M3P Data Analysis Timepoints

Elise

2024-06-09

Set the right working directory.

```
rm(list = ls())

setwd("C:/Users/elise/Documents/Mémoire/Main/Data/Templates/M3P")
platform <- "M3P"
```

Data importation

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

```
list.files()
```

```
##  [1] "endpoint.txt"
##  [2] "plant_info.txt"
##  [3] "S_timeseries.txt"
##  [4] "T_timeseries.txt"
##  [5] "timeseries.txt"
##  [6] "timeseries_Leaf_number.txt"
##  [7] "timeseries_Ligulated_leaf_number.txt"
##  [8] "timeseries_Plant_emergence.txt"
##  [9] "timeseries_Plant_transpiration.txt"
## [10] "timeseries_Soil_water_potential.txt"
## [11] "timeseries_Water.txt"
```

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

# timeseries
# No data for M3P

# S_timeseries
matching_cols <- intersect(names(S_timeseries), names(plant_info))
S_timeseries[, matching_cols] <- lapply(S_timeseries[, matching_cols], factor)
S_timeseries$Timestamp <- as.POSIXct(S_timeseries$Timestamp, format = "%Y-%m-%d %H:%M:%
S")
S_timeseries$Date <- date(S_timeseries$Date)

# T_timeseries
# No data

platform <- "M3P"

# endpoint
df <- endpoint[,colSums(is.na(endpoint))<nrow(endpoint)]
genotype_index <- which(colnames(df) == "Genotype")
variables <- "DW_plant_g"

# timeseries
# too much data

# S_timeseries
df_S_timeseries <- S_timeseries[,colSums(is.na(S_timeseries))<nrow(S_timeseries)]
genotype_index <- which(colnames(df_S_timeseries) == "Genotype")
variables_S <- colnames(df_S_timeseries[, c(5:(genotype_index - 1))]) # We remove the t
hree first columns that are "Unit.ID","Time" and "Date"

# T_timeseries
# no data

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

```
## [1] "M3P : The variables for endpoint are DW_plant_g"
```

```
print(paste(platform, ": The variables for S_timeseries are", paste(variables_S, collap
se = ", "), sep = " "))
```

```
## [1] "M3P : The variables for S_timeseries are S_Height_cm, S_Leaf_area_cmsquared"
```

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



# We add a Plant_type variable that is H or L, with T being L
S_timeseries$Plant_type <- substr(S_timeseries$Genotype, nchar(as.character(S_timeserie
s$Genotype)), nchar(as.character(S_timeseries$Genotype)))

