UCL Data Analysis Timepoints

Elise

2024-06-09

Set the right working directory.

```
setwd("C:/Users/elise/Documents/Mémoire/Main/Data/Templates/UCL")
```

Data importation

Reimport the data sets extracted from the Data Preparation and Data Analysis R Markdown.

```
list.files()
```

```
## [1] "endpoint.txt"     "plant_info.txt"   "S_timeseries.txt" "T_timeseries.txt"
## [5] "timeseries.txt"
```

```
plant_info <- read.table("plant_info.txt", header = TRUE, sep = "\t")
endpoint <- read.table("endpoint.txt", header = TRUE, sep = "\t")

# plant_info
plant_info <- lapply(plant_info, factor)

# endpoint
matching_cols <- intersect(names(endpoint), names(plant_info))
endpoint[, matching_cols] <- lapply(endpoint[, matching_cols], factor)
endpoint$Date <- date(endpoint$Date)
endpoint$Timestamp <- NA

platform <- "UCL"

# endpoint
df <- endpoint[,colSums(is.na(endpoint))<nrow(endpoint)]
genotype_index <- which(colnames(df) == "Genotype")
variables <- colnames(df[, c(3:(genotype_index - 1))]) # We remove the two first column
s that are "Unit.ID" and "Date"

print(paste(platform, ": The variables for endpoint are", paste(variables, collapse =
", "), sep = " "))
```

```
## [1] "UCL : The variables for endpoint are DW_shoot_g, FW_shoot_g, DW_root_g, FW_root
_g"
```

```
endpoint$Plant_type <- substr(endpoint$Genotype, nchar(as.character(endpoint$Genotyp
e)), nchar(as.character(endpoint$Genotype)))
```

Get the cleaned endpoint data

```
endpoint_clean <- endpoint
# Run the function on the dataset for all the variables
endpoint_clean <- detect_replace_outliers_by_genotype(endpoint_clean)
```

Time point objects

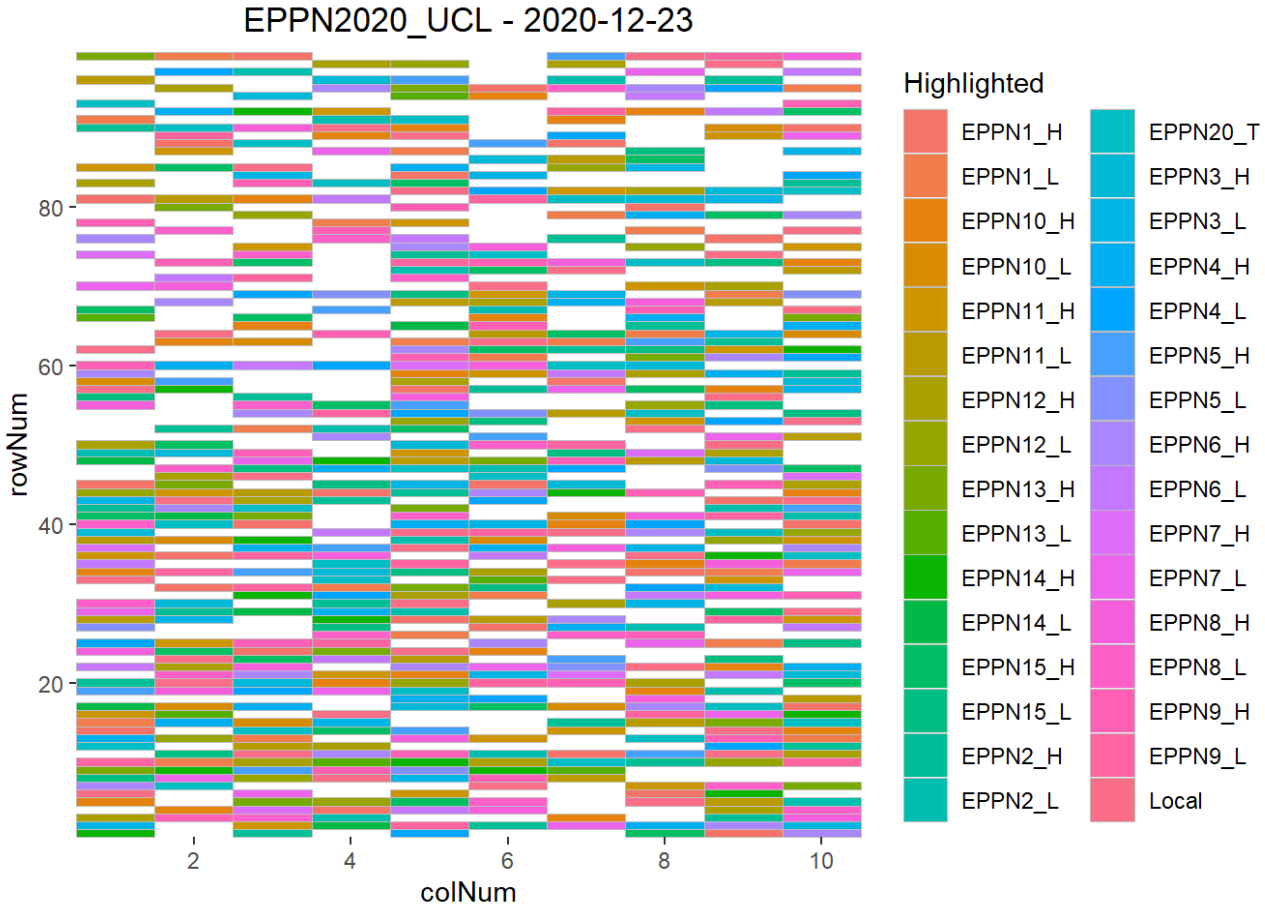
Generation of the timePoints objects using the function “createTimePoints”.

```
endpoint <- endpoint %>%
  group_by(Date, Unit.ID) %>%
  slice(1) %>%
  ungroup()


timePoint_endpoint <- createTimePoints(dat = endpoint,
                                       experimentName = "EPPN2020_UCL",
                                       genotype = "Genotype",
                                       timePoint = "Date",
                                       plotId = "Unit.ID",
                                       rowNum = "Row",
                                       colNum = "Column",
                                       repId = "Replication")


endpoint_clean <- endpoint_clean %>%
  group_by(Date, Unit.ID) %>%
  slice(1) %>%
  ungroup()

timePoint_endpoint_clean <- createTimePoints(dat = endpoint_clean,
                                       experimentName = "EPPN2020_UCL",
                                       genotype = "Genotype",
                                       timePoint = "Date",
                                       plotId = "Unit.ID",
                                       rowNum = "Row",
                                       colNum = "Column",
                                       repId = "Replication")
```

