```
##############################################
# Third Assignment WS22/23
#
# Variable energy supply system in the remote region
#
# Project Members:
# Carl Rupprich
# Erfan Akbarpour
# Johannes Häufler
# Lukas Heidenreich
# Simon Maramangalam
#
##############################################

rm (list = ls())
library(timelineS)
library(lubridate)
library(ggplot2)
library(dplyr)
library(reshape2)
library(rPref)
library(mco)
library(MASS)
par(mar=c(1,1,1,1))

#1st function
plot.timeline <- function(lifetime, events.name, start.Date, plot.Name) {
  dataP <- data.frame(Events = events.name, Event_Dates = ymd(start.Date) +
                        years(lifetime))
  timelineS(dataP, main = plot.Name,
            labels = events.name, label.direction = "up", label.position =
3)
}

#2nd function

dist.Events <- function (lifetime, events, start.Date, option.Name) {
  #sort the events
  events <- sort(events, decreasing = T)
  #create the distribution of the events
  distribution.events <- sapply(events, seq, from = 0, to = lifetime)
  #all events unlisted
  all.events <- melt(distribution.events)
  colnames(all.events) <- c("frequency", "event")
  #sort the events ascending
  all.events <- all.events[order(all.events$frequency),]
  #get the unique sequence of events
  unique.events <- all.events[!duplicated(all.events$frequency),]
  unique.events$event[which(unique.events$frequency == 0)] <- "DC"
  #plot the timeline
  #plot.timeline(unique.events$frequency, unique.events$event, start.Date,
option.Name)
  return(unique.events)
  }


combine.lifeTimelines1 <- function(product.1, p1.dur ) {
  #function body
  base <- list(Names = c(), frequency = c(), duration = c())
  base$frequency <- sort(unique(c(product.1$frequency
```

```
                                  )),
                           decreasing = FALSE)
  base$Names[1] <- "DC"
  base$duration[1] <- 0
  base$Names[length(base$frequency)] <- "END"
  base$duration[length(base$frequency)] <- 0
  for(index in 2:(length(base$frequency) - 1)) {
    phase.1 <- product.1$event[which(product.1$frequency ==
                                       base$frequency[index])]
    if(length(phase.1) != 0 ) {
      base$Names[index] <- paste(phase.1)
      base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.1)])
    } else {
      base$duration[index] <- 0
    }
  }
  return(base)
  }

#3rd function
combine.lifeTimelines <- function(product.1, product.2, product.3, product.
4, product.5, p1.dur, p2.dur, p3.dur, p4.dur, p5.dur) {

  base <- list(Names = c(), frequency = c(), duration = c())
  base$frequency <- sort(unique(c(product.1$frequency,
                                  product.2$frequency,
                                  product.3$frequency,
                                  product.4$frequency,
                                  product.5$frequency)),
                           decreasing = FALSE)

  base$Names[1] <- "DC"
  base$duration[1] <- 0
  base$Names[length(base$frequency)] <- "END"
  base$duration[length(base$frequency)] <- 0

  for(index in 2:(length(base$frequency) - 1)) {
    phase.1 <- product.1$event[which(product.1$frequency ==
                                       base$frequency[index])]
    phase.2 <- product.2$event[which(product.2$frequency ==
                                       base$frequency[index])]
    phase.3 <- product.3$event[which(product.3$frequency ==
                                       base$frequency[index])]
    phase.4 <- product.4$event[which(product.4$frequency ==
                                       base$frequency[index])]
    phase.5 <- product.5$event[which(product.5$frequency ==
                                       base$frequency[index])]


      #all not 0
     if(length(phase.1) != 0 && length(phase.2) != 0 && length(phase.3) !=
0 && length(phase.4) != 0 && length(phase.5) != 0) {
      base$Names[index] <- paste(phase.1, phase.2, phase.3,phase.4, phase.
5)
      base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.1)],
                                  p2.dur[which(names(p2.dur) == phase.2)],
                                  p3.dur[which(names(p3.dur) == phase.3)],
                                  p4.dur[which(names(p4.dur) == phase.4)],
                                  p5.dur[which(names(p5.dur) == phase.5)])
     }
```

```
      # only 1 = 0
      else if(length(phase.1) == 0 && length(phase.2) != 0 && length(phase.
3) != 0 && length(phase.4) != 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste(phase.2, phase.3, phase.4, phase.5)
        base$duration[index] <- max(p2.dur[which(names(p2.dur) == phase.
2)],
                                    p3.dur[which(names(p3.dur) == phase.
3)],
                                    p4.dur[which(names(p4.dur) == phase.
