```
#####################
### Control panel ###
#####################

# Define if øko or not
switch_variable <- "yes"  # "yes" or "no"

# Define the vector of years
years_vector <- c(2017,2018,2019,2020,2021,2022)  # Add your desired years here

#Set working directory
setwd("P:/2024/1243_PAF_Klima_MAJH_Økologisk_Planteavl_som_nationalt_Virkemiddel_for_klima/01_Arbejdsmappe/Vare/R script")


#################
### Opsætning ###
#################

if (!require(tidyr))install.packages("tidyr")
if (!require(data.table))install.packages("data.table")
if (!require(dplyr))install.packages("dplyr")
if (!require(writexl))install.packages("writexl")
if (!require(bit64))install.packages("bit64")
if (!require(readxl))install.packages("readxl")
if (!require(openxlsx))install.packages("openxlsx")





options(scipen = 999)


###################
### Import data ###
###################


# Import the appropriate data based on the switch
if (switch_variable == "yes") {
  Field_info_RAW <- fread("Field_Info_Eco.csv", sep = ";", dec = ",")#import Øko field data
} else {
  Field_info_RAW <- fread("Field_Info_conventional.csv", sep = ";", dec = ",")#import conventional field data
}



#Import Index data from the CSV file "Index data" into the script as a dataframe with the name "Index_data"
Index_data<-read.csv("Index_Data.csv")


Crop_Index<-read.csv("Crop_Index.csv")


Leftovercrop_Index <- read_excel("Leftovercrop_Index.xlsx")


CropYield_Index <- read_excel("CropYield_Index.xlsx")


ECO_data<-read_excel("MarkOnline_øko_normtal.xlsx")


LBST_ECO_Area<-read_excel("LBST_ECO_Area.xlsx")

#Remove the duplicate unique kg value for Kg
CropYield_Index$Enhed <- gsub("kg", "Kg", CropYield_Index$Enhed, ignore.case = TRUE)



####################
### Prepare data ###
####################

### Filter ###

###### Sæt til enten organic (=0) eller inorganic (=1) ######
# Remove rows where isOrganic is = x
# Import the appropriate data based on the switch
if (switch_variable == "yes") {
  Field_info_RAW <- Field_info_RAW %>%
    filter(isOrganic != 0)#Remove Non øko fields
} else {
  Field_info_RAW <- Field_info_RAW %>%
    filter(isOrganic != 1)#Remove øko fields
}




# Define what columns from the RAW data is carried over to the script
Field_info <- Field_info_RAW[, c("farmid","harvestYear", "isOrganic", "fieldId","fieldYearId","preCropName",
                                 "directorateCropCode","directorateCropName","area","harmonyArea","mainCropYield",
                                 "catchCropDirectorateCode","catchCropDirectorateName",
                                 "SuppliedOrganicN","SuppliedCommercialFertilizerN","NTotal","jb","isMainCrop",
                                 "EcoStartDate")]
```

```r
# Replace NA values with 0 for specific columns
Field_info$SuppliedOrganicN[is.na(Field_info$SuppliedOrganicN)] <- 0
Field_info$SuppliedCommercialFertilizerN[is.na(Field_info$SuppliedCommercialFertilizerN)] <- 0



# Step 1: Find common years between Field_info and LBST_ECO_Area
common_years <- intersect(Field_info$harvestYear, LBST_ECO_Area$Year)

# Step 2: Calculate total area and kg N/ha only for common years
area_nitrogen_summary <- Field_info %>%
  filter(harvestYear %in% common_years) %>%
  group_by(harvestYear) %>%
  summarize(
    filter_area = sum(area, na.rm = TRUE),  # Total area for each harvestYear
    kg_N_per_ha = sum((SuppliedOrganicN + SuppliedCommercialFertilizerN) * area, na.rm = TRUE) / filter_area, # kg N/ha
    .groups = "drop"
  )

# Step 3: Join the summary with LBST_ECO_Area, adding filter_area and kg N/ha columns only for matching years
LBST_ECO_Area <- LBST_ECO_Area %>%
  left_join(area_nitrogen_summary, by = c("Year" = "harvestYear"))

# Calculate percentage_filter_area and add it as a new column in LBST_ECO_Area
LBST_ECO_Area <- LBST_ECO_Area %>%
  mutate(percentage_filter_area = (filter_area / Area) * 100) %>%
  relocate(percentage_filter_area, .after = filter_area)

LBST_ECO_Area <- LBST_ECO_Area %>%
  rename(noFilter_area = filter_area,percentage_filter_1_area = percentage_filter_area, filter_1_kg_N_per_ha = kg_N_per_ha)




# Select only columns with names starting with "manureTypeName" from Field_info_RAW
manure_columns <- Field_info_RAW %>%
  select(starts_with("manureTypeName"))

# Bind the selected columns to Field_info
Field_info <- bind_cols(Field_info, manure_columns)


# Check if the first value in Field_info$isOrganic is 1
if (Field_info$isOrganic[1] == 1) {
  # Remove rows where any manureTypeName column contains "Handelsgødning"
  Field_info <- Field_info %>%
    filter(!if_any(starts_with("manureTypeName"), ~ . == "Handelsgødning"))
}



# Remove all columns in Field_info that start with "manureTypeName"
Field_info <- Field_info %>%
  select(-starts_with("manureTypeName"))



# Step 1: Find common years between Field_info and LBST_ECO_Area
common_years <- intersect(Field_info$harvestYear, LBST_ECO_Area$Year)

# Step 2: Calculate total area and kg N/ha only for common years
area_nitrogen_summary <- Field_info %>%
  filter(harvestYear %in% common_years) %>%
  group_by(harvestYear) %>%
  summarize(
    filter_area = sum(area, na.rm = TRUE),  # Total area for each harvestYear
    kg_N_per_ha = sum((SuppliedOrganicN + SuppliedCommercialFertilizerN) * area, na.rm = TRUE) / filter_area, # kg N/ha
    .groups = "drop"
  )

# Step 3: Join the summary with LBST_ECO_Area, adding filter_area and kg N/ha columns only for matching years
LBST_ECO_Area <- LBST_ECO_Area %>%
  left_join(area_nitrogen_summary, by = c("Year" = "harvestYear"))

# Calculate percentage_filter_area and add it as a new column in LBST_ECO_Area
LBST_ECO_Area <- LBST_ECO_Area %>%
  mutate(percentage_filter_area = (filter_area / Area) * 100) %>%
  relocate(percentage_filter_area, .after = filter_area)

LBST_ECO_Area <- LBST_ECO_Area %>%
  rename(CommercialFertalizer_Filter_area = filter_area,percentage_filter_2_area = percentage_filter_area,
filter_2_kg_N_per_ha = kg_N_per_ha)







# Check if the first value of isOrganic is 1
if (Field_info$isOrganic[1] == 1) {

  # Convert EcoStartDate to a Date object (if it isn't already)
```

```
  Field_info$EcoStartDate <- as.Date(Field_info$EcoStartDate)

  # Filter the data frame to keep rows where EcoStartDate is on or before the start of the harvest year
  Field_info <- Field_info %>%
    filter(EcoStartDate <= as.Date(paste(harvestYear, "-01-01", sep = "")))
}

#Remove the omlægnings date (EcoStartDate)
Field_info <- Field_info %>% select(-EcoStartDate)



# Step 1: Find common years between Field_info and LBST_ECO_Area
common_years <- intersect(Field_info$harvestYear, LBST_ECO_Area$Year)

# Step 2: Calculate total area and kg N/ha only for common years
area_nitrogen_summary <- Field_info %>%
  filter(harvestYear %in% common_years) %>%
  group_by(harvestYear) %>%
  summarize(
    filter_area = sum(area, na.rm = TRUE),  # Total area for each harvestYear
    kg_N_per_ha = sum((SuppliedOrganicN + SuppliedCommercialFertilizerN) * area, na.rm = TRUE) / filter_area, # kg N/ha
    .groups = "drop"
  )

