Re-Livestock

RESILIENT FARMING SYSTEMS

Practical work: Editing raw data from sniffer

Oscar González-Recio and Coralia Manzanilla-Pech

WAGENINGEN UNIVERSITY & RESEARCH

INIA CSIC

Zaragoza, February 25th, 2025

Overview: First part-Connecting

1. Starting with Jupyter Notebook in Annuna (WUR)
2. Opening Github to see the files
3. Cloning files from Github to Jupyter Notebook
4. Ready for practicals

Preliminars: How to connect to the WUR server

https://notebook.anunna.wur.nl

Github

Oscar's: https://ogrecio.github.io/RelivestockMethaneCourse/

Coralia's:https://github.com/cmanzanillap

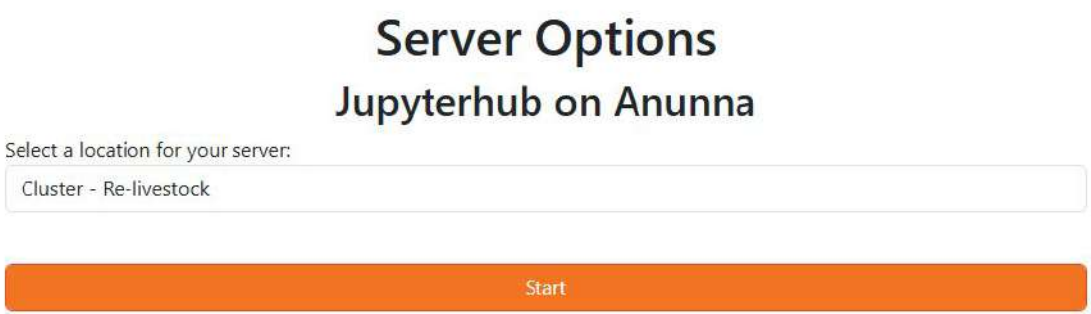
Ester's: https://github.com/estermt/sniffer_Data_management

Environments

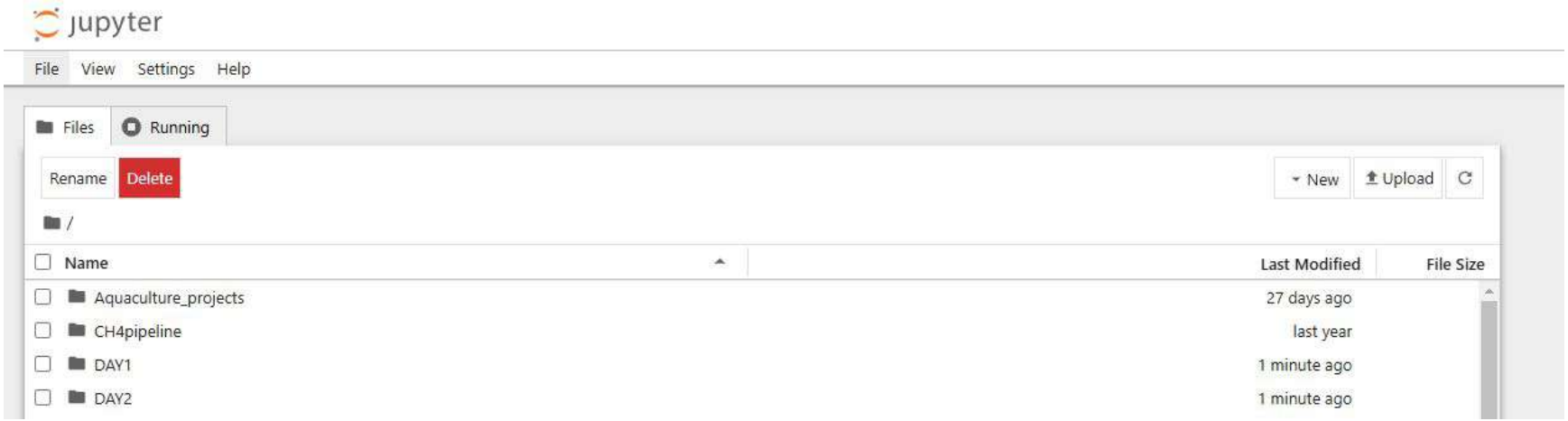
Practical environment

https://notebook.anunna.wur.nl

test.ipynb



How it looks


jupyter
File
View
Settings
Help
Files
Running
Rename
Delete
New
Upload
/
Name
Last Modified
File Size
Aquaculture_projects
27 days ago
CH4pipeline
last year
DAY1
1 minute ago
DAY2
1 minute ago

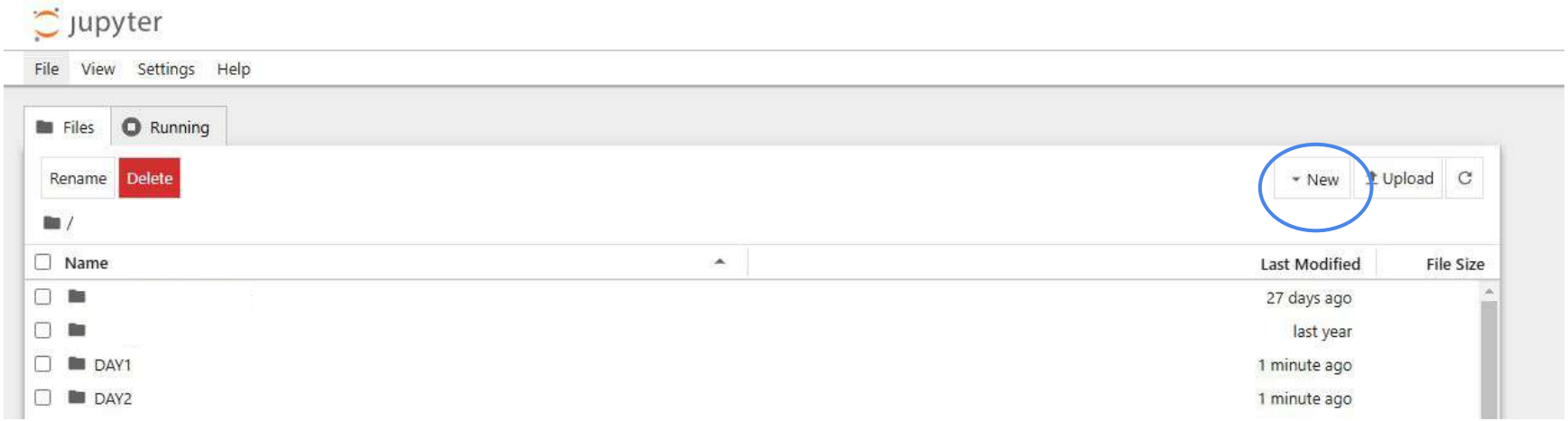
How to clone files/folders from Github to Jupyter

1. Open Jupyter Notebook online in your web browser.
2. Click on the “New” button in the top right corner of the screen.
3. Select “Terminal” from the dropdown menu.
4. In the terminal window, type the following command:

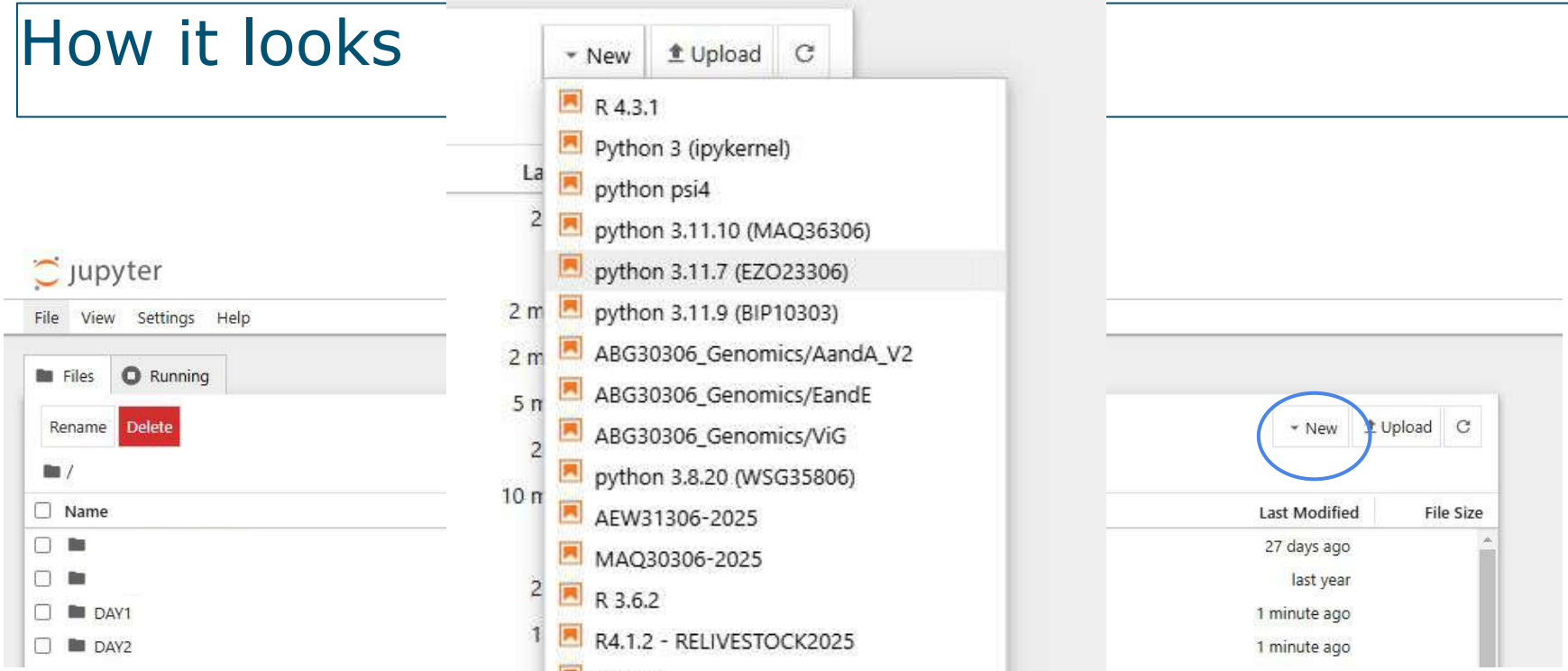
git clone https://github.com/ogrecio/RelivestockMethaneCourse

How it looks

RelivestockMetaneCourse/Day1



How it looks


jupyter
File View Settings Help
Files Running
Rename Delete
/
Name
DAY1
DAY2
New Upload
R 4.3.1
Python 3 (ipykernel)
python psi4
python 3.11.10 (MAQ36306)
python 3.11.7 (EZO23306)
python 3.11.9 (BIP10303)
ABG30306_Genomics/AandA_V2
ABG30306_Genomics/EandE
ABG30306_Genomics/ViG
python 3.8.20 (WSG35806)
AEW31306-2025
MAQ30306-2025
R 3.6.2
R4.1.2 - RELIVESTOCK2025
R 4.4.2
Terminal
Console
New File
New Folder
3 months ago
New Upload
Last Modified
File Size
27 days ago
last year
1 minute ago
1 minute ago