4)],
                                    p5.dur[which(names(p5.dur) == phase.
5)])
      }


      # only 2 = 0
      else if(length(phase.1) !=0 && length(phase.2) == 0 && length(phase.
3) != 0 && length(phase.4) != 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste(phase.1, phase.3, phase.4, phase.5)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.
1)],
                                    p3.dur[which(names(p3.dur) == phase.
3)],
                                    p4.dur[which(names(p4.dur) == phase.
4)],
                                    p5.dur[which(names(p5.dur) == phase.
5)])
      }

      # only 3 = 0
      else if(length(phase.1) != 0 && length(phase.2) != 0 && length(phase.
3) == 0 && length(phase.4) != 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste(phase.2, phase.1, phase.4, phase.5)
        base$duration[index] <- max(p2.dur[which(names(p2.dur) == phase.
2)],
                                    p1.dur[which(names(p1.dur) == phase.
1)],
                                    p4.dur[which(names(p4.dur) == phase.
4)],
                                    p5.dur[which(names(p5.dur) == phase.
5)])
      }

      # only 4 = 0
      else if(length(phase.1) != 0 && length(phase.2) != 0 && length(phase.
3) != 0 && length(phase.4) == 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste(phase.2, phase.3, phase.1, phase.5)
        base$duration[index] <- max(p2.dur[which(names(p2.dur) == phase.
2)],
                                    p3.dur[which(names(p3.dur) == phase.
3)],
                                    p1.dur[which(names(p1.dur) == phase.
1)],
                                    p5.dur[which(names(p4.dur) == phase.
5)])
      }
```

```
      # only 5 = 0
      else if(length(phase.1) != 0 && length(phase.2) != 0 && length(phase.
3) != 0 && length(phase.4) != 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste(phase.2, phase.3, phase.4, phase.1)
        base$duration[index] <- max(p2.dur[which(names(p2.dur) == phase.
2)],
                                    p3.dur[which(names(p3.dur) == phase.
3)],
                                    p4.dur[which(names(p4.dur) == phase.
4)],
                                    p1.dur[which(names(p1.dur) == phase.
1)])
      }

      #2 numbers
      #1,2 =0
      else if(length(phase.1) == 0 && length(phase.2) == 0 && length(phase.
3) != 0 && length(phase.4) != 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.3, phase.4, phase.5)
        base$duration[index] <- max(p3.dur[which(names(p3.dur) == phase.
3)],
                                    p4.dur[which(names(p4.dur) == phase.
4)],
                                    p5.dur[which(names(p5.dur) == phase.
5)])
      }


      #1,3 =0
      else if(length(phase.1) == 0 && length(phase.2) != 0 && length(phase.
3) == 0 && length(phase.4) != 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.2, phase.4, phase.5)
        base$duration[index] <- max(p2.dur[which(names(p2.dur) == phase.
2)],
                                    p4.dur[which(names(p4.dur) == phase.
4)],
                                    p5.dur[which(names(p5.dur) == phase.
5)])
      }

      #1,4 =0
      else if(length(phase.1) == 0 && length(phase.2) != 0 && length(phase.
3) != 0 && length(phase.4) == 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.3, phase.2, phase.5)
        base$duration[index] <- max(p3.dur[which(names(p3.dur) == phase.
3)],
                                    p2.dur[which(names(p2.dur) == phase.
2)],
                                    p5.dur[which(names(p5.dur) == phase.
5)])
      }

      #1,5 =0
      else if(length(phase.1) == 0 && length(phase.2) != 0 && length(phase.
3) != 0 && length(phase.4) != 0 && length(phase.5) == 0)
```

```
      {
        base$Names[index] <- paste( phase.3, phase.4, phase.2)
        base$duration[index] <- max(p3.dur[which(names(p3.dur) == phase.
3)],
                                    p4.dur[which(names(p4.dur) == phase.
4)],
                                    p2.dur[which(names(p2.dur) == phase.
2)])
      }

      #2,3 =0
      else if(length(phase.1) != 0 && length(phase.2) == 0 && length(phase.
3) == 0 && length(phase.4) != 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.1, phase.4, phase.5)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.
1)],
                                    p4.dur[which(names(p4.dur) == phase.
4)],
                                    p5.dur[which(names(p5.dur) == phase.
5)])
      }

      #2,4 =0
      else if(length(phase.1) != 0 && length(phase.2) == 0 && length(phase.
3) != 0 && length(phase.4) == 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.1, phase.3, phase.5)
        base$duration[index] <- max(p3.dur[which(names(p3.dur) == phase.
