

Decision Analysis for Agriculture Development

Introducing a SSB Tax in Germany

Nutrition group

Karin Altvater, Santosh Bhandari, Patrick Frey, Sanghyo Moon

Sugar sweetened beverages (SSBs)

- soft drinks, energy drinks, fruit nectar drinks etc.
- one of the main sources of added sugar intake
- many associated health issues
 - decrease life quality of individual people
 - burden on the public health system

SSB tax

- to internalize the external costs of high sugar consumption
- higher prices of SSBs
 - lower consumption: proven for several countries
- tax returns can be reinvested in health sector

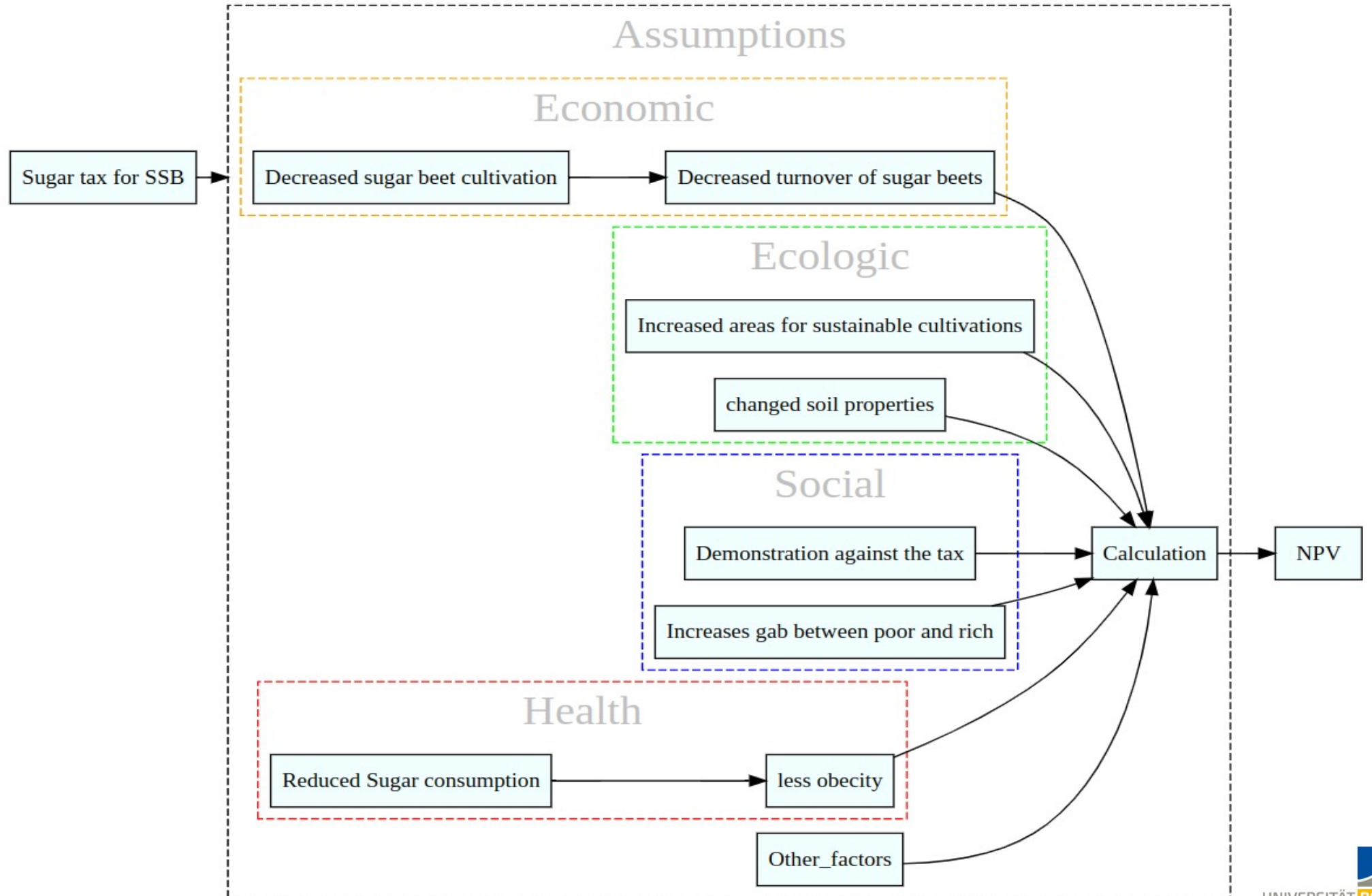
Countries with SSB -Tax



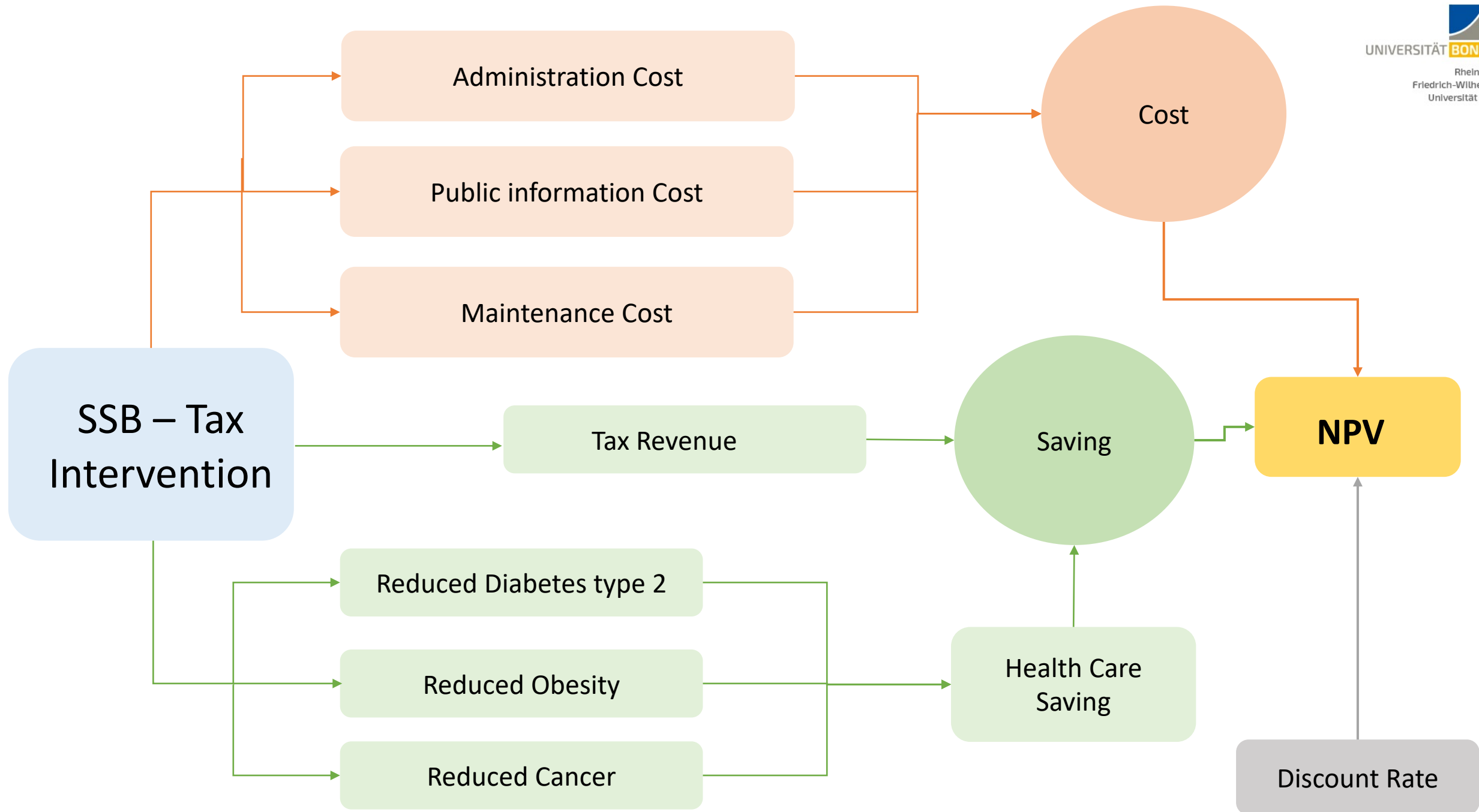
Decision: Introducing a tax on SSBs in Germany

Decision maker: German government

General Modell



For the decision analysis, we are focusing on health care costs



Input table

description	label	variable	distribution	lower	median	upper	unit
Establishment costs	Government implementation [€]	implementation_costs_admin	posnorm	1371265500.00		1782645150.00	Euro
	Costs for accompanying campaigns/public information [€]	implementation_costs_info	posnorm	1000000.00		5000000.00	Euro
Maintenance cost	Maintenance cost [€]	maintenance_intervention_SSB_Tax	posnorm	27000000		36000000	Euro
