```
# 05.02.2020
# Niklas Moehring and Leonie Vidensky
# Agricultural Economics and Policy Group, Eth Zurich


# Calclulating the Danish Pesticide Load Indicator
# (Kudsk et al., 2018; Moehring et al., 2019)
# with three example products from Switzerland

#' @importFrom stats aggregate

required_columns_products <- c(
  "crop",
  "formula",
  "product",
  "reference.sum.risk.scores",
  "sum.risk.score"
)

optional_columns_products <- c("amount.applied", "standard.doses")

required_columns_substances <- c(
  "Algae.Acute.72hr.EC50.Growth.mg.l",
  "Aquatic.Invertebrates.Acute.48hr.EC50.mg.l",
  "Aquatic.Invertebrates.Chronic.21d.NOEC.mg.l.correted",
  "Aquatic.Plants.Acute.7d.EC50.mg.l",
  "BCF",
  "BeesLD50",
  "Birds.Acute.LD50.mg.kg",
  "Earthworms.Acute.14d.LC50.mg.kg",
  "Earthworms.Chronic.14d.NOEC..Reproduction.mg.kg.corrected",
  "Fish.Acute.96hr.LC50.mg.l",
  "Fish.Chronic.21d.NOEC.mg.l.corrected",
  "Load.Factor.Algae",
  "Load.Factor.Aquatic.Invertebrates",
  "Load.Factor.Aquatic.Invertebrates.Chronic",
  "Load.Factor.Aquatic.Plants",
  "Load.Factor.BCF",
  "Load.Factor.Bees",
  "Load.Factor.Birds",
  "Load.Factor.Earthworms",
  "Load.Factor.Earthworms.Chronic",
  "Load.Factor.Fish",
  "Load.Factor.Fish.Chronic",
  "Load.Factor.Mammals",
  "Load.Factor.SCI",
  "Load.Factor.SoilDT50",
  "Mammals.Acute.Oral.LD50.mg.kg.BW.day",
  "Reference.BCF",
  "Reference.SCI-Grow",
  "Reference.SoilDT50",
  "Reference.Value.Algae",
  "Reference.Value.Aquatic.Invertebrates",
  "Reference.Value.Aquatic.Invertebrates.Chronic",
  "Reference.Value.Aquatic.Plants",
  "Reference.Value.Bees",
  "Reference.Value.Birds",
  "Reference.Value.Earthworms",
  "Reference.Value.Earthworms.Chronic",
  "Reference.Value.Fish",
  "Reference.Value.Fish.Chronic",
  "Reference.Value.Mammals",
  "SCI-Grow",
  "SoilDT50",
  "concentration",
  "product",
  "substance",
  "water.phase.DT50.days"
)
```

```
#' @title Compute Pesticide Load Indicator
#'
#' @param substances Dataframe with substance data
#' @param products Dataframe with products data
#' @return products Dataframe with added columns
#'
#' @export

compute_pesticide_load_indicator <- function(substances, products) {
  check_substance_column_names(substances)
  check_products_column_names(products)

  substances <- compute_fate_load(substances)
  substances <- compute_toxity_load(substances)

  products <- compute_health_load(products)
  products <- compute_pesticide_load(products, substances)

  if (all(optional_columns_products %in% names(products))) {
    products <- compute_load_index(products)
  }

  return(products)
}


#' @title Check if column names of substances are valid
#'
#' @param substances Dataframe with substance data
#'
#' @export

check_substance_column_names <- function(substances)
{
    check_columns(substances,
                  required_columns_substances,
                  c(),
                  "substances")
}


#' @title Check if column names of products are valid
#'
#' @param products Dataframe with substance data
#'
#' @export

check_products_column_names <- function(products)
{
    check_columns(products,
                  required_columns_products,
                  optional_columns_products,
                  "products")
}


check_columns <- function(data_frame, required, optional, name) {
  found_columns <- names(data_frame)

  if (setequal(found_columns, required)) {
    return(data_frame)
  }


  if (setequal(found_columns, union(required, optional))) {
    return(data_frame)
  }

  missing <- setdiff(required, found_columns)
  unknown <- setdiff(found_columns, union(required, optional))
  missing_str <- paste(missing, collapse = ", ")
  unknown_str <- paste(unknown, collapse = ", ")

  if (length(missing_str) == 0) {
    missing_str <- "none"
  }
  if (length(unknown_str) == 0) {
    unknown_str <- "none"
  }

  message <- sprintf(
    "%s data frame is not valid. missing columns: %s, unknown columns: %s",
    name, missing_str, unknown_str
  )

  stop(message)
}


compute_fate_load <- function(substances) {
  degradation <- (substances$`SCI-Grow`
    / substances$`Reference.SCI-Grow`
    * substances$Load.Factor.SCI)

  bioaccumulation <- (substances$BCF
    / substances$Reference.BCF
    * substances$Load.Factor.BCF)

  sci_growth_index <- (substances$SoilDT50
    / substances$Reference.SoilDT50
    * substances$Load.Factor.SoilDT50)

  substances$U <- degradation
  substances$B <- bioaccumulation
  substances$P <- sci_growth_index
  substances$Fate.Load.substances <- substances$U + substances$B + substances$P

  return(substances)
}


compute_toxity_load <- function(substances) {
  short_term_effect_birds <- (substances$Reference.Value.Birds
    / substances$Birds.Acute.LD50.mg.kg
    * substances$Load.Factor.Birds)

  substances$Fa <- ifelse(is.finite(short_term_effect_birds),
    short_term_effect_birds,