Gentoypic layout

Check the layout of the platforms’ genotypes.

```
genotypes_list <- as.character(unique(endpoint$Genotype))

plot(timePoint_endpoint,
     plotType = "layout",
     highlight =  genotypes_list,
     showGeno = FALSE)
```



1. endpoint

Comparisons between raw and cleaned data

View timePoint object.

```
summary(timePoint_endpoint)
```

```
## timePoint_endpoint contains data for experiment EPPN2020_UCL.
##
## It contains 1 time points.
## First time point: 2020-12-23
## Last time point: 2020-12-23
##
## No check genotypes are defined.
```

```
getTimePoints(timePoint_endpoint)
```

```
##   timeNumber  timePoint
## 1          1 2020-12-23
```

Count the number of observations per trait.

```
traits <- variables

for (trait_name in traits) {
  print(paste("How many data observations for", trait_name))
  num_observations <- countValid(timePoint_endpoint, trait_name)
  print(num_observations)
}
```

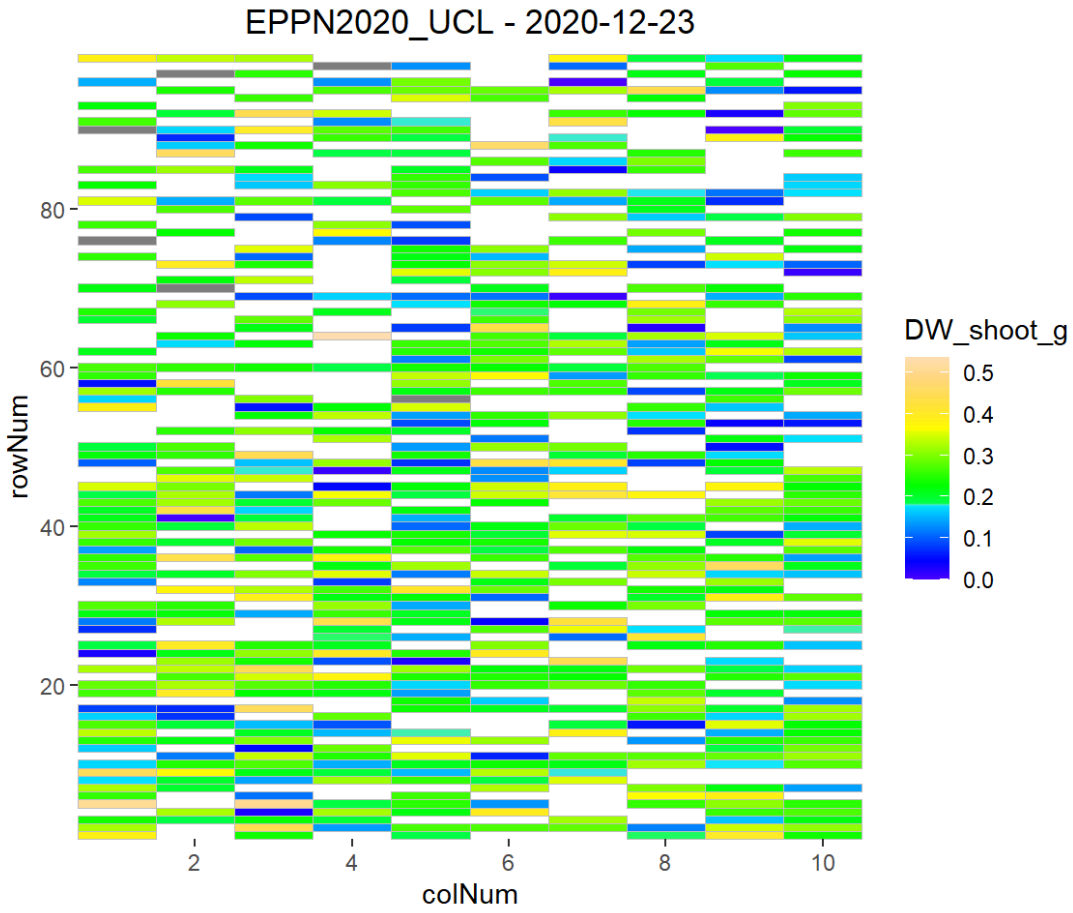
```
## [1] "How many data observations for DW_shoot_g"
## 2020-12-23
##        613
## [1] "How many data observations for FW_shoot_g"
## 2020-12-23
##        615
## [1] "How many data observations for DW_root_g"
## 2020-12-23
##        618
## [1] "How many data observations for FW_root_g"
## 2020-12-23
##        617
```

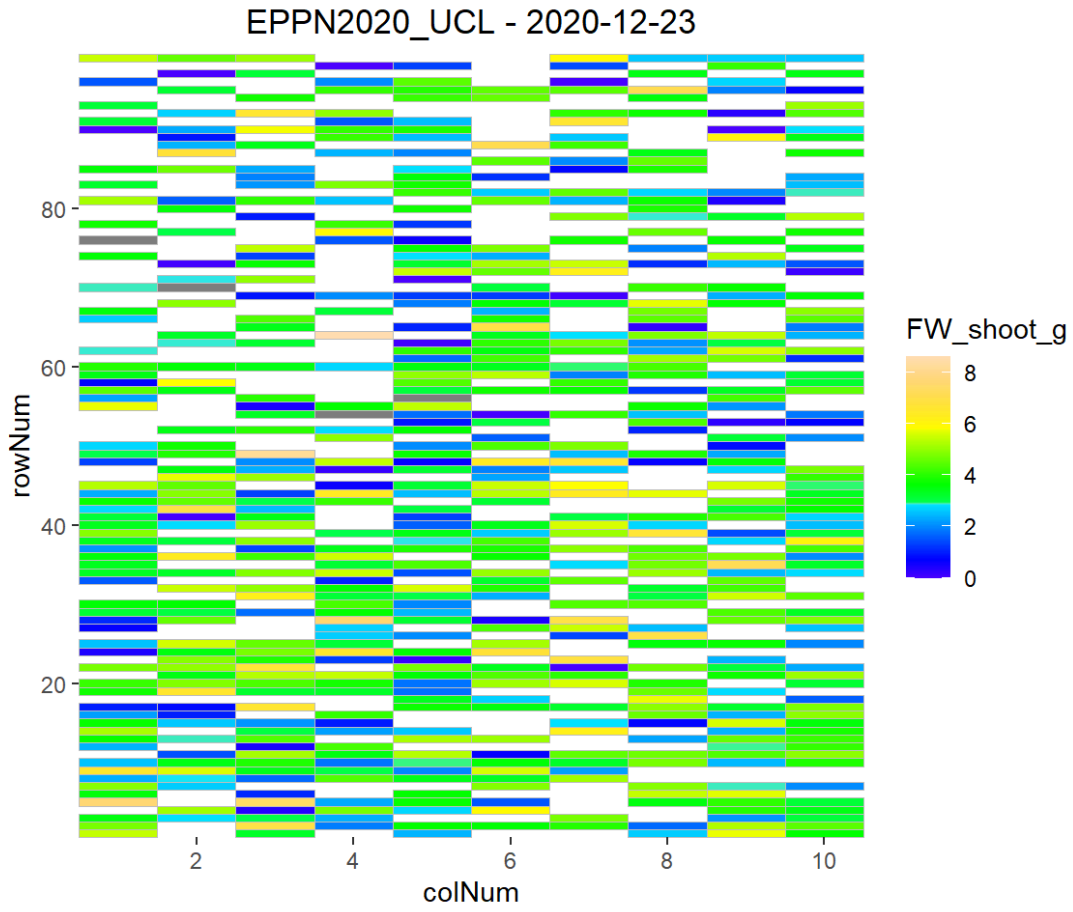
```
for (trait_name in traits) {
  print(paste("How many cleaned data observations for", trait_name))
  num_observations <- countValid(timePoint_endpoint_clean, trait_name)
  print(num_observations)
}
```