Reduced incidences of	Reduced diabetes type 2 [Case]	red_diabetes_case	posnorm	155400.00		218100.00	Case
	Reduced obesity [Case]	red_obesity_case	posnorm	1604000.00		2857700.00	Case
	Reduced cancer [Case]	red_cancer_case	posnorm	16040.00		28577.00	Case
Total case	Total diabetes type 2 [Case]	total_diabetes_case	posnorm	6100000		7000000	Case
	Total obesity [Case]	total_obesity_case	posnorm	15000000		17000000	Case
	Total cancer [Case]	total_cancer_case	posnorm	500000		550000	Case
health care costs per case	Diabetes type 2 [€]	cost_diabetes_euro	posnorm	4524.30		4917.20	Euro
	Obesity [€]	cost_obesity_euro	posnorm	95.19		95.34	Euro
	Cancer [€]	cost_cancer_euro	posnorm	95189.27		95340.08	Euro
Proportion of cases per year due to sugar	Proportion Cases	proportion_cases_from_sugar	tnorm_0_1	0.01		0.1	Percent
Tax income for the government	Tax revenue [€]	revenue_tax_euro	const	19200000.00		19200000	Euro
Discount rate (%)	Discount rate (%)	discount_rate	posnorm	1.00		5.00	%
Years from the start o the implementation of the intervention to the end.	Project time horizon (years)	n_years	const	10		10	
Coefficient of variation, ratio of the standard deviation to the mean (a measure of relative variability).	coeff. Variation	var_CV	posnorm	5		20	

