

Case Study - Abdominal aortic aneurysm

Xavier G.L.V. Pouwels

Introduction

This vignette showcases the functionalities of `pacheck` on the [AAA_DES_model](#), which is described in Sweeting et al. (2018) . The probabilistic set of inputs and outputs has been obtained by running the `Main file_men 30years_FullModel.R` file under the `models` folder of the R project. To limit computational time, we only simulated 1,000 iterations containing 1,000 individuals (which may not be enough to obtain stable results).

Loading packages and data

```
#library(pacheck)
library(purrr)
devtools::load_all()
```

i Loading pacheck

Warning: replacing previous import 'glmnet::na.replace' by 'gtools::na.replace' when loading 'pacheck'

Warning: replacing previous import 'boot::logit' by 'gtools::logit' when loading 'pacheck'

Warning: replacing previous import 'boot::inv.logit' by 'gtools::inv.logit' when loading 'pacheck'

Warning: replacing previous import 'magrittr::is_less_than' by 'testthat::is_less_than' when loading 'pacheck'

Warning: replacing previous import 'magrittr::not' by 'testthat::not' when loading 'pacheck'

Warning: replacing previous import 'magrittr::equals' by 'testthat::equals' when loading 'pacheck'

Warning: replacing previous import 'dplyr::matches' by 'testthat::matches' when loading 'pacheck'

Warning: replacing previous import 'assertthat::has_name' by 'tibble::has_name' when loading 'pacheck'

Warning: replacing previous import 'magrittr::extract' by 'tidyr::extract' when loading 'pacheck'

Warning: replacing previous import 'testthat::matches' by 'tidyr::matches' when loading 'pacheck'

```
set.seed(1234)
data("l_psa_aaa")
```

Preparing dataset for use in pacheck

The following code chunk prepares the inputs and outputs list of the AAA DES model to use the `pacheck` tests. In this example, only a subset of all inputs is used for convenience.

```
# Inputs
## Input parameters are stored in a list and
## have attributes that are used here to retrieve
## the different types of parameters
## using the `map_depth` function from the `purrr` package
v_attr_type <- unlist(lapply(l_psa_aaa$v2values[[1]], function(l_obj)
  attr(l_obj, which = "type")), use.names = F)
v_probnames <- names(l_psa_aaa$v2value[[1]])[which(v_attr_type == "probability")]
v_ratenames <- names(l_psa_aaa$v2value[[1]])[grep("rate", v_attr_type)]
v_qalynames <- names(l_psa_aaa$v2value[[1]])[grep("qaly", v_attr_type)]
v_costnames <- names(l_psa_aaa$v2value[[1]])[grep("cost", v_attr_type)]
v_parmodnames <- names(l_psa_aaa$v2value[[1]])[grep("par", v_attr_type)]
m_aaa_inputs <- sapply(c(v_probnames,
```

```

        v_ratenames,
        # v_qalynames, # omitted because fixed
        # v_costnames, # omitted because fixed
        v_parmodnames), function(name)
            unlist(purrr::map_depth(l_psa_aaa$v2values, 1, name), use.names =
# Outputs
## Outputs are stored in the 3D arra
m_aaa_res <- l_psa_aaa$psaQuantities
dim(m_aaa_res) <- c(unname(
  dim(l_psa_aaa$psaQuantities)[[1]] * dim(l_psa_aaa$psaQuantities)[[2]]
), unname(dim(l_psa_aaa$psaQuantities)[[3]]))
m_aaa_res <- t(m_aaa_res)
colnames(m_aaa_res) <- paste0(unlist(dimnames(l_psa_aaa$psaQuantities)[1]), "_", rep(unlist(
  l_psa_aaa$psaQuantities
)[2]), each = length(unlist(
  dimnames(l_psa_aaa$psaQuantities)[1]
))))

df_aaa <- data.frame(cbind(m_aaa_inputs, m_aaa_res))

```

Testing plausibility of model inputs and outputs

In this section, we test the plausibility of multiple groups of parameters and outcomes using the `do_quick_check` function. We also check whether discounted results are lower than corresponding non-discounted results using the `do_discount_check` function.

```

# Quick checks of groups of input parameter values
do_quick_check(df = df_aaa,
  v_probs = v_probnames,
  v_r = v_ratenames,
  v_costs = c("noScreening_cost",
    "screening_cost",
    "noScreening_discountedCost",
    "screening_discountedCost"))