S_timeseries$Plant_type <- ifelse(S_timeseries$Plant_type %in% c("T", "L"), "Line",
                                  ifelse(S_timeseries$Plant_type == "H", "Hybrid", S_ti
meseries$Plant_type))
```

Time point objects

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

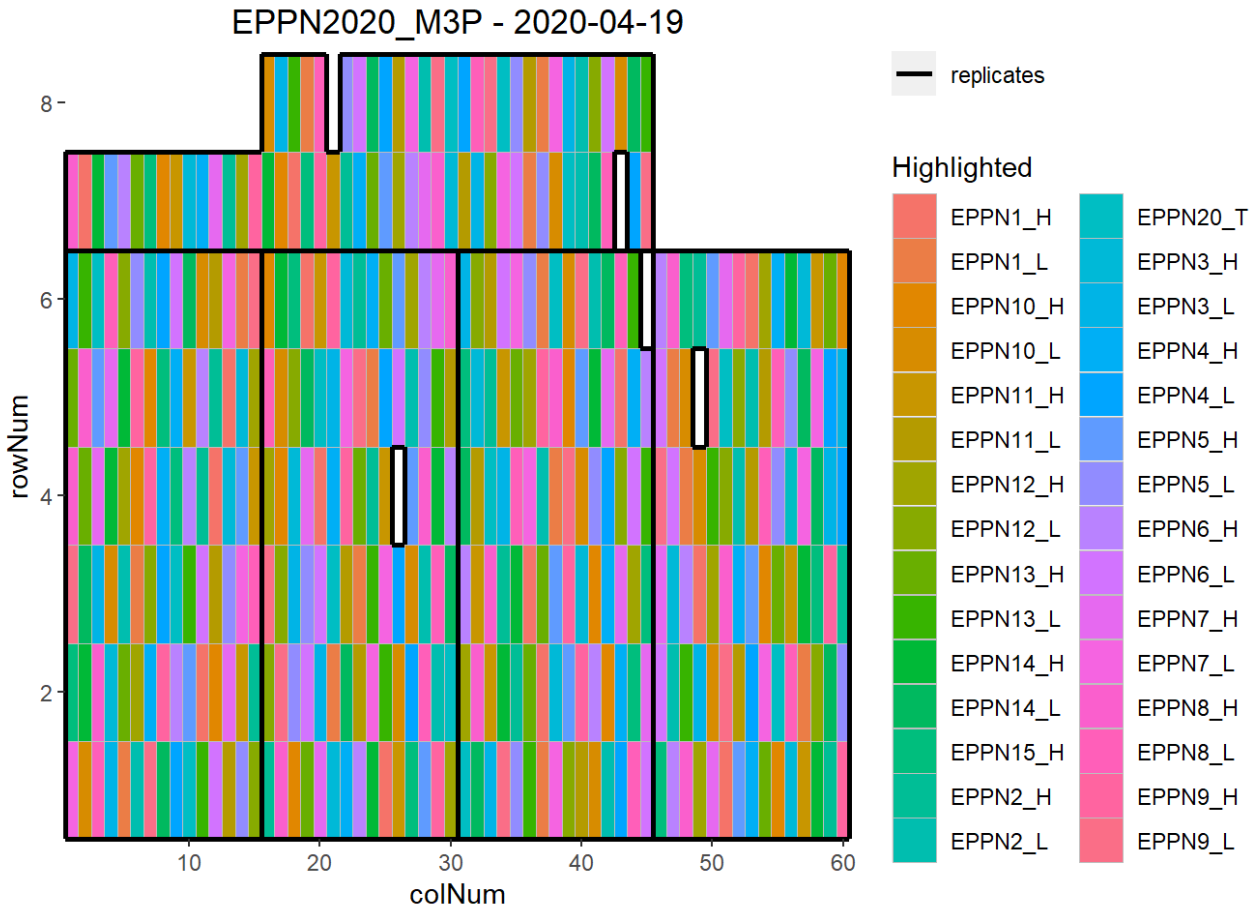
```
timePoint_endpoint <- createTimePoints(dat = endpoint,
                                       experimentName = "EPPN2020_M3P",
                                       genotype = "Genotype",
                                       timePoint = "Date",
                                       plotId = "Unit.ID",
                                       rowNum = "Row",
                                       colNum = "Column",
                                       repId = "Replication")

timePoint_endpoint_clean <- createTimePoints(dat = endpoint_clean,
                                       experimentName = "EPPN2020_M3P",
                                       genotype = "Genotype",
                                       timePoint = "Date",
                                       plotId = "Unit.ID",
                                       rowNum = "Row",
                                       colNum = "Column",
                                       repId = "Replication")

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

timePoint_S <- createTimePoints(dat = S_timeseries,
                                experimentName = "EPPN2020_M3P",
                                genotype = "Genotype",
                                timePoint = "Date",
                                plotId = "Unit.ID",
                                rowNum = "Row",
                                colNum = "Column",
                                repId = "Replication",
                                addCheck = TRUE,
                                checkGenotypes = "EPPN20_T")
```

```
## Warning: The following plotIds have observations for less than 50% of the time point
s:
## 148, 426
```

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_M3P.
##
## It contains 1 time points.
## First time point: 2020-04-19
## Last time point: 2020-04-19
##
## No check genotypes are defined.
```

```
getTimePoints(timePoint_endpoint)
```

```
##   timeNumber  timePoint
## 1          1 2020-04-19
```