# Step 3: Join the summary with LBST_ECO_Area, adding filter_area and kg N/ha columns only for matching years
LBST_ECO_Area <- LBST_ECO_Area %>%
  left_join(area_nitrogen_summary, by = c("Year" = "harvestYear"))

# Calculate percentage_filter_area and add it as a new column in LBST_ECO_Area
LBST_ECO_Area <- LBST_ECO_Area %>%
  mutate(percentage_filter_area = (filter_area / Area) * 100) %>%
  relocate(percentage_filter_area, .after = filter_area)

LBST_ECO_Area <- LBST_ECO_Area %>%
  rename(ECOstartDateSameYear_Filter_area = filter_area,percentage_filter_3_area = percentage_filter_area,
filter_3_kg_N_per_ha = kg_N_per_ha)









#Remove rows that isn't for main crop
Field_info <- Field_info %>%
  filter(isMainCrop == 1)



# Step 1: Find common years between Field_info and LBST_ECO_Area
common_years <- intersect(Field_info$harvestYear, LBST_ECO_Area$Year)

# Step 2: Calculate total area and kg N/ha only for common years
area_nitrogen_summary <- Field_info %>%
  filter(harvestYear %in% common_years) %>%
  group_by(harvestYear) %>%
  summarize(
    filter_area = sum(area, na.rm = TRUE),  # Total area for each harvestYear
    kg_N_per_ha = sum((SuppliedOrganicN + SuppliedCommercialFertilizerN) * area, na.rm = TRUE) / filter_area, # kg N/ha
    .groups = "drop"
  )

# Step 3: Join the summary with LBST_ECO_Area, adding filter_area and kg N/ha columns only for matching years
LBST_ECO_Area <- LBST_ECO_Area %>%
  left_join(area_nitrogen_summary, by = c("Year" = "harvestYear"))

# Calculate percentage_filter_area and add it as a new column in LBST_ECO_Area
LBST_ECO_Area <- LBST_ECO_Area %>%
  mutate(percentage_filter_area = (filter_area / Area) * 100) %>%
  relocate(percentage_filter_area, .after = filter_area)

LBST_ECO_Area <- LBST_ECO_Area %>%
  rename(isMainCrop_Filter_area = filter_area,percentage_filter_4_area = percentage_filter_area, filter_4_kg_N_per_ha =
kg_N_per_ha)









# Create a reference yield based on the logic of isOrganic
reference_yield <- ifelse(
  Field_info$isOrganic == 1,
  # For organic crops, check if there's a value in ECO_data first, otherwise fallback to Crop_Index
  ifelse(
    !is.na(ECO_data$`Øko-normudbytte`[match(Field_info$directorateCropCode, ECO_data$Crop_ID)]),
    ECO_data$`Øko-normudbytte`[match(Field_info$directorateCropCode, ECO_data$Crop_ID)],
    Crop_Index$`Normudbytte..udbytteenhed.ha`[match(Field_info$directorateCropCode, Crop_Index$Crop_ID)]
  ),
  # For non-organic crops, use the value from Crop_Index
```

```
  Crop_Index$`Normudbytte..udbytteenhed.ha`[match(Field_info$directorateCropCode, Crop_Index$Crop_ID)]
)



# Remove rows where mainCropYield is more than 10 times higher than the reference yield
Field_info <- Field_info %>%
  filter(mainCropYield <= 10 * reference_yield | is.na(mainCropYield))


# Replace NA values with 0 in all columns except mainCropYield using dplyr
Field_info <- Field_info %>%
  mutate(across(-mainCropYield, ~ replace_na(., 0)))


# Filter Field_info based on the years in the vector
Field_info <- Field_info %>%
  filter(harvestYear %in% years_vector)



# Remove rows where the sum of SuppliedOrganicN and SuppliedCommercialFertilizerN is 500 or more
Field_info <- Field_info %>%
  filter((SuppliedOrganicN + SuppliedCommercialFertilizerN) < 500)


# Step 1: Find common years between Field_info and LBST_ECO_Area
common_years <- intersect(Field_info$harvestYear, LBST_ECO_Area$Year)

# Step 2: Calculate total area and kg N/ha only for common years
area_nitrogen_summary <- Field_info %>%
  filter(harvestYear %in% common_years) %>%
  group_by(harvestYear) %>%
  summarize(
    filter_area = sum(area, na.rm = TRUE),  # Total area for each harvestYear
    kg_N_per_ha = sum((SuppliedOrganicN + SuppliedCommercialFertilizerN) * area, na.rm = TRUE) / filter_area, # kg N/ha
    .groups = "drop"
  )

# Step 3: Join the summary with LBST_ECO_Area, adding filter_area and kg N/ha columns only for matching years
LBST_ECO_Area <- LBST_ECO_Area %>%
  left_join(area_nitrogen_summary, by = c("Year" = "harvestYear"))

# Calculate percentage_filter_area and add it as a new column in LBST_ECO_Area
LBST_ECO_Area <- LBST_ECO_Area %>%
  mutate(percentage_filter_area = (filter_area / Area) * 100) %>%
  relocate(percentage_filter_area, .after = filter_area)

LBST_ECO_Area <- LBST_ECO_Area %>%
  rename("500KgN_Filter_area" = filter_area,percentage_filter_5_area = percentage_filter_area, filter_5_kg_N_per_ha =
kg_N_per_ha)




# Remove rows where area is less than 0.2
Field_info <- Field_info %>%
  filter(area >= 0.2)

# Step 1: Find common years between Field_info and LBST_ECO_Area
common_years <- intersect(Field_info$harvestYear, LBST_ECO_Area$Year)

# Step 2: Calculate total area and kg N/ha only for common years
area_nitrogen_summary <- Field_info %>%
  filter(harvestYear %in% common_years) %>%
  group_by(harvestYear) %>%
  summarize(
    filter_area = sum(area, na.rm = TRUE),  # Total area for each harvestYear
    kg_N_per_ha = sum((SuppliedOrganicN + SuppliedCommercialFertilizerN) * area, na.rm = TRUE) / filter_area, # kg N/ha
    .groups = "drop"
  )

# Step 3: Join the summary with LBST_ECO_Area, adding filter_area and kg N/ha columns only for matching years
LBST_ECO_Area <- LBST_ECO_Area %>%
  left_join(area_nitrogen_summary, by = c("Year" = "harvestYear"))

# Calculate percentage_filter_area and add it as a new column in LBST_ECO_Area
LBST_ECO_Area <- LBST_ECO_Area %>%
  mutate(percentage_filter_area = (filter_area / Area) * 100) %>%
  relocate(percentage_filter_area, .after = filter_area)

LBST_ECO_Area <- LBST_ECO_Area %>%
  rename("0.2ha_Filter_area" = filter_area,percentage_filter_6_area = percentage_filter_area, filter_6_kg_N_per_ha =
kg_N_per_ha)


# Create the ResultsAlt directory if it doesn't exist
dir.create("ResultsAlt", showWarnings = FALSE)

# Check the first entry of Field_info$isOrganic
if (Field_info$isOrganic[1] == 1) {
  # Define the file path for the organic file in the ResultsAlt folder
  file_path <- "ResultsAlt/AndelFiltreret.xlsx"

  # Save the file as Field_info_filtered_ECO.xlsx if the first entry is organic
  write.xlsx(LBST_ECO_Area, file = file_path)
```

```
  # Print confirmation message
  cat("File saved as:", file_path, "\n")
} else {
}
```

```
####
```

```
Field_info_copy<-Field_info

ECO_data_copy<-ECO_data

Crop_Index_copy<-Crop_Index


# Convert to data.table if they are not already
setDT(Field_info_copy)
setDT(ECO_data_copy)
setDT(Crop_Index_copy)

# Initialize replacement flags
Field_info_copy[, replaced_by_ECO := FALSE]
Field_info_copy[, replaced_by_CropIndex := FALSE]

# Step 1: Replace missing mainCropYield values based on switch_variable
if (switch_variable == "yes") {
  # Merge ECO_data_copy based on directorateCropCode for ECO data replacement
  Field_info_copy <- merge(Field_info_copy, ECO_data_copy[, .(Crop_ID, `Øko-normudbytte`)],
                           by.x = "directorateCropCode", by.y = "Crop_ID",
                           all.x = TRUE)

  # Update ECO data replacement flag and apply ECO yield replacement
  Field_info_copy[, replaced_by_ECO := !is.na(`Øko-normudbytte`) & is.na(mainCropYield)]
  Field_info_copy[, mainCropYield := fifelse(is.na(mainCropYield), `Øko-normudbytte`, mainCropYield)]