Estimation for our decision

- Total health care cost without SSB-Tax
 - Total cost diabetes = (total diabetes case * proportion) * cost for diabetes
 - Total cost obesity = (total obesity case * proportion) * cost for obesity
 - Total cost cancer = (total cancer case * proportion) * cost for cancer
- total health care cost with SSB-Tax
 - Total cost diabetes - (saved diabetes case * proportion) * cost for diabetes
 - Total cost obesity - (saved obesity case * proportion) * cost for obesity
 - Total cost cancer – (saved cancer case * proportion) * cost for cancer
- Implementation cost
 - Implementation cost for SSB Tax
 - Costs for accompanying campaigns/public information
- Tax revenue

R code for function

```
###Intervention of implementation SSB Tax ###
```

```
for (decision_implementation_SSB_Tax in c(FALSE,TRUE)){
```

```
  if (decision_implementation_SSB_Tax){
```

```
    implementation_SSB_Tax <- TRUE
```

```
    implementation_admin <- TRUE
```

```
    no_implementation <- FALSE
```

```
  } else
```

```
{
```

```
  implementation_SSB_Tax <- FALSE
```

```
  implementation_admin <- FALSE
```

```
  no_implementation <- TRUE
```

```
}
```

```
##cost for intervention##
```

```
if (implementation_admin){
```

```
  cost_implementation_govern <-
```

```
    implementation_costs_admin + implementation_costs_info
```

```
} else{
```

```
  cost_implementation_govern <- 0
```

```
}
```

```
# calculating the maintenance costs,
```

```
  #initializing the array with 0 costs for the first year:
```

```
maintenance_cost <- rep(0, n_years)
```

```
if (implementation_SSB_Tax){
```

```
  maintenance_cost <-
```

```
    maintenance_cost + vv(maintenance_intervention_SSB_Tax,  
                          var_CV, n_years)
```

```
intervention_cost <- maintenance_cost
```

```
#intervention cost for first year
```

```
intervention_cost[1] <-
```

```
  cost_implementation_govern
```

```
}
```

R code for function

```
# health care cost with intervention of SSB Tax ####

implementation_SSB_Tax_Di <-
  as.numeric(implementation_SSB_Tax) * precalc_HC_with_interv_Tax_DI
implementation_SSB_Tax_OB <-
  as.numeric(implementation_SSB_Tax) * precalc_HC_with_interv_Tax_OB
implementation_SSB_Tax_Ca <-
  as.numeric(implementation_SSB_Tax) * precalc_HC_with_interv_Tax_CA

total_healthcare_cost_with_SSB_Tax <-
  implementation_SSB_Tax_Di + implementation_SSB_Tax_OB + implementation_SSB_Tax_Ca

implementation_SSB_Tax_tax_revenue <-
  as.numeric(implementation_SSB_Tax) * precalc_tax_revenue

#Health care cost without SSB Tax

cost_diabetes_no_SSB_Tax <- as.numeric(no_implementation)*cost_diabetes
cost_obesities_no_SSB_Tax <- as.numeric(no_implementation)* cost_obesities
cost_cancer_no_SSB_Tax <- as.numeric(no_implementation)*cost_cancer

total_health_care_cost_no_SSB_Tax <-
  cost_diabetes_no_SSB_Tax + cost_obesities_no_SSB_Tax + cost_cancer_no_SSB_Tax
```