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)
}
```

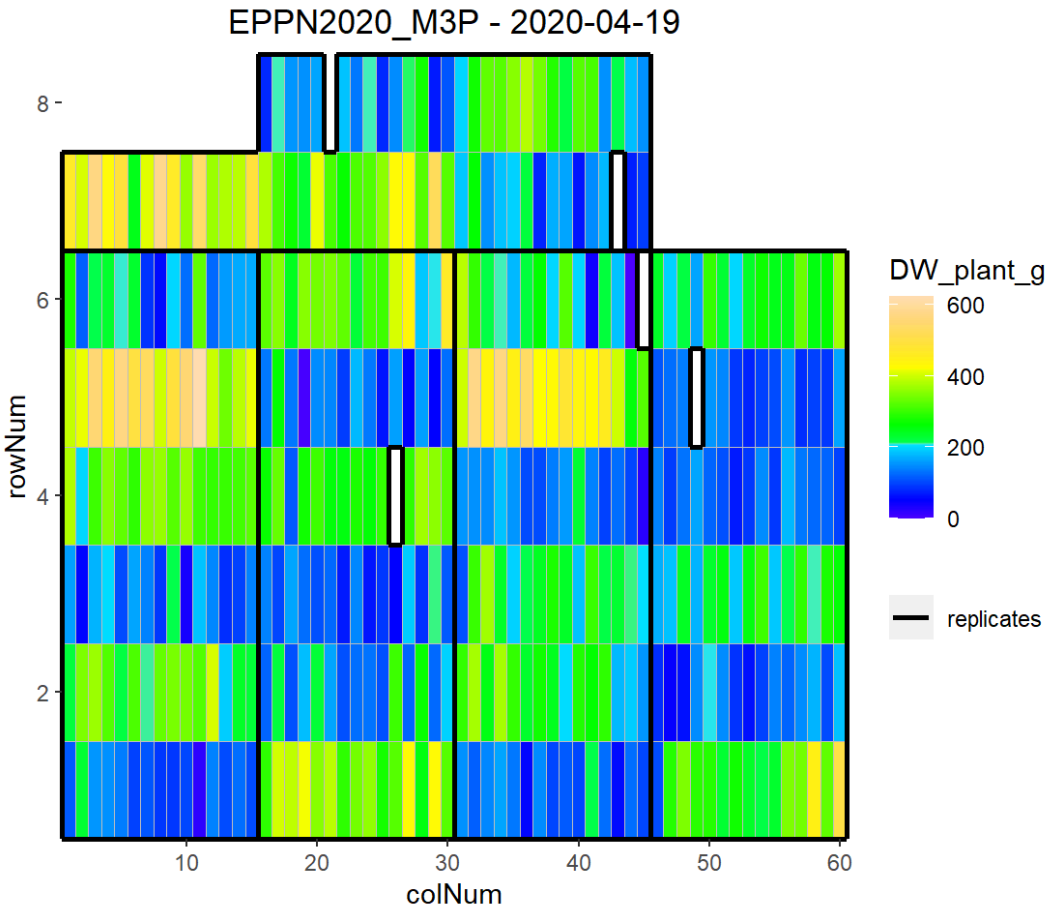
```
## [1] "How many data observations for DW_plant_g"
## 2020-04-19
##        430
```