}

# Step 2: Join with Crop_Index_copy for conventional crop yield replacement
Field_info_copy <- merge(Field_info_copy, Crop_Index_copy[, .(Crop_ID, `Normudbytte..udbytteenhed.ha`)],
                         by.x = "directorateCropCode", by.y = "Crop_ID",
                         all.x = TRUE)

# Update Crop_Index_copy replacement flag and apply conventional yield replacement
Field_info_copy[, replaced_by_CropIndex := is.na(mainCropYield) & !is.na(`Normudbytte..udbytteenhed.ha`)]
Field_info_copy[, mainCropYield := fifelse(is.na(mainCropYield), `Normudbytte..udbytteenhed.ha`, mainCropYield)]

# Calculate the total area for each harvest year
total_area_df <- Field_info_copy[, .(Total_Area = sum(area, na.rm = TRUE)), by = harvestYear]

# Step 3: Calculate the areas where replacements occurred
sum_replaced_df <- Field_info_copy[!is.na(mainCropYield) & (replaced_by_ECO | replaced_by_CropIndex),
                                   .(Sum_replaced_Total = sum(area, na.rm = TRUE),
                                     Sum_replaced_ECO = sum(area[replaced_by_ECO], na.rm = TRUE),
                                     Sum_replaced_Conv = sum(area[replaced_by_CropIndex], na.rm = TRUE)),
                                   by = harvestYear]

# Merge with total area data
sum_replaced_df <- merge( total_area_df, by = "harvestYear",sum_replaced_df, all.x = TRUE)

#Calculate percentages
sum_replaced_df$Percentage_replaced_total <-(sum_replaced_df$Sum_replaced_Total/sum_replaced_df$Total_Area)*100

sum_replaced_df$Percentage_replaced_ECO <-(sum_replaced_df$Sum_replaced_ECO/sum_replaced_df$Sum_replaced_Total)*100


# Create the ResultsAlt directory if it doesn't exist
dir.create("ResultsAlt", showWarnings = FALSE)

# Check the first entry of Field_info$isOrganic
if (Field_info$isOrganic[1] == 1) {
  # Define the file path for the organic file in the ResultsAlt folder
  file_path <- "ResultsAlt/Andel_normudbytte_ØKO.xlsx"

  # Save the file as Field_info_filtered_ECO.xlsx if the first entry is organic
  write.xlsx(sum_replaced_df, file = file_path)

  # Print confirmation message
  cat("File saved as:", file_path, "\n")
} else {
  # Define the file path for the conventional file in the ResultsAlt folder
  file_path <- "ResultsAlt/Andel_normudbytte_Con.xlsx"

  # Save the file as Field_info_filtered_Con.xlsx if the first entry is conventional
  write.xlsx(sum_replaced_df, file = file_path)

  # Print confirmation message
  cat("File saved as:", file_path, "\n")
}
```

```
Field_info$mainCropYield <- ifelse(
  is.na(Field_info$mainCropYield),
  # Check if the crop is organic
  ifelse(
    Field_info$isOrganic == 1,
    # Try to use organic yield data from ECO_data
    ifelse(
      !is.na(ECO_data$`Øko-normudbytte`[match(Field_info$directorateCropCode, ECO_data$Crop_ID)]),
      {
        Field_info$Sum_replaced_ECO <- Field_info$area  # Track area replaced by ECO_data
        ECO_data$`Øko-normudbytte`[match(Field_info$directorateCropCode, ECO_data$Crop_ID)]
      },
      # If there's no organic yield, use conventional data
      ifelse(
        !is.na(Crop_Index$`Normudbytte..udbytteenhed.ha`[match(Field_info$directorateCropCode, Crop_Index$Crop_ID)]),
        {
          Field_info$Sum_replaced_CropIndex <- Field_info$area  # Track area replaced by Crop_Index
          Crop_Index$`Normudbytte..udbytteenhed.ha`[match(Field_info$directorateCropCode, Crop_Index$Crop_ID)]
        },
        NA  # If both are NA, keep it NA
      )
    ),
    # If not organic, use conventional data
    ifelse(
      !is.na(Crop_Index$`Normudbytte..udbytteenhed.ha`[match(Field_info$directorateCropCode, Crop_Index$Crop_ID)]),
      {
        Field_info$Sum_replaced_CropIndex <- Field_info$area  # Track area replaced by Crop_Index
        Crop_Index$`Normudbytte..udbytteenhed.ha`[match(Field_info$directorateCropCode, Crop_Index$Crop_ID)]
      },
      NA  # If it's NA, keep it NA
    )
  ),
  Field_info$mainCropYield  # Keep existing value if it's not NA
)


if (switch_variable == "yes") {
     Field_info <- Field_info %>%
       select(-Sum_replaced_ECO, -Sum_replaced_CropIndex)
   } else {

     }







# make remaining NA values in mainCropYield = 0
Field_info[is.na(Field_info)] <- 0



### Name crop grouping ###

colnames(Crop_Index)

# Update 'Under.kategori' based on 'CropID'
Crop_Index <- Crop_Index %>%
  mutate(Under.kategori = case_when(
    Crop_ID %in% c(1, 2, 3, 4, 5, 6, 8, 19, 55, 56, 58) ~ "Vårkorn",
    Crop_ID %in% c(25, 26, 27, 30, 31, 32, 54, 35, 36, 215) ~ "Bælgsæd",
    Crop_ID %in% c(7, 18, 214, 217, 234) ~ "Blandsæd (bælgsæd+korn)",
    Crop_ID %in% c(210,211,212,213,230,701,702,703,704,705) ~ "Helsæd/grønkorn (oftest efterfulgt af en efterafgrøde))",
    Crop_ID %in% c(216,218) ~ "Majs",
    Crop_ID %in% c(149,150,151,154,155,156,157) ~ "Kartofler",
    Crop_ID %in% c(160,161,162,280,281,283) ~ "Sukkerroer mm",
    Crop_ID %in% c(21, 24,180,182,40,41,42,51,52,53,282) ~ "Andre landbrugsafgrøder (små)",
    Crop_ID %in% c(400:434) ~ "Grønsager",
    Crop_ID %in% c(9,10,11,13,14,15,16,17,57) ~ "Vinterkorn",
    Crop_ID %in% c(220,221,222,223,224,235,706,707,708,709,710,711) ~ "Vinterkorn, helsæd",
    Crop_ID %in% c(22,23) ~ "Olieplanter (raps)",
    Crop_ID %in% c(266,267,268,260,284,261,262,263,269,270,264,285,170,174,171,172,173,236,237,362,975) ~ "Græs og
kløvergræs mm (omdrift)",
    Crop_ID %in% c(101,102,103,104,116,117,105,106,107,108,109,110,111,112,113,114,115,118,120,121) ~ "Frøgræs",
    Crop_ID %in% c(122,123,124,440,448,449,496,650:668) ~ "Andre frø",
    Crop_ID %in% c(250,251,252,259,276,286,287,363,255,256,257,272,274,278,279,305,306) ~ "Permanent græs (5 år eller
ældre)",
    Crop_ID %in% c(310,319,324,327,328,334,338,342,344,345,271,486,579,247,254,312,316,317,318,321,322,326,361,567) ~
"Brak,MVJ",
    Crop_ID %in% c(501:509,489,491,492,493,494,537,538,495,540,541,542,543,544,545,547,497,548,499,563,564) ~ "Andre
(planteskolekulturer mm)",
    Crop_ID %in%
c(510,551,552,553512,513,514,515,516,517,518,519,532,570,520,521,522,523,524,525,526,527,490,528,529,530,533,534,535,539,531)
~ "Frugt og bær",
    Crop_ID %in%
c(311,482,483,484,485,487,488,565,580,575,576,577,578,581,582,583,585,586,587,589,590,591,592,593,594,596,597,598,599) ~
"Skov",
    Crop_ID %in% c(900,903,907,920,921) ~ "Natur",
    Crop_ID %in% c(943,944,945,946,963,964,965,966) ~ "Efterafgrøder (med bælgplanter)",
    Crop_ID %in% c(960,961,962,968,970) ~ "Efterafgrøder (uden bælgplanter)",
    TRUE ~ Under.kategori  # Keep existing value if no condition is met
  ))
```

```r
# Update 'Under.kategori' based on 'CropID'
Crop_Index <- Crop_Index %>%
  mutate(Hoved.kategori = case_when(
    Under.kategori %in% c("Vårkorn","Bælgsæd","Blandsæd (bælgsæd+korn)",
                          "Helsæd/grønkorn (oftest efterfulgt af en efterafgrøde))","Majs","Kartofler","Sukkerroer mm",
                          "Andre landbrugsafgrøder (små)","Grønsager") ~ "Forårssået",
    Under.kategori %in% c("Vinterkorn","Vinterkorn, helsæd","Olieplanter (raps)") ~ "Efterårssået",