    0
  )

  short_term_effect_mammals <- (substances$Reference.Value.Mammals
    / substances$Mammals.Acute.Oral.LD50.mg.kg.BW.day
    * substances$Load.Factor.Mammals)

  substances$Pa <- ifelse(is.finite(short_term_effect_mammals),
    short_term_effect_mammals,
    0
  )

  short_term_effect_fish <- (substances$Reference.Value.Fish
    / substances$Fish.Acute.96hr.LC50.mg.l
    * substances$Load.Factor.Fish)

  substances$Fla <- ifelse(is.finite(short_term_effect_fish),
    short_term_effect_fish,
    0
  )
  short_term_effect_daphina <- (substances$Reference.Value.Aquatic.Invertebrates
    / substances$Aquatic.Invertebrates.Acute.48hr.EC50.mg.l
    * substances$Load.Factor.Aquatic.Invertebrates)

  substances$Da <- ifelse(is.finite(short_term_effect_daphina),
    short_term_effect_daphina,
    0
  )

  short_term_effect_algae <- (substances$Reference.Value.Algae
    / substances$Algae.Acute.72hr.EC50.Growth.mg.l
    * substances$Load.Factor.Algae)

  substances$Aa <- ifelse(is.finite(short_term_effect_algae),
    short_term_effect_algae,
    0
  )
  short_term_effect_aquatic_plants <- (substances$Reference.Value.Aquatic.Plants
    / substances$Aquatic.Plants.Acute.7d.EC50.mg.l
    * substances$Load.Factor.Aquatic.Plants)

  substances$Vp <- ifelse(is.finite(short_term_effect_aquatic_plants),
    short_term_effect_aquatic_plants,
    0
  )

  # short term effect Earthworms
  short_term_effect_earthworms <- (substances$Reference.Value.Earthworms
    / substances$Earthworms.Acute.14d.LC50.mg.kg
    * substances$Load.Factor.Earthworms)

  substances$Ra <- ifelse(is.finite(short_term_effect_earthworms),
    short_term_effect_earthworms,
    0
  )

  short_term_effect_bees <- (substances$Reference.Value.Bees
    / substances$BeesLD50
    * substances$Load.Factor.Bees)

  substances$Ba <- ifelse(is.finite(short_term_effect_bees),
    short_term_effect_bees,
    0
  )

  degradation_factor_water <- (1 - exp(-log(2) / substances$water.phase.DT50.days * 7)
  / (log(2) / substances$water.phase.DT50.days * 7))

  substances$Degradation.Factor.Water <- (
    ifelse(substances$water.phase.DT50.days == 0 | substances$water.phase.DT50.days == 70
8,
      1,
      degradation_factor_water
    )
  )

  long_term_effect_fish <- (substances$Reference.Value.Fish.Chronic
    / substances$Fish.Chronic.21d.NOEC.mg.l.corrected
    * substances$Load.Factor.Fish.Chronic
    * substances$Degradation.Factor.Water)

  substances$Flk <- ifelse(is.finite(long_term_effect_fish),
    long_term_effect_fish,
    0
  )

  long_term_effect_daphina <- (substances$Reference.Value.Aquatic.Invertebrates.Chronic
    / substances$Aquatic.Invertebrates.Chronic.21d.NOEC.mg.l.correted
    * substances$Load.Factor.Aquatic.Invertebrates.Chronic
    * substances$Degradation.Factor.Water)
  substances$Dk <- ifelse(is.finite(long_term_effect_daphina),
    long_term_effect_daphina,
    0
  )

  degradation_factor_soil <-
    (1 - exp((-log(2) / substances$SoilDT50) * 180)) / ((log(2) / substances$SoilDT50) *
180)

  substances$Degradation.Factor.Soil <- ifelse(substances$SoilDT50 == 0 | substances$Soil
DT50 == 708,
    1,
    degradation_factor_soil
  )

  long_term_effect_earthworms <- (substances$Reference.Value.Earthworms.Chronic
    / substances$Earthworms.Chronic.14d.NOEC..Reproduction.mg.kg.corrected
    * substances$Load.Factor.Earthworms.Chronic
    * substances$Degradation.Factor.Soil)

  substances$Rk <- ifelse(is.finite(long_term_effect_earthworms),
    long_term_effect_earthworms,
    0
  )

  substances$Environmental.Toxicity.Substance <- (
    substances$Fa
      + substances$Pa
      + substances$Fla
      + substances$Da
      + substances$Aa
      + substances$Vp
      + substances$Ra
      + substances$Ba
      + substances$Flk
      + substances$Dk
      + substances$Rk
  )

  return(substances)
}


compute_health_load <- function(products) {
  products$HL <- (products$formula
    * products$sum.risk.score
    / products$reference.sum.risk.scores)

  return(products)
}


compute_pesticide_load <- function(products, substances) {
  TL.products <- substances$concentration * substances$Environmental.Toxicity.Substance
  FL.products <- substances$concentration * substances$Fate.Load.substances

  TL <- aggregate(TL.products, by = list(products = substances$product), FUN = sum)
  FL <- aggregate(FL.products, by = list(products = substances$product), FUN = sum)

  products$TL <- TL$x[TL$products == products$product]
  products$FL <- FL$x[FL$products == products$product]

  products$L <- products$HL + products$TL + products$FL

  return(products)
}


compute_load_index <- function(products) {
  sti_qutient <- products$amount.applied / products$standard.doses
  load_index <- sti_qutient * products$L

  products$STI <- sti_qutient
  products$LI <- load_index

  return(products)
}
```