```
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)
}
```

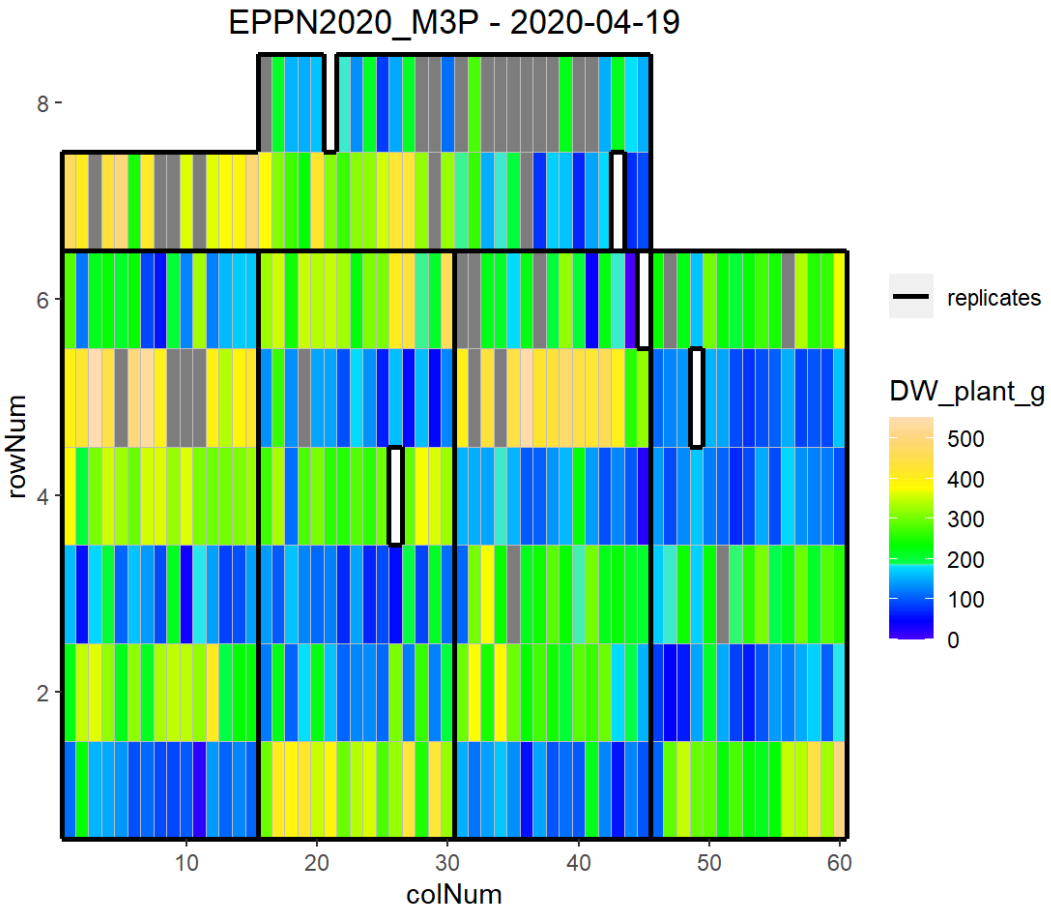
```
## [1] "How many cleaned data observations for DW_plant_g"
## 2020-04-19
##        398
```

Check the heatmap of the data at harvest

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




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



2. S_timeseries

Raw data

View timePoint object

```
summary(timePoint_S)
```

```
## timePoint_S contains data for experiment EPPN2020_M3P.
##
## It contains 42 time points.
## First time point: 2020-02-14
## Last time point: 2020-04-01
##
## The following genotypes are defined as check genotypes: EPPN20_T.
```

```
getTimePoints(timePoint_S)
```

```
##    timeNumber  timePoint
## 1           1 2020-02-14
## 2           2 2020-02-15
## 3           3 2020-02-16
## 4           4 2020-02-17
## 5           5 2020-02-18
## 6           6 2020-02-19
## 7           7 2020-02-20
## 8           8 2020-02-21
## 9           9 2020-02-22
## 10         10 2020-02-23
## 11         11 2020-02-24
## 12         12 2020-02-25
## 13         13 2020-02-26
## 14         14 2020-02-28
## 15         15 2020-02-29
## 16         16 2020-03-01
## 17         17 2020-03-02
## 18         18 2020-03-03
## 19         19 2020-03-04
## 20         20 2020-03-05
## 21         21 2020-03-06
## 22         22 2020-03-07
## 23         23 2020-03-08
## 24         24 2020-03-11
## 25         25 2020-03-12
## 26         26 2020-03-13
## 27         27 2020-03-14
## 28         28 2020-03-15
## 29         29 2020-03-18
## 30         30 2020-03-19
## 31         31 2020-03-20
## 32         32 2020-03-21
## 33         33 2020-03-22
## 34         34 2020-03-24
## 35         35 2020-03-25
## 36         36 2020-03-26
## 37         37 2020-03-27
## 38         38 2020-03-28
## 39         39 2020-03-29
## 40         40 2020-03-30
## 41         41 2020-03-31
## 42         42 2020-04-01
```