3)],
                                    p1.dur[which(names(p1.dur) == phase.
1)],
                                    p5.dur[which(names(p5.dur) == phase.
5)])
      }

      #2,5 =0
      else if(length(phase.1) != 0 && length(phase.2) == 0 && length(phase.
3) != 0 && length(phase.4) != 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.1, phase.3, phase.4)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.
1)],
                                    p4.dur[which(names(p4.dur) == phase.
4)],
                                    p3.dur[which(names(p3.dur) == phase.
3)])
      }

      #3,4 =0
      else if(length(phase.1) != 0 && length(phase.2) != 0 && length(phase.
3) == 0 && length(phase.4) == 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.1, phase.2, phase.5)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.
1)],
                                    p2.dur[which(names(p2.dur) == phase.
2)],
                                    p5.dur[which(names(p5.dur) == phase.
5)])
```

```
      }

      #3,5 =0
      else if(length(phase.1) != 0 && length(phase.2) != 0 && length(phase.
3) == 0 && length(phase.4) != 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.1, phase.2, phase.4)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.
1)],
                                    p2.dur[which(names(p2.dur) == phase.
2)],
                                    p4.dur[which(names(p4.dur) == phase.
4)])
      }

      #4,5 =0
      else if(length(phase.1) != 0 && length(phase.2) != 0 && length(phase.
3) != 0 && length(phase.4) == 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.1, phase.2, phase.3)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.
1)],
                                    p2.dur[which(names(p2.dur) == phase.
2)],
                                    p3.dur[which(names(p3.dur) == phase.
3)])
      }

      # 3 numbers
      #1,2,3 =0
      else if(length(phase.1) == 0 && length(phase.2) == 0 && length(phase.
3) == 0 && length(phase.4) != 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.4, phase.5)
        base$duration[index] <- max(p4.dur[which(names(p4.dur) == phase.
4)],
                                    p5.dur[which(names(p5.dur) == phase.5)]
)
      }

      #1,2,4 =0
      else if(length(phase.1) == 0 && length(phase.2) == 0 && length(phase.
3) != 0 && length(phase.4) == 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.3, phase.5)
        base$duration[index] <- max(p3.dur[which(names(p3.dur) == phase.
3)],
                                    p5.dur[which(names(p5.dur) == phase.5)]
)
      }
      # 1,2,5 =0
      else if(length(phase.1) == 0 && length(phase.2) == 0 && length(phase.
3) != 0 && length(phase.4) != 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.4, phase.3)
        base$duration[index] <- max(p4.dur[which(names(p4.dur) == phase.
4)],
                                    p3.dur[which(names(p3.dur) == phase.3)]
)
      }
```

```
      #1,3,4 =0
      else if(length(phase.1) == 0 && length(phase.2) != 0 && length(phase.
3) == 0 && length(phase.4) == 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.2, phase.5)
        base$duration[index] <- max(p2.dur[which(names(p2.dur) == phase.
2)],
                                    p5.dur[which(names(p5.dur) == phase.5)]
)
      }
      #1,3,5 =0
      else if(length(phase.1) == 0 && length(phase.2) != 0 && length(phase.
3) == 0 && length(phase.4) != 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.4, phase.2)
        base$duration[index] <- max(p4.dur[which(names(p4.dur) == phase.
4)],
                                    p2.dur[which(names(p2.dur) == phase.2)]
)
      }
      #1,4,5 =0
      else if(length(phase.1) == 0 && length(phase.2) != 0 && length(phase.
3) != 0 && length(phase.4) == 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.2, phase.3)
        base$duration[index] <- max(p2.dur[which(names(p2.dur) == phase.
2)],
                                    p3.dur[which(names(p3.dur) == phase.3)]
)
      }
      #2,3,4 =0
      else if(length(phase.1) != 0 && length(phase.2) == 0 && length(phase.
3) == 0 && length(phase.4) == 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.1, phase.5)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.
1)],
                                    p5.dur[which(names(p5.dur) == phase.5)]
)
      }
      #2,3,5 = 0
      else if(length(phase.1) != 0 && length(phase.2) == 0 && length(phase.
3) == 0 && length(phase.4) != 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.1, phase.4)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.
1)],
                                    p4.dur[which(names(p4.dur) == phase.4)]
)
      }
      #3,4,5 =0
      else if(length(phase.1) != 0 && length(phase.2) != 0 && length(phase.
3) == 0 && length(phase.4) == 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.1, phase.2)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.
1)],
                                    p2.dur[which(names(p2.dur) == phase.2)]
)
      }
```

```
      #2,4,5 =0
      else if(length(phase.1) != 0 && length(phase.2) == 0 && length(phase.