R code for function

```
if(implementation_SSB_Tax){
  imple_tax_revenue <- implementation_SSB_Tax_tax_revenue
} else imple_tax_revenue <- 0

## Total cost with SSB tax##
if (decision_implementation_SSB_Tax){

  net_health_care_cost_with_imple <-
    total_healthcare_cost_with_SSB_Tax + intervention_cost - implementation_SSB_Tax_tax_revenue

  result_imple <- net_health_care_cost_with_imple
}

if (!decision_implementation_SSB_Tax){
  result_no_imple <- total_health_care_cost_no_SSB_Tax
}

}#close implementation

#discount rate

NPV_imple <-
  discount(result_imple, discount_rate, calculate_NPV = T)

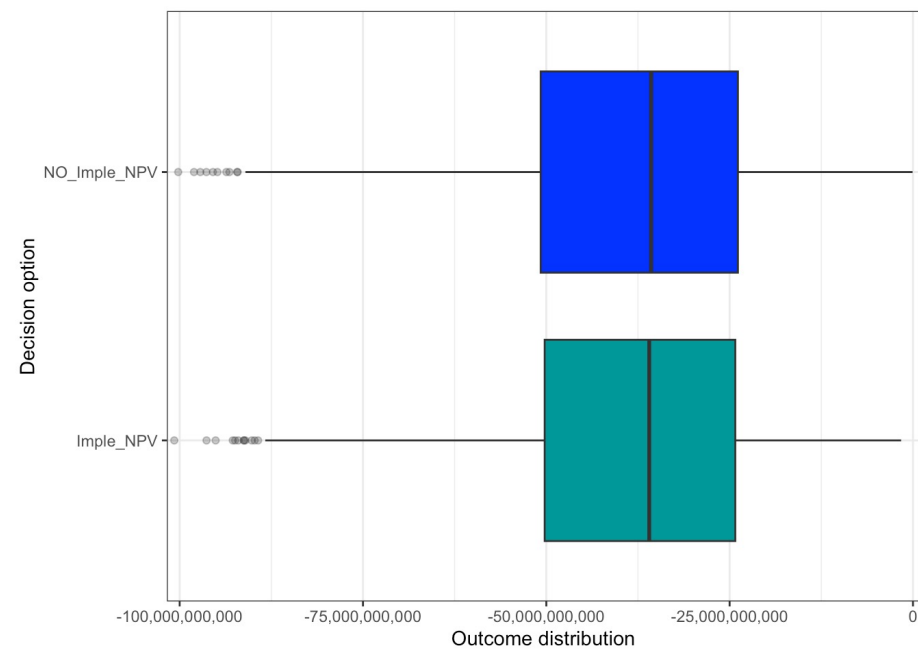
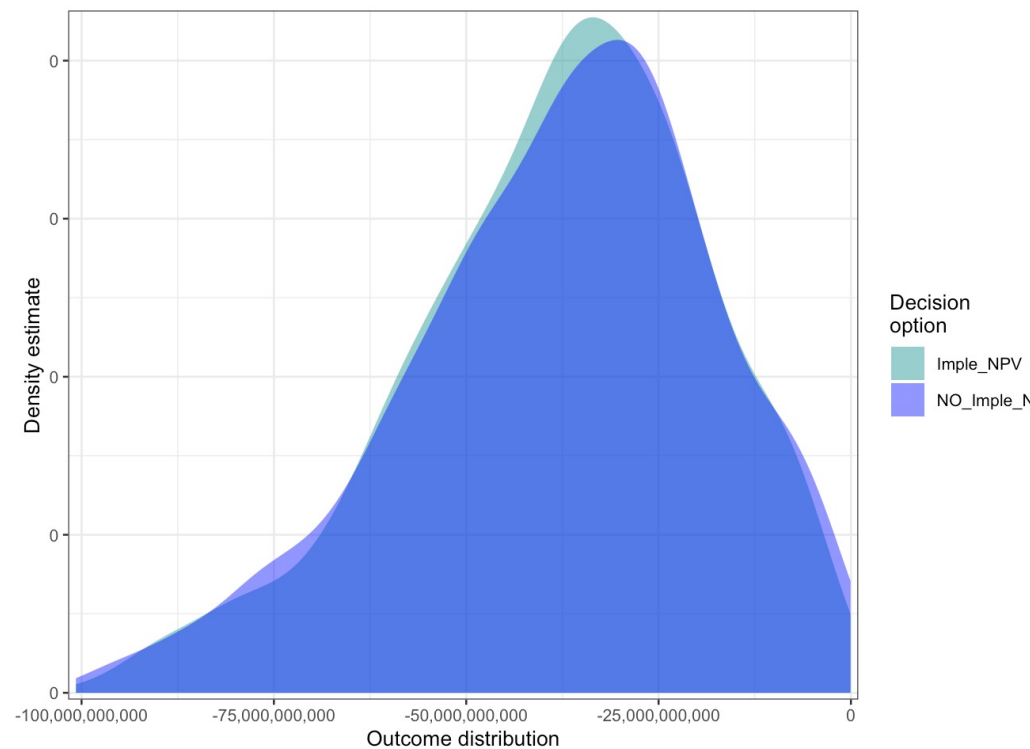
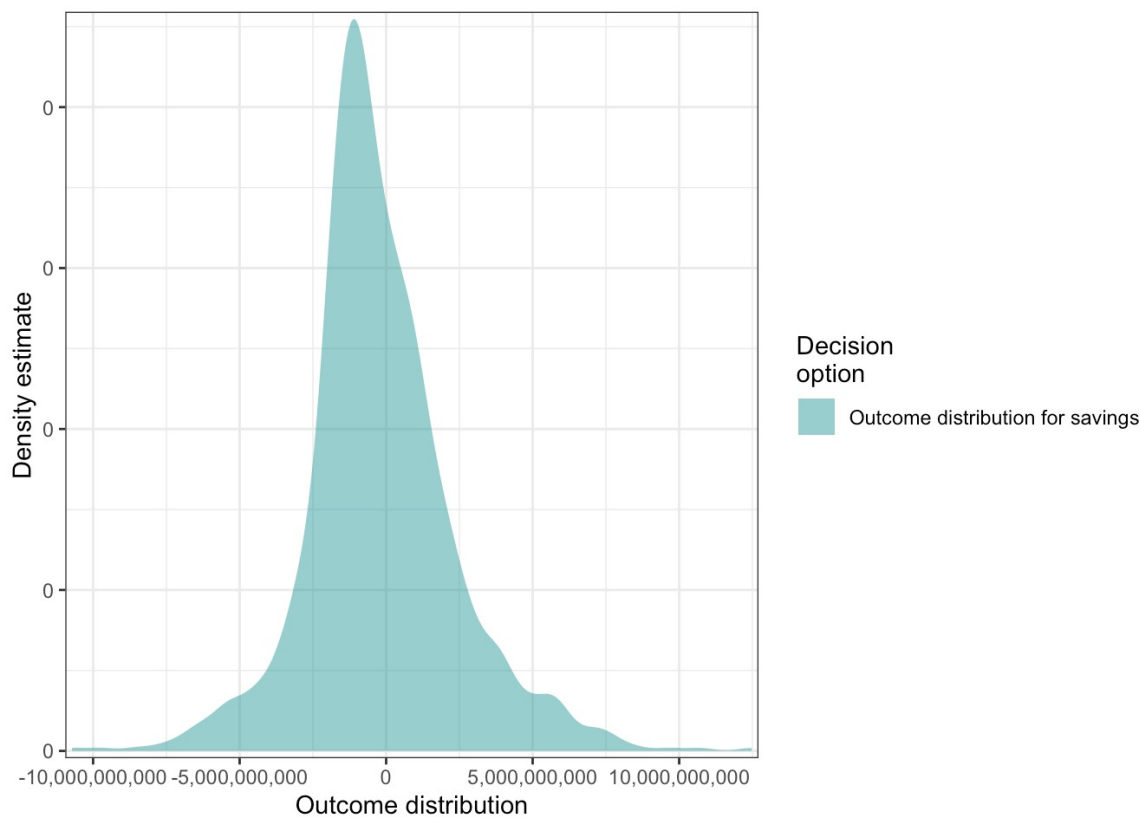
NPV_no_imple <-
  discount(result_no_imple, discount_rate, calculate_NPV = T)

return(list(Imple_NPV = - NPV_imple, #invert because this is the spending
            NO_Imple_NPV = - NPV_no_imple, #invert because this is the spending
            NPV_decision_SSB_Tax = NPV_no_imple - NPV_imple, #how much they will save
            Cashflow_decision_SSB_Tax = result_no_imple - result_imple)) #how much they will save
}
```

