

Assignment 1: Decision trees and cost-effectiveness

YOUR NAME HERE

2023-08-10

Table of contents

Section 1: Decision trees	2
1a Expected value calculations	2
1b Peptic ulcer closure decision model	5
Section 2: Economic evaluation	9
2a Discounting	9
2b Incremental analysis	10
2c Identifying maximum net monetary benefit	16
Last two questions	17

```
# Load packages
#. Use install.packages("XXXX") if you don't have any of these installed
library(rdecision) #decision trees
library(flextable) #Formatting tables to display (https://davidgohel.github.io/flextable/r
library(ggplot2) #Plotting
library(readxl) #for read_excel()
library(dplyr) # I use mutate at one point
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

```
intersect, setdiff, setequal, union
```

```
theme_set(theme_bw()) #Makes ggplots look better
```

Section 1: Decision trees

We'll use the [rdecision package](#) to develop and visualize decision trees. There are a few vignettes on the CRAN page. [This introductory one](#) will probably be sufficient for you to understand how we are using the package in this assignment.

1a Expected value calculations

This first code chunk is just an example; you don't need to edit it. Here, using the [rdecision package](#), I have created a decision tree, visualized it, and "rolled it back" to calculate the expected cost and QALYs associated with the two alternatives. Review it carefully, because in 1b, you'll do similar calculations for a different decision tree.

```
# Decision problem: Should we use diet or exercise to reduce chance of needing a stent in

# Parameters
c_diet <- 50 #cost of diet
c_exercise <- 750 #cost of exercise
c_stent <- 5000 #cost of a stent
u_stent <- 0.75 #utility of getting a stent (relative to 1.0)
p_stent_diet <- (68 - 12)/68 #probability needing stent if we diet
p_stent_exercise <- (58 - 18)/58 #probability of needing a stent if we exercise

#Build model using rdecision package

#Create decision and chance nodes
decision_node <- DecisionNode$new("Diet or exercise")
chance_node_diet <- ChanceNode$new("Stent?")
chance_node_exercise <- ChanceNode$new("Stent?")

#Create leaf nodes
leaf_node_diet_no_stent <- LeafNode$new("No stent")
leaf_node_diet_stent <- LeafNode$new("Stent", utility = u_stent)
```

```

leaf_node_exercise_no_stent <- LeafNode$new("No stent")
leaf_node_exercise_stent <- LeafNode$new("Stent", utility = u_stent)

#Create 'actions', paths from your decision node(s)
action_diet <- Action$new(
  decision_node, chance_node_diet, cost = c_diet, label = "Diet"
)
action_exercise <- Action$new(
  decision_node, chance_node_exercise, cost = c_exercise, label = "Exercise"
)

#Create 'reactions', paths from your chance node(s)
reaction_diet_success <- Reaction$new(
  chance_node_diet, leaf_node_diet_no_stent,
  p = 1-p_stent_diet, cost = 0.0, label = "Did not need stent")

reaction_diet_failure <- Reaction$new(
  chance_node_diet, leaf_node_diet_stent,
  p = p_stent_diet, cost = c_stent, label = "Needed stent")

reaction_exercise_success <- Reaction$new(
  chance_node_exercise, leaf_node_exercise_no_stent,
  p = 1-p_stent_exercise, cost = 0.0, label = "Did not need stent")

reaction_exercise_failure <- Reaction$new(
  chance_node_exercise, leaf_node_exercise_stent,
  p = p_stent_exercise, cost = 5000.0, label = "Needed stent")

#Create, draw, and evaluate the tree
DT1 <- DecisionTree$new(
  V = list(decision_node, #vertices (nodes)
    chance_node_diet,
    chance_node_exercise,
    leaf_node_diet_no_stent,
    leaf_node_diet_stent,
    leaf_node_exercise_no_stent,
    leaf_node_exercise_stent),
  E = list(action_diet, #edges (paths between nodes)
    action_exercise,
    reaction_diet_success,
    reaction_diet_failure,