3) != 0 && length(phase.4) == 0 && length(phase.5) == 0)
      {
      base$Names[index] <- paste( phase.1, phase.3)
      base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.1)],
                                  p3.dur[which(names(p3.dur) == phase.3)] )
      }


      #4 numbers
      #1,2,3,4 =0

      else if(length(phase.1) == 0 && length(phase.2) == 0 && length(phase.
3) == 0 && length(phase.4) == 0 && length(phase.5) != 0)
      {
        base$Names[index] <- paste( phase.5)
        base$duration[index] <- max(p5.dur[which(names(p5.dur) == phase.5)]
)
      }
         #2,3,4,5=0
      else if(length(phase.1) != 0 && length(phase.2) == 0 && length(phase.
3) == 0 && length(phase.4) == 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.1)
        base$duration[index] <- max(p1.dur[which(names(p1.dur) == phase.1)]
)
      }

      #1,3,4,5=0
      else if(length(phase.1) == 0 && length(phase.2) != 0 && length(phase.
3) == 0 && length(phase.4) == 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.2)
        base$duration[index] <- max(p2.dur[which(names(p2.dur) == phase.
2)])
      }
      #1,2,4,5=0
      else if(length(phase.1) == 0 && length(phase.2) == 0 && length(phase.
3) != 0 && length(phase.4) == 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.3)
        base$duration[index] <- max(p3.dur[which(names(p3.dur) == phase.3)]
)
      }

      #1,2,3,5=0
      else if(length(phase.1) == 0 && length(phase.2) == 0 && length(phase.
3) == 0 && length(phase.4) != 0 && length(phase.5) == 0)
      {
        base$Names[index] <- paste( phase.4)
        base$duration[index] <- max(p4.dur[which(names(p4.dur) == phase.
4)])
      }

     else {
      base$duration[index] <- 0
    }
  }
```

```
  return(base)
}

#ploting the lifetime
par(mfrow = c(2, 1))
start.Date = "2023-01-01"
lifetime <- 100

events.Op1.Photovoltaik = c(p.1 = 3, p.2 = 15, p.3 =30, END = lifetime) #p.
1 = professional maintenance, p.2 = replacement of inverters and batteries,
p.3 = complete renewal
duration.ev.Photovoltaik <- c(p.1 = 3, p.2 = 6, p.3 = 20)

events.Op1.WindTurbine <- c(w.1 = 4, w.2 = 6, w.3 = 25, END = lifetime) #w.
1 = Change sacrificial cathodic protection, w.2=repair scouring protection,
w.3 = complete renewal

duration.ev.WindTurbine <- c(w.1 = 3, w.2 = 8, w.3 = 30)

events.Op1.Dam = c(d.1 = 6, d.2 = 15, d.3 = 60, END = lifetime) #d.1= minor
repair(crack closure), d.2= partial renewal, d.3= complete renewal )
duration.ev.Dam <- c(d.1 = 20, d.2 = 40, d.3 = 80)

events.Op1.HeatingSystem = c(h.1 = 3, h.2 = 6, h.3 = 12, END = lifetime)
#h.1 = Regular maintenance, h.2=  Distribution system maintenace, h3. main
engine change
duration.ev.HeatingSystem <- c(h.1 = 2, h.2 = 6, h.3 = 4)

events.Op1.RetainingWall = c(r.1 = 5, r.2 = 8, r.3 = 10, END = lifetime)
#r.1 = maintenance, r.2= overturning check , r.3 = sliding check
duration.ev.RetainingWall <- c(r.1 = 1, r.2 = 11, r.3 = 22)

design.Options.Photovoltaik <- list(desing.Op1.Photovoltaik <-
"Photovoltaik")
design.Options.WindTurbine <- list(desing.Op1.WindTurbine <- "Wind
Turbine")
design.Options.Dam <- list(desing.Op1.Dam <- "Hydropower Dam")  # rockfill
concrete dam
design.Options.HeatingSystem <- list(desing.Op1.HeatingSystem <- "Heating
System from Building") # baseboard resistance with electricity
design.Options.RetainingWall <- list(desing.Op1.RetainingWall <- "Retaining
Wall") #reinforced concrete wall

maintenance.Photovoltaik <- dist.Events(lifetime, events.Op1.Photovoltaik,
start.Date, design.Options.Photovoltaik)
maintenance.WindTurbine <- dist.Events(lifetime, events.Op1.WindTurbine,
start.Date, design.Options.WindTurbine)
maintenance.Dam <- dist.Events(lifetime, events.Op1.Dam, start.Date,
design.Options.Dam)
maintenance.HeatingSystem <- dist.Events(lifetime,
events.Op1.HeatingSystem, start.Date, design.Options.HeatingSystem)