How it looks Terminal

type cd Day1

useful commands Linux

ls (to see files) ls -d */ (to see directories)

less (to see the files inside) less -S (useful for genotypes) q (to quit from less)

head (head of the file)

wc -l file (number of rows)

awk '{print NF; exit}' file (number of columns)

cd folder (change directory) cd .. (comeback to the previous level)

rm -rf directory or rm file

Files in RelivestockMethaneCourse/Day1 folder

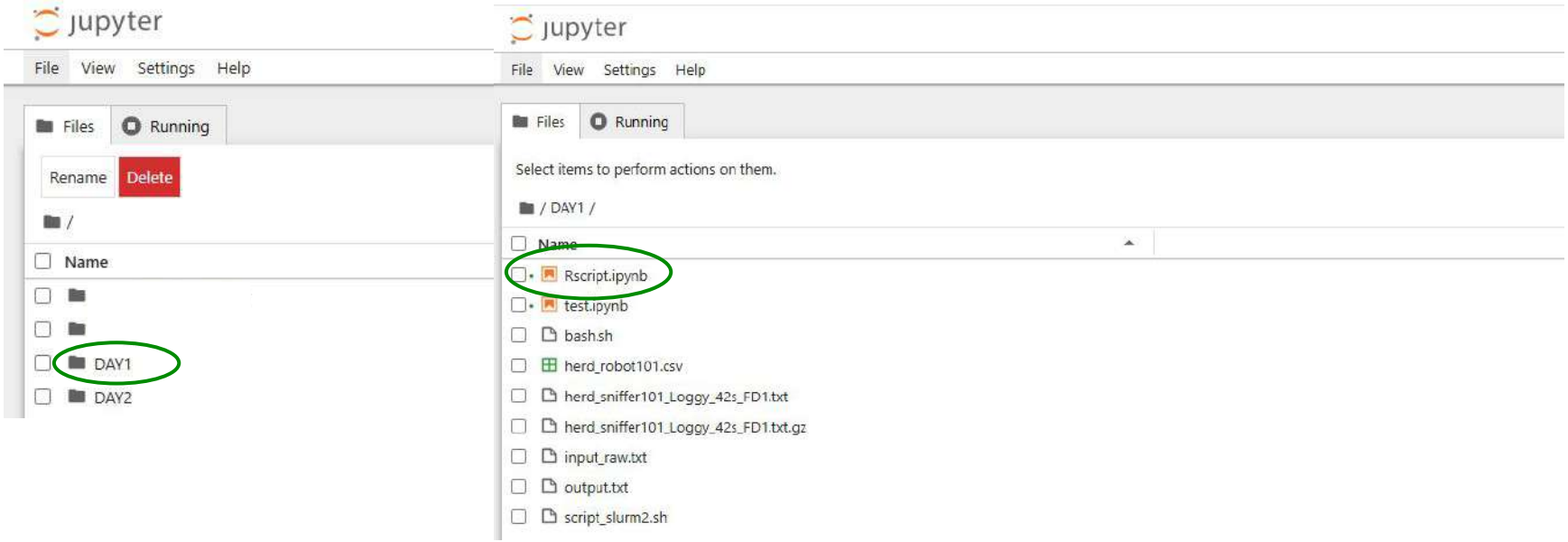
1. Database to read in R: input_raw.txt
2. Jupyter R Notebook: Rscript.ipynb, merge_sniffer_testday.ipynb
3. Databases to run Java program: herd_robot101.csv
   herd_sniffer101_Loggy_42s_FD1.txt
4. Script to run in a bash file: bash.sh

Overview: Second part- Practical in R

1. Installing packages
2. Read data input_raw.txt
3. Get familiar with the data (check number records, number cows, duration, min max etc)
4. Create event (visit), plotting events to see how data looks
5. Background approaches
6. Calculate different phenotypes
7. Calculate correlations between them

How it looks

RelivestockMetaneCourse/Day1



Raw CH_4 data from sniffer

jupyter output.txt Last Checkpoint: 2 minutes ago

File Edit View Settings Help

```
cow,date,ch4,co2
59,2022/18/02 00:00:01,0.042,0.512
59,2022/18/02 00:00:02,0.042,0.52
59,2022/18/02 00:00:03,0.041,0.521
59,2022/18/02 00:00:04,0.041,0.526
59,2022/18/02 00:00:05,0.041,0.527
59,2022/18/02 00:00:06,0.041,0.529
59,2022/18/02 00:00:07,0.041,0.533
59,2022/18/02 00:00:08,0.041,0.534
59,2022/18/02 00:00:09,0.04,0.532
59,2022/18/02 00:00:10,0.04,0.534
59,2022/18/02 00:00:11,0.04,0.538
59,2022/18/02 00:00:12,0.04,0.542
59,2022/18/02 00:00:13,0.04,0.542
59,2022/18/02 00:00:14,0.04,0.544
59,2022/18/02 00:00:15,0.04,0.545
59,2022/18/02 00:00:16,0.04,0.549
59,2022/18/02 00:00:17,0.04,0.551
59,2022/18/02 00:00:18,0.04,0.549
59,2022/18/02 00:00:19,0.04,0.549
59,2022/18/02 00:00:20,0.04,0.547
59,2022/18/02 00:00:21,0.041,0.549
59,2022/18/02 00:00:22,0.042,0.549
```

columns:

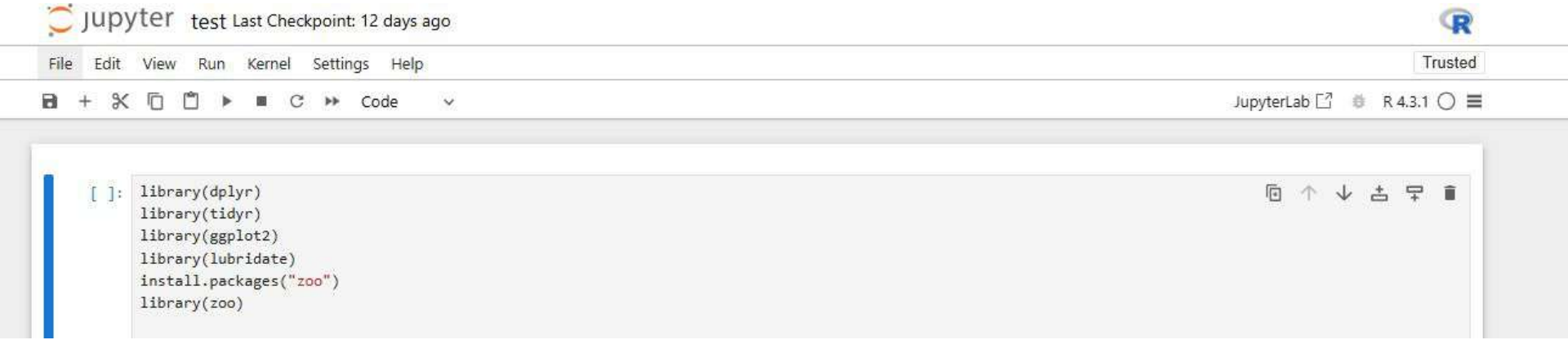
cow

date

ch4

co2

Call the following packages



Read the output.txt file

```
rawCH4 <- read.table("input_raw.txt", sep = ",", stringsAsFactors = F, header = T)
colnames(rawCH4)
head(rawCH4)
dim(rawCH4)
summary(rawCH4)
```

Multiplicate to get ppm & format date and time

```r
# Multiply to get ppm
rawCH4$ch4ppm <- -rawCH4$ch4 * 10000
rawCH4$co2ppm <- -rawCH4$co2 * 10000

# Parse datetime
rawCH4$datetime <- as.POSIXct(rawCH4$date, format = "%Y/%d/%m %H:%M:%S")
rawCH4$date <- as.Date.character(rawCH4$date, "%Y/%d/%m")
head(rawCH4)
```

Create an event per animal

```
#create an event per entrance to the AMS per day per animal
event_ch4 <- rawCH4 %>%
  select(ch4ppm, co2ppm, cow, date, datetime) %>%
  mutate(eventID = 1 + cumsum(cow != lag(cow, default = first(cow))))

hist(event_ch4$eventID)
head(event_ch4)
```

Get familiar with the data

```
num_cows <- event_ch4 %>%
  summarise(unique_cows = n_distinct(cow))
 print(num_cows)

cow_counts <- rawCH4 %>%
  group_by(cow) %>%
  summarise(count = n())
head(cow_counts)
```

Calculating duration of the visit

```
########## calculating duration of the visit ##############
event_duration <- event_ch4 %>%
  group_by(eventID) %>%
  summarize(
    start_time = min(datetime),
    end_time = max(datetime),
    lenght = as.numeric(difftime(max(datetime), min(datetime), units = "secs"))
  )
head(event_duration)
summary(event_duration)
hist(event_duration$lenght)
```

Calculating number of visits per cow

```
#Calculating how many events (visit) per cow

event_counts <- event_ch4 %>%
  group_by(cow) %>%
  summarise(unique_events = n_distinct(eventID))

# View the result
print(event_counts)
```

Calculating background Approach 1

```
############ calculating the average of the lowest 3 or 5 measurements per visit to use it as background ########
event_5low <- event_ch4 %>%
  group_by(eventID) %>%
  arrange(ch4) %>%
  slice_head(n = 5) %>%
  summarise(avg_ch4 = mean(ch4, na.rm = TRUE))
head(event_5low)
summary(event_5low)
```

```
event_3low <- event_ch4 %>%
  group_by(eventID) %>%
  arrange(ch4) %>%
  slice_head(n = 3) %>%
  summarise(avg_ch4 = mean(ch4, na.rm = TRUE))
head(event_3low)
summary(event_3low)
```

Calculating background Approach 2

```
############ calculating the 0.001 quantile per visit to use it as background ############
event_quant <- event_ch4 %>%
  group_by(eventID) %>%
  summarise(ch4_quantile_0_001 = quantile(ch4, probs = 0.001, na.rm = TRUE))
head(event_quant)
summary(event_quant)
```

Calculating average concentration for CH4 - CO2

```
################ calculating the mean for ch4 and co2 per visit #################
event_means <- event_ch4 %>%
  group_by(eventID) %>%
  summarise(
    mean_ch4 = mean(ch4, na.rm = TRUE),
    mean_co2 = mean(co2, na.rm = TRUE)
  )
head(event_means)
summary(event_means)
```

Now for the cut visit length

```
############## calculating the mean for ch4 only for 60-300 sec per visit ######## losing one event
event_cut <- event_ch4 %>%
  # Trim any spaces from the TimeStamp column
  mutate(datetime = trimws(datetime)) %>%
  # Convert TimeStamp to datetime object
  mutate(datetime = ymd_hms(datetime)) %>%
  group_by(eventID) %>%
  # Calculate the time difference in seconds from the first TimeStamp per eventID
  mutate(time_diff_sec = as.numeric(difftime(datetime, min(datetime), units = "secs"))) %>%
  # Filter to keep only rows where the time difference is between 60 and 300 seconds
  filter(time_diff_sec >= 60 & time_diff_sec <= 300) %>%
  # Optionally, summarize the data (for example, calculating means of ch4 and co2)
  summarise(
    mean_ch4 = mean(ch4, na.rm = TRUE),
    mean_co2 = mean(co2, na.rm = TRUE)
  )
head(event_cut)
summary(event_cut)
dim(event_cut)
```

Plotting events to see patterns

```r
#Printing cows numbers
cow_ids<- unique(event_ch4$cow)
print(cow_ids)
```

```r
#################### function to read the events of n cow ##################
# Function to get all events of a specific cow
get_unique_cow_event_ids <- function(cow_number) {
  cow_events <- subset(event_ch4, cow == cow_number)
  unique_event_ids <- unique(cow_events$eventID)
  return(unique_event_ids)
}

cow_number <- 72
unique_event_ids <- get_unique_cow_event_ids(cow_number)
print(unique_event_ids)
```

Function: Detecting peaks

```r
# Function to detect peaks
detectPeaks <- function(ch4, windowSize = 5, threshold = 0.0005) {
  peaksIni <- c()
  peaksFin <- c()
  stage <- FALSE
  peak_decreasing <- FALSE

  for (i in 1:(length(ch4) - windowSize)) {
    tempW <- ch4[i:(i + windowSize - 1)]
    tempX <- 1:windowSize

    mean_tempW <- mean(tempW, na.rm = TRUE)
    mean_tempX <- mean(tempX, na.rm = TRUE)

    cov_xy <- sum((tempW - mean_tempW) * (tempX - mean_tempX), na.rm = TRUE)
    var_x <- sum((tempX - mean_tempX) * (tempX - mean_tempX), na.rm = TRUE)

    pendiente <- cov_xy / var_x

    if (pendiente > threshold && !stage) {
      peaksIni <- c(peaksIni, i)
      stage <- TRUE
      peak_decreasing <- FALSE
    } else if (stage && pendiente < -threshold) {
      peak_decreasing <- TRUE
    } else if (stage && peak_decreasing && pendiente > -threshold / 2) {
      peaksFin <- c(peaksFin, i)
      stage <- FALSE
      peak_decreasing <- FALSE
    } else if (length(peaksIni) == 0 && pendiente > threshold / 2) {
      # Handle case where no peaks have been detected yet
    } else if (i == (length(ch4) - windowSize) && peak_decreasing) {
      peaksFin <- c(peaksFin, i)
    }
  }

  list(peaksIni = peaksIni, peaksFin = peaksFin)
}
```