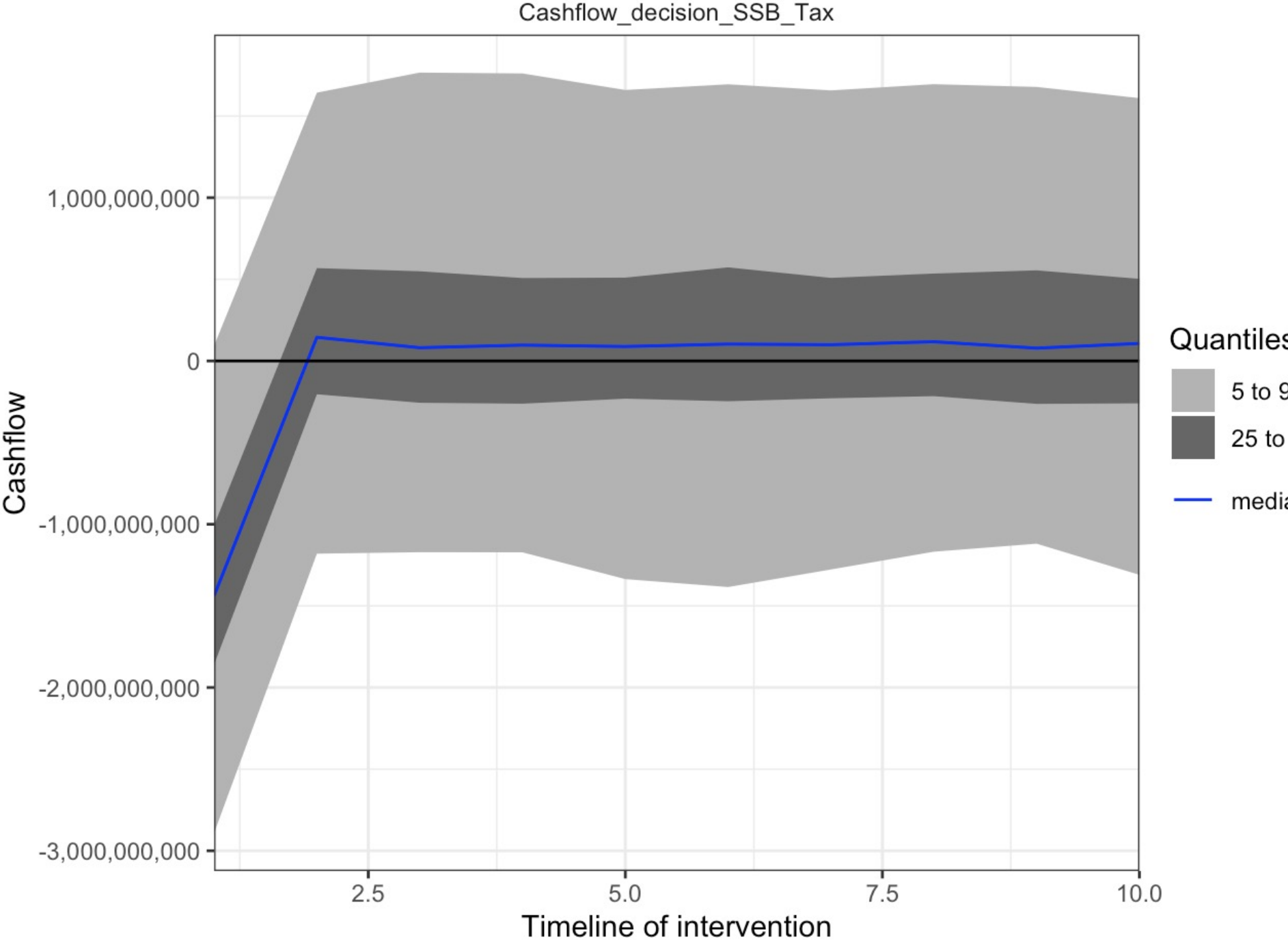
Simulation

```
mcSimulation_results1 <- mcSimulation(estimate =  
    estimate_read_csv("Presentation/input_table_updated_3.csv"),  
    model_function = decision_function,  
    numberOfModelRuns = 1000,  
    functionSyntax = "plainNames"  
)
```