maintenance.RetainingWall <- dist.Events(lifetime,
events.Op1.RetainingWall, start.Date, design.Options.RetainingWall)



integrated.interv <- combine.lifeTimelines( maintenance.Photovoltaik,
maintenance.WindTurbine, maintenance.Dam, maintenance.HeatingSystem,
maintenance.RetainingWall,
```

```
                                          duration.ev.Photovoltaik,
duration.ev.WindTurbine,  duration.ev.Dam, duration.ev.HeatingSystem,
duration.ev.RetainingWall)

integrated.interv1 <- combine.lifeTimelines1( maintenance.Photovoltaik,
duration.ev.Photovoltaik)
integrated.interv2 <- combine.lifeTimelines1( maintenance.WindTurbine,
duration.ev.WindTurbine)
integrated.interv3<- combine.lifeTimelines1( maintenance.Dam,
duration.ev.Dam)
integrated.interv4 <- combine.lifeTimelines1( maintenance.HeatingSystem,
duration.ev.HeatingSystem)
integrated.interv5 <- combine.lifeTimelines1( maintenance.RetainingWall,
duration.ev.RetainingWall)

sum(integrated.interv1$duration)
sum(integrated.interv2$duration)
sum(integrated.interv3$duration)
sum(integrated.interv4$duration)
sum(integrated.interv5$duration)


sum(integrated.interv$duration)


#Automate more scenarios
design.explore <- function(events1, events2, events3, events4, events5) {
  results <- c()
  for(i in 1: dim(events1)[1]) {
    ev1 <- unlist(events1[i, ])
    dist.1 <- dist.Events(lifetime, ev1, start.Date,
design.Options.Photovoltaik)
    dur.ev1 <- ev1/5
    for (j in 1: dim(events1)[1]) {
      ev2 <- unlist(events2[j, ])
      dist.2 <- dist.Events(lifetime, ev2, start.Date,
design.Options.WindTurbine)
      dur.ev2 <- ev2/5
      for(k in 1: dim(events3)[1]) {
        ev3 <- unlist(events3[k, ])
        dist.3 <- dist.Events(lifetime, ev3, start.Date,
design.Options.Dam)
        dur.ev3 <- ev3/5
        for(l in 1: dim(events4)[1]) {
          ev4 <- unlist(events4[l, ])
          dist.4 <- dist.Events(lifetime, ev4, start.Date,
design.Options.HeatingSystem)
          dur.ev4 <- ev4/5
          for(m in 1: dim(events5)[1]) {
            ev5 <- unlist(events5[m, ])
            dist.5 <- dist.Events(lifetime, ev5, start.Date,
design.Options.RetainingWall)
            dur.ev5 <- ev5/5
      combined.lifetime <- combine.lifeTimelines(dist.1, dist.2, dist.3,
dist.4, dist.5,
                                                 dur.ev1, dur.ev2, dur.ev3,
dur.ev4, dur.ev5)
      min.dist.int <- min(abs(combined.lifetime$frequency[1:
(length(combined.lifetime$frequency) - 1)]
```

```
                                -
combined.lifetime$frequency[2:length(combined.lifetime$frequency)]
                                -
combined.lifetime$frequency[2:length(combined.lifetime$frequency)]
                                -
combined.lifetime$frequency[2:length(combined.lifetime$frequency)]
                                -
combined.lifetime$frequency[2:length(combined.lifetime$frequency)]))
      results <- rbind(results, c(ev1, ev2, ev3, ev4, ev5, dur =
sum(combined.lifetime$duration), dist.inter = min.dist.int))
          }
        }
      }
    }
  }
  return(as.data.frame(results))
}


#Design space
n.grid <- 2


events.grid.Photovoltaik <- expand.grid(p.1 = sample(seq(2, 7, by = 1),
n.grid),
                                p.2 = sample(seq(12,28,1), n.grid),
                                p.3 = sample(seq(30, 50, 1), n.grid))
events.grid.Photovoltaik

events.grid.WindTurbine <- expand.grid(w.1 = sample(seq(3,6, by = 1),
n.grid),
                                  w.2 = sample(seq(5,8,1), n.grid),
                                  w.3 = sample(seq(25, 35, 1), n.grid))
events.grid.WindTurbine

events.grid.Dam <- expand.grid(d.1 = sample(seq(3, 10, by = 1), n.grid),
                                d.2 = sample(seq(10,22,1), n.grid),
                                d.3 = sample(seq(40, 80, 1), n.grid))
events.grid.Dam