Function: Detecting peaks

```r
# Function to detect peaks
detectPeaks <- function(ch4, windowSize = 5, threshold = 0.0005) {
  peaksIni <- c()
  peaksFin <- c()
  stage <- FALSE
  peak_decreasing <- FALSE

  for (i in 1:(length(ch4) - windowSize)) {
    tempW <- ch4[i:(i + windowSize - 1)]
    tempX <- 1:windowSize

    mean_tempW <- mean(tempW, na.rm = TRUE)
    mean_tempX <- mean(tempX, na.rm = TRUE)

    cov_xy <- sum((tempW - mean_tempW) * (tempX - mean_tempX), na.rm = TRUE)
    var_x <- sum((tempX - mean_tempX) * (tempX - mean_tempX), na.rm = TRUE)

    pendiente <- cov_xy / var_x

    if (pendiente > threshold && !stage) {
      peaksIni <- c(peaksIni, i)
      stage <- TRUE
      peak_decreasing <- FALSE
    } else if (stage && pendiente < -threshold) {
      peak_decreasing <- TRUE
    } else if (stage && peak_decreasing && pendiente > -threshold / 2) {
      peaksFin <- c(peaksFin, i)
      stage <- FALSE
      peak_decreasing <- FALSE
    } else if (length(peaksIni) == 0 && pendiente > threshold / 2) {
      # Handle case where no peaks have been detected yet
    } else if (i == (length(ch4) - windowSize) && peak_decreasing) {
      peaksFin <- c(peaksFin, i)
    }
  }

  list(peaksIni = peaksIni, peaksFin = peaksFin)
}
```

```r
# Apply the peak detection to the entire database
peak_det <- event_ch4 %>%
  group_by(eventID) %>%
  summarise(num_peaks = length(detectPeaks(ch4)$peaksIni))

# Print the results
head(peak_det)
summary(peak_det)
```

Exercise: calculate number of peaks per minute

```
######################### EXERCISE ########################
#1)Calculate the num of peaks per  minute
    #HINT: Use merge, and lenght of event (visit)
```

Function: Detect peaks and calc sum2maxpeaks

```r
# Function to detect peaks and calculate sum2maxpeaks
detectPeaksAndSum <- function(ch4, windowSize = 5, threshold = 0.0005) {
  peaksIni <- c()
  peaksFin <- c()
  stage <- FALSE
  peak_decreasing <- FALSE

  for (i in 1:(length(ch4) - windowSize)) {
    tempW <- ch4[i:(i + windowSize - 1)]
    tempX <- 1:windowSize

    mean_tempW <- mean(tempW, na.rm = TRUE)
    mean_tempX <- mean(tempX, na.rm = TRUE)

    cov_xy <- sum((tempW - mean_tempW) * (tempX - mean_tempX), na.rm = TRUE)
    var_x <- sum((tempX - mean_tempX) * (tempX - mean_tempX), na.rm = TRUE)

    pendiente <- cov_xy / var_x

    if (pendiente > threshold && !stage) {
      peaksIni <- c(peaksIni, i)
      stage <- TRUE
      peak_decreasing <- FALSE
    } else if (stage && pendiente < -threshold) {
      peak_decreasing <- TRUE
    } else if (stage && peak_decreasing && pendiente > -threshold / 2) {
      peaksFin <- c(peaksFin, i)
      stage <- FALSE
      peak_decreasing <- FALSE
    } else if (length(peaksIni) == 0 && pendiente > threshold / 2) {
      # Handle case where no peaks have been detected yet
    } else if (i == (length(ch4) - windowSize) && peak_decreasing) {
      peaksFin <- c(peaksFin, i)
    }
  }

  sum2maxpeaks <- 0
  if (length(peaksIni) > 0 && length(peaksFin) > 0) {
    for (j in 1:length(peaksIni)) {
      start <- peaksIni[j]
      end <- ifelse(j <= length(peaksFin), peaksFin[j], NA)
      if (!is.na(end) && end > start) {
        peak_values <- ch4[start:end]
        max_values <- sort(peak_values, decreasing = TRUE)[1:2]
        avg_max_values <- mean(max_values)
        sum2maxpeaks <- sum2maxpeaks + avg_max_values
      }
    }
```

```r
# Apply the peak detection to the entire database
peak_phen <- event_ch4 %>%
  group_by(eventID) %>%
  summarise(
    num_peaks = {
      result <- detectPeaksAndSum(ch4)
      length(result$peaksIni)
    },
    sum2maxpeaks = {
      result <- detectPeaksAndSum(ch4)
      result$sum2maxpeaks
    }
  )

# Print the results
head(peak_phen)
```

Adding area under the curve

```
sum2maxpeaks <- 0
area_under_curve <- 0
if (length(peaksIni) > 0 && length(peaksFin) > 0) {
  for (j in 1:length(peaksIni)) {
    start <- peaksIni[j]
    end <- ifelse(j <= length(peaksFin), peaksFin[j], NA)
    if (!is.na(end) && end > start) {
      peak_values <- ch4[start:end]
      max_values <- sort(peak_values, decreasing = TRUE)[1:2]
      avg_max_values <- mean(max_values)
      sum2maxpeaks <- sum2maxpeaks + avg_max_values

      # Calculate area under the curve above ground level (assumed to be the minimum value in ch4)
      ground_level <- min(ch4)
      area_under_curve <- area_under_curve + sum(peak_values - ground_level)
    }
  }
}

list(peaksIni = peaksIni, peaksFin = peaksFin, sum2maxpeaks = sum2maxpeaks, area_under_curve = area_under_curve)


# Apply the peak detection to the entire database
AUC <- event_ch4 %>%
  group_by(eventID) %>%
  summarise(
    num_peaks = {
      result <- detectPeaksAndSum(ch4)
      length(result$peaksIni)
    },
    sum2maxpeaks = {
      result <- detectPeaksAndSum(ch4)
      result$sum2maxpeaks
    },
    area_under_curve = {
      result <- detectPeaksAndSum(ch4)
      result$area_under_curve
    }
  )
```

Exercise: Calculate all phenotypes for cut visit

```r
# Apply the peak detection to the entire database, considering only data between 60 sec to 300 sec per eventID
peak_phen2 <- event_ch42 %>%
  filter(time_diff_sec >= 60 & time_diff_sec <= 300) %>%
  group_by(eventID) %>%
  summarise(
    num_peaks = {
      result <- detectPeaksAndSum(ch4)
      length(result$peaksIni)
    },
    sum2maxpeaks = {
      result <- detectPeaksAndSum(ch4)
      result$sum2maxpeaks
    },
    area_under_curve = {
      result <- detectPeaksAndSum(ch4)
      result$area_under_curve
    }
  )

# Print the results
head(peak_phen2)
dim(peak_phen2)
```

Moving average. play with the window size

```r
# Function to calculate the average of the moving average of CH4 values
calculate_moving_avg <- function(ch4, window_size) {
  moving_avg <- rollmean(ch4, k = window_size, fill = NA)
  mean(moving_avg, na.rm = TRUE)
}

# Apply the peak detection to the entire database, considering only data between 60 sec to 300 sec per eventID
all_phen <- event_ch42 %>%
  filter(time_diff_sec >= 60 & time_diff_sec <= 300) %>%
  group_by(eventID) %>%
  summarise(
    num_peaks = {
      result <- detectPeaksAndSum(ch4)
      length(result$peaksIni)
    },
    sum2maxpeaks = {
      result <- detectPeaksAndSum(ch4)
      result$sum2maxpeaks
    },
    area_under_curve = {
      result <- detectPeaksAndSum(ch4)
      result$area_under_curve
    },
    avg_moving_avg_ch4 = calculate_moving_avg(ch4, window_size = 10) # Change window_size as needed
  )

# Print the results
head(all_phen)
dim(all_phen)
```

Exercise: Calc correlations between phenotypes

```
#calculate all the phenotypes with cut and not cut and calculate correlations among the phenotypes
#HINT1: use merge or join
merged<- merge(event_5low, event_quant, by="eventID")
head(merged)
merged2<- left_join(event_5low, event_3low, by="eventID")
head(merged2)
#HINT2: use cor and choose the method kendall, spearman or pearson
cor(event_5lowavg_ch4, event_3lowavg_ch4, method="kendall", use="complete.obs")
cor(event_meansmean_ch4, event_meansmean_co2, method="pearson", use="complete.obs")
```

Re-Livestock

RESILIENT FARMING SYSTEMS

Spain software example

Birgit Gredler-Grandl, Coralia Manzanilla-Pech, Ester Teran and Oscar González-Recio

WAGENINGEN UNIVERSITY & RESEARCH

INIA CSIC

Zaragoza, February 25th, 2025

Now let's use an automatic software (Oscar)

1. Open settings → New → Terminal
2. cd DAY1
3. load module

   ml use /lustre/shared/Courses/RELIVESTOCK2025/modules

   module load SnifferAnalyzer
4. run the program giving the input files or using bash bash.sh

   sniffer_analyzer herd_robot101.csv herd_sniffer101_Loggy_42s_FD1.txt 42
5. check the output.txt

Combining sniffer and AMS data in Spain

Raw Data

Sincronize

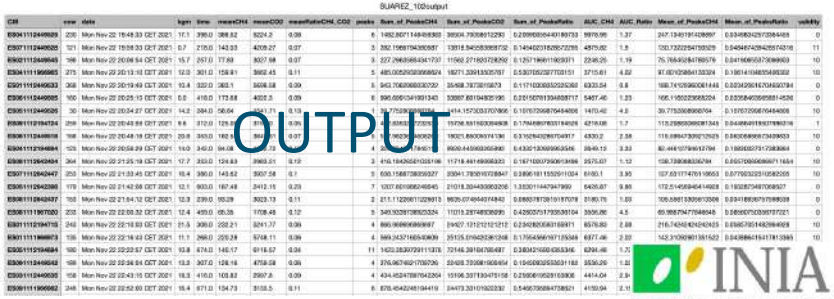
SANCHEZdic21_101_Loggy_42

Date and time	CH4	CO2
03/12/2021 11:50:00	0,018	0,095
03/12/2021 11:50:01	0,018	0,095
03/12/2021 11:50:02	0,018	0,093
03/12/2021 11:50:03	0,018	0,094
03/12/2021 11:50:04	0,018	0,093
03/12/2021 11:50:05	0,018	0,095
03/12/2021 11:50:06	0,018	0,093
03/12/2021 11:50:07	0,018	0,095
03/12/2021 11:50:08	0,018	0,095
03/12/2021 11:50:09	0,018	0,095
03/12/2021 11:50:10	0,018	0,095
03/12/2021 11:50:11	0,018	0,095
03/12/2021 11:50:12	0,019	0,095
03/12/2021 11:50:13	0,019	0,095
03/12/2021 11:50:14	0,019	0,094
03/12/2021 11:50:15	0,019	0,093
03/12/2021 11:50:16	0,019	0,094
03/12/2021 11:50:17	0,02	0,095
03/12/2021 11:50:18	0,02	0,095
03/12/2021 11:50:19	0,02	0,095
03/12/2021 11:50:20	0,02	0,095
03/12/2021 11:50:21	0,02	0,095
03/12/2021 11:50:22	0,02	0,097
03/12/2021 11:50:23	0,02	0,096
03/12/2021 11:50:24	0,021	0,095
03/12/2021 11:50:25	0,021	0,095

SUAREZ_102

Numero vaca	Addres	Fecha/Hora de visita	tiempo en cubiculo	Produccion de leche	Tiempo	Descripcion	CIB
161	102	22/11/2021 0:02:00	4:26	18.8	6:04	Correcto	ES031112194660
148	102	22/11/2021 0:15:00	10:08	10.9	11:39	Correcto	ES021112642431
193	102	22/11/2021 0:23:00	6:01	15.8	7:38	Correcto	ES091111967021
110	102	22/11/2021 0:28:00	3:45	14.4	5:04	Correcto	ES041112449478
136	102	22/11/2021 0:35:00	5:37	15.1	7:11	Correcto	ES091112194713
158	102	22/11/2021 0:43:00	5:42	15.9	7:29	Correcto	ES031112449535
257	102	22/11/2021 0:48:00	3:41	11.3	5:20	Correcto	ES091112642438
178	102	22/11/2021 0:54:00	3:44	14.6	5:24	Correcto	ES061112449527
142	102	22/11/2021 1:01:00	5:01	15	6:35	Correcto	ES031112194693