    Under.kategori %in% c("Græs og kløvergræs mm (omdrift)","Frøgræs","Andre frø","Permanent græs (5 år eller ældre)",
                          "Brak,MVJ","Andre (planteskolekulturer mm)","Frugt og bær","Skov","Natur") ~ "Flerårige",
    Under.kategori %in% c("Efterafgrøder (med bælgplanter)","Efterafgrøder (uden bælgplanter)") ~ "Efterafgrøder",
    TRUE ~ Hoved.kategori  # Keep existing value if no condition is met
  ))

#write_xlsx(Crop_Index, "Crop_Index_filtered.xlsx")



match_indices <- match(Field_info$directorateCropCode, Crop_Index$Crop_ID)


matching_values_Hoved.kategori <- Crop_Index$Hoved.kategori[match_indices]

matching_values_Under.kategori <- Crop_Index$Under.kategori[match_indices]

Field_info$`Hoved.kategori`<-matching_values_Hoved.kategori

Field_info$`Under.kategori`<-matching_values_Under.kategori


Field_info <- Field_info %>%
  relocate(Hoved.kategori, .after = directorateCropName) %>%
  relocate(Under.kategori, .after = Hoved.kategori)




#Make Ntotal be SuppliedCommercialFertilizerN + SuppliedOrganicN
Field_info$NTotal<-Field_info$SuppliedCommercialFertilizerN+Field_info$SuppliedOrganicN


###### Standardize units#####


# Rename columns for consistency if needed
names(CropYield_Index)[which(names(CropYield_Index) == "cropCode")] <- "directorateCropCode"

# Merge Field_info with CropYield_Index to get the units
Field_info <- Field_info %>%
  left_join(CropYield_Index %>% select(directorateCropCode, Enhed), by = "directorateCropCode")

# Rename "Enhed" to "Unit"
Field_info <- Field_info %>%
  rename(Unit = Enhed)



#omregn sødlupin til kg baseret på om det enten er foderenheder (yield over 1000) eller i hkg (yield 1000 eller under)
Field_info$StandardYield <- ifelse(
  Field_info$directorateCropCode == 32 & Field_info$mainCropYield > 1000,
  Field_info$mainCropYield * 1.6,                    # Case 1: Multiply by 1.6 if yield is above 1000
  ifelse(
    Field_info$directorateCropCode == 32 & Field_info$mainCropYield <= 1000,
    Field_info$mainCropYield * 100,                    # Case 2: Multiply by 100 if yield is 1000 or below
    Field_info$mainCropYield                           # Keep original yield for all other cases
  )
)


Field_info$Unit[Field_info$directorateCropCode == 32] <- "kg"


# Create the new column StandardYield
Field_info <- Field_info %>%
  mutate(StandardYield = case_when(
    Unit == "HKg" ~ mainCropYield * 100,
    Unit == "Kg" ~ mainCropYield,
    Unit == "Ton" ~ mainCropYield * 1000,
    Unit == "FEN" ~ mainCropYield,
    TRUE ~ NA_real_  # Handle cases where Unit doesn't match any of the specified values
  ))



# Move the new "Unit" and "StandardYield" columns next to "mainCropYield"
Field_info <- Field_info %>%
  relocate(Unit, StandardYield, .after = mainCropYield)

### use "normtal" for when yield is not provided"

# Merge Field_info with Leftovercrop_Index to get the standard yields
Field_info <- Field_info %>%
  left_join(Leftovercrop_Index %>% select(Afgrødekode, `Normudbytte, udbytteenhed/ha`),
            by = c("directorateCropCode" = "Afgrødekode"))
```

```
#Filter based on 10x standard yield
# Perform the replacement
Field_info$StandardYield <- ifelse(
  Field_info$StandardYield >= 10 * Field_info$`Normudbytte, udbytteenhed/ha`,
  Field_info$`Normudbytte, udbytteenhed/ha`,
  Field_info$StandardYield
)




# Remove the temporary Normudbytte, udbytteenhed/ha column if no longer needed
Field_info <- Field_info %>%
  select(-`Normudbytte, udbytteenhed/ha`)



### use units from Leftovercrop_Index

Leftovercrop_Index <- Leftovercrop_Index %>%
  mutate(Udbytteenhed = ifelse(Udbytteenhed == "kg", "Kg", Udbytteenhed))

# Merge Field_info with Leftovercrop_Index to get the unit values
Field_info <- Field_info %>%
  left_join(Leftovercrop_Index %>% select(Afgrødekode, Udbytteenhed),
            by = c("directorateCropCode" = "Afgrødekode"))

# Fill in NA values in Unit with Udbytteenhed
Field_info <- Field_info %>%
  mutate(Unit = ifelse(is.na(Unit), Udbytteenhed, Unit))


# Remove the temporary Udbytteenhed column if no longer needed
Field_info <- Field_info %>%
  select(-Udbytteenhed)

# Adjust StandardYield based on Unit
Field_info$StandardYield <- ifelse(Field_info$Unit == "tus.",
                                   Field_info$mainCropYield,
                                   Field_info$StandardYield)



# make remaining NA values in mainCropYield = 0
Field_info[is.na(Field_info)] <- 0



# Create the ResultsAlt directory if it doesn't exist
if (!dir.exists("ResultsAlt")) {
  dir.create("ResultsAlt")
}

# Check the first entry of Field_info$isOrganic
if (Field_info$isOrganic[1] == 1) {
  # Save the file as Field_info_filtered_ECO.xlsx in the ResultsAlt folder if the first entry is organic
  write.xlsx(Field_info, file = "ResultsAlt/Field_info_filtered_ECO.xlsx")
} else {
  # Save the file as Field_info_filtered_Con.xlsx in the ResultsAlt folder if the first entry is conventional
  write.xlsx(Field_info, file = "ResultsAlt/Field_info_filtered_Con.xlsx")
}


# Step 1: Identify crops with non-NA ECO data for leaching
crops_with_eco_data <- ECO_data %>%
  filter(!is.na(`Udvaskning, kg N/ha, Øko100N`)) %>%
  pull(Crop_ID)

# Step 2: Calculate total and ECO-explained area per harvest year
area_summary <- Field_info %>%
  group_by(harvestYear) %>%
  summarise(
    Total_Area = sum(area, na.rm = TRUE),
    ECO_Explained_Area = sum(area[directorateCropCode %in% crops_with_eco_data], na.rm = TRUE)
  )


#Calculate percentages
area_summary$Percentage_ECO<-(area_summary$ECO_Explained_Area/area_summary$Total_Area)*100




# Check the first entry of Field_info$isOrganic
if (Field_info$isOrganic[1] == 1) {
  # Create the ResultsAlt directory if it doesn't exist
  dir.create("ResultsAlt", showWarnings = FALSE)

  # Define the file path for the Excel file in the ResultsAlt folder
  file_path <- "ResultsAlt/Andel_udvaskning_normtal_ØKO.xlsx"

  # Save the file in the ResultsAlt subfolder if the first entry is organic
```

```
  write.xlsx(area_summary, file = file_path)

  # Print confirmation message
  cat("File saved as:", file_path, "\n")
} else {
  cat("The first entry of isOrganic is not 1. File not exported.\n")
}



# Check the first entry of Field_info$isOrganic
if (Field_info$isOrganic[1] == 1) {
  # Save the file as Field_info_filtered_ECO.xlsx if the first entry is organic
  write.xlsx(area_summary, file = "Andel_udvaskning_normtal_ØKO.xlsx")
} else {

}






####################
### Calculations ###
####################

### part 1: Fertilizer ###

### part 2: Afgrøde rester ###

#### part 3: Nitrat udvaskning ###

#### part 4: Kulstofbalance ###

#### part 5: Diesel forbrug ###


####################
#### Fertilizer ####
####################


### Calculate Fertilizer_soil_emissions ###
EF_N2O <- Index_data$Value[Index_data$Info == "EF_N2O"]


Field_info$`Direct_Fertilizer_soil_emissions_N2O` <-Field_info$NTotal*Field_info$area*EF_N2O*(44/28)


### Calculate Fertilizer_ammonia_emissions ###
EF_NH3_husdyrgoedning <- Index_data$Value[Index_data$Info == "EF_NH3_husdyrgoedning"]

EF_NH3_handelsgoedning <- Index_data$Value[Index_data$Info == "EF_NH3_handelsgoedning"]