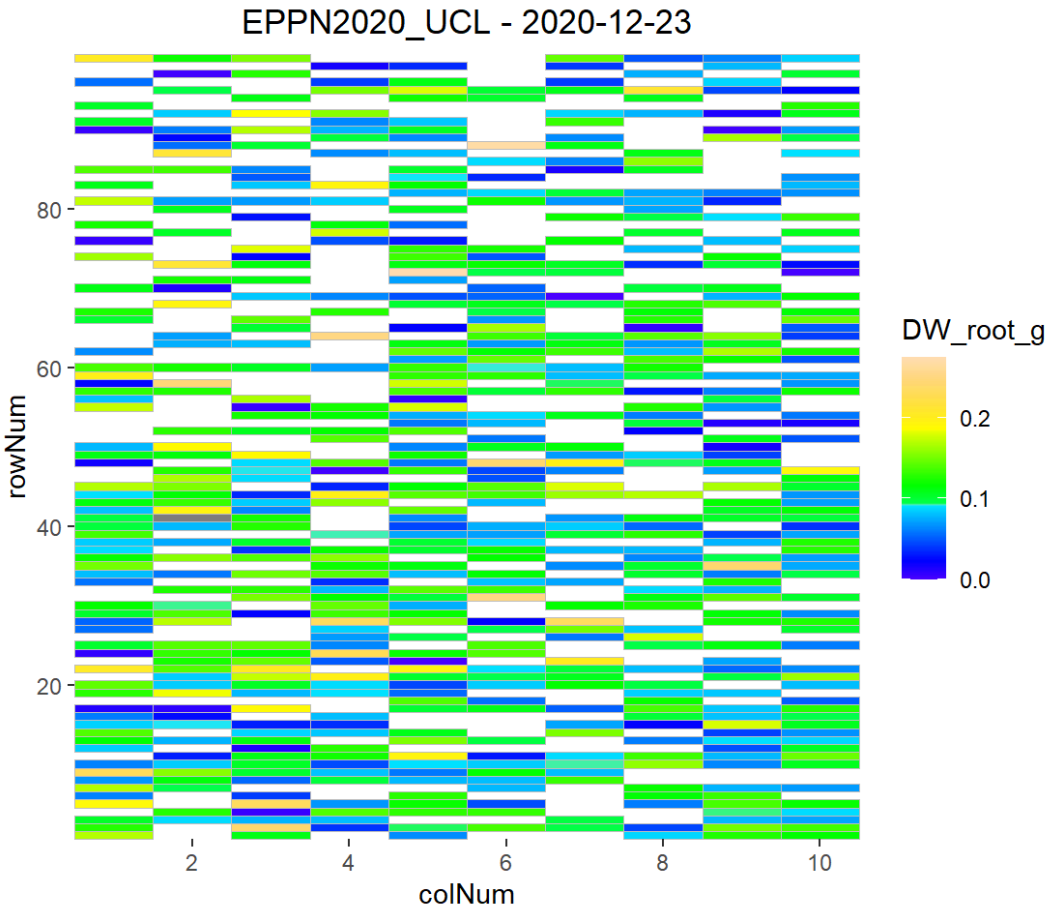
```
## [1] "How many cleaned data observations for DW_shoot_g"
## 2020-12-23
##        580
## [1] "How many cleaned data observations for FW_shoot_g"
## 2020-12-23
##        585
## [1] "How many cleaned data observations for DW_root_g"
## 2020-12-23
##        593
## [1] "How many cleaned data observations for FW_root_g"
## 2020-12-23
##        585
```