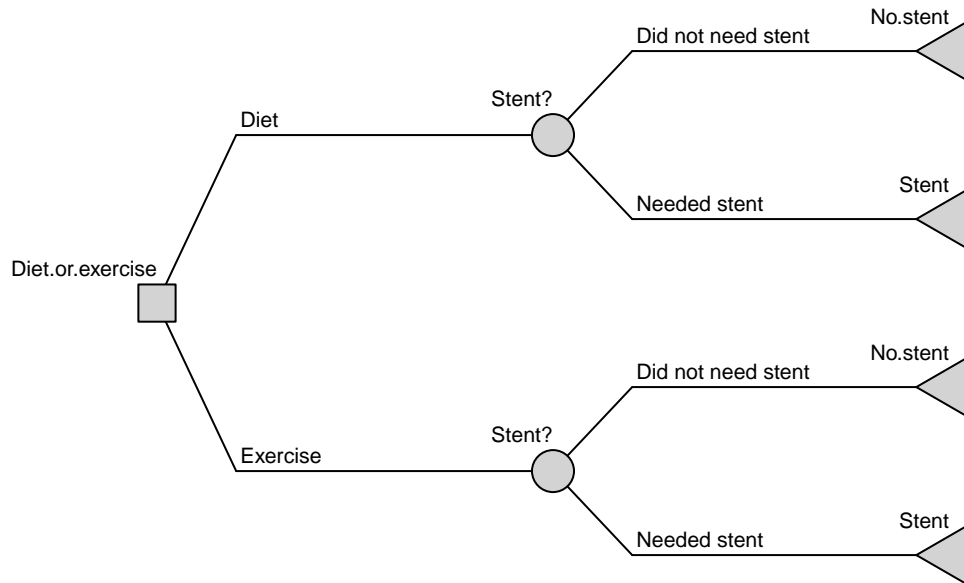
```

```

        reaction_exercise_success,
        reaction_exercise_failure)
    )

DT1$draw() #Plot it

```



```

DT1_evaluation <- DT1$evaluate() #calculate it
DT1_evaluation |> flextable()

```

Diet.or.exercise	Run	Probability	Cost	Benefit	Utility	QALY
Diet	1	1	4,167.647	0	0.7941176	0.7941176
Exercise	1	1	4,198.276	0	0.8275862	0.8275862

In the code chunk below, using the variables defined in the last code chunk (e.g., `p_stent_diet`), calculate the expected utility and the expected cost of the diet arm without using the `rdecision` package (summing/multiplying).

```

expected_cost_diet <- NA
expected_utility_diet <- NA

```

```
#Print the values  
expected_cost_diet
```

```
[1] NA
```

```
expected_utility_diet
```

```
[1] NA
```

1b Peptic ulcer closure decision model

You will now develop a decision analysis to inform whether a newer clip should be used to close bleeding peptic ulcers in the gastrointestinal tract during an upper GI endoscopy, which is also called an EGD (stands for esophagogastroduodenoscopy). The newer clips are called over-the-scope clips, abbreviated OTSc. Randomized trial data show that rebleeding rates are lower following endoscopic closure with OTSc as compared to when standard therapy clips are used, but OTSc are significantly more expensive. You will build a decision tree to determine whether OTSc are ‘worth’ the added expense, either as a first line therapy (i.e. to treat all peptic ulcer bleeds initially) or only for rebleeds (i.e. only if a standard therapy clip failed, resulting in a ‘rebleed’).

We assume all peptic ulcer bleeds are fully resolved during a short hospitalization; we therefore use a time horizon of only 30 days. No discounting is needed due to the short time horizon. Because this analysis involves very small differences in utility, our parameters are in quality-adjusted life days (QALDs) instead of quality-adjusted life years (QALYs): 1 QALD = 1/365 QALY, and 1 QALY = 365 QALD.

All model parameters are provided for you in the “OTSc” sheet of the excel file `params_assign1.xlsx`. You can open it to take a look. For your assignment to render without error, this `.qmd` file and that `.xlsx` file must be in the same folder. The next chunk of code reads in the `.xlsx` file and generates a parameter table, using the `flextable` package for formatting (you don’t have to do anything in this code chunk).

NOTE: You might need to close the Excel file for the `read_excel` function to run properly.