# If field is organic assume all fertilizer (NTotal) is organic, else calculate ammonia separately.
ifelse(Field_info$isOrganic==1,
Field_info$`Fertilizer_ammonia_emissions_N2O` <-Field_info$NTotal * Field_info$area * EF_NH3_husdyrgoedning*EF_N2O*(44 /
28),
Field_info$`Fertilizer_ammonia_emissions_N2O` <- (Field_info$SuppliedCommercialFertilizerN * Field_info$area *
                                                    EF_NH3_handelsgoedning*EF_N2O * (44 / 28))+
                                                          (Field_info$SuppliedOrganicN * Field_info$area
                                                           * EF_NH3_husdyrgoedning *EF_N2O* (44 / 28))
)



### Calculate Fertilizer_NOx_emissions ###
EF_NOx <- Index_data$Value[Index_data$Info == "EF_NOx"]

Field_info$`Fertilizer_NOx_emissions_N2O` <-((Field_info$NTotal*Field_info$area*EF_NOx)/(46/14))*EF_N2O*(44/28)


### calculate total N20_Fertilizer_emissions

Field_info$`N20_Fertilizer_emissions` <- Field_info$Direct_Fertilizer_soil_emissions_N2O +
                                          Field_info$Fertilizer_ammonia_emissions_N2O +
                                         Field_info$Fertilizer_NOx_emissions_N2O




### Calculate total Fertilizer_CO2e for the field###
N20_CO2e <- Index_data$Value[Index_data$Info == "Nitrox oxide to CO2-e conversion rate"]

Field_info$`Fertilizer_CO2e` <-Field_info$N20_Fertilizer_emissions*N20_CO2e


# Calculate the sum of the "Fertilizer_CO2e" column
Fertilizer_CO2e_total <- sum(Field_info$Fertilizer_CO2e, na.rm = TRUE)




# Define the file path based on the switch_variable
file_path <- if (switch_variable == "yes") {
```

```
  file.path("Mellemregninger", "Gødning_mellemregning_ECO.xlsx")
} else {
  file.path("Mellemregninger", "Gødning_mellemregning_con.xlsx")
}

# Write the dataframe to the dynamically set path
write_xlsx(Field_info, file_path)




########################
#### Afgrøde rester ####
########################


#Make Drymatter fraction

Match_FieldInfo_Leftovercrop <- match(Field_info$directorateCropCode, Leftovercrop_Index$Afgrødekode)

# Extract the relevant Tørstof fraktion values from Leftovercrop_Index
tørstof_FEN <- Leftovercrop_Index$`Tørstof fraktion af høstet produkt (FEN)`[Match_FieldInfo_Leftovercrop]
tørstof_kg <- Leftovercrop_Index$`Tørstoffraktion af høstet produkt (kg)`[Match_FieldInfo_Leftovercrop]


# Calculate Drymatter
Field_info$Drymatter <- ifelse(Field_info$Unit %in% c("FEN", "tus."),
                               Field_info$StandardYield * tørstof_FEN,
                               ifelse(Field_info$Unit %in% c("HKg", "Kg", "Ton"),
                                      Field_info$StandardYield * tørstof_kg,
                                      NA))

Field_info$'Yield_Kg_Drymatter'<-Field_info$Drymatter

Match_FieldInfo_Leftovercrop <- match(Field_info$directorateCropCode, Leftovercrop_Index$Afgrødekode)

###Define Hu###

# Extract the relevant Halmudbytte, kg ts values from Leftovercrop_Index
halmudbytte_kg_ts <- Leftovercrop_Index$`Halmudbytte, kg ts`[Match_FieldInfo_Leftovercrop]

# Replace NA values with 0
halmudbytte_kg_ts[is.na(halmudbytte_kg_ts)] <- 0

# Add the new column "Hu" to Field_info
Field_info$Hu <- halmudbytte_kg_ts

###Define HF###

# Extract the relevant Halmudbytte, kg ts values from Leftovercrop_Index
Halmfraktion_ift_udbytte_ts <-
  Leftovercrop_Index$`Halmfraktion, ift. udbytte (ts)`[Match_FieldInfo_Leftovercrop]

# Replace NA values with 0
Halmfraktion_ift_udbytte_ts[is.na(Halmfraktion_ift_udbytte_ts)] <- 0

# Add the new column "HF" to Field_info
Field_info$Hf <- Halmfraktion_ift_udbytte_ts


###Define X1###

# Extract the relevant Intercept  values from Leftovercrop_Index
Nedmuldning_faktor  <- Leftovercrop_Index$`Neduldning faktor` [Match_FieldInfo_Leftovercrop]

Nedmuldning_faktor[is.na(Nedmuldning_faktor)] <- 0


# Add the new column "X1" to Field_info
Field_info$X1 <- Nedmuldning_faktor

###Define X2###

# Extract the relevant Halmudbytte, kg ts values from Leftovercrop_Index
Janej_udbytte_nedmuldes <- Leftovercrop_Index$`Ja/nej udbytte nedmuldes`[Match_FieldInfo_Leftovercrop]

# Replace NA values with 0
Janej_udbytte_nedmuldes[is.na(Janej_udbytte_nedmuldes)] <- 0

# Add the new column "Hu" to Field_info
Field_info$X2 <- Janej_udbytte_nedmuldes


###Define S###

# Extract the relevant Slope values from Leftovercrop_Index
Slope <- Leftovercrop_Index$Slope[Match_FieldInfo_Leftovercrop]

# Add the new column "S" to Field_info
Field_info$S <- Slope


###Define I###

# Extract the relevant Intercept  values from Leftovercrop_Index
Intercept  <- Leftovercrop_Index$Intercept [Match_FieldInfo_Leftovercrop]
```

```
# Add the new column "S" to Field_info
Field_info$I <- Intercept


Field_info[is.na(Field_info)] <- 0


### Calculate Drymater above ground
Field_info$AboveGround_Main_T <- ((Field_info$Drymatter+Field_info$Hu)*Field_info$S+Field_info$I)+
                                   (Field_info$X1-1)*Field_info$Hf*Field_info$Drymatter+
                                    Field_info$X1*Field_info$Hu+
                                    Field_info$X2*Field_info$Drymatter


#Define drymatter to N factor for above ground
Field_info$AboveGround_N_factor<-
  Leftovercrop_Index$`N indhold i afgrøderester over jord, kg N/kg ts` [Match_FieldInfo_Leftovercrop]

### calculate N above ground
Field_info$AboveGround_N<-Field_info$AboveGround_Main_T*Field_info$AboveGround_N_factor


### N bellow ground

#Define above ground to bellow ground factor
Field_info$AboveBellow_Factor<-
  Leftovercrop_Index$`Forhold underjordisk biomasse til overjordisk biomasse`[Match_FieldInfo_Leftovercrop]


Field_info$BellowGround_Main_T <-(Field_info$Drymatter+Field_info$Hu+
                                  ((Field_info$Drymatter+Field_info$Hu)*
                                     Field_info$S+Field_info$I))*Field_info$AboveBellow_Factor

# Replace NA values with 0 in the specified columns
Field_info <- Field_info %>%
  mutate(
    AboveGround_Main_T = coalesce(AboveGround_Main_T, 0),
    BellowGround_Main_T = coalesce(BellowGround_Main_T, 0)
  )

### Note leftovercrop drymatter###
Field_info$MainCropLeftoverDrymatter<-Field_info$AboveGround_Main_T+Field_info$BellowGround_Main_T

#Define drymatter to N factor for bellow ground
Field_info$BellowGround_N_factor<-
  Leftovercrop_Index$`N indhold i underjordiske afgrøderester, kg N/kg ts` [Match_FieldInfo_Leftovercrop]

### calculate N above ground
Field_info$BellowGround_N<-Field_info$BellowGround_Main_T*Field_info$BellowGround_N_factor




####### Calculate total N for main crop

#Define omdrift
Field_info$Omdrift<-
  Leftovercrop_Index$`Ompløjningsfrekvens, Default (angivet som antal år mellem pløjninger)`[Match_FieldInfo_Leftovercrop]

Field_info$TotalNLeftoverMainCrop<-
((Field_info$AboveGround_N+Field_info$BellowGround_N)/Field_info$Omdrift)*Field_info$area

###Calculate Leftover crop as CO2e

EF_N2O <- Index_data$Value[Index_data$Info == "EF_N2O"]

N20_CO2e<- Index_data$Value[Index_data$Info == "Nitrox oxide to CO2-e conversion rate"]

Field_info$TotalNLeftoverMainCrop_N20<-Field_info$TotalNLeftoverMainCrop*EF_N2O*(44/28)

Field_info$TotalNLeftoverMainCrop_CO2e<-Field_info$TotalNLeftoverMainCrop_N20*N20_CO2e


# Define the file path based on the switch_variable
file_path <- if (switch_variable == "yes") {