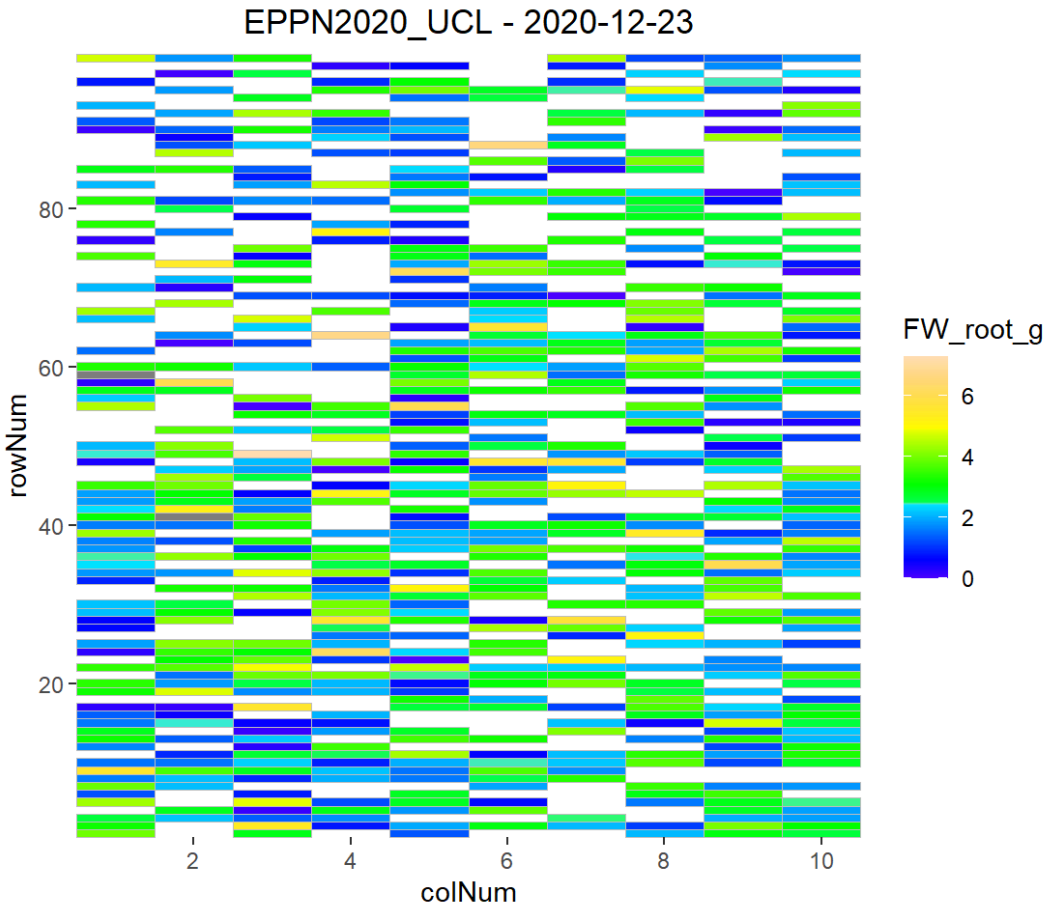
Check the heatmap of the data at harvest

```
for (trait_name in traits) {
  plot(timePoint_endpoint,
     plotType = "layout",
     timePoints = 1,
     traits = trait_name)
}
```