OUTPUT



SUAREZ_102output

WAGENINGEN UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

Data processing

Java Program	Function
SnifferAnalyzer	1. Assign events to cows according to time footprint and lag
	2. Calculate background (average of 5 lowest measurements from opening of the AMS gate to cow exit).
	3. Detect eructation peaks
	4. Calculate traits
	5. Write output

Executing SnifferAnalyzer Program

./SnifferAnalyzer_linux <AMS_traffic_file> <sniffer_file> <lag_time>

```
/SnifferAnalyzer_linux herd_robot101.csv herd_sniffer101_Loggy_42s_FD1.txt 42
```

WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

What does SnifferAnalyzer do

	A	B	C	D	E	F	G	H
1	cow	Address	time	time	milk	time	Descripcion	CIB
2	59	101	18/02/2022 0:05:31	05:03	15,4	05:03	Correcto	59
3	236	101	18/02/2022 0:11:10	04:57	13	04:57	Correcto	236
4	31	101	18/02/2022 0:16:33	05:18	15,1	05:18	Correcto	31
5	231	101	18/02/2022 0:26:45	09:21	18,7	09:21	Correcto	231
6	532	101	18/02/2022 0:37:19	05:44	14,8	05:44	Correcto	532
7	240	101	18/02/2022 0:43:43	05:31	17,8	05:31	Correcto	240
8	259	101	18/02/2022 0:52:21	06:42	11,8	06:42	Correcto	259
9	208	101	18/02/2022 0:58:00	06:09	0,7	06:09	Tiempo de conexiÔøΩn	208

	A	B	C
1	Date and time	CH4	CO2
325	18/02/2022 00:05:24	0,032	0,534
326	18/02/2022 00:05:25	0,032	0,531
327	18/02/2022 00:05:26	0,032	0,527
328	18/02/2022 00:05:27	0,031	0,514
329	18/02/2022 00:05:28	0,03	0,385
330	18/02/2022 00:05:29	0,03	0,335
331	18/02/2022 00:05:30	0,03	0,331
332	18/02/2022 00:05:31	0,029	0,318
333	18/02/2022 00:05:32	0,029	0,298
334	18/02/2022 00:05:33	0,028	0,296
335	18/02/2022 00:05:34	0,028	0,3
336	18/02/2022 00:05:35	0,029	0,572
337	18/02/2022 00:05:36	0,029	1.002
338	18/02/2022 00:05:37	0,029	1.456
339	18/02/2022 00:05:38	0,03	1.663
340	18/02/2022 00:05:39	0,03	1,61
341	18/02/2022 00:05:40	0,03	1.553
342	18/02/2022 00:05:41	0,03	1,49
343	18/02/2022 00:05:42	0,029	1.438
344	18/02/2022 00:05:43	0,029	1.142
345	18/02/2022 00:05:44	0,029	0,955
346	18/02/2022 00:05:45	0,029	0,717
347	18/02/2022 00:05:46	0,028	0,661
348	18/02/2022 00:05:47	0,028	0,623
349	18/02/2022 00:05:48	0,028	0,597
350	18/02/2022 00:05:49	0,027	0,579
351	18/02/2022 00:05:50	0,027	0,566
352	18/02/2022 00:05:51	0,027	0,564
353	18/02/2022 00:05:52	0,027	0,577
354	18/02/2022 00:05:53	0,027	0,572
355	18/02/2022 00:05:54	0,028	0,567

WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

Data management: Sniffer output

Date and time	Input 1 (2) [V/mA]
2023-11-06 12:29:00	0,038
2023-11-06 12:29:01	0,038
2023-11-06 12:29:02	0,038
2023-11-06 12:29:03	0,038
2023-11-06 12:29:04	0,038
2023-11-06 12:29:05	0,039
2023-11-06 12:29:06	0,039
2023-11-06 12:29:07	0,039
2023-11-06 12:29:08	0,039
2023-11-06 12:29:09	0,039
2023-11-06 12:29:10	0,039

METHANE PHENOTYPING

- Phenotype definition (weekly averages)

1. Mean CH_4 (by second and every 5 s)
2. Sum of peaks CH_4 (by second and every 5 s)
3. Sum of max peaks CH_4
4. Area under the curve (AUC CH_4)
5. Ratio of (mean) CH_4/CO_2
6. CH_4 grams per day (Madsen et al., 2010)

$$\text{Prod } CH_4\left(\frac{g}{d}\right) = 0.714 * \boldsymbol{ratio(ppm)} * 180 * 24 * 0.001 * (5.6 * kg\ body\ mass^{0.75}) + 22 * \boldsymbol{ECM} + 1.6{*}10^{-5} * days\ in\ pregnancy$$

7. CH_4 grams per day (in-house*, based on observed CO2 and CH4/CO2 ratio, *to be published*)

Large genetic correlation between phenotypes

WAGENINGEN UNIVERSITY & RESEARCH

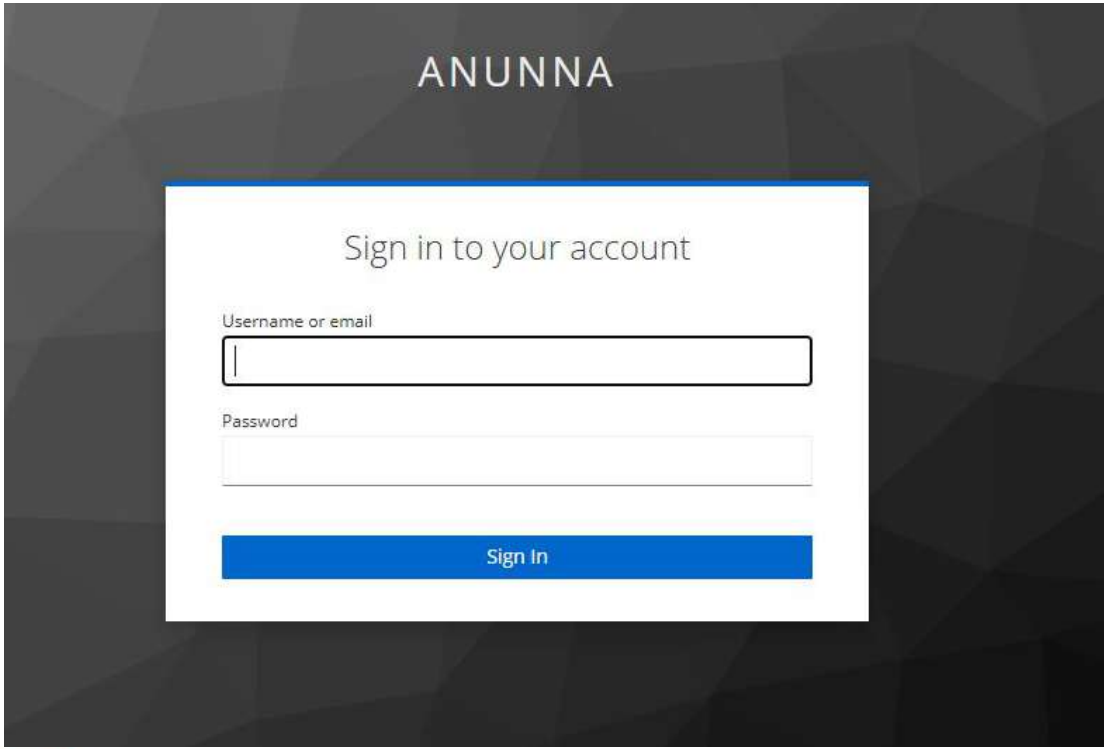
INIA Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

How to clone files/folders from Github to Jupyter

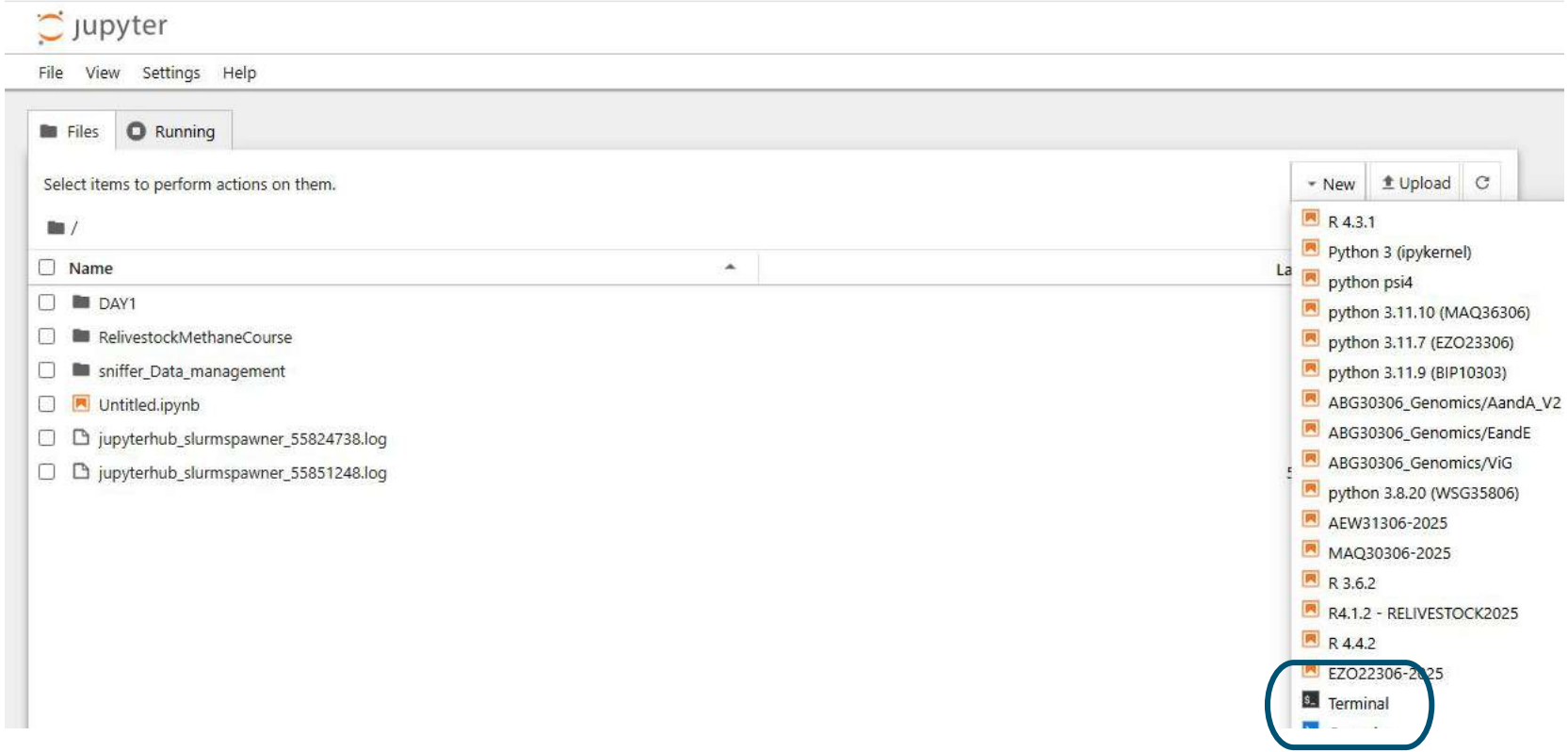
1. Open Jupyter Notebook online in your web browser (https://notebook.anunna.wur.nl) .
2. Click on the “New” button in the top right corner of the screen.
3. Select “Terminal” from the dropdown menu.
4. In the terminal window, type the following command:

git clone https://github.com/ogrecio/RelivestockMethaneCourse

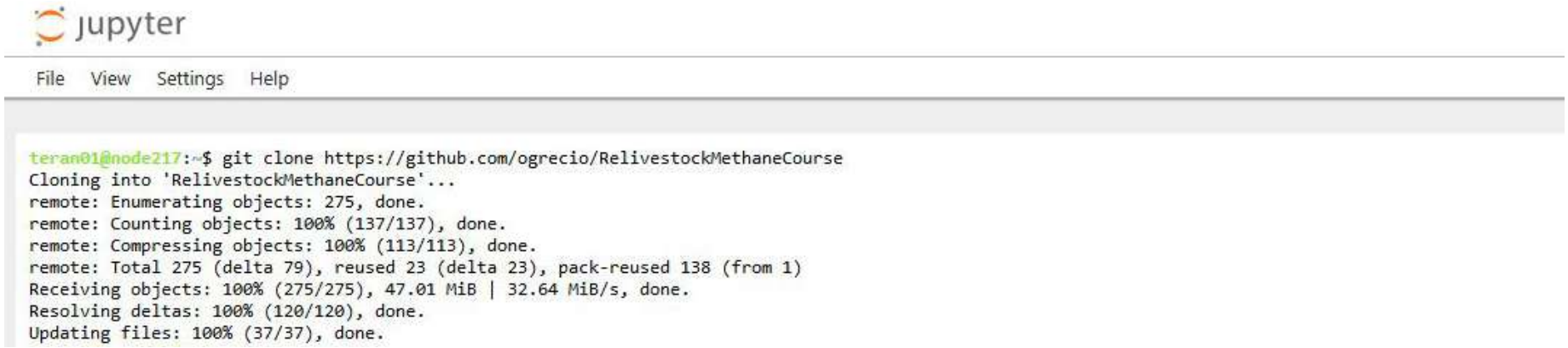
Step 1- Open Jupyter Notebook and login



Step 2 and 3- Open the terminal



Step 4- Clone the repository of the course

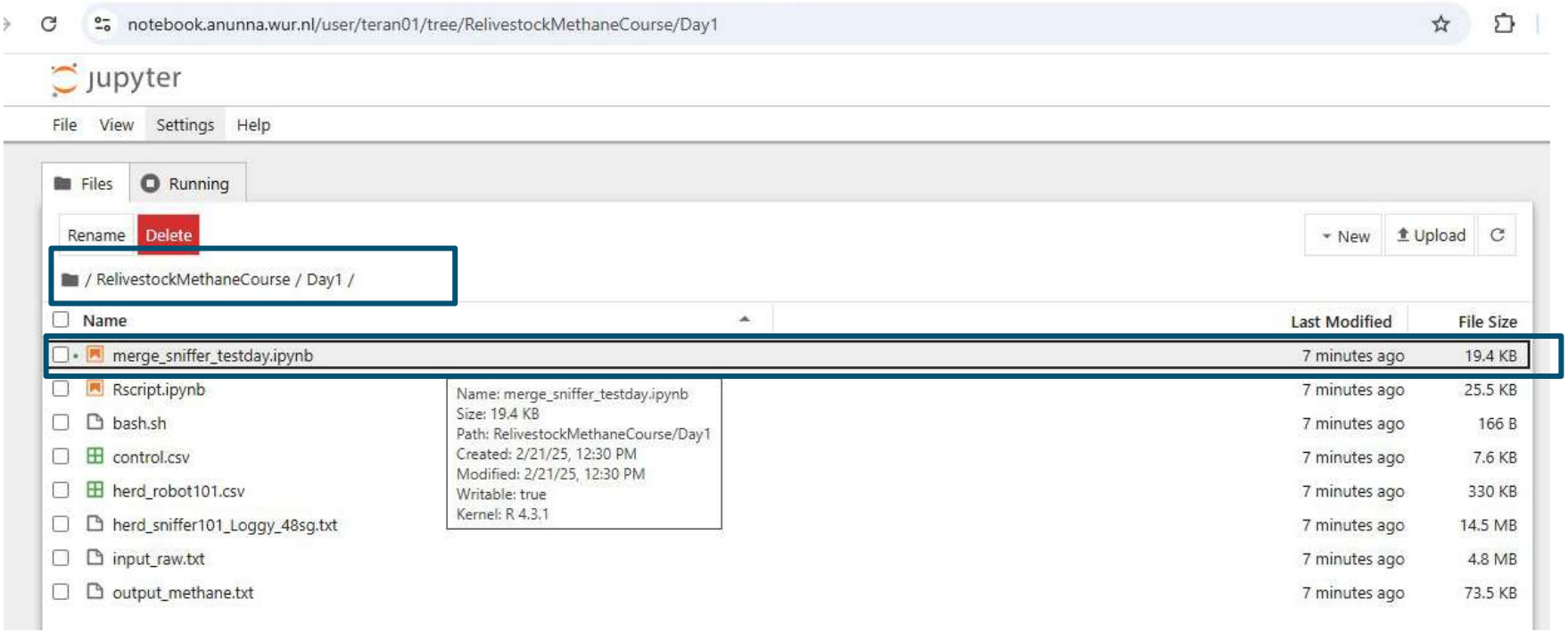


Set the working directory

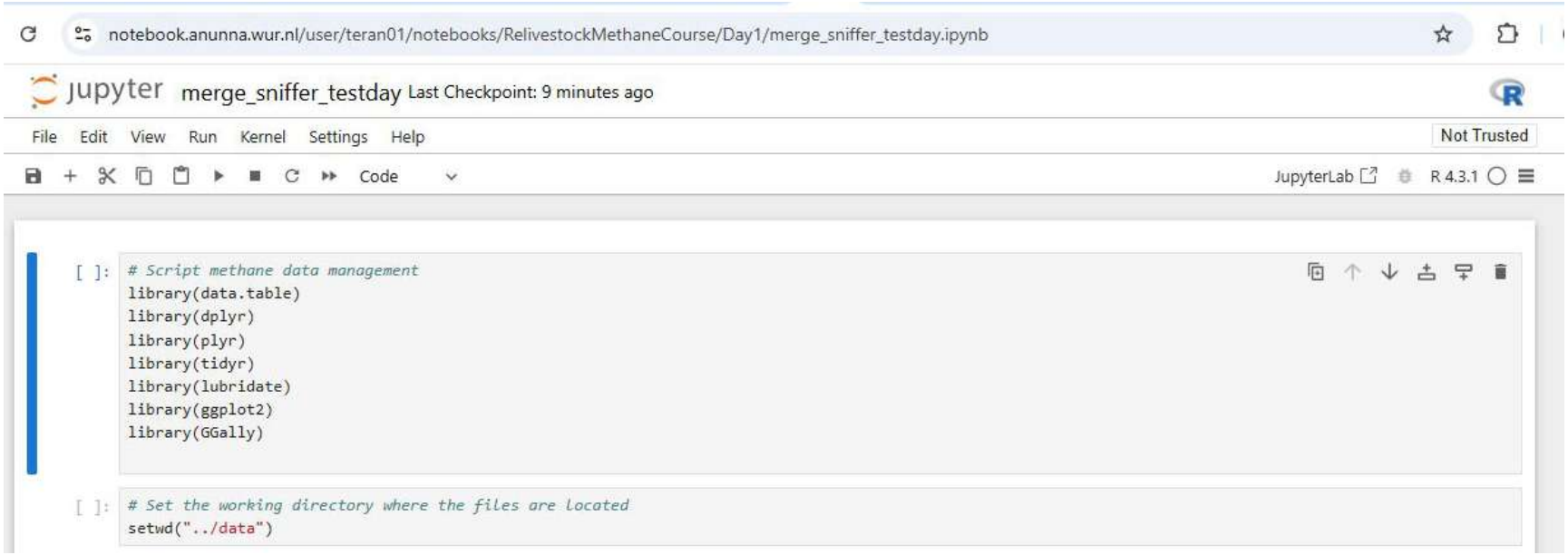
- cd RelivestockMethaneCourse/Day1
- ls

```
Updating files: 100% (37/37), done.
teran01@node217:~$ ls
DAY1  RelivestockMethaneCourse  Untitled.ipynb  jupyterhub_slurmspawner_55824738.log  jupyterhub_slurmspawner_55851248.log  sniffer_Data_management
teran01@node217:~$ cd RelivestockMethaneCourse
teran01@node217:~/RelivestockMethaneCourse$ ls
ASReml-4.2-Functional-Specification.pdf  Day1  LICENSE    Rscript.ipynb  Timetable-ing_v2_Zaragoza_Final.pdf  bash.sh  programs             scripts
ASReml-4.2-Structural-Specification.pdf  Day2  README.md  Rscripts       _config.yml                          data     reading_rawCH4data.R  slides
teran01@node217:~/RelivestockMethaneCourse$ cd Day1
teran01@node217:~/RelivestockMethaneCourse/Day1$ ls
Rscript.ipynb  bash.sh  control.csv  herd_robot101.csv  herd_sniffer101_Loggy_48sg.txt  input_raw.txt  merge_sniffer_testday.ipynb  output_methane.txt
teran01@node217:~/RelivestockMethaneCourse/Day1$
```

Back to Notebook and open the script



Merge_sniffer_testday



Data management: R

Call the libraries and set the work direction

In [ ]:

```
# Script methane data management
library(data.table)
library(dplyr)
library(plyr)
library(tidyr)
library(lubridate)
library(ggplot2)
library(GGally)
```

Read files and check they were correctly imported

In [ ]:

```
# Read the sniffer output
bd=read.table("output_methane.txt",sep=",",header=T) #500 rows & 20 cols
bd_id=unique(bd[,1]) #63 individuals
summary(bd)
```

WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

Data management: Output format

- One farm – 1 robot
- 500 records per visit
- 63 cows

Phenotypes

	cow	date	kgm	time	meanCH4	meanCO2	meanRatioCH4_CO2	AUC_CH4	AUC_Ratio	peaks	peaks_per_minut
1	5425	Tue Aug 09 17:31:52 CEST 2022	15.84	327	146.47	2313.80	0.09	14653.21	8.39	5	
2	6502	Tue Aug 09 17:37:21 CEST 2022	14.39	358	395.57	4347.48	0.09	30028.49	5.72	3	
3	6734	Tue Aug 09 17:43:21 CEST 2022	13.85	418	145.07	2814.31	0.05	5263.64	1.67	3	
4	6493	Tue Aug 09 17:50:21 CEST 2022	13.72	418	235.07	2594.37	0.09	14608.71	4.72	3	
5	7431	Tue Aug 09 17:59:40 CEST 2022	15.86	459	426.97	4366.50	0.08	27266.67	5.05	5	
6	9765	Tue Aug 09 18:09:57 CEST 2022	10.55	322	194.63	2952.69	0.04	19926.71	4.04	4	
7	2504	Tue Aug 09 18:15:26 CEST 2022	14.31	353	245.03	6197.70	0.05	19440.00	3.07	4	
8	2514	Tue Aug 09 18:21:21 CEST 2022	18.76	418	621.93	6548.15	0.11	47065.55	8.07	4	
9	7436	Tue Aug 09 18:34:53 CEST 2022	8.34	266	493.27	6448.67	0.09	48298.65	7.55	4	
10	9750	Tue Aug 09 18:39:21 CEST 2022	18.33	418	375.19	4027.56	0.10	21424.02	6.02	3	
11	7441	Tue Aug 09 18:48:03 CEST 2022	9.56	496	261.73	4234.37	0.04	17648.95	2.51	5	

Read files and check they were correctly imported

In [ ]:

```
# Read test day file
test=read.table("control.csv",sep=";",header=T)#107 rows
test_id=unique(test[,1])#64 #july and sept
#obtain date from test day records
testtest_date1 = dmy(testtest_date)
# Obtain kgm of protein and fat

testkgmfat = testfat*test$milk/100
testkgmprotein = testprotein*test$milk/100
```

Data management: data test description

- Test date → To join to sniffer data
- N of calving → Model
- Calving date → Days in milking
- Productive traits → Genetic correlations
- Weight → Methane production

	cow	test_date	numpar	calving_date	milk	fat	protein	RCS	ETS	lactose	urea	bhb	weight	te
1	586	16/09/2022	2	01/08/2022 0:00	39.32	3.34	3.00	106	8.49	4.73	121	0.05	571.0743	
2	635	19/07/2022	1	NA	21.20	3.93	3.15	74	8.94	5.06	285	0.04	NA	
3	635	16/09/2022	1	NA	21.60	3.92	3.32	40	9.07	5.03	129	NULL	NA	
4	4109	19/07/2022	1	NA	26.80	3.96	3.26	18	8.88	4.95	329	0.04	NA	
5	4109	16/09/2022	1	NA	26.00	3.57	3.53	15	9.24	4.84	149	0.05	NA	
6	4110	19/07/2022	1	07/07/2022 0:00	23.23	3.26	3.12	51	8.56	4.61	200	0.07	598.4168	
7	4110	16/09/2022	1	07/07/2022 0:00	25.74	5.06	3.37	19	9.04	4.99	155	NULL	598.4168	
8	5422	16/09/2022	3	16/08/2022 0:00	43.31	3.66	2.83	14	8.47	4.89	108	0.1	580.8212	
9	5423	16/09/2022	4	NA	30.40	4.67	3.18	11	9.00	5.00	130	0.05	NA	
10	5424	16/09/2022	3	31/01/2022 0:00	28.43	3.84	3.29	173	8.79	4.63	155	0.07	514.0000	
11	5424	19/07/2022	3	31/01/2022 0:00	37.66	2.74	2.93	24	8.61	4.90	265	0.04	514.0000	