  file.path("Mellemregninger", "Hovedafgrøderest_ECO.xlsx")
} else {
  file.path("Mellemregninger", "Hovedafgrøderest_con.xlsx")
}

# Write the dataframe to the dynamically set path
write_xlsx(Field_info, file_path)




#### Afgrøde rester catchcrop ####


Match_FieldInfoCatchCrop_Leftovercrop <- match(Field_info$catchCropDirectorateCode, Leftovercrop_Index$Afgrødekode)

Field_info$Catchcrop_Yield<-Leftovercrop_Index$`Normudbytte, udbytteenhed/ha` [Match_FieldInfoCatchCrop_Leftovercrop]

Field_info$Catchcrop_Yield_Unit<-Leftovercrop_Index$Udbytteenhed [Match_FieldInfoCatchCrop_Leftovercrop]
```

```
tørstof_FEN <- Leftovercrop_Index$`Tørstof fraktion af høstet produkt (FEN)`[Match_FieldInfoCatchCrop_Leftovercrop]
tørstof_kg <- Leftovercrop_Index$`Tørstoffraktion af høstet produkt (kg)`[Match_FieldInfoCatchCrop_Leftovercrop]


# Calculate Drymatter
Field_info$Drymatter <- ifelse(Field_info$Catchcrop_Yield_Unit %in% c("FEN"),
                               Field_info$Catchcrop_Yield * tørstof_FEN,
                               ifelse(Field_info$Catchcrop_Yield_Unit %in% c("Kg"),
                                      Field_info$Catchcrop_Yield * tørstof_kg,
                                      NA))

Field_info[is.na(Field_info)] <- 0

###Define Hu###

# Extract the relevant Halmudbytte, kg ts values from Leftovercrop_Index
halmudbytte_kg_ts <- Leftovercrop_Index$`Halmudbytte, kg ts`[Match_FieldInfoCatchCrop_Leftovercrop]



# Replace NA values with 0
halmudbytte_kg_ts[is.na(halmudbytte_kg_ts)] <- 0

# Add the new column "Hu" to Field_info
Field_info$Hu <- halmudbytte_kg_ts

###Define HF###

# Extract the relevant Halmudbytte, kg ts values from Leftovercrop_Index
Halmfraktion_ift_udbytte_ts <-
  Leftovercrop_Index$`Halmfraktion, ift. udbytte (ts)`[Match_FieldInfoCatchCrop_Leftovercrop]

# Replace NA values with 0
Halmfraktion_ift_udbytte_ts[is.na(Halmfraktion_ift_udbytte_ts)] <- 0

# Add the new column "Hu" to Field_info
Field_info$Hf <- Halmfraktion_ift_udbytte_ts



###Define X1###

# Extract the relevant Intercept  values from Leftovercrop_Index
Field_info$X1  <- 0


###Define X2###

# Extract the relevant Halmudbytte, kg ts values from Leftovercrop_Index
Janej_udbytte_nedmuldes <- Leftovercrop_Index$`Ja/nej udbytte nedmuldes`[Match_FieldInfoCatchCrop_Leftovercrop]

# Replace NA values with 0
Janej_udbytte_nedmuldes[is.na(Janej_udbytte_nedmuldes)] <- 0

# Add the new column "Hu" to Field_info
Field_info$X2 <- Janej_udbytte_nedmuldes


###Define S###

# Extract the relevant Slope values from Leftovercrop_Index
Field_info$S  <- Leftovercrop_Index$Slope[Match_FieldInfoCatchCrop_Leftovercrop]



###Define I###

# Extract the relevant Intercept  values from Leftovercrop_Index
Field_info$I  <- Leftovercrop_Index$Intercept [Match_FieldInfoCatchCrop_Leftovercrop]




### Calculate Drymater above ground
Field_info$AboveGround_Main_T <- ((Field_info$Drymatter+Field_info$Hu)*Field_info$S+Field_info$I)+
  (Field_info$X1-1)*Field_info$Hf*Field_info$Drymatter+
  Field_info$X1*Field_info$Hu+
  Field_info$X2*Field_info$Drymatter


#Define drymatter to N factor for above ground
Field_info$AboveGround_N_factor<-
  Leftovercrop_Index$`N indhold i afgrøderester over jord, kg N/kg ts` [Match_FieldInfoCatchCrop_Leftovercrop]

### calculate N above ground
Field_info$AboveGround_N<-Field_info$AboveGround_Main_T*Field_info$AboveGround_N_factor


### N bellow ground

#Define above ground to bellow ground factor
Field_info$AboveBellow_Factor<-
  Leftovercrop_Index$`Forhold underjordisk biomasse til overjordisk biomasse` [Match_FieldInfoCatchCrop_Leftovercrop]
```

```
Field_info$BellowGround_Main_T <-(Field_info$Drymatter+Field_info$Hu+
                                   ((Field_info$Drymatter+Field_info$Hu)*
                                     Field_info$S+Field_info$I))*Field_info$AboveBellow_Factor

# Replace NA values with 0 in the specified columns
Field_info <- Field_info %>%
  mutate(
    AboveGround_Main_T = coalesce(AboveGround_Main_T, 0),
    BellowGround_Main_T = coalesce(BellowGround_Main_T, 0)
  )

### Note leftovercrop drymatter###
Field_info$CatchCropLeftoverDrymatter<-Field_info$AboveGround_Main_T+Field_info$BellowGround_Main_T


#Define drymatter to N factor for bellow ground
Field_info$BellowGround_N_factor<-
  Leftovercrop_Index$`N indhold i underjordiske afgrøderester, kg N/kg ts` [Match_FieldInfoCatchCrop_Leftovercrop]

### calculate N above ground
Field_info$BellowGround_N<-Field_info$BellowGround_Main_T*Field_info$BellowGround_N_factor


### Calculate total N for catch crop

#Define omdrift
Field_info$Omdrift<-
  Leftovercrop_Index$`Ompløjningsfrekvens, Default (angivet som antal år mellem
pløjninger)`[Match_FieldInfoCatchCrop_Leftovercrop]

Field_info$TotalNLeftoverCatchCrop<-
((Field_info$AboveGround_N+Field_info$BellowGround_N)/Field_info$Omdrift)*Field_info$area

###Calculate Leftover crop as CO2e

Field_info$TotalNLeftoverCatchCrop_N20<-Field_info$TotalNLeftoverCatchCrop*EF_N2O*(44/28)

Field_info$TotalNLeftoverCatchCrop_CO2e<-Field_info$TotalNLeftoverCatchCrop_N20*N20_CO2e




###Calculate total drymatter from leftover crop matter###
Field_info$TotalLeftoverCropDrymatter<-Field_info$MainCropLeftoverDrymatter+Field_info$CatchCropLeftoverDrymatter




# Replace NA values with 0 in the specified columns
Field_info <- Field_info %>%
  mutate(
    TotalNLeftoverMainCrop_N20 = coalesce(TotalNLeftoverMainCrop_N20, 0),
    TotalNLeftoverCatchCrop_N20 = coalesce(TotalNLeftoverCatchCrop_N20, 0),
    TotalNLeftoverMainCrop_CO2e = coalesce(TotalNLeftoverMainCrop_CO2e, 0),
    TotalNLeftoverCatchCrop_CO2e = coalesce(TotalNLeftoverCatchCrop_CO2e, 0)
    )

Field_info$TotalNLeftoverCrop_N20<-Field_info$TotalNLeftoverMainCrop_N20+Field_info$TotalNLeftoverCatchCrop_N20

Field_info$TotalNLeftoverCrop_CO2e<-Field_info$TotalNLeftoverMainCrop_CO2e+Field_info$TotalNLeftoverCatchCrop_CO2e




# Define the file path based on the switch_variable
file_path <- if (switch_variable == "yes") {
  file.path("Mellemregninger", "Efterafgrøderest_mellemregning_ECO.xlsx")
} else {
  file.path("Mellemregninger", "Efterafgrøderest_mellemregning_con.xlsx")
}

# Write the dataframe to the dynamically set path
write_xlsx(Field_info, file_path)



# Remove the specified columns using select
Field_info <- Field_info %>%
  select(-Drymatter, -Hu, -Hf, -X1, -X2, -S, -I,
         -AboveGround_Main_T,-AboveGround_N_factor,
         -AboveGround_N, -AboveBellow_Factor, -BellowGround_Main_T,
         -BellowGround_N_factor, -BellowGround_N,
         -Omdrift#,
         #-TotalNLeftoverMainCrop,-TotalNLeftoverCatchCrop,
  )



#write.csv(Field_info, "temp_results.csv", row.names = FALSE)



#########################
#### Kulstofbalance ####
#########################
```

```
#Match
Match_FieldInfo_Leftovercrop <- match(Field_info$directorateCropCode, Leftovercrop_Index$Afgrødekode)

C_frac<- Index_data$Value[Index_data$Info == "Cfrac"]

Fhus<-Index_data$Value[Index_data$Info == "Fhus"]


Field_info$`C_organic`<-Field_info$SuppliedOrganicN*Fhus


Field_info$`C_Maincrop`<-Field_info$MainCropLeftoverDrymatter*C_frac


# Replace NA values with 0 in the specified column of Leftovercrop_Index
Leftovercrop_Index$`Ja/nej - skal relativeres til DK gennemsnit (kulstof)`[is.na(Leftovercrop_Index$`Ja/nej - skal
relativeres til DK gennemsnit (kulstof)`)] <- 0

matching_values_crop.averageC<-