events.grid.HeatingSystem <- expand.grid(h.1 = sample(seq(1, 5, by = 1),
n.grid),
                                h.2 = sample(seq(3,12,1), n.grid),
                                h.3 = sample(seq(10, 25, 1), n.grid))
events.grid.HeatingSystem

events.grid.RetainingWall <- expand.grid(r.1 = sample(seq(2, 9, by = 1),
n.grid),
                                r.2 = sample(seq(7 ,19,1), n.grid),
                                r.3 = sample(seq(5, 15, 1), n.grid))
events.grid.RetainingWall

response.space <- design.explore(events.grid.WindTurbine,
events.grid.Photovoltaik, events.grid.Dam, events.grid.HeatingSystem,
events.grid.RetainingWall)
response.space
l<-length(response.space$w.1)
l
#Design preference
p <- low(dur)* high(dist.inter)
```

```
sky <- psel(response.space, p)
sky

#Alternative ranks
pareto2 <- psel(response.space, p, top = nrow(response.space))

#ploting
ggplot(response.space, aes(x = dur, y = dist.inter)) +
  geom_point(shape = 21) +
  geom_point(data = pareto2, size = 3, aes(color = factor(pareto2$.level)))


#Find the Pareto frontier for different preferences (more of a sorting)
show_front <- function(pref) {
  plot(response.space$dur, response.space$dist.inter)
  sky <- psel(response.space, pref)
  plot_front(response.space, pref, col = rgb(0, 0, 1))
  points(sky$dur, sky$dist.inter, lwd = 3)
}
show_front(p)




LCA.system <- function(i, energy, CO2, NOx, SO2, dist.event) {

  if (i == 1){
    count <- table(dist.event$event)
    interventions <- c(count["p.1"], count["p.2"], count["p.3"])
    interventions[is.na(interventions)] <- 0
    energy <- energy + 0.001 * energy * interventions[1] + 0.01 * energy *
interventions[2] + 0.8 * energy * interventions[3]
    CO2 <- CO2 +  0.001 * CO2 * interventions[1] + 0.01 * CO2 *
interventions[2] + 0.8 * CO2 * interventions[3]
    NOx <- NOx +  0.001 * NOx * interventions[1] + 0.01 * NOx *
interventions[2] + 0.8 * NOx * interventions[3]
    SO2 <- SO2 +  0.001 * SO2 * interventions[1] + 0.01 * SO2 *
interventions[2] + 0.8 * SO2 * interventions[3]

  } else if (i == 2) {
    count <- table(dist.event$event)
    interventions <- c(count["w.1"], count["w.2"], count["w.3"])
    interventions[is.na(interventions)] <- 0
    energy <- energy + 0.005 * energy * interventions[1] + 0.05 * energy *
interventions[2] + 0.8 * energy * interventions[3]
    CO2 <- CO2 +  0.005 * CO2 * interventions[1] + 0.05 * CO2 *
interventions[2] + 0.8 * CO2 * interventions[3]
    NOx <- NOx +  0.005 * NOx * interventions[1] + 0.05 * NOx *
interventions[2] + 0.8 * NOx * interventions[3]
    SO2 <- SO2 +  0.005 * SO2 * interventions[1] + 0.05 * SO2 *
interventions[2] + 0.8 * SO2 * interventions[3]

  } else if (i == 3) {
    count <- table(dist.event$event)
    interventions <- c(count["d.1"], count["d.2"], count["d.3"])
    interventions[is.na(interventions)] <- 0
    energy <- energy + 0.001 * energy * interventions[1] + 0.005 * energy *
interventions[2] + 0.01 * energy * interventions[3]
```

```
    CO2 <- CO2 +  0.001 * CO2 * interventions[1] + 0.005 * CO2 *
interventions[2] + 0.01 * CO2 * interventions[3]
    NOx <- NOx +  0.001 * NOx * interventions[1] + 0.005 * NOx *
interventions[2] + 0.01 * NOx * interventions[3]
    SO2 <- SO2 +  0.001 * SO2 * interventions[1] + 0.005 * SO2 *
interventions[2] + 0.01 * SO2 * interventions[3]
  } else if (i == 4) {
    count <- table(dist.event$event)
    interventions <- c(count["h.1"], count["h.2"], count["h.3"])
    interventions[is.na(interventions)] <- 0
    energy <- energy + 0.01 * energy * interventions[1] + 0.005 * energy *
interventions[2] + 0.8 * energy * interventions[3]
    CO2 <- CO2 +  0.01 * CO2 * interventions[1] + 0.005 * CO2 *
interventions[2] + 0.8 * CO2 * interventions[3]
    NOx <- NOx +  0.01 * NOx * interventions[1] + 0.005 * NOx *
interventions[2] + 0.8 * NOx * interventions[3]
    SO2 <- SO2 +  0.01 * SO2 * interventions[1] + 0.005 * SO2 *
interventions[2] + 0.8 * SO2 * interventions[3]
  } else if (i == 5) {
    count <- table(dist.event$event)
    interventions <- c(count["r.1"], count["r.2"], count["r.3"])
    interventions[is.na(interventions)] <- 0
    energy <- energy + 0.005 * energy * interventions[1] + 0.01 * energy *
interventions[2] + 0.03 * energy * interventions[3]
    CO2 <- CO2 +  0.005 * CO2 * interventions[1] + 0.01 * CO2 *
interventions[2] + 0.03 * CO2 * interventions[3]
    NOx <- NOx +  0.005 * NOx * interventions[1] + 0.01 * NOx *
interventions[2] + 0.03 * NOx * interventions[3]
    SO2 <- SO2 +  0.005 * SO2 * interventions[1] + 0.01 * SO2 *
interventions[2] + 0.03 * SO2 * interventions[3]
  } else {
    0
  }

  LCA.results <- c(energy,
                   CO2,
                   NOx,
                   SO2)
  return(LCA.results)
}


Option.Photovoltaik <- LCA.system(1, 90000, 1500, 15, 15,
maintenance.Photovoltaik) #(i, energy, CO2, NOx, SO2, dist.event)
Option.WindTurbine <- LCA.system(2, 276748.74, 197.76, 5.74, 0.106,
maintenance.WindTurbine)
Option.Dam <- LCA.system(3, 19695950, 2395.85, 550.83, 3664.94,
maintenance.Dam)
Option.HeatingSystem <- LCA.system(4, 111415, 220, 4, 20,
maintenance.HeatingSystem)
Option.RetainingWall <- LCA.system(5, 550000, 150, 5, 16,
maintenance.RetainingWall)

integrated.Design <- as.data.frame(list(Energy = Option.Photovoltaik[1] +
Option.WindTurbine[1] + Option.Dam[1] + Option.HeatingSystem[1] +
Option.RetainingWall[1],
                                        CO2 = Option.Photovoltaik[2] +
Option.WindTurbine[2] + Option.Dam[2] + Option.HeatingSystem[2] +
Option.RetainingWall[2],
```

```
                                        NOx = Option.Photovoltaik[3] +
Option.WindTurbine[3] + Option.Dam[3] + Option.HeatingSystem[3] +
Option.RetainingWall[3],
                                        SO2 = Option.Photovoltaik[4] +
Option.WindTurbine[4] + Option.Dam[4] + Option.HeatingSystem[4] +
Option.RetainingWall[4]))
Energy.costs <- 0.2 # Euro per MJ
CO2.unitcosts <- 55 # Euro per metric tone
NOx.unitcosts <- 42 # Euro per metric tone
SO2.unitcosts <- 85 # Euro per metric tone

integrated.Design <- mutate(integrated.Design,
                            Costs = (Energy * Energy.costs +
                                       CO2*CO2.unitcosts +
                                       NOx * NOx.unitcosts +
                                       SO2 * SO2.unitcosts)/10^0)


#Multi-Objective Optimization
fitness <- function(x) {
  #define the output dimension
  z <- numeric(7)
  x <- round(x, 0)
  y <- expand.grid(p.1 = x[1], p.2 = x[2], p.3 = x[3], w.1 = x[4], w.2 =
x[5], w.3 = x[6],
                   d.1 = x[7], d.2 = x[8], d.3 = x[9], h.1 = x[10], h.2 =
x[11], h.3 = x[12],
                   r.1 = x[13], r.2 = x[14], r.3 = x[15])

  dur.ev1 <- unlist(y[1:3] / 5)
  dur.ev2 <- unlist(y[4:6] / 5)
  dur.ev3 <- unlist(y[7:9] / 5)
  dur.ev4 <- unlist(y[10:12] / 5)
  dur.ev5 <- unlist(y[13:15] / 5)

  dist.1 <- apply(y[1:3], 1, FUN = dist.Events, lifetime = lifetime,
start.Date = start.Date)
  dist.2 <- apply(y[4:6], 1, FUN = dist.Events, lifetime = lifetime,
start.Date = start.Date)
  dist.3 <- apply(y[7:9], 1, FUN = dist.Events, lifetime = lifetime,
start.Date = start.Date)
  dist.4 <- apply(y[10:12], 1, FUN = dist.Events, lifetime = lifetime,
start.Date = start.Date)
  dist.5 <- apply(y[13:15], 1, FUN = dist.Events, lifetime = lifetime,
start.Date = start.Date)

  results <- combine.lifeTimelines(dist.1[[1]], dist.2[[1]], dist.3[[1]],
dist.4[[1]], dist.5[[1]], dur.ev1, dur.ev2, dur.ev3, dur.ev4, dur.ev5)
  results
  z[1] <- sum(results[["duration"]])
  z[2] <- -min(abs(results$frequency[1:(length(results$frequency) - 1)]
                   - results$frequency[2:length(results$frequency)]))


   Product.1.Photovoltaik <- LCA.system(1, 90000, 1500, 15, 15, dist.