WAGENINGEN UNIVERSITY & RESEARCH

INIA Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

Join sniffer file with test file by ID and closest date

```
# Obtain date from output
bd<-tidyr::separate(data = bd, col ="date",into =  c("day", "month","date","time","z","year"),
                    sep = " ", extra = "merge")
bd$sniffer_date=paste(bd$date, bd$month, bd$year, sep = "/")
bd$sniffer_date1<-dmy(bd$sniffer_date)


# Join the two datatables-
bd_full <- lapply(intersect(bd$cow,test$cow),function(id) {
  d1 <- subset(bd,cow==id)
  d2 <- subset(test,cow==id)

  d1$indices <- sapply(as.Date(d1$sniffer_date1),function(d) which.min(abs(as.Date(d2$test_date1) - d)))
  d2$indices <- 1:nrow(d2)

  merge(d1,d2,by=c('cow','indices'))
}) ###

bd_full <- do.call(rbind,bd_full) #342
bd_full$indices <- NULL
```

Check the format of date column

Check the number of animals that have not joined

```
############################################## Check the number of animals that have not joined #
bd_antijoin=anti_join(bd,bd_full,by="cow") #158, 3 individuals
bd_full$dif_days=abs(bd_full$sniffer_date1-bd_full$test_date1)
# check the difference of days ###choose less than 40 days
table(bd_full$dif_days)
```

Days in milking and threshold

```
# Obtain days in milking
bd_full=tidyr::separate(data=bd_full,col ="calving_date", into=c('calving_date', 'calving_time'),sep=' ')
bd_full$calving_date1=dmy(bd_full$calving_date)
bd_full$daysinmilking=bd_full$sniffer_date1-bd_full$calving_date1
bd_full$daysinmilking=gsub("[a-z]","",bd_full$daysinmilking)#retain only numbers
bd_full$daysinmilking=as.numeric(bd_full$daysinmilking)
# Check difference of days
table(bd_full$daysinmilking)
# Choose The threshold the difference of days , 365
bd_full1=bd_full %>% filter(daysinmilking <= 365)
```

Week of lactation and/or state of lactation

```
# Obtain week of lactation
bd_full1$week_lactation=floor(bd_full1$daysinmilking / 7)
# Codify days in milking as levels
bd_full1<-mutate(bd_full1,state_lactation=case_when(
  daysinmilking < 91 ~ "1",
  daysinmilking > 90 & daysinmilking < 151 ~ "2",
  daysinmilking >150 ~ "3"))
table(bd_full1$state_lactation)
```

Number of calving

```
# Codify number of calving in three or four levels
hist(bd_full1$numpar)
table(bd_full1$numpar)
# If there is data of more than 3 calvings
bd_full1<-mutate(bd_full1,num_calving=case_when(
  numpar <= 1 ~ "1",
  numpar > 1 & numpar < 3  ~ "2",
  numpar > 2 ~ "3"))
table(bd_full1$num_calving)
```

Data management: data output description

Methane phenotypes : ppm

- Ratio CH4/CO2
- Methane production: grams per day (Madsen et.al, 2010)

$$\textbf{Prod } CH_4\textbf{(g/d)} = 0.714 * \textbf{ratio} * 180 * 24 * 0.001 * (5.6 * \text{kg body mass}^{0.75}) + 22 * \textbf{ECM} + 1.6 * 10^{-5} * \text{days in pregnancy}$$

- Methane production: Tier 2. Dependent of milk, fat, protein and weight

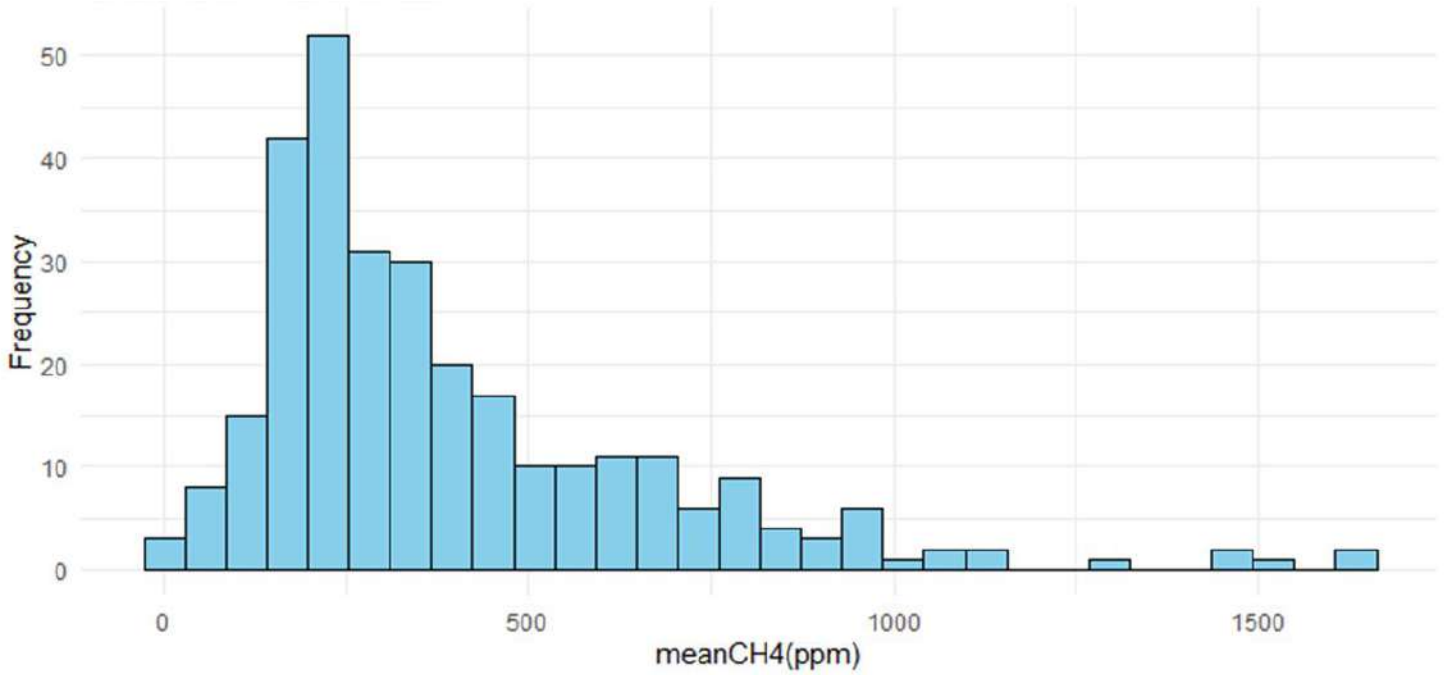
WAGENINGEN UNIVERSITY & RESEARCH

INIA Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

Ratio and grams per day

```
# Ratio mean CH4/CO2
bd_full1$ratioCH4CO2=bd_full1$meanCH4/bd_full1$meanCO2
mean(bd_full1$ratioCH4CO2)

# Obtain grams per day
# Madsen equation
bd_full1$ECM=bd_full1$milk*(0.25+0.122*bd_full1$fat+0.077*bd_full1$protein)
bd_full1$days_inpregnancy=0
bd_full1$gd_madsen<-(0.714*bd_full1$ratioCH4CO2)*180*24*0.001*(5.6*bd_full1$weight**0.75+22*bd_full1$ECM+
                                                               1.6*0.00001*bd_full1$days_inpregnancy**3)
mean(bd_full1$gd_madsen,na.rm=T)

# Tier 2 equation
Cf<-0.386*1.2 #coefficient for feeding situation, lactating
Ca<-0 #Coefficient for activity
Cp<-0.10 #coefficient for pregnancy
DE<-80 #Digestible energy per gross energy in cows fed low quality forage and concentrate
NEm<-Cf*bd_full1$weight**0.75
NEa<-NEm*Ca
NEl<-bd_full1$milk*(1.47+0.4*bd_full1$fat)
NEp<-ifelse(bd_full1$daysinmilking>100, Cp*NEm,0)
REM<-1.123-(0.004092)*DE+0.00001126*(DE)**2-25.4/DE
GE<- 1000*((NEm+NEa+NEl+NEp)/REM)/(DE/100) #g/d
bd_full1$CH4_tier2<-GE*0.065/55.65
mean(bd_full1$CH4_tier2,na.rm=T)
```

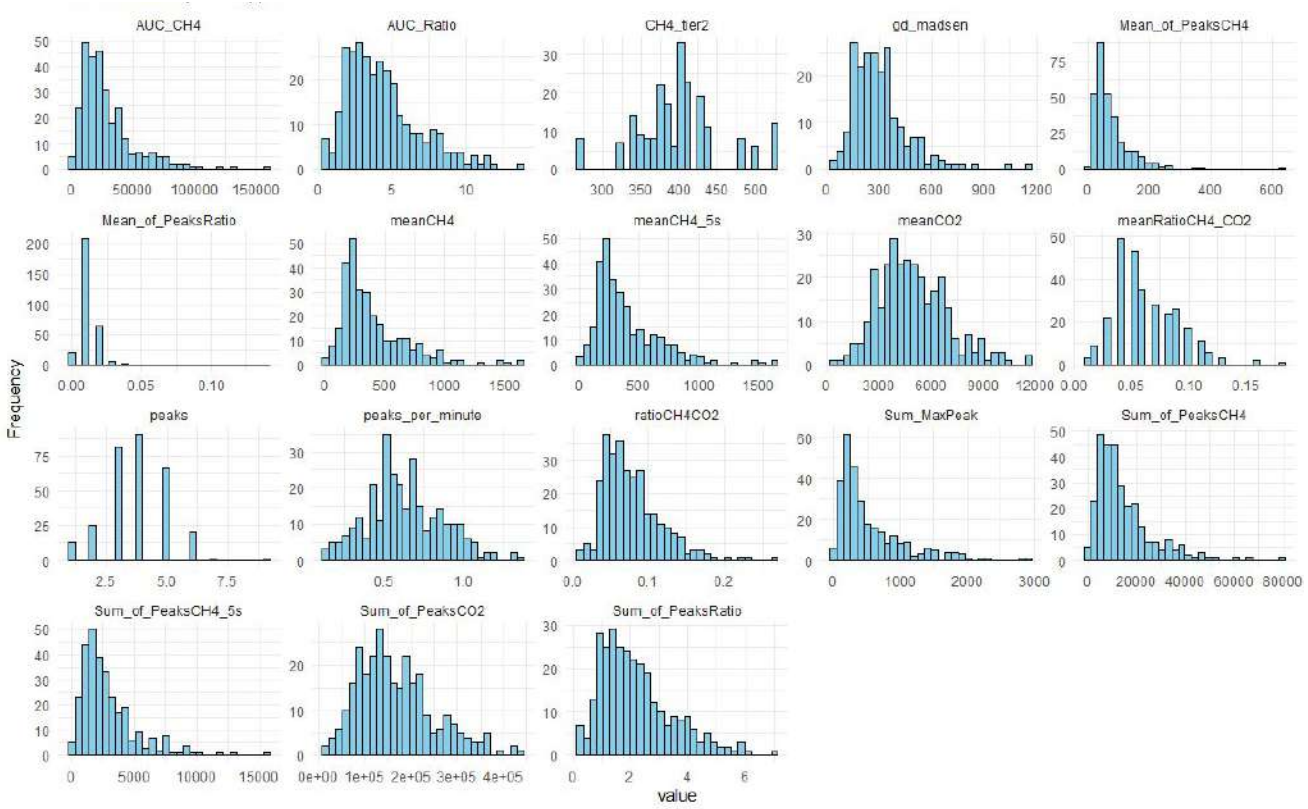
Data management: Phenotypes distribution



WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación
y Tecnología Agraria y Alimentaria