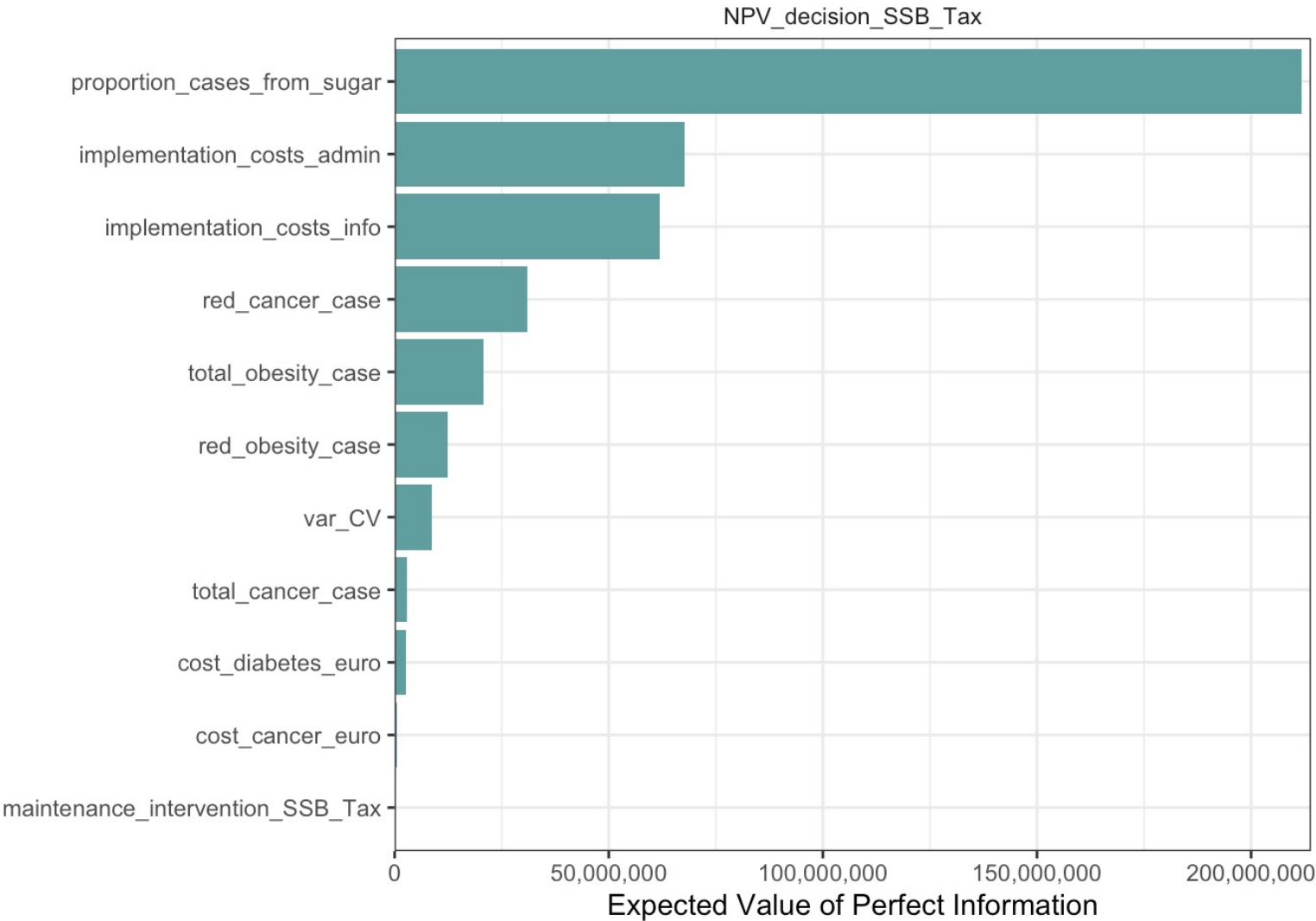
Net present Value (NPV) distributions



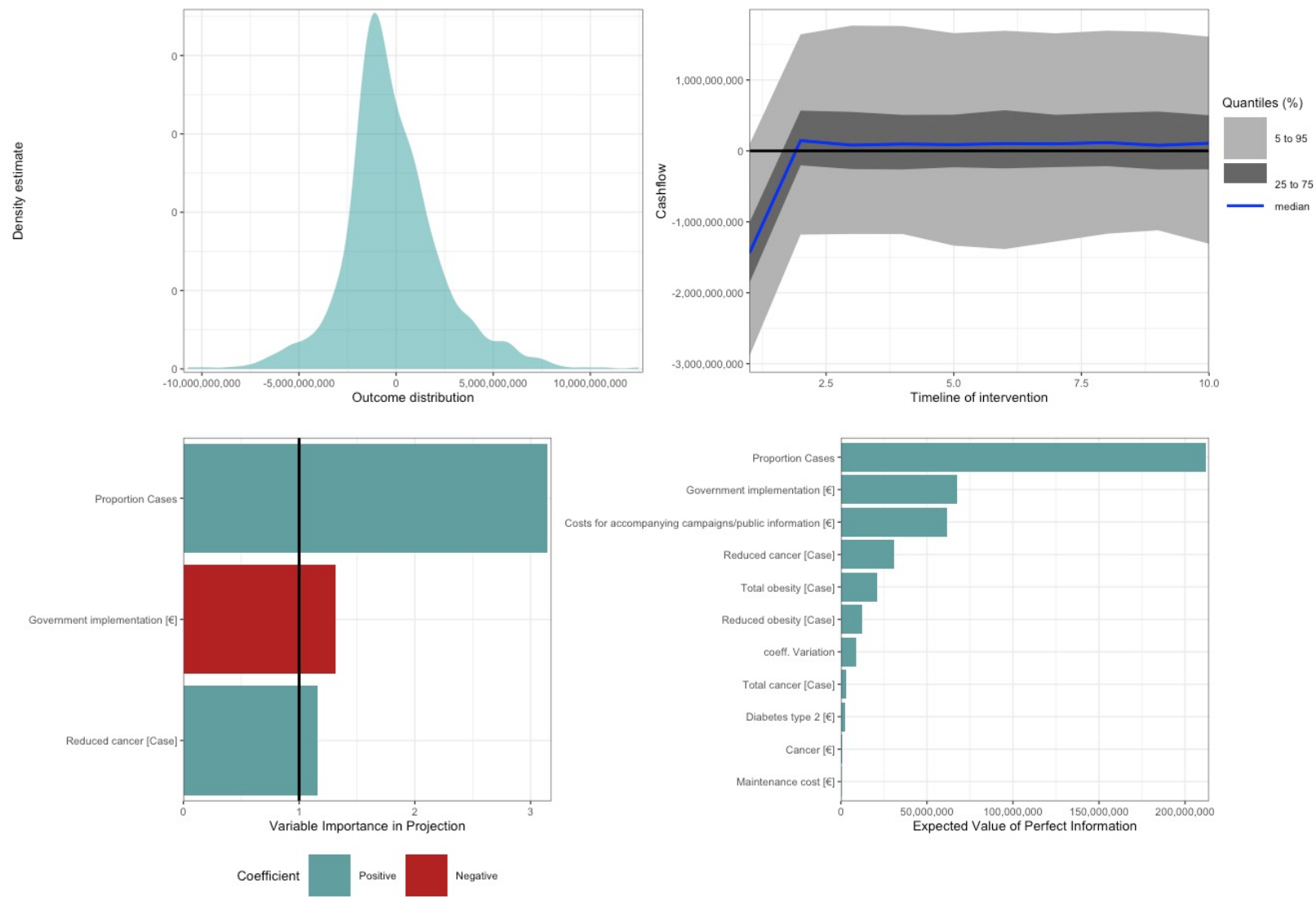
Cashflow analysis