```

```

Test passed
Test passed
Test passed
Test passed

```

	Test	Result
1	All probabilities are positive	TRUE
2	All probabilities are lower or equal to 1	TRUE
3	All utility values are positive	NOT PERFORMED
4	All utility values are lower or equal to 1	NOT PERFORMED
5	All costs parameters are positive	TRUE
6	All hazard ratios are positive	NOT PERFORMED
7	All relative risks are positive	NOT PERFORMED
8	All rates are positive	TRUE
9	All outcomes are positive	NOT PERFORMED

```
# Discounted results lower than undiscounted results
do_discount_check(
  df = df_aaa,
  v_outcomes = c(
    "noScreening_lifeYears",
    "screening_lifeYears",
    "noScreening_qalys",
    "screening_qalys",
    "noScreening_cost",
    "screening_cost"
  ),
  v_outcomes_d = c(
    "noScreening_discountedLifeYears",
    "screening_discountedLifeYears",
    "noScreening_discountedQalys",
    "screening_discountedQalys",
    "noScreening_discountedCost",
    "screening_discountedCost"
  )
)
```

Test passed

	Test	Result
1	All discounted outcomes are lower than undiscounted outcomes	TRUE

Drawing health economic figures

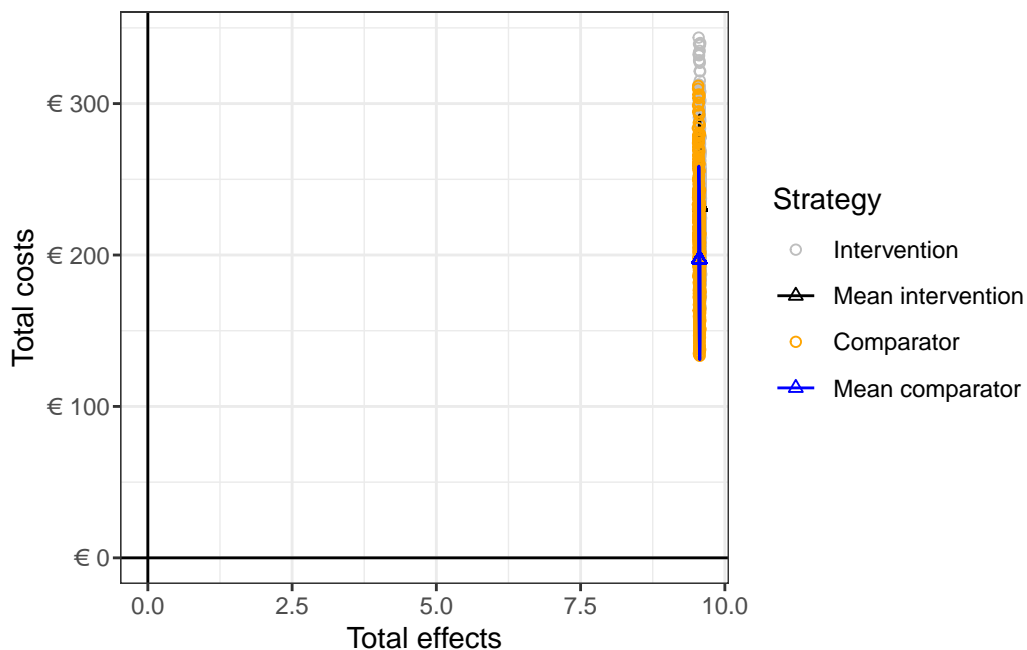
pacheck also contains figures to draw standard figures for health economic evaluations, such as the cost-effectiveness plane (`plot_ce` function), the incremental cost-effectiveness plane

(`plot_ice` function), the cost-effectiveness acceptability curves (`plot_ceac` function), and net benefit curves (`plot_nb` function). Both the `plot_ceac` and `plot_nb` functions require data frames containing specific variable names, which can respectively be obtained with the accompanying functions `calculate_ceac` and `calculate_nb`.