```
num_timepoints <- getTimePoints(timePoint_S)
```

Count the number of observations per trait and time point

We focus on the Height [cm] and Leaf area, because these are the two most common among the platforms.

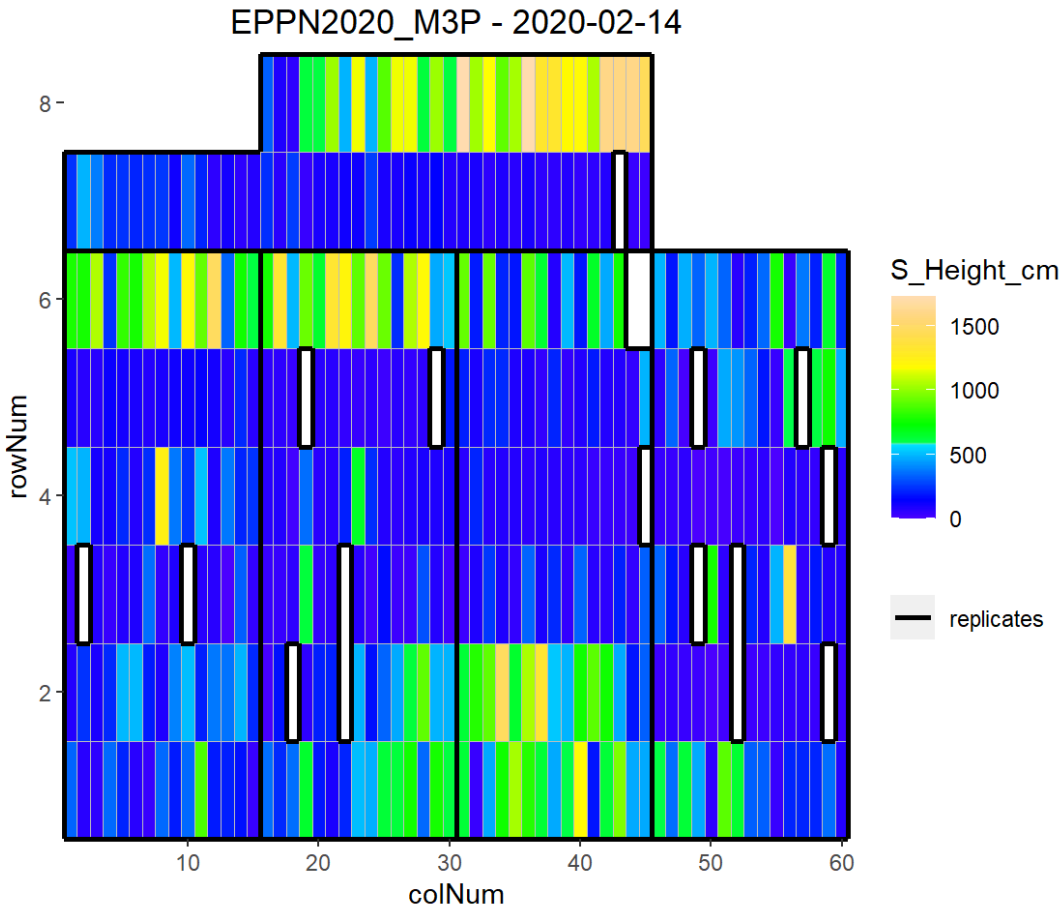
Height is computed for 6 platforms out of 9 and area for 4 out of 9.

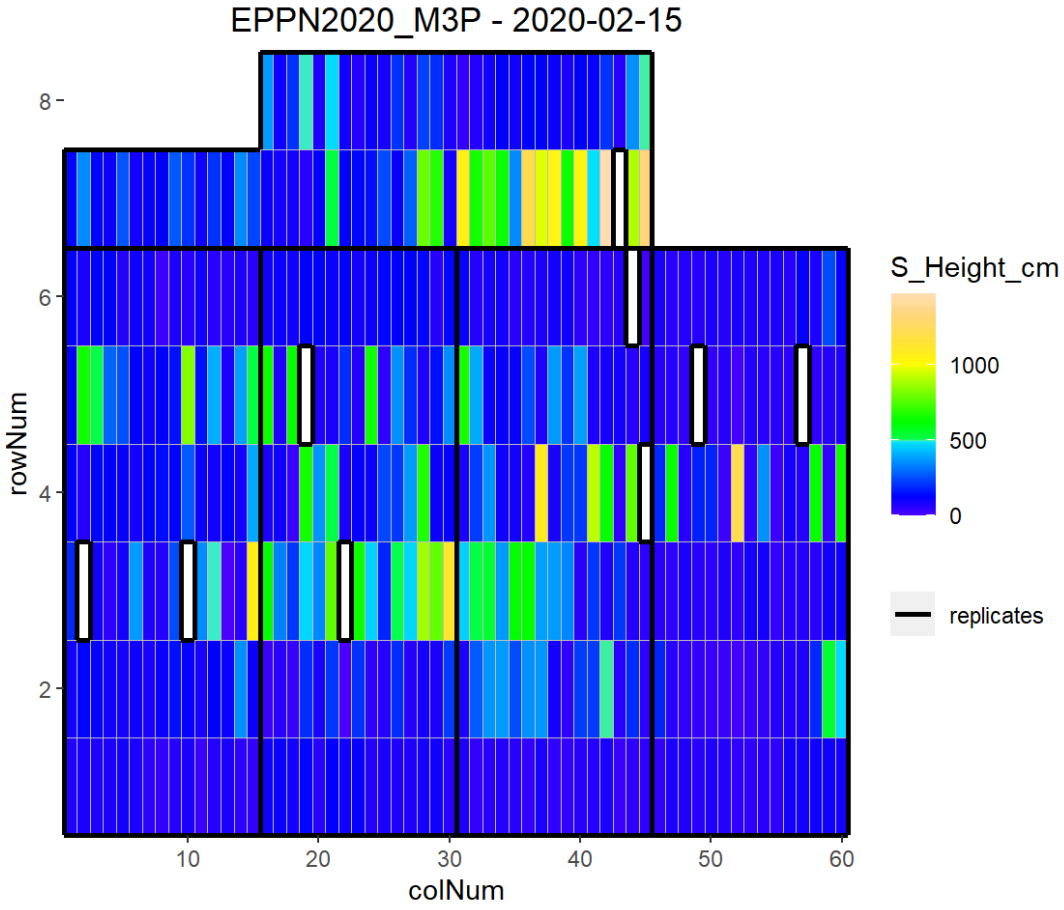
```
var_voulues <- c(variables_S[1], variables_S[2])
traits <- var_voulues