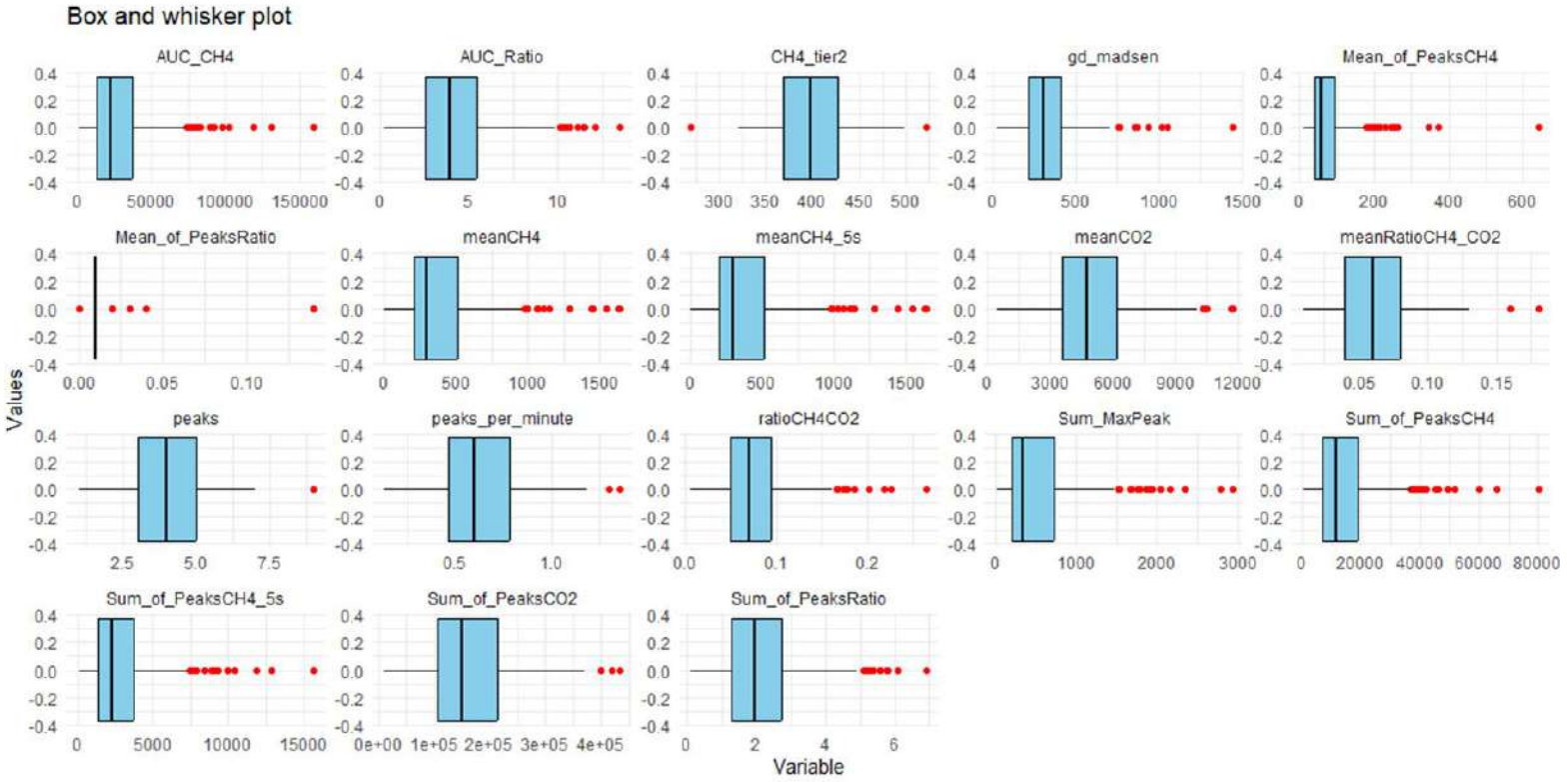
Data management: Phenotypes distribution



WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

Data management: Outlier visualization



WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

Data management: filtering

- CO_2 concentration lower limit, as technical error < 2500
- Filter applied to ratio $CH_4/CO_2 > 0.3$
- Filter applied to CH_4 g/d > 1200 values
- Detection of outliers
- Data correction of outliers ± 3SD

Filtering out

- Methane records
  CO2<2500.
  > +- 3SD in the comparison group
  <5 or >35 records per week (remove all records in that week)
  Remove weeks with <5 records, and animals with only 1 week of records
- CO2 records
  <2500
  > +- 3SD in the comparison group
- CH4/CO2 records
  ratio>0.3
  > +- 3SD in the comparison group

Average data per week (day), for consistency
(more reliable heritability estimates)

WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación
y Tecnología Agraria y Alimentaria

Data management: filtering

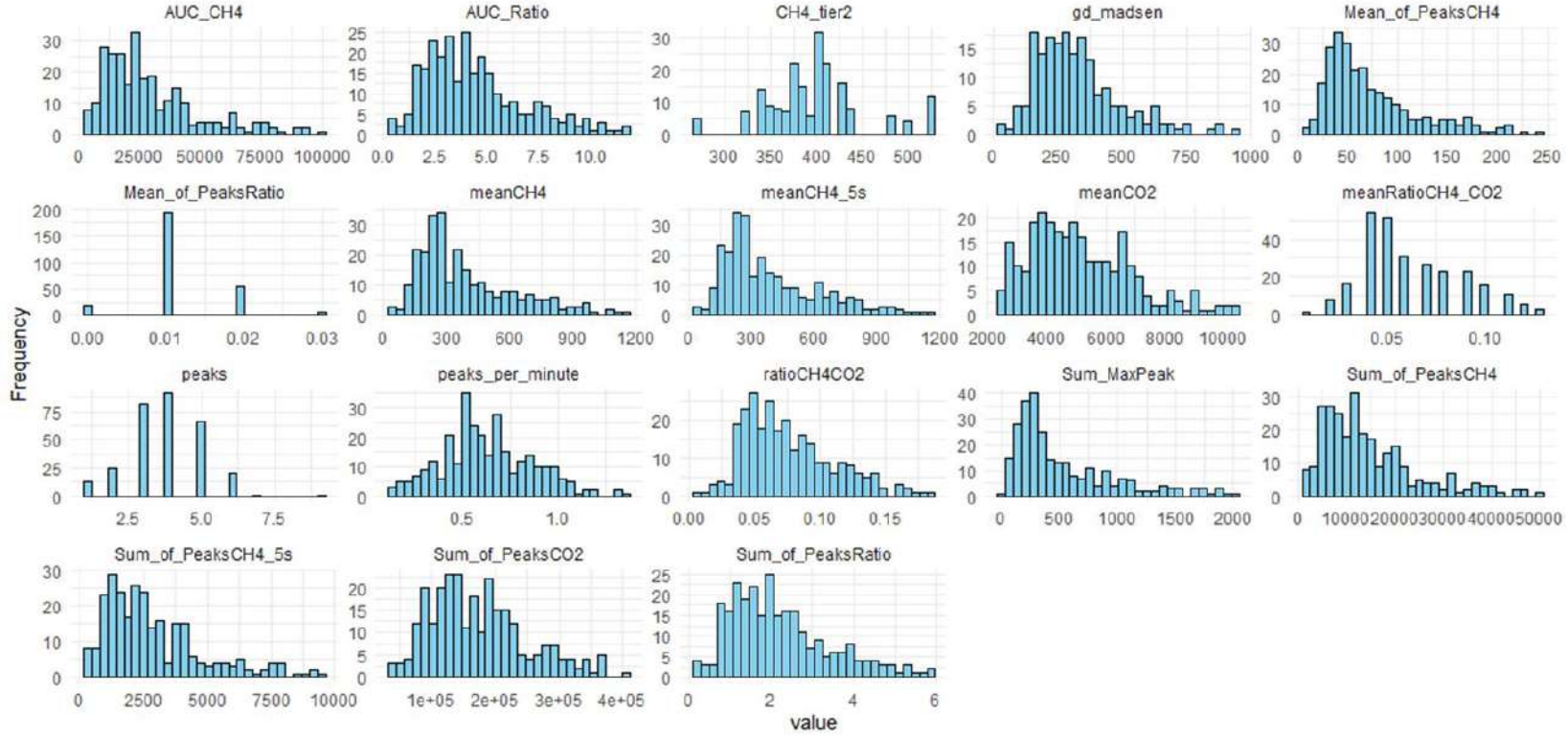
Set as NAs the values in the different phenotypes where the corresponding meanCO2 is lower than 2500

```
# Threshold by CO2 concentration
bd_full2 <- bd_full1 %>%
  dplyr::mutate(across(c(meanCH4, meanCH4_5s,meanCO2,meanRatioCH4_CO2,AUC_CH4,AUC_Ratio,
                         Sum_of_PeaksCH4,Sum_of_PeaksCH4_5s,Sum_of_PeaksCO2,Sum_of_PeaksRatio,
                         Mean_of_PeaksCH4,Mean_of_PeaksRatio,Sum_MaxPeak, ratioCH4CO2, gd_madsen,
                         CH4_tier2), ~ ifelse(meanCO2 < 2500, NA, .)))

colSums(is.na(bd_full1))
colSums(is.na(bd_full2))
```

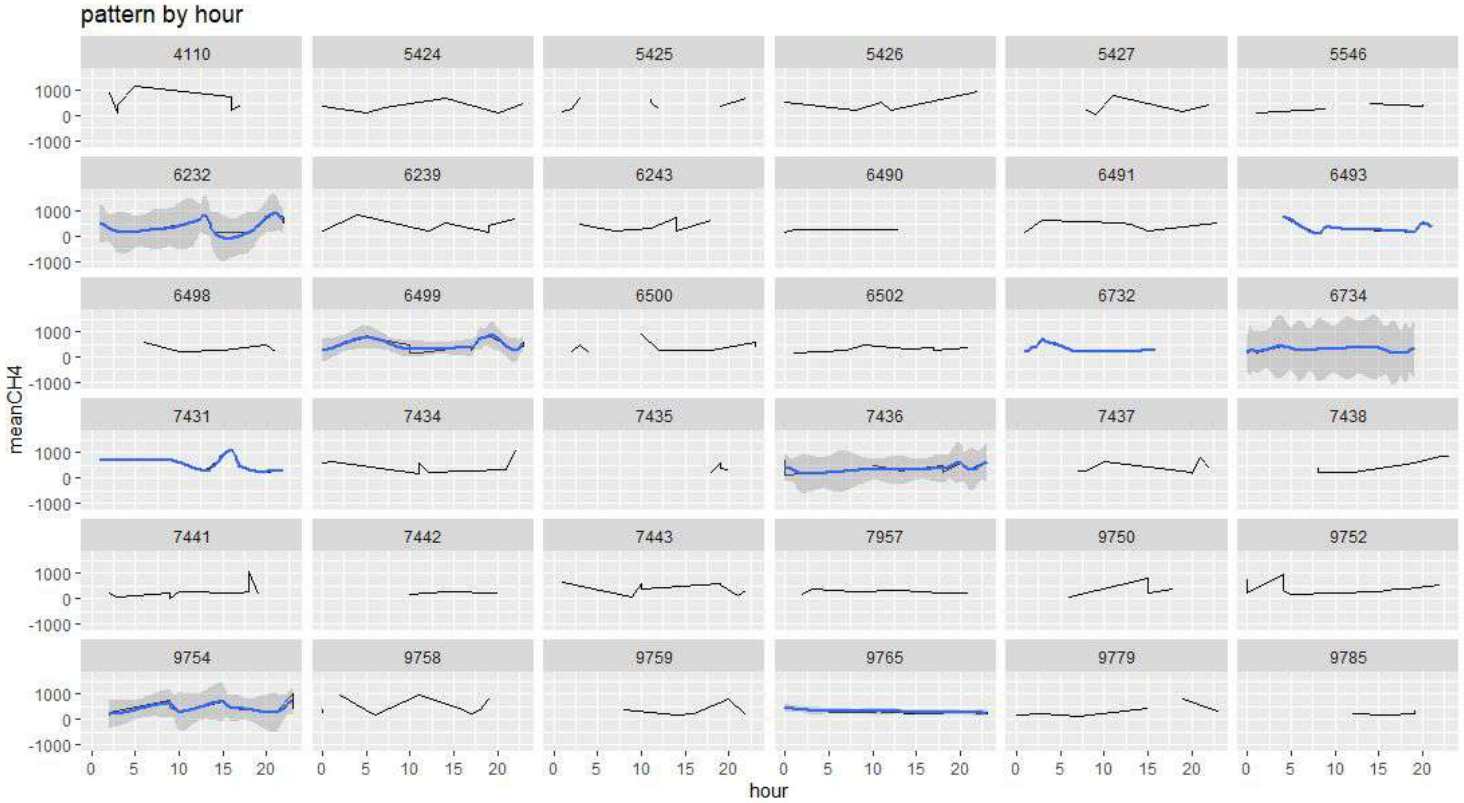
Data management: filtering

```r
# Function for correction of data ±3 SD
function_outliers <- function(x) {
  mean_x <- mean(x, na.rm = TRUE)
  sd_x <- sd(x, na.rm = TRUE)

  # Set upper and lower lim (± 3 SD)
  lower_lim <- mean_x - 3 * sd_x
  upper_lim <- mean_x + 3 * sd_x

  # Replace out-of-limit values with Nas
  x <- ifelse(x < lower_lim | x > upper_lim, NA, x)

  return(x)
}
#Apply the function
data_corrected <- bd_full2 %>%
  dplyr::mutate(across(
    c(meanCH4, meanCH4_5s,meanCO2,meanRatioCH4_CO2,AUC_CH4,AUC_Ratio,
      Sum_of_PeaksCH4,Sum_of_PeaksCH4_5s,Sum_of_PeaksCO2,Sum_of_PeaksRatio,
      Mean_of_PeaksCH4,Mean_of_PeaksRatio,Sum_MaxPeak, ratioCH4CO2, gd_madsen ,
      CH4_tier2,milk,kgmfat,kgmprotein), #Apply the function to these variables
    function_outliers    # Function to replace outliers with NAs
  ))
colSums(is.na(data_corrected))
```

