



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání



Zdroj: <https://predmety.fbmi.cvut.cz/cs/doktorske-bme>

Balíček „heemod“

Příkazy pro tvorbu modelů

Definování parametrů modelu:

- `define_parameters(...)`

Definování matice přechodu:

- `define_transition(..., state_names)`

Definování hodnot stavů:

- `define_state(..., starting_values = define_starting_values())`

Definování intervencí (strategií):

- `define_strategy(..., transition = define_transition(), starting_values = define_starting_values())`

Spuštění simulace

- `run_model(..., parameters = define_parameters(), init = c(1000L, rep(0L, get_state_number(get_states(list(...)[[1]])) - 1)), cycles = 1, ,cost = NULL, effect = NULL, state_time_limit = NULL, central_strategy = NULL, inflow = rep(0L, get_state_number(get_states(list(...)[[1]]))))`

Definování deterministické analýzy citlivosti:

- `define_dsa(...)`

Spuštění deterministické analýzy citlivosti:

- `run_dsa(model, dsa)`

Definování probabilistické analýzy citlivosti:

- `define_psa(..., correlation)`

Spuštění probabilistické analýzy citlivosti:

- `run_psa(model, psa, N, resample)`

Přehled výsledků (simulací, analýz citlivosti):

- `summary()`

Zobrazení grafů:

- `plot()`

Ukázka příkazů:

Definování pravděpodobností přechodů:

- `define_transition` – funkce pro definování pravděpodobností
 - `mat_trans` – vytvořená proměnná - název matice (ten si můžete nazvat dle sebe)
 - `matice_A <- define_transition(.9, .1, .2, .8)`
- `matice_A <- define_transition(.9, .1, .2, .8, state_names = c("A", "B"))`
 - definování pravděpodobností přechodů a vlastních názvů stavů: `state_names`

Zobrazení modelové struktury:

- `plot` – příkaz pro vytvoření grafu, lze ho použít i u jiných funkcí (viz dále)
 - `plot(matice_A)`



Přiřazení hodnot stavům:

- `define_state` – funce pro definování hodnot
- `c` – náklady (ale můžeme si je zde pojmenovat jak chceme)
- `qol` – kvalita života (ale můžeme si je zde pojmenovat jak chceme)
- `A` – vytvořená proměnná pro přiřazení hodnot stavu `A` (můžeme si ji nazvat jak chceme)
 - `A <- define_state(cost = 1234, utility = 0.85)`
 - `B <- define_state(c = 4321, qol = 0.5)`

Kombinace informací v modelu:

- `define_strategy` – funkce pro přiřazení definovaných hodnot strategiím (porovnávaným intervencím)
- `inter` – vytvořená proměnná pro název strategie (můžete si zvolit jaký název chcete)
- `transition` – pro přiřazení nadefinovaných pravděpodobností přechodů
- `matice_A` – vytvořená proměnná s pravděpodobnostmi přechodu
- `state_A + state_B` – vytvořené proměnné s nadefinovanými hodnotami
 - `inter <- define_strategy(transition = mat_trans, state_A, state_B)`
- Pokud máme jména stavů můžeme použít:
 - `inter <- define_strategy(transition = mat_trans, A= state_A, B= state_B)`

Spuštění modelu:

- `run_model` – funkce pro spuštění modelu
- `results` – proměnná pro uložení výsledků modelu (můžeme si pojmenovat jakkoliv)
- `inter` – proměnná obsahující informace o modelovaných hodnotách strategie (naše pojmenování)
- `cycles` – určení počtu cyklů
- `cost` – určení počítaných nákladů (zde musíme určit pod jakou proměnnou máme uložené náklady, v našem případě `c`)
- `effect` – určení počítaného efektu (zde musíme určit pod jakou proměnnou máme uložené efekty, v našem případě kvalitu života – `qol`)
 - `results <- run_model(strat, cycles = 10, cost=cost, effect = utility)`
- Program počítá v základním nastavení výsledky pro kohortu 1000 pacientů (`A = 1000L`) vstupujících do modelu skrze stav `A`. Toto nastavení lze změnit pomocí `init`.

Analýza výsledků:

- `plot` – zobrazí graficky výsledky (je zde více možností jak graficky zobrazit výsledky)
- `head` – zobrazí výsledky pro jednotlivé kroky
 - `plot(results)`
- Výsledky pro jednotlivé cykly – `counts` – počty pacientů, `values` - hodnoty
 - `head(get_counts(results))`
 - `head(get_values(results))`

Definování parametrů:

- `par` – proměnná kde jsou uloženy parametry
 - `par <- define_parameters(c_komp=5000, c_L1=17500, c_L2=33500, qol_komp=0.5, qol_L1=0.7, qol_L2=0.75)`

Definování a spuštění deterministické analýzy citlivosti:

- `det` – proměnná s nastavením analýzy



- *det <- define_dsa(c_L1, 15750, 19250, c_L2, 30150, 36850, c_komp, 4500, 5500, qol_L1, 0.63, 0.77, qol_L2, 0.675, 0.825, qol_komp, 0.45, 0.55)*
- vysledky_det – proměnná kam ukládám výsledky
 - *vysledky_det <- run_dsa(model=model, dsa=det)*
- Grafické zobrazení
 - *plot(vysledky_det, strategy="Lecba1", result="cost", typ="simple")*
 - *plot(vysledky_det, strategy="Lecba2", result="cost", typ="simple")*
 - *plot(vysledky_det, strategy="Lecba2", result="cost", typ="difference")*
 - *plot(vysledky_det, strategy="Lecba2", result="icer", typ="difference")*

Definování a spuštění deterministické analýzy citlivosti:

- prb – proměnná s nastavením analýzy
 - *prb <- define_psa(c_komp ~ normal(5000, 500), c_L1 ~ normal(17500, 3000), c_L2 ~ normal(33500, 5000), qol_komp ~ normal(0.5, 0.1), qol_L1 ~ normal(0.7, 0.1), qol_L2 ~ normal(0.75, 0.1))*
- vysledky_prb – proměnná kam ukládám výsledky
 - *vysledky_prb <- run_psa(model=model, psa=prb, N=100)*
- Grafické zobrazení
 - *plot(vysledky_prb, typ="ce")*
 - *plot(vysledky_prb, typ="ac", max_wtp=65000, log_scale=FALSE)*