```
#read table from Excel  
t_params <- read_excel('params_assign1.xlsx', sheet = "OTSc")  
#Display it nicely  
t_params |>  
  flextable() |> #turn into flextable object
```

```
merge_v(j=1) |> #Merge cells in first column with same value (group probabilities, costs)
theme_box() |> #Apply a theme for aesthetics
autofit() #automatically set column widths to reasonable values
```

category	name	description
Probability	p_ST_rebleed	Continued bleeding after an initial EGD using standard therapy clip
	p_OTSc_rebleed	Continued bleeding after an initial EGD using OTSc
	p_ST_rpt_IR	Interventional Radiology (IR) procedure needed after second line EGD using standard therapy clip
	p_OTSc_rpt_IR	Interventional Radiology (IR) procedure needed after second line EGD using OTSc
Cost	c_EGD_MD	EGD physician fee (CPT code)
	c_STclip	Cost of standard therapy clip
	c_OTScip	Cost of over-the-scope clip
	c_hosp_noCC	Hospitalization cost of single successful EGD, based on DRG 379: GI hemorrhage
	c_hosp_CC	Hospitalization cost of needing two EGDs, based on DRG 377, GI hemorrhage
	c_hosp_MCC	Hospitalization cost of two failed EGDs followed by an interventional radiology procedure
	c_IR_MD	Interventional radiology physician fee
QALY	q_single_EGD	QALYs experienced in one-year period if single EGD needed (loss of 4.75 quality-adjusted life years)
	q_double_EGD	QALYs experienced in one-year period if two EGDs needed (loss of 1.1 quality-adjusted life years)
	q_IR	QALYs experienced in one-year period if two EGDs and an IR procedure needed (loss of 0.1 quality-adjusted life years)

In the code chunk below, build the model using the `rdecision` package with the ‘base_case’ value for each parameter (we won’t be using the ‘lower_bound’ or ‘upper_bound’ columns for anything in this problem set).

Your model should start with a decision node with three options:

- **OTSc first:** use OTS clip for first line therapy; if a rebleed occurs use a standard clip for second line therapy
- **ST first:** use a standard clip for first line therapy; if a rebleed occurs use an OTS clip for second line therapy
- **ST only:** use a standard clip for first line therapy and, if a rebleed occurs, for second line therapy.

All three arms from this decision node (“actions” in the rdecision package) lead into a chance node corresponding to the outcomes of the first line EGD. From the initial chance node, there are two possibilities:

- **Success:** no re-bleeding occurs after the first line procedure (ends in a leaf node).
- **Second-line therapy:** A second EGD is needed to try and close the peptic ulcer bleed (leads to a chance node corresponding to the outcome of the second line EGD).

After second-line therapy, you have another chance node with two possibilities:

- **Success:** second line therapy worked (ends in a leaf node).
- **Further bleeding; IR procedure needed:** After 2 failed EGDs, patients undergo an interventional radiology procedure, which we assume is always successful (ends in a leaf node).

Costs: You will accumulate costs along the action and reaction paths:

- In the actions from this initial decision node, patients incur the cost for the physician fee to do the EGD (`c_EGD_MD`) and the cost of whichever clip was used (`c_STclip` or `c_OTScip`)
- If the first line EGD was successful, we add in the cost of a hospitalization with no major complications or comorbidities (`c_hosp_noCC`).
- If the first line EGD was unsuccessful and we need second-line therapy, we add in the costs of a second EGD: physician fee plus cost of whichever clip was used (`c_STclip` or `c_OTScip`)
- If the second line EGD was successful, we add in the cost of a hospitalization with minor complications and comorbidities (`c_hosp_CC`)
- If the second line EGD was unsuccessful and an IR procedure was needed, we add in the cost of the IR physician fee (`c_IR_MD`), and the cost of a hospitalization with major complications/comorbidities (`c_hosp_MCC`).

Quality-adjusted life years: We will use rdecision’s default time horizon of one year. You will need to assign one of the three parameters `q_single_EGD`, `q_double_EGD`, and `q_IR` as the utility at each leaf node, depending on which procedures the patient underwent in order to arrive at that leaf. Note that `q_IR` already factors in the QALY loss from the two failed EGDs that occurred before the interventional radiology procedure.

Tips:

- Draw the tree on paper before you start coding it.
- Focus on tree structure and probabilities first, plot it, and then add costs and utilities.