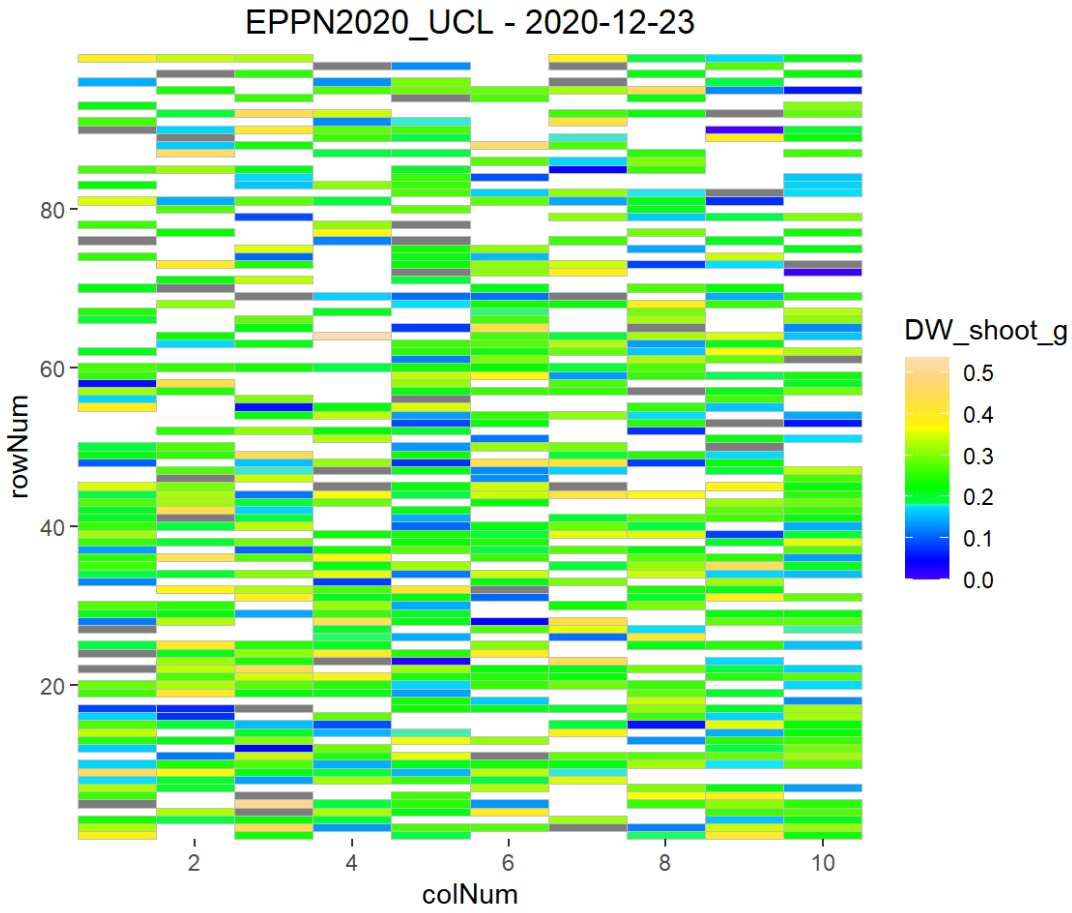

EPPN2020_UCL - 2020-12-23
rowNum
80
60
40
20
2
4
6
8
10
colNum
DW_shoot_g
0.5
0.4
0.3
0.2
0.1
0.0