1[[1]])
   Product.2.WindTurbine <- LCA.system(2, 276748.74, 197.76, 5.74, 0.106,
dist.2[[1]])
   Product.3.Dam <- LCA.system(3, 19695950, 2395.85, 550.83, 3664.94, dist.
3[[1]])
```

```
   Product.4.HeatingSystem <- LCA.system(4, 111415, 220, 4, 20, dist.
4[[1]])
   Product.5.RetainingWall <- LCA.system(5, 550000, 150, 5, 16, dist.
5[[1]])



  integrated.system <- as.data.frame(list(Energy = Product.
1.Photovoltaik[1] + Product.2.WindTurbine[1] + Product.3.Dam[1] + Product.
4.HeatingSystem[1] + Product.5.RetainingWall[1],
                                          CO2 = Product.1.Photovoltaik[2] +
Product.2.WindTurbine[2] + Product.3.Dam[2] + Product.4.HeatingSystem[2] +
Product.5.RetainingWall[2],
                                          NOx = Product.1.Photovoltaik[3] +
Product.2.WindTurbine[3] + Product.3.Dam[3] + Product.4.HeatingSystem[3] +
Product.5.RetainingWall[3],
                                          SO2 = Product.1.Photovoltaik[4] +
Product.2.WindTurbine[4] + Product.3.Dam[4] + Product.4.HeatingSystem[4] +
Product.5.RetainingWall[4]))
  integrated.system <- mutate(integrated.system,
                              costs = (Energy * Energy.costs +
                                         CO2*CO2.unitcosts +
                                         NOx * NOx.unitcosts +
                                         SO2 * SO2.unitcosts)/10^0)

  z[3] <- integrated.system$Energy
  z[4] <- integrated.system$CO2
  z[5] <- integrated.system$NOx
  z[6] <- integrated.system$SO2
  z[7] <- integrated.system$costs

  return(z)
}

#GA optimization function
r2 <- nsga2(fitness, idim = 15, odim = 7,
            generations=10, popsize=100,
            lower.bounds=c(2, 12, 30, 3, 5, 25, 3, 10, 40, 1, 3, 10, 2, 7,
5),
            upper.bounds= c(7, 28, 50, 6, 8, 35, 10, 22, 80, 5, 12, 25, 9,
19, 15))

r2


#Plot the r2 results
r2Results <- as.data.frame(r2$value)

outNames <- c("duration", "interv.dist", "energy", "co2", "nox", "so2",
"cost" )
colnames(r2Results) <- outNames

pareto3 <- as.data.frame(paretoFront(r2))
colnames(pareto3) <- outNames

ggplot(r2Results, aes(x = duration, y = cost)) + geom_point(shape = 21) +
  geom_point(data = pareto3, size = 3, color="red") +
  geom_line(data = pareto3, color="blue")

#accumulated plot
```

```
input.params <- round(r2$par, 0)
all.results <- cbind(input.params, r2Results, r2$pareto.optimal)
colnames(all.results) <- c("p.1", "p.2", "p.3",
                           "w.1", "w.2", "w.3",
                           "d.1", "d.2", "d.3",
                           "h.1", "h.2", "h.3",
                           "r.1", "r.2", "r.3",
                           "duration", "interv.dist", "energy", "co2",
"nox", "so2", "cost", "pareto")
par(mfrow = c(1,1))
parcoord(all.results[, 1:23], var.label = T,
         col = ifelse(all.results$pareto == TRUE, "indianred", "skyblue2"),
         lty = ifelse(all.results$pareto == TRUE, 1, 3),
         lwd = ifelse(all.results$pareto == TRUE, 3, 1))
```