```

# I like to turn the parameters from my table into a named list
# Then I can refer to the parameter I want by, e.g., `params$c_OTs`
params <- as.list(t_params$base_case)
names(params) <- t_params$name

#Create root decision node
decision_node <- NA

# Create chance nodes for whether first-line therapy was successful in each scenario

#Create chance nodes for whether second-line therapy was successful in each scenario

#Create leaf nodes
# (9 total. "1st line success", "2nd line success", and "IR needed" for each scenario)

#Create 'actions', paths from your decision node(s).

#Create 'reactions', paths from your chance node(s)
# 12 total:
# - Reactions for success after first-line therapy
# - Reactions for failure (rebleed) after first-line therapy
# - Reactions for success after second-line therapy
# - Reactions for failure (IR needed) after second-line therapy

#First line success:

#First line failure:

#Second line success:

#Second line failure:

#Create the tree
DT2 <- NA

```



```
# Draw the tree
#DT2$draw()

# Create table with the expected cost and utility (in QALYs or QALD) of each alternative
#DT2_evaluation <- DT2$evaluate() #calculate it
#DT2_evaluation |> flextable()
```

The differences in QALYs experienced in a one-year period is hard to see. We can instead convert the QALYs experienced to lost quality-adjusted life days lost (QALDs lost). I wrote the code for you; you just need to uncomment it after completing the previous code block.

```
# DT2_evaluation <- DT2_evaluation |>
#   mutate(QALD_lost = (1 - QALY)*365)
# DT2_evaluation |> flextable()
```

Using a willingness to pay threshold of \$50,000 per QALY (\$137 per QALD), indicate whether each intervention is dominated with strong dominance, Dominated by extended dominant, not preferred, or preferred (exactly one intervention should be preferred). Indicate the ICER for any non-dominated interventions when applicable.

ST only:

ST first:

OTSc first:

In case you're curious, this is an adapted/simplified version of [a paper I published as a PhD student](#).

Section 2: Economic evaluation

2a Discounting

The vector `costs_by_year` represents costs by year, starting with year 0 (this year, which should not be discounted). Calculate the net present cost with an annual discount rate of 2% and of 5%.

```
costs_by_year <- c(100, 200, 500, 350, 700) #costs 0, 1, 2, 3, and 4 years into the future

npc_2percent <- NA #Calculate the net present cost with a discount rate of 2 percent
```

```
npc_5percent <- NA #Calculate the net present cost with a discount rate of 5 percent

npc_2percent
```

```
[1] NA
```

```
npc_5percent
```

```
[1] NA
```

2b Incremental analysis

In this section, we'll use loops to conduct incremental analysis, the process of identifying dominated strategies and calculating the incremental cost-effectiveness ratios (ICERs) between the remaining non-dominated strategies. Coding this yourself should give you a strong understanding of the concept. Let's get started!

First, I've written the code for you that reads in the data on 9 different strategies we wish to compare (from the `params_assign1.xlsx` file). For each strategy, you are provided with the estimated QALYs experienced and the estimated cost across three categories: costs paid by the insurer, out-of-pocket costs paid by the patient, and the cost of lost productivity.

```
#No edits needed to this code chunk

#read table from Excel
t_CE <- read_excel('params_assign1.xlsx', sheet = "CEdata")
#Display it nicely
t_CE |>
  flextable() |> #turn into flextable object
  theme_box() |> #Apply a theme for aesthetics
  autofit() #automatically set column widths to reasonable values
```

Strategy	QALYs	Cost_insurer	Cost_patient	Cost_productivity
G	1.21	110,000	1,080	1,318
A	2.03	120,000	10,800	2,212
B	3.04	747,000	224,100	3,070
I	3.43	504,000	5,940	3,430

Strategy	QALYs	Cost_insurer	Cost_patient	Cost_productivity
E	5.03	414,000	124,200	4,828
C	5.49	594,000	64,800	5,105
H	5.64	310,000	23,760	5,414
F	6.10	324,000	81,000	5,396
D	8.39	648,000	110,880	8,641

Insurer perspective: In this code chunk, you'll calculate the cost-effectiveness from the insurer's perspective (only including costs in the `Cost_insurer` column). First, we detect any strategies that are dominated by strong dominance, then, we check for costs that are dominated by extended dominance. Lastly, we calculate the ICERs between all non-dominated interventions. I've provided some of the code, but you'll need to complete it.

```
#Create table with columns "strategy", "QALYs", and "Cost", where cost is the Cost_insurer
t_ICERS <- data.frame(
  Strategy = t_CE$Strategy,
  QALYs = t_CE$QALYs,
  Cost = t_CE$Cost_insurer,
  Dominated = "No", #we will check and some to 'yes' if dominated in the for loop below
  ICER = 0.0
)

#Sort t_ICERS ascending by QALYs (from low to high)
t_ICERS <- t_ICERS #EDIT THIS

#We need to reset the row index names
row.names(t_ICERS) <- NULL

### REMOVE INTERVENTIONS DOMINATED BY STRONG DOMINANCE
# We have to loop through rows to identify dominated strategies. If we find any,
# we need to look through them again in case there are more, so we use a while loop.

continue = T #If this is still T at the end of the while loop, it will restart

while(continue){
  continue = F #if no dominated strategies found in for loop below, exit while loop

  #Get the row index for all rows that aren't dominated
  idx_nondom_rows <- as.numeric(row.names(t_ICERS[t_ICERS$Dominated=="No",]))
```

```

#Get number of non-dominated rows
n_nondom <- 2 #EDIT THIS

#LOOP over each row starting with the second non-dominated row
# compare it to previous non-dominated row to see if it is dominated
for (i in 2:n_nondom){
  row_this <- idx_nondom_rows[i] #index of the current row
  row_last <- idx_nondom_rows[i-1] #index of last undominated row

  if (F){ #Replace 'F' with the correct condition
    #Strategy in row t_ICERS[row_last, ] is strong dominated
    # set Dominated value to "Strong"
    # set ICER value to NA
    # set continue to T so that we do another loop

    #YOUR CODE HERE
  }
}

### REMOVE INTERVENTIONS DOMINATED BY EXTENDED DOMINANCE
# We have to loop through rows to identify extended dominated strategies. If we find any,
# we need to loop through again in case there are more. We will use a while loop.
continue = T

while(continue){
  continue = F #if no dominated strategies are found, exit loop

  #Get the row index for all rows that aren't dominated
  idx_nondom_rows <- as.numeric(rownames(t_ICERS[t_ICERS$Dominated=="No",]))

  #Get number of non-dominated rows
  n_nondom <- 2 #EDIT THIS

  #LOOP over each row starting with the second non-dominated row
  # compare it to previous non-dominated row to see if it is extended dominated
  for (i in 2:n_nondom){
    row_this <- idx_nondom_rows[i] #index of the current row
    row_last <- idx_nondom_rows[i-1] #index of last undominated row

    #Calculate ICER for row_this compared to last non-dominated row

```

```

# Put it in t_ICERS[row_this,"ICER"]

#YOUR CODE HERE

if(F){ #REPLACE F WITH THE CORRECT CONDITION
  #Strategy in row t_ICERS[row_last, ] is extended dominated
  # set Dominated value to "Extended"
  # set ICER value to NA
  # set continue to T so that we do another loop

  #YOUR CODE HERE

}
}
}

#Display it nicely
t_ICERS |>
  flextable() |> #turn into flextable object
  theme_box() |> #Apply a theme for aesthetics
  autofit() #automatically set column widths to reasonable values

```

Strategy	QALYs	Cost	Dominated	ICER
G	1.21	110,000	No	0
A	2.03	120,000	No	0
B	3.04	747,000	No	0
I	3.43	504,000	No	0
E	5.03	414,000	No	0
C	5.49	594,000	No	0
H	5.64	310,000	No	0
F	6.10	324,000	No	0
D	8.39	648,000	No	0

Function-ize it: The following code chunk contains the shell of a function that takes three vectors (strategy_name, cost, and QALY), puts them into a table called t_ICERS, does calculations, and then returns the table with ICER and dominated columns added. Complete the function by copying and pasting code from the previous code chunk

```

#INPUT: 3 equal-length vectors strategy_name, cost, and QALYs
#OUTPUT: A data.frame with a column for each of the input vectors,
#. plus new "Dominated" and "ICER" columns
generate_ICER_table <- function(strategy_name, cost, QALYs){
  t_ICERS <- data.frame(
    Strategy = strategy_name,
    QALYs = QALYs,
    Cost = cost,
    Dominated = "No",
    ICER = 0.0)

  #
  #YOUR CODE HERE
  #

  return(t_ICERS)
}

```

Other perspectives: using the function above, complete the incremental analysis from three cost perspectives:

- **Insurer:** same as above
- **Healthcare costs** includes insurer and patient out-of-pocket costs
- **Societal:** healthcare costs plus productivity loss

```

t_ICER_insurer <- t_CE #REPLACE WITH CORRECT CODE

t_ICER_healthcare_cost <- t_CE #REPLACE WITH CORRECT CODE

t_ICER_societal <- t_CE #REPLACE WITH CORRECT CODE

#Display them nicely

#INSURER PERSPECTIVE
t_ICER_insurer |>
  flextable() |> #turn into flextable object
  theme_box() |> #Apply a theme for aesthetics
  autofit() #automatically set column widths to reasonable values

```

Strategy	QALYs	Cost_insurer	Cost_patient	Cost_productivity
G	1.21	110,000	1,080	1,318

Strategy	QALYs	Cost_insurer	Cost_patient	Cost_productivity
A	2.03	120,000	10,800	2,212
B	3.04	747,000	224,100	3,070
I	3.43	504,000	5,940	3,430
E	5.03	414,000	124,200	4,828
C	5.49	594,000	64,800	5,105
H	5.64	310,000	23,760	5,414
F	6.10	324,000	81,000	5,396
D	8.39	648,000	110,880	8,641

```
#HEALTHCARE COST PERSPECTIVE
```

```
t_ICER_healthcare_cost |>
  flextable() |> #turn into flextable object
  theme_box() |> #Apply a theme for aesthetics
  autofit() #automatically set column widths to reasonable values
```

Strategy	QALYs	Cost_insurer	Cost_patient	Cost_productivity
G	1.21	110,000	1,080	1,318
A	2.03	120,000	10,800	2,212
B	3.04	747,000	224,100	3,070
I	3.43	504,000	5,940	3,430
E	5.03	414,000	124,200	4,828
C	5.49	594,000	64,800	5,105
H	5.64	310,000	23,760	5,414
F	6.10	324,000	81,000	5,396
D	8.39	648,000	110,880	8,641

```
#SOCIAL PERSPECTIVE
```

```
t_ICER_societal |>
  flextable() |> #turn into flextable object
  theme_box() |> #Apply a theme for aesthetics
```

```
autofit() #automatically set column widths to reasonable values
```

Strategy	QALYs	Cost_insurer	Cost_patient	Cost_productivity
G	1.21	110,000	1,080	1,318
A	2.03	120,000	10,800	2,212
B	3.04	747,000	224,100	3,070
I	3.43	504,000	5,940	3,430
E	5.03	414,000	124,200	4,828
C	5.49	594,000	64,800	5,105
H	5.64	310,000	23,760	5,414
F	6.10	324,000	81,000	5,396
D	8.39	648,000	110,880	8,641

2c Identifying maximum net monetary benefit

Calculate NMB at a range of WTP thresholds *from a societal perspective*, and then plot them.

```
#Vector of willingness-to-pay values for which to calculate NMB
wtp_vec = seq(0, 2.5e5, 1e5)

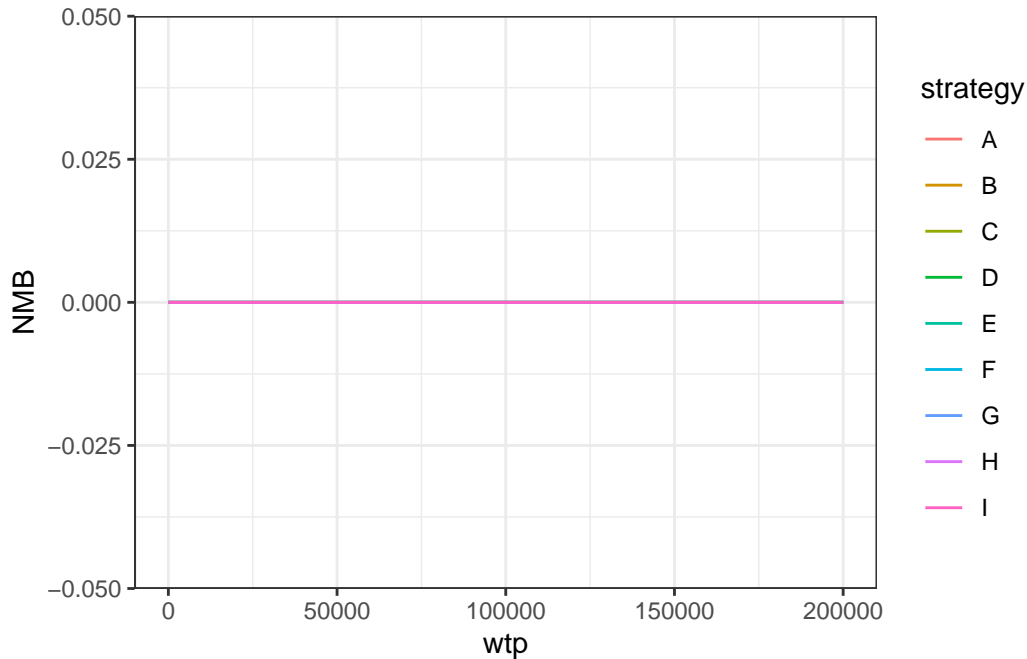
#Calculate societal cost and add it as a column to the t_CE table
t_CE$cost_societal <- 0 #REPLACE WITH CORRECT CODE

#Create a table to contain the NMB calculations
t_nmb_plot <- data.frame(
  strategy = rep(t_CE$Strategy, each = length(wtp_vec)),
  wtp = rep(wtp_vec, times = nrow(t_CE)),
  cost = rep(t_CE$cost_societal, each = length(wtp_vec)),
  QALYs = rep(t_CE$QALYs, each = length(wtp_vec))
)

#Calculate NMB
t_nmb_plot$NMB <- 0 #REPLACE WITH CORRECT CODE
```



```
#Plot it with ggplot
ggplot(t_nmb_plot, aes(x = wtp, y = NMB, color = strategy))+
  geom_line()
```



Comparing the net monetary benefit plot to your incremental analysis from the societal perspective, you should observe the following:

- All non-dominated interventions maximize net monetary benefit for some willingness-to-pay level (i.e., their line is the highest on the Y axis for some segment of the X axis).
- The willingness-to-pay level at which we ‘flip’ from preferring non-dominated strategy i to non-dominated strategy j can be derived from the NMB analysis or the incremental analysis:
 - In incremental analysis, this corresponds to the ICER for intervention j vs. i
 - In the NMB plot, this corresponds to the WTP level (x-axis value) at which the line for strategy j crosses from below to above the line for strategy i.

Last two questions

- About how much time did you spend on the assignment? **Replace with your answer**

- Did you find any errors or have suggestions to improve it? **Replace with your answer**

Fin.