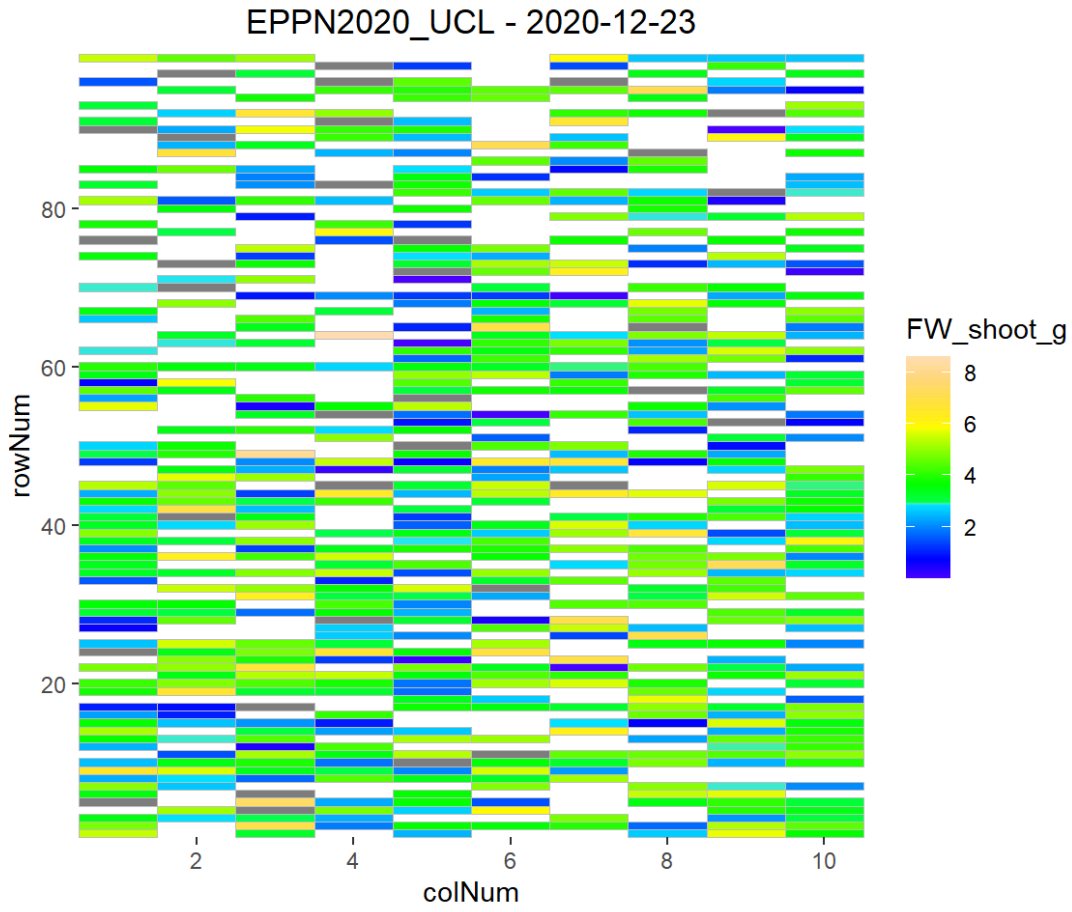

EPPN2020_UCL - 2020-12-23
rowNum
80
60
40
20
2
4
6
8
10
colNum
FW_shoot_g
8
6
4
2
0