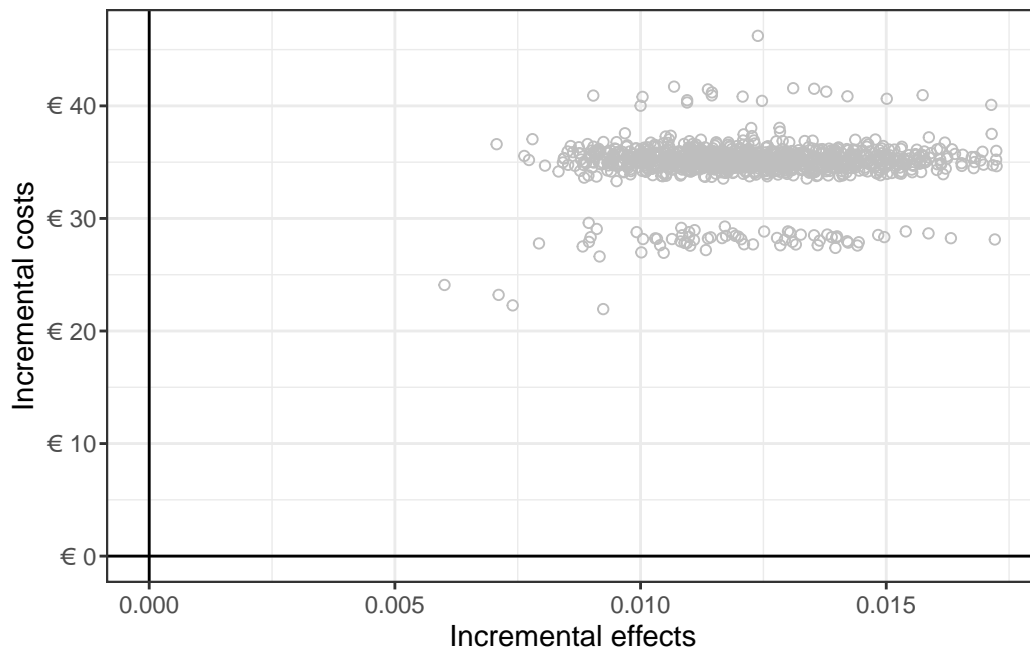
```
plot_ce(
  df = df_aaa,
  e_int = "screening_discountedQalys",
  e_comp = "noScreening_discountedQalys",
  c_int = "screening_discountedCost",
  c_comp = "noScreening_discountedCost"
)
```

Warning in `ggplot2::geom_point(data = df, ggplot2::aes_string(x = mean(df[, : All aesthetics`
i Please consider using ``annotate()`` or provide this layer with data containing
a single row.

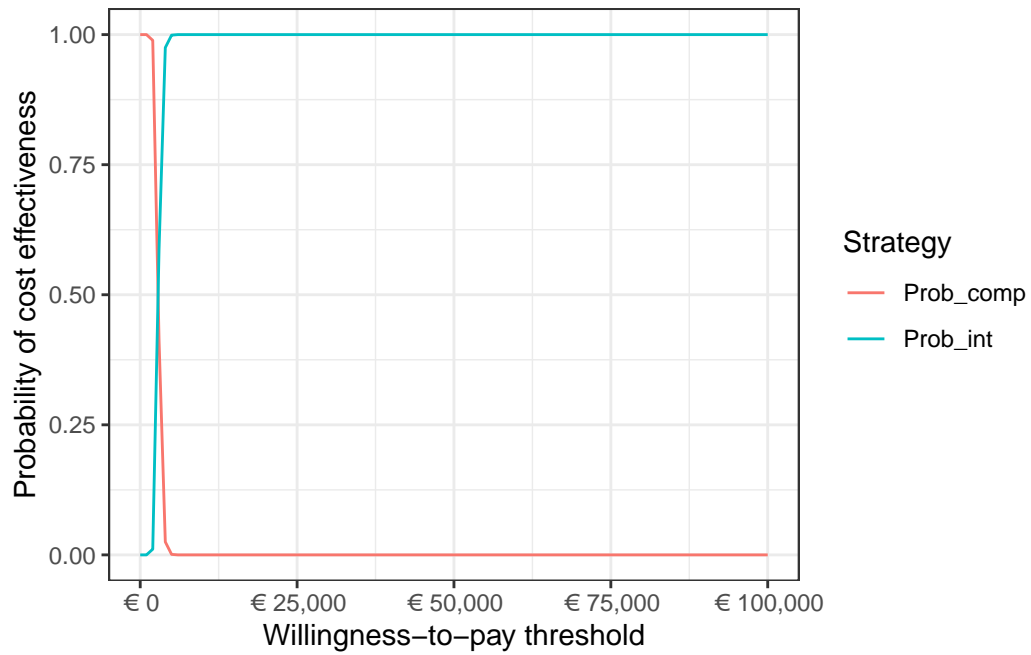
Warning in `ggplot2::geom_point(data = df, ggplot2::aes_string(x = mean(df[, : All aesthetics`
i Please consider using ``annotate()`` or provide this layer with data containing
a single row.