  Leftovercrop_Index$`Ja/nej - skal relativeres til DK gennemsnit (kulstof)`[Match_FieldInfo_Leftovercrop]

Udk<-Index_data$Value[Index_data$Info == "Udk"]

Field_info$`Carbon_BalanceMainCrop` <- ifelse(matching_values_crop.averageC == "1",
                                       ((Field_info$C_Maincrop+Field_info$C_organic-Udk)*Field_info$area*
                                         (44/12)*0.097*(-1)),
                                       ifelse(matching_values_crop.averageC == "0",
                                              (Field_info$C_Maincrop+Field_info$C_organic)*Field_info$area*
                                                (44/12)*0.097*(-1),0
                                       )
)



Field_info$`C_Catchcrop`<-Field_info$CatchCropLeftoverDrymatter*C_frac

Match_FieldInfoCatchCrop_Leftovercrop <- match(Field_info$catchCropDirectorateCode, Leftovercrop_Index$Afgrødekode)

matching_values_crop.averageC_Catchcrop<-
  Leftovercrop_Index$`Ja/nej - skal relativeres til DK gennemsnit (kulstof)`[Match_FieldInfo_Leftovercrop]

# Calculate Carbon_BalanceCatchCrop only if C_Catchcrop > 0
Field_info$Carbon_BalanceCatchCrop <- ifelse(Field_info$C_Catchcrop > 0,
                                             ifelse(matching_values_crop.averageC_Catchcrop == "1",
                                                    ((Field_info$C_Catchcrop) * Field_info$area *
                                                       (44 / 12) * 0.097 * (-1)),
                                                    ifelse(matching_values_crop.averageC_Catchcrop == "0",
                                                           Field_info$C_Catchcrop * Field_info$area *
                                                             (44 / 12) * 0.097 * (-1), 0
                                                    )),
                                             0)



Field_info$TotalCarbon_Balance_CO2e<-Field_info$`Carbon_BalanceMainCrop`+Field_info$`Carbon_BalanceCatchCrop`


sum(Field_info$TotalCarbon_Balance_CO2e)

# Define the file path based on the switch_variable
file_path <- if (switch_variable == "yes") {
  file.path("Mellemregninger", "Kulstofbalance_mellemregning_ECO.xlsx")
} else {
  file.path("Mellemregninger", "Kulstofbalance_mellemregning_con.xlsx")
}

# Write the dataframe to the dynamically set path
write_xlsx(Field_info, file_path)




Field_info <- Field_info %>%
  select(-MainCropLeftoverDrymatter,
         -CatchCropLeftoverDrymatter
  )



############################
### Kvælstof udvaskning ###
############################


# Main Crop T value calculation
T_main_crop <- ifelse(
  Field_info$isOrganic == 1,
  # For organic fields, prioritize ECO_data, then fallback to Crop_Index if needed
  ifelse(
    !is.na(ECO_data$`Udvaskning, kg N/ha, Øko100N`[match(Field_info$directorateCropCode, ECO_data$Crop_ID)]),
    ECO_data$`Udvaskning, kg N/ha, Øko100N`[match(Field_info$directorateCropCode, ECO_data$Crop_ID)],
    Crop_Index$Nitratudvaskning.kg.N.ha[match(Field_info$directorateCropCode, Crop_Index$Crop_ID)]
  ),
  # For non-organic fields, use the value from Crop_Index
```

```
  Crop_Index$Nitratudvaskning.kg.N.ha[match(Field_info$directorateCropCode, Crop_Index$Crop_ID)]
)

# Catch Crop T value calculation
T_catch_crop <- ifelse(
  Field_info$isOrganic == 1,
  # For organic fields, prioritize ECO_data, then fallback to Crop_Index if needed
  ifelse(
    !is.na(ECO_data$`Udvaskning, kg N/ha, Øko100N`[match(Field_info$catchCropDirectorateCode, ECO_data$Crop_ID)]),
    ECO_data$`Udvaskning, kg N/ha, Øko100N`[match(Field_info$catchCropDirectorateCode, ECO_data$Crop_ID)],
    Crop_Index$Nitratudvaskning.kg.N.ha[match(Field_info$catchCropDirectorateCode, Crop_Index$Crop_ID)]
  ),
  # For non-organic fields, use the value from Crop_Index
  Crop_Index$Nitratudvaskning.kg.N.ha[match(Field_info$catchCropDirectorateCode, Crop_Index$Crop_ID)]
)

# Main Crop Nitrate Leaching Calculation
Field_info$N20_Nitrateleaching_emissions_maincrop <- T_main_crop * 0.0075 * (44 / 28) * Field_info$area

# Catch Crop Nitrate Leaching Calculation
Field_info$N20_Nitrateleaching_emissions_catchcrop <- T_catch_crop * 0.0075 * (44 / 28) * Field_info$area




Field_info[is.na(Field_info)] <- 0

Field_info$`N20_Nitrateleaching_emissions`<-
  Field_info$N20_Nitrateleaching_emissions_maincrop+Field_info$N20_Nitrateleaching_emissions_catchcrop

N20_CO2e <- Index_data$Value[Index_data$Info == "Nitrox oxide to CO2-e conversion rate"]

Field_info$`Nitrateleaching_CO2e` <-Field_info$N20_Nitrateleaching_emissions*N20_CO2e




# Define the file path based on the switch_variable
file_path <- if (switch_variable == "yes") {
  file.path("Mellemregninger", "Nitratudvaskning_mellemregning_ECO.xlsx")
} else {
  file.path("Mellemregninger", "Nitratudvaskning_mellemregning_con.xlsx")
}

# Write the dataframe to the dynamically set path
write_xlsx(Field_info, file_path)




#########################
#### Diesel forbrug ####
#########################

# extract diesel emissionsfactor from Index_data
Diesel_CO2e <- Index_data$Value[Index_data$Info == "DieselCO2e"]


match_indices_maincrop <- match(Field_info$directorateCropCode, Crop_Index$Crop_ID)

match_indices_catchcrop <- match(Field_info$catchCropDirectorateCode, Crop_Index$Crop_ID)

# Extract the corresponding match_indices
matching_values_diesel_maincrop <- Crop_Index$Diesel.forbrug[match_indices_maincrop]

matching_values_diesel_catchcrop <- Crop_Index$Diesel.forbrug[match_indices_catchcrop]



#Calculate Maincrop Diesel CO2e
Field_info$`Diesel_Maincrop_CO2e` <-(matching_values_diesel_maincrop*Field_info$area)*Diesel_CO2e

# Replace NA with 0 in the 'Diesel_Maincrop_CO2e' column
Field_info$Diesel_Maincrop_CO2e[is.na(Field_info$Diesel_Maincrop_CO2e)] <- 0


Field_info$`Diesel_catchcrop_CO2e` <-(matching_values_diesel_catchcrop*Field_info$area)*Diesel_CO2e

# Replace NA with 0 in the 'Diesel_Maincrop_CO2e' column
Field_info$Diesel_catchcrop_CO2e[is.na(Field_info$Diesel_catchcrop_CO2e)] <- 0


Field_info$`DieselCO2e` <-(Field_info$Diesel_Maincrop_CO2e+Field_info$Diesel_catchcrop_CO2e)




# Define the file path based on the switch_variable
file_path <- if (switch_variable == "yes") {
  file.path("Mellemregninger", "Dieselforbrug_mellemregning_ECO.xlsx")
} else {
  file.path("Mellemregninger", "Dieselforbrug_mellemregning_con.xlsx")
}

# Write the dataframe to the dynamically set path
write_xlsx(Field_info, file_path)
```

```
##########################
#### Samlet udledning ####
##########################

Field_info[is.na(Field_info)] <- 0


Field_info$TotalN20<-(Field_info$N20_Fertilizer_emissions+Field_info$TotalNLeftoverCrop_N20+
                        Field_info$N20_Nitrateleaching_emissions)

TotalN20_CO2e<- Index_data$Value[Index_data$Info == "Nitrox oxide to CO2-e conversion rate"]

Field_info$TotalN20_CO2e<-(Field_info$TotalN20*N20_CO2e)

Field_info$TotalCO2e<-(Field_info$Fertilizer_CO2e+Field_info$TotalNLeftoverCrop_CO2e+Field_info$Nitrateleaching_CO2e+
                        Field_info$TotalCarbon_Balance_CO2e+Field_info$DieselCO2e)


Field_info$TotalCO2eWithoutCarbonBalance<-(Field_info$TotalCO2e-
                                           Field_info$TotalCarbon_Balance_CO2e)






########################
#### Under/over N65 ####
########################

Field_info_N65<- Field_info