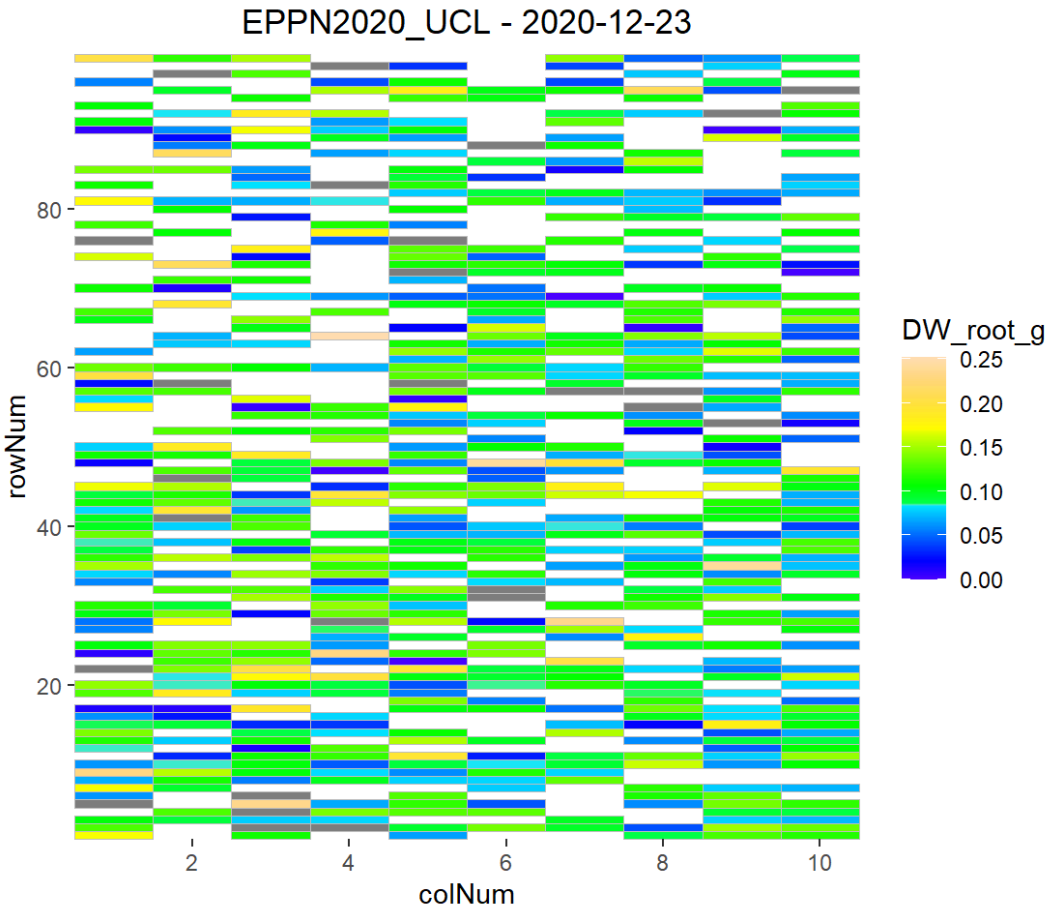

EPPN2020_UCL - 2020-12-23
rowNum
colNum
DW_root_g
0.2
0.1
0.0

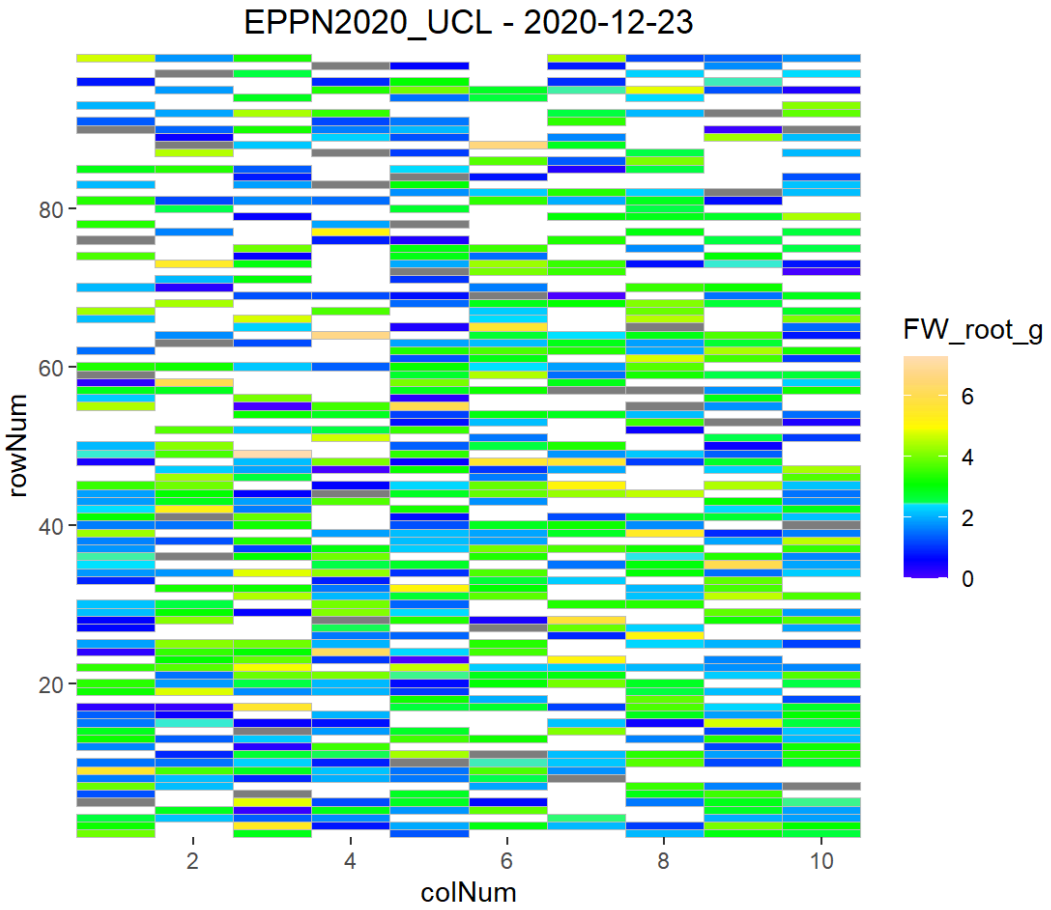

EPPN2020_UCL - 2020-12-23
rowNum
colNum
FW_root_g
6
4
2
0

```
for (trait_name in traits) {
  plot(timePoint_endpoint_clean,
     plotType = "layout",
     timePoints = 1,
     traits = trait_name)
}
```


EPPN2020_UCL - 2020-12-23
rowNum
colNum
DW_shoot_g
0.5
0.4
0.3
0.2
0.1
0.0
20
40
60
80
2
4
6
8
10


EPPN2020_UCL - 2020-12-23
rowNum
colNum
FW_shoot_g
8
6
4
2
20
40
60
80
2
4
6
8
10


EPPN2020_UCL - 2020-12-23
rowNum
colNum
DW_root_g
0.25
0.20
0.15
0.10
0.05
0.00
20
40
60
80
2
4
6
8
10


EPPN2020_UCL - 2020-12-23
rowNum
colNum
FW_root_g
6
4
2
0
20
40
60
80
2
4
6
8
10