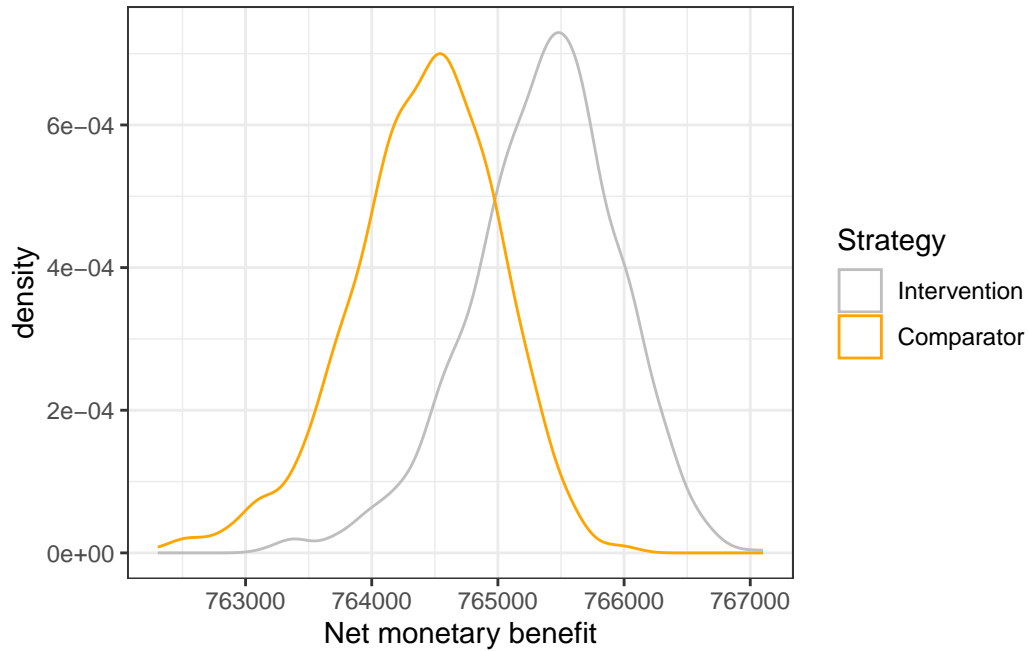
```
plot_ice(
  df = df_aaa,
  e_int = "screening_discountedQalys",
  e_comp = "noScreening_discountedQalys",
  c_int = "screening_discountedCost",
  c_comp = "noScreening_discountedCost"
)
```



```
df_ceac <- calculate_ceac(
  df = df_aaa,
  e_int = "screening_discountedQalys",
  e_comp = "noScreening_discountedQalys",
  c_int = "screening_discountedCost",
  c_comp = "noScreening_discountedCost"
)
plot_ceac(df = df_ceac, name_wtp = "WTP_threshold")
```



```
df_nb <- calculate_nb(  
  df = df_aaa,  
  e_int = "screening_discountedQalys",  
  e_comp = "noScreening_discountedQalys",  
  c_int = "screening_discountedCost",  
  c_comp = "noScreening_discountedCost",  
  wtp = 80000  
)  
plot_nb(df = df_nb)
```



Reference

Sweeting, Michael J., Katya L. Masconi, Edmund Jones, Pinar Ulug, Matthew J. Glover, Jonathan A. Michaels, Matthew J. Bown, Janet T. Powell, and Simon G. Thompson. 2018. "Analysis of Clinical Benefit, Harms, and Cost-Effectiveness of Screening Women for Abdominal Aortic Aneurysm." *The Lancet* 392 (10146): 487–95. [https://doi.org/10.1016/S0140-6736\(18\)31222-4](https://doi.org/10.1016/S0140-6736(18)31222-4).