# Step 1: Calculate total N and total area per farm per year
farm_year_summary <- Field_info_N65 %>%
  group_by(farmid, harvestYear) %>%
  summarise(
    Total_N = sum(NTotal * area, na.rm = TRUE),  # Sum of NTotal * area per farm per year
    Total_Area = sum(area, na.rm = TRUE),       # Sum of area per farm per year
    .groups = "drop"
  )

# Step 2: Join Total_N and Total_Area back to Field_info and calculate FarmUdnyttetN
Field_info_N65 <- Field_info_N65 %>%
  left_join(farm_year_summary, by = c("farmid", "harvestYear")) %>%
  mutate(FarmUdnyttetN = (Total_N / Total_Area) * 0.7)  # Calculate and scale FarmUdnyttetN

# View the updated data frame
print(Field_info_N65)


# Create Field_info$N65_N107 based on the conditions
#If N under 65 = 1, if between 65 and 107 =2 if over 107=0
Field_info_N65 <- Field_info_N65 %>%
  mutate(
    N65_N107 = case_when(
      FarmUdnyttetN <= 65 ~ 1,
      FarmUdnyttetN > 65 & FarmUdnyttetN <= 107 ~ 2,
      FarmUdnyttetN > 107 ~ 0
    )
  )

# Reorder columns to place FarmUdnyttetN and N65_N107 right after Total_N
Field_info_N65 <- Field_info_N65 %>%
  relocate(FarmUdnyttetN, N65_N107, .after = NTotal)

#Remove Total_N to not confuse with NTotal later
Field_info_N65$Total_N <- NULL


#Remove fields which assosiated farm has a harmony area of 0
Field_info_N65 <- Field_info_N65[Field_info_N65$harmonyArea != 0, ]



# Splitting the data into two data frames based on if the are N65 or not
Field_info_N65_under <- Field_info_N65[Field_info_N65$FarmUdnyttetN <= 65, ]
Field_info_N65_over <- Field_info_N65[Field_info_N65$FarmUdnyttetN > 65, ]




# Check if the first value of isOrganic in Field_info_N65_under is 1
if (Field_info_N65_under$isOrganic[1] == 1) {
  # Create the directory if it doesn't exist
  dir.create("ResultsAlt", showWarnings = FALSE)

  # Define file paths
  file_path_under <- "ResultsAlt/Field_info_N65_under.xlsx"
  file_path_over <- "ResultsAlt/Field_info_N65_over.xlsx"

  # Export the data frames to Excel
  write_xlsx(Field_info_N65_under, file_path_under)
  write_xlsx(Field_info_N65_over, file_path_over)

  # Print confirmation
  cat("Data exported successfully to:\n",
```

```
      file_path_under, "\n",
      file_path_over, "\n")
} else {
  cat("The first value of isOrganic is not 1. Data not exported.\n")
}



#######################
###### Eksporter ######
#######################




#### På mark plan


# Check the first value of Field_info$isOrganic
folder_name <- ifelse(Field_info$isOrganic[1] == 1, "Results_Øko",
                      "Results_Konventionel")

# Create the folder if it doesn't exist
if (!dir.exists(folder_name)) {
  dir.create(folder_name)
}

# Create the file path
file_path <- paste0(folder_name, "/Field_results.xlsx")

# Export the Field_info dataframe as an xlsx file
write.xlsx(Field_info, file = file_path)



#### På gårds plan



Farm_info <- Field_info %>%
  group_by(farmid, harvestYear) %>%
  summarize(
    numberfields = n_distinct(fieldId),
    numbercrops = n_distinct(directorateCropCode),
    Yield_Kg_Drymatter = sum(Yield_Kg_Drymatter*area, na.rm = TRUE),
    NTotal = sum(NTotal*area, na.rm = TRUE),
    SuppliedOrganicN= sum(SuppliedOrganicN*area, na.rm = TRUE),
    SuppliedCommercialFertilizerN= sum(SuppliedCommercialFertilizerN*area, na.rm = TRUE),
    area = sum(area, na.rm = TRUE),
    Fertilizer_N2O = sum(N20_Fertilizer_emissions, na.rm = TRUE),
    Fertilizer_CO2e = sum(Fertilizer_CO2e, na.rm = TRUE),
    Direct_Fertilizer_soil_emissions_N2O = sum(Direct_Fertilizer_soil_emissions_N2O, na.rm = TRUE),
    Fertilizer_ammonia_emissions_N2O = sum(Fertilizer_ammonia_emissions_N2O, na.rm = TRUE),
    Fertilizer_NOx_emissions_N2O = sum(Fertilizer_NOx_emissions_N2O, na.rm = TRUE),
    Leftovercrop_N2O = sum(TotalNLeftoverCrop_N20, na.rm = TRUE),
    LeftoverCrop_CO2e = sum(TotalNLeftoverCrop_CO2e, na.rm = TRUE),
    Nitrateleaching_N2O = sum(N20_Nitrateleaching_emissions, na.rm = TRUE),
    Nitrateleaching_CO2e = sum(Nitrateleaching_CO2e, na.rm = TRUE),
    Carbon_Balance_CO2e = sum(TotalCarbon_Balance_CO2e, na.rm = TRUE),
    DieselCO2e = sum(DieselCO2e, na.rm = TRUE),
    TotalN20 = sum(TotalN20, na.rm = TRUE),
    TotalN20_CO2e = sum(TotalN20_CO2e, na.rm = TRUE),
    TotalCO2e = sum(TotalCO2e, na.rm = TRUE),
    TotalCO2eWithoutCarbonBalance= sum(TotalCO2e, na.rm = TRUE)-sum(Carbon_Balance_CO2e, na.rm = TRUE)
  ) %>%
  ungroup()



# Check the first value of Field_info$isOrganic
folder_name <- ifelse(Field_info$isOrganic[1] == 1, "Results_Øko", "Results_Konventionel")

# Create the folder if it doesn't exist
if (!dir.exists(folder_name)) {
  dir.create(folder_name)
}

# Create the file path
file_path <- paste0(folder_name, "/Farm_results.xlsx")

# Export the Field_info dataframe as an xlsx file
write.xlsx(Farm_info, file = file_path)



#### På afgrøde plan


Crop_info <- Field_info %>%
  group_by(harvestYear, Under.kategori, Hoved.kategori) %>%
  summarize(
    numberfields = n_distinct(fieldId),
    numbercrops = n_distinct(directorateCropCode),
    Yield_Kg_Drymatter = sum(Yield_Kg_Drymatter, na.rm = TRUE),
    NTotal = sum(NTotal, na.rm = TRUE),
```

```
    SuppliedOrganicN= sum(SuppliedOrganicN, na.rm = TRUE),
    SuppliedCommercialFertilizerN= sum(SuppliedCommercialFertilizerN, na.rm = TRUE),
    area = sum(area, na.rm = TRUE),
    Fertilizer_N2O = sum(N2O_Fertilizer_emissions, na.rm = TRUE),
    Fertilizer_CO2e = sum(Fertilizer_CO2e, na.rm = TRUE),
    Direct_Fertilizer_soil_emissions_N2O = sum(Direct_Fertilizer_soil_emissions_N2O, na.rm = TRUE),
    Fertilizer_ammonia_emissions_N2O = sum(Fertilizer_ammonia_emissions_N2O, na.rm = TRUE),
    Fertilizer_NOx_emissions_N2O = sum(Fertilizer_NOx_emissions_N2O, na.rm = TRUE),
    Leftovercrop_N2O = sum(TotalNLeftoverCrop_N2O, na.rm = TRUE),
    LeftoverCrop_CO2e = sum(TotalNLeftoverCrop_CO2e, na.rm = TRUE),
    Nitrateleaching_N2O = sum(N2O_Nitrateleaching_emissions, na.rm = TRUE),
    Nitrateleaching_CO2e = sum(Nitrateleaching_CO2e, na.rm = TRUE),
    Carbon_Balance_CO2e = sum(TotalCarbon_Balance_CO2e, na.rm = TRUE),
    DieselCO2e = sum(DieselCO2e, na.rm = TRUE),
    TotalN2O = sum(TotalN2O, na.rm = TRUE),
    TotalN2O_CO2e = sum(TotalN2O_CO2e, na.rm = TRUE),
    TotalCO2eWithoutCarbonBalance= sum(TotalCO2e-DieselCO2e, na.rm = TRUE),
    TotalCO2e = sum(TotalCO2e, na.rm = TRUE)         # Total of TotalCO2e
  ) %>%
  ungroup()


# Check the first value of Field_info$isOrganic
folder_name <- ifelse(Field_info$isOrganic[1] == 1, "Results_Øko", "Results_Konventionel")

# Create the folder if it doesn't exist
if (!dir.exists(folder_name)) {
  dir.create(folder_name)
}

# Create the file path
file_path <- paste0(folder_name, "/Crop_results.xlsx")

# Export the Field_info dataframe as an xlsx file
write.xlsx(Crop_info, file = file_path)


####################
### Print script ###
####################


# Create ResultsAlt folder if it doesn't exist
dir.create("ResultsAlt", showWarnings = FALSE)

# Get the current working directory
current_dir <- getwd()

# Specify the path to the R script
script_path <- file.path(current_dir, "MainScript.R")

# Output path for the PDF
output_pdf <- file.path(current_dir, "ResultsAlt", "script_print.pdf")

# Write the content of the script to a text file
text_path <- file.path(current_dir, "ResultsAlt", "script_print.txt")
file.copy(script_path, text_path, overwrite = TRUE)
```