Data management: data distribution after filtering



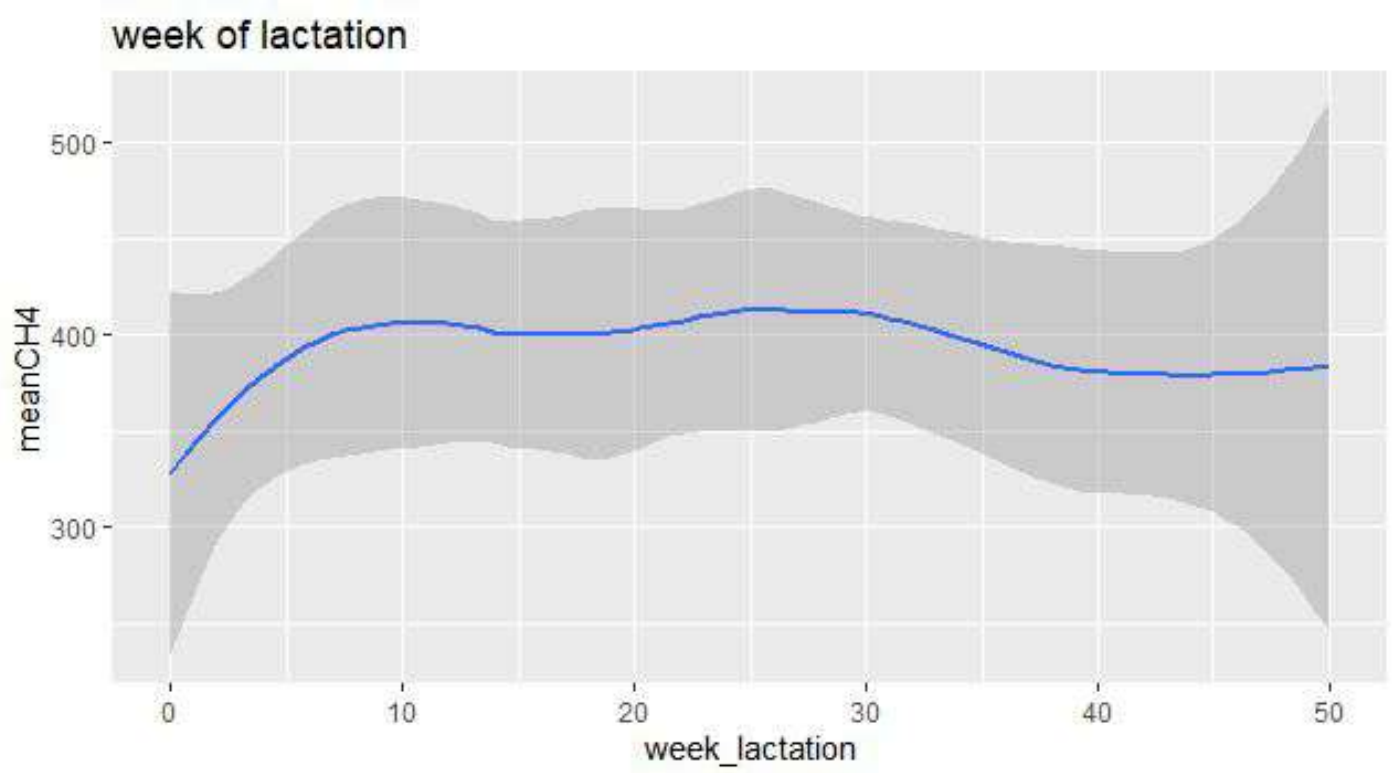
Data management:pattern by hour



WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación
y Tecnología Agraria y Alimentaria

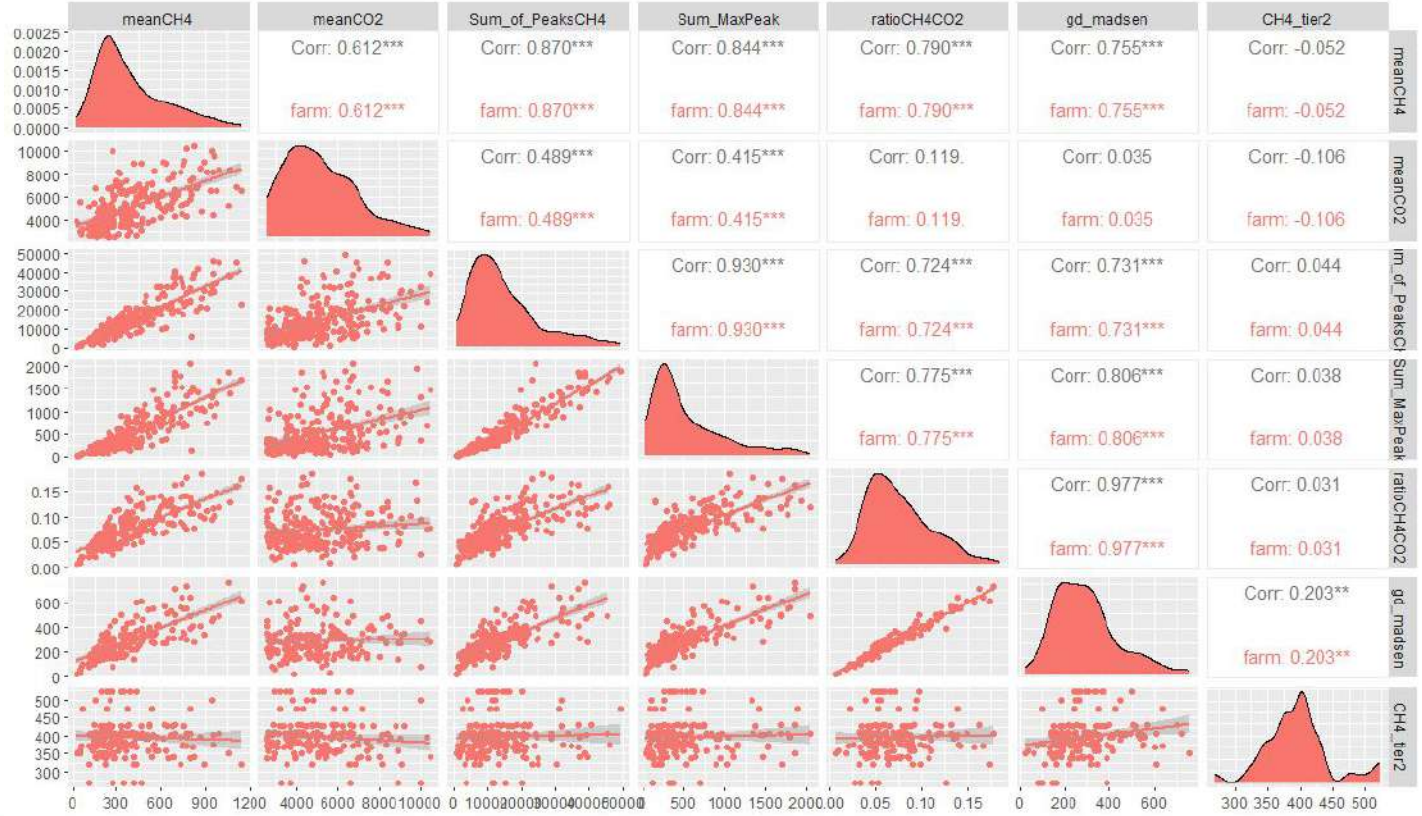
Data management:trend by week of lactation



WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación
y Tecnología Agraria y Alimentaria

Correlation between phenotypes



WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

Data management: mean per week

- Data more consistent.
- Check the number of data by individual and comparison group (At least 2 per individual and 3/5 per group).

```
######################################################################## MEAN PER WEEK ####################
# Obtain epiweek with lubridate
data_corrected$epiweek=epiweek(data_corrected$sniffer_date1)
data_corrected$epiyear=epiyear(data_corrected$sniffer_date1)
#If there are more than one robot by farm, codify them
data_corrected$robot="1"
data_corrected$farm="1"
```

Data management: mean per week

- Data more consistent.
- Check the number of data by individual and comparison group (At least 2 per individual and 3/5 per group).

```
# Calculate the mean per week per animal, considering robot and year if there is more than one p
mean_week <- data_corrected %>%
  group_by(cow,epiweek,epiyear) %>% #some farms could be studied in two or more periods
  dplyr::summarise(across((where(is.numeric)), list(mean = ~mean(.,na.rm=T)), .names = "{.col}_m
                   count = n()) %>%
  dplyr::left_join(data_corrected, by = c("cow","epiweek","epiyear"))#keep the other variables (are necessary fo

# Filter by count per week, keep count > 3/5 records pe week
table(mean_week$count)
mean_week1=mean_week[mean_week$count > 3, ]
```

Add robot if it's necessary

Keep the information for further analysis

WAGENINGEN UNIVERSITY & RESEARCH

INIA Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria

Data management: mean per week

```
# Choose the variables to keep
names(mean_week1)
mean_week2=mean_week1[,c(1:3,5:38,86,87,94,95)] # ID, averages of phenotypes and variables to further analysis
mean_week3=mean_week2[!duplicated(mean_week2),]
# Sort the columns
names(mean_week3)
mean_week4=mean_week3[,c(1:3,41,40,39,38,4:30,33,36,37)] #38 rows
```

Data management: mean per week

```
# Check the number of animals by comparison group. i.e =herd-season-year or herd-robot-week-year
mean_week5=mean_week5%>%
  group_by(farm,robot,epiweek,epiyear)%>%
  dplyr::mutate(N_group=n())
table(mean_week5$N_group)
# Check the number of records per animal (to be considered in the model, repeated measurements)
mean_week5=mean_week5%>%
  group_by(cow)%>%
  dplyr::mutate(N_group=n())
table(mean_week5$N_anim)
```

Data management: Final

	cow	epiweek	epiyear	state_lactation	num_calving	robot	farm	meanCH4_mean_week	meanCO2_mean_week	meanRatioCH4_CO2_
1	4110	32	2022	1	1	1	1	564.4129	6502.296	
2	5424	32	2022	3	3	1	1	348.6350	4372.060	
3	5425	32	2022	3	3	1	1	444.1787	4900.474	
4	5426	32	2022	2	3	1	1	493.4380	5193.396	
5	5427	32	2022	3	3	1	1	321.1300	4112.284	
6	5546	32	2022	1	1	1	1	321.0280	6483.494	
7	6232	32	2022	2	2	1	1	481.0820	5271.495	
8	6239	32	2022	1	3	1	1	435.8071	6298.911	
9	6243	32	2022	3	3	1	1	484.4729	6602.339	
10	6490	32	2022	2	2	1	1	398.3675	5782.205	

ID

Effects

Phenotypes

WAGENINGEN
UNIVERSITY & RESEARCH

INIA
Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria