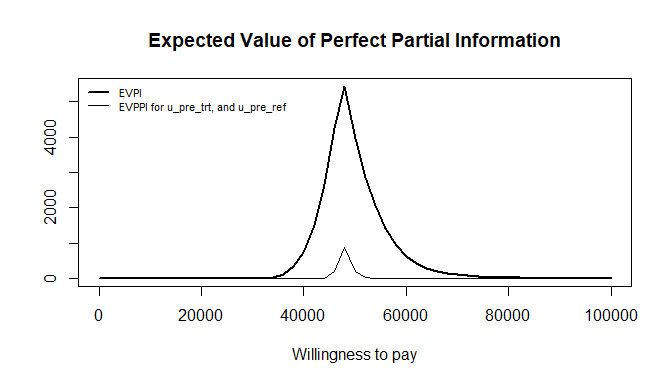
##   
## Calculating fitted values for the GAM regression   
##   
## Calculating fitted values for the GAM regression   
## Calculating EVPPI



##   
## Calculating fitted values for the GAM regression   
##   
## Calculating fitted values for the GAM regression   
## Calculating EVPPI



##   
## Calculating fitted values for the GAM regression   
##   
## Calculating fitted values for the GAM regression   
## Calculating EVPPI



# Redo PSA, but for predictive distribution (new tissue) - add uncertainty

Add predictive uncertainty wrt HRs, and costs.

Keep the baseline model (which per se would be expected to vary too!).

## Run the PSA and plot the cost-effectiveness plane

Summaries of distributions used for parameter uncertainty. In particular with respect to reference treatment.

# HR OS, PFS ~ Beta(1.5, 3.5) => Expectation: 3/10, but very uncertain  
qbeta(1.5, 3.5, p = c(0.025, 0.975))

## [1] 0.0284709 0.7162483

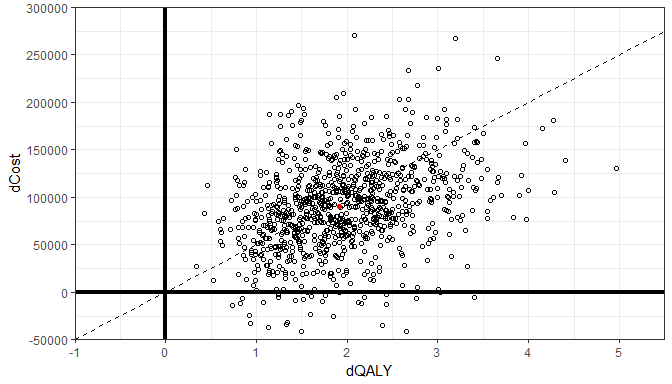
# costs pre-progression state new treatment  
qlnorm(meanlog = log(model\_pars$c\_pre\_trt), sdlog = 0.3, p = c(0.025, 0.25, 0.5, 0.75, 0.975))

## [1] 2499.494 3675.652 4500.000 5509.227 8101.641

# costs pre-progression state ref treatment  
# median: 1000  
qlnorm(meanlog = log(model\_pars$c\_pre\_ref), sdlog = 0.75, p = c(0.025, 0.25, 0.5, 0.75, 0.975))

## [1] 689.7951 1808.9471 3000.0000 4975.2698 13047.3530

## Warning: Removed 14 rows containing missing values (geom\_point).



## Summarize PSA outputs and (re)analyze the PSA outputs with the BCEA package

# PSA outputs  
e2 <- model\_psa2[, c("QALYs.trt", "QALYs.ref")] # ref in bcea terminology is our trt!!  
c2 <- model\_psa2[, c("costs.trt", "costs.ref")] # caution to this !!  
me2 <- apply(e2, MAR = 2, FUN = mean)  
me2

## QALYs.trt QALYs.ref   
## 3.165825 1.205403

mc2 <- apply(c2, MAR = 2, FUN = mean)  
mc2

## costs.trt costs.ref   
## 136331.60 48525.19

diff(mc2) / diff(me2)

## costs.ref   
## 44789.56

# Compare with basecase analysis  
ee\_trt

## total.costs total.QALYs   
## 129936.397800 3.094613

ee\_ref

## total.costs total.QALYs   
## 39151.669666 1.182232

delta

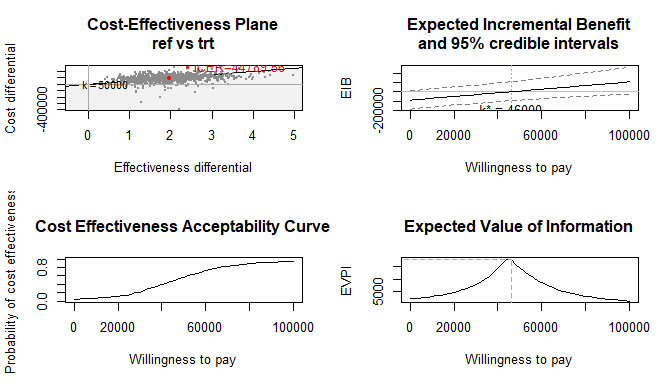
## delta.total.costs delta.total.QALYs   
## 90784.728134 1.912381

ICER

## costs.per.QALY   
## 47472.1

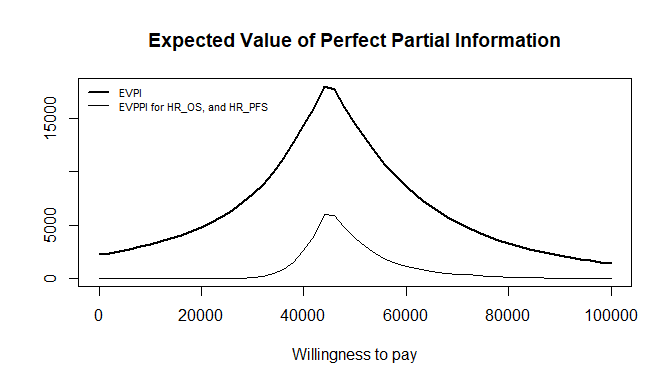
# re-analyze with bcea  
res2 <- bcea(e2, c2, interventions = c("ref", "trt"),   
 Kmax = 100000, wtp = seq(0, 100000, 2000))  
  
summary.bcea(res2, wtp = 50000)

##   
## Cost-effectiveness analysis summary   
##   
## Reference intervention: ref  
## Comparator intervention: trt  
##   
##   
##   
## Analysis for willingness to pay parameter k = 50000  
##   
## Expected utility  
## ref 21960  
## trt 11745  
##   
## EIB CEAC ICER  
## ref vs trt 10215 0.565 44790  
##   
## Optimal intervention (max expected utility) for k=50000: ref  
##   
## EVPI 14485

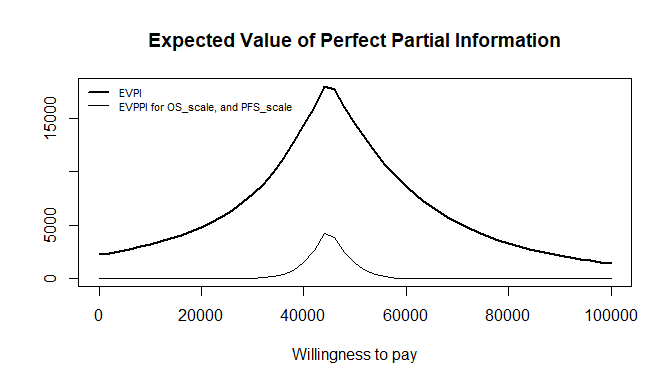


## Calculate EVPPI

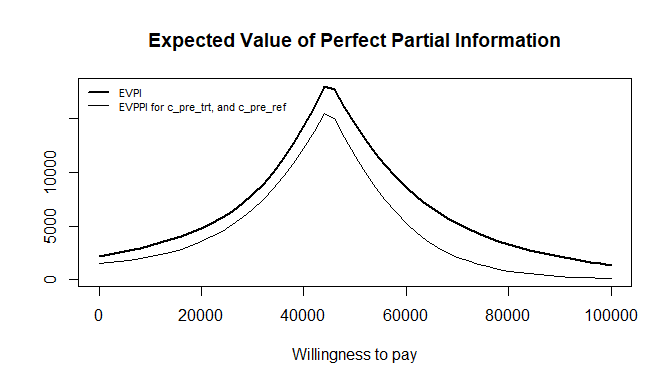
##   
## Calculating fitted values for the GAM regression   
##   
## Calculating fitted values for the GAM regression   
## Calculating EVPPI



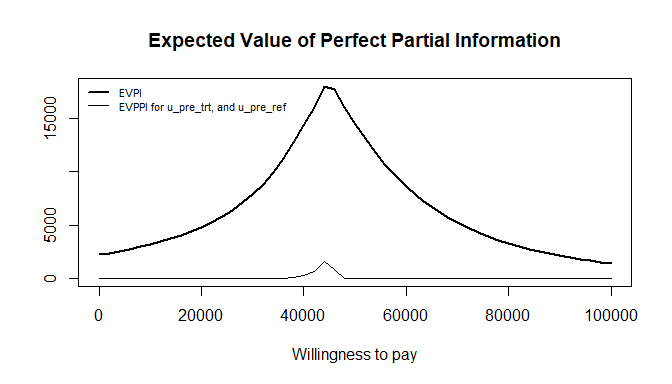
##   
## Calculating fitted values for the GAM regression   
##   
## Calculating fitted values for the GAM regression   
## Calculating EVPPI



##   
## Calculating fitted values for the GAM regression   
##   
## Calculating fitted values for the GAM regression   
## Calculating EVPPI



##   
## Calculating fitted values for the GAM regression   
##   
## Calculating fitted values for the GAM regression   
## Calculating EVPPI



# Session info

## [1] "C:/GIT/punta"

## R version 3.4.2 (2017-09-28)  
## Platform: x86\_64-w64-mingw32/x64 (64-bit)  
## Running under: Windows 10 x64 (build 17763)  
##   
## Matrix products: default  
##   
## locale:  
## [1] LC\_COLLATE=English\_United Kingdom.1252   
## [2] LC\_CTYPE=English\_United Kingdom.1252   
## [3] LC\_MONETARY=English\_United Kingdom.1252  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United Kingdom.1252   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] BCEA\_2.3-1.1 ggplot2\_2.2.1 tidyr\_0.8.3 dplyr\_0.8.0.1 readxl\_1.3.1   
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_1.0.1 knitr\_1.28 magrittr\_1.5 munsell\_0.4.3   
## [5] tidyselect\_0.2.5 lattice\_0.20-35 colorspace\_1.3-2 R6\_2.2.2   
## [9] rlang\_0.3.4 plyr\_1.8.4 stringr\_1.4.0 tools\_3.4.2   
## [13] grid\_3.4.2 nlme\_3.1-131 gtable\_0.2.0 mgcv\_1.8-20   
## [17] xfun\_0.12 htmltools\_0.3.6 lazyeval\_0.2.1 yaml\_2.2.0   
## [21] digest\_0.6.12 assertthat\_0.2.0 tibble\_2.1.1 crayon\_1.3.4   
## [25] Matrix\_1.2-11 purrr\_0.3.2 glue\_1.3.1 evaluate\_0.14   
## [29] rmarkdown\_2.1 labeling\_0.3 stringi\_1.4.3 compiler\_3.4.2   
## [33] pillar\_1.4.3 cellranger\_1.1.0 scales\_0.5.0 pkgconfig\_2.0.3