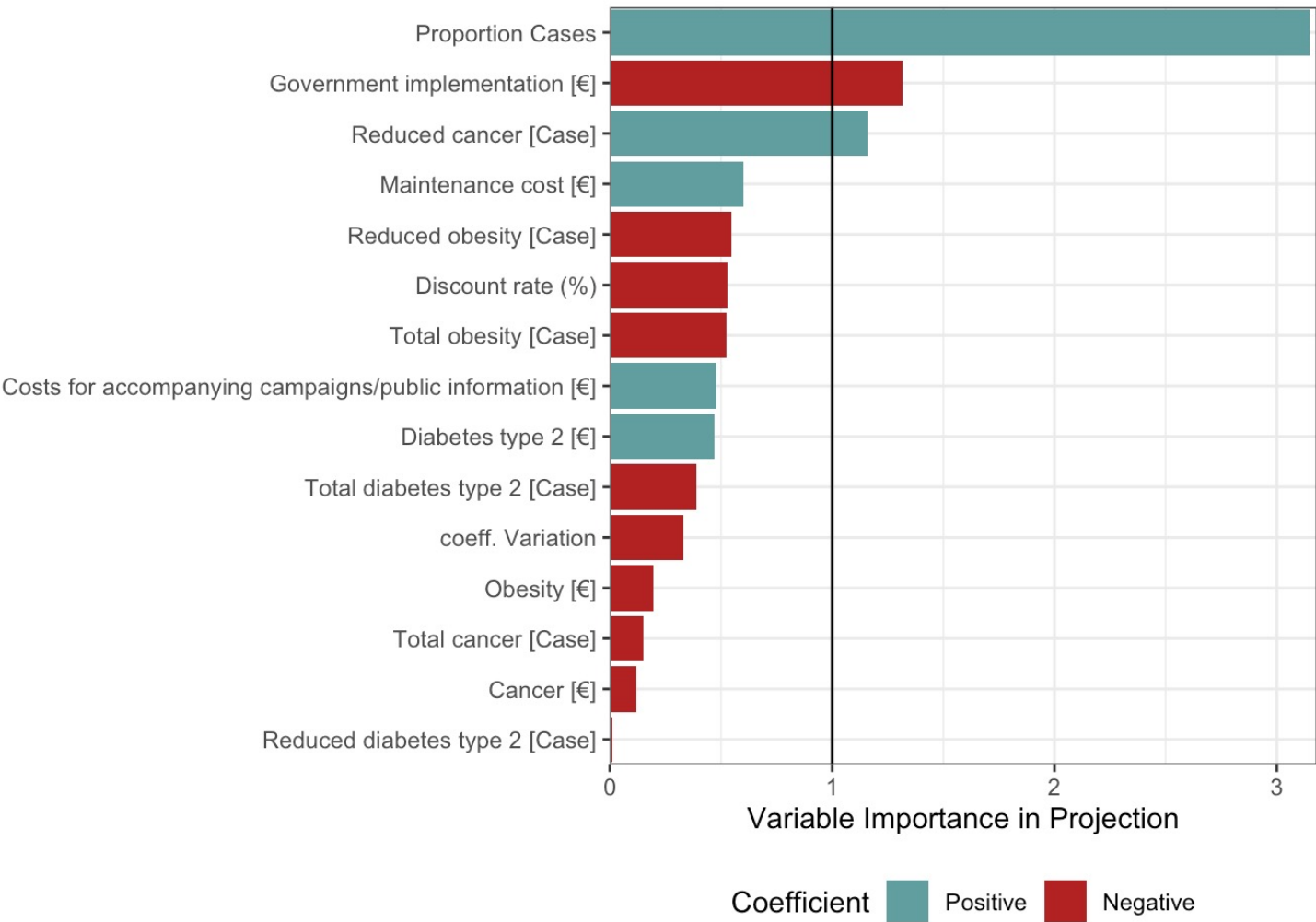
Value of Information
(Vol) analysis



Results with plots



Projection to Latent
Structures (PLS)
analysis



Results with plots