for (trait_name in traits) {
  print(paste("How many observations for", trait_name))
  valid_count <- countValid(timePoint_S, trait_name)
  print(valid_count)
}
```

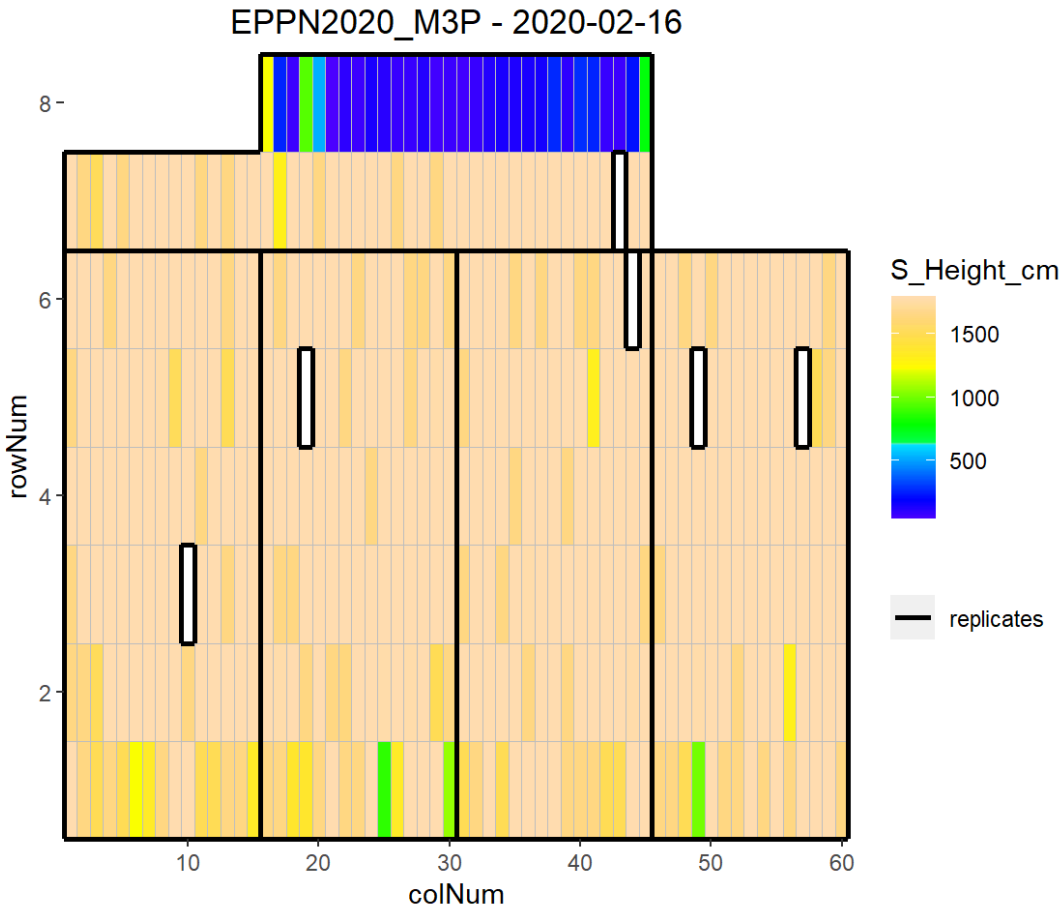
```
## [1] "How many observations for S_Height_cm"
## 2020-02-14 2020-02-15 2020-02-16 2020-02-17 2020-02-18 2020-02-19 2020-02-20
##        417        426        429        430        430        430        429
## 2020-02-21 2020-02-22 2020-02-23 2020-02-24 2020-02-25 2020-02-26 2020-02-28
##        430        430          8        430        430        430        271
## 2020-02-29 2020-03-01 2020-03-02 2020-03-03 2020-03-04 2020-03-05 2020-03-06
##        314        273        114        430        313        117        377
## 2020-03-07 2020-03-08 2020-03-11 2020-03-12 2020-03-13 2020-03-14 2020-03-15
##         46        418        379         51        428        213        214
## 2020-03-18 2020-03-19 2020-03-20 2020-03-21 2020-03-22 2020-03-24 2020-03-25
##        371         58        356         72        428        429        429
## 2020-03-26 2020-03-27 2020-03-28 2020-03-29 2020-03-30 2020-03-31 2020-04-01
##        401        429        429        368        429        429        429
## [1] "How many observations for S_Leaf_area_cmsquared"
## 2020-02-14 2020-02-15 2020-02-16 2020-02-17 2020-02-18 2020-02-19 2020-02-20
##        408        426        429        430        430        430        429
## 2020-02-21 2020-02-22 2020-02-23 2020-02-24 2020-02-25 2020-02-26 2020-02-28
##        430        430          8        430        430        430        271
## 2020-02-29 2020-03-01 2020-03-02 2020-03-03 2020-03-04 2020-03-05 2020-03-06
##        314        273        114        430        313        117        377
## 2020-03-07 2020-03-08 2020-03-11 2020-03-12 2020-03-13 2020-03-14 2020-03-15
##         46        418        379         51        428        213        214
## 2020-03-18 2020-03-19 2020-03-20 2020-03-21 2020-03-22 2020-03-24 2020-03-25
##        371         58        356         72        428        429        429
## 2020-03-26 2020-03-27 2020-03-28 2020-03-29 2020-03-30 2020-03-31 2020-04-01
##        401        429        429        368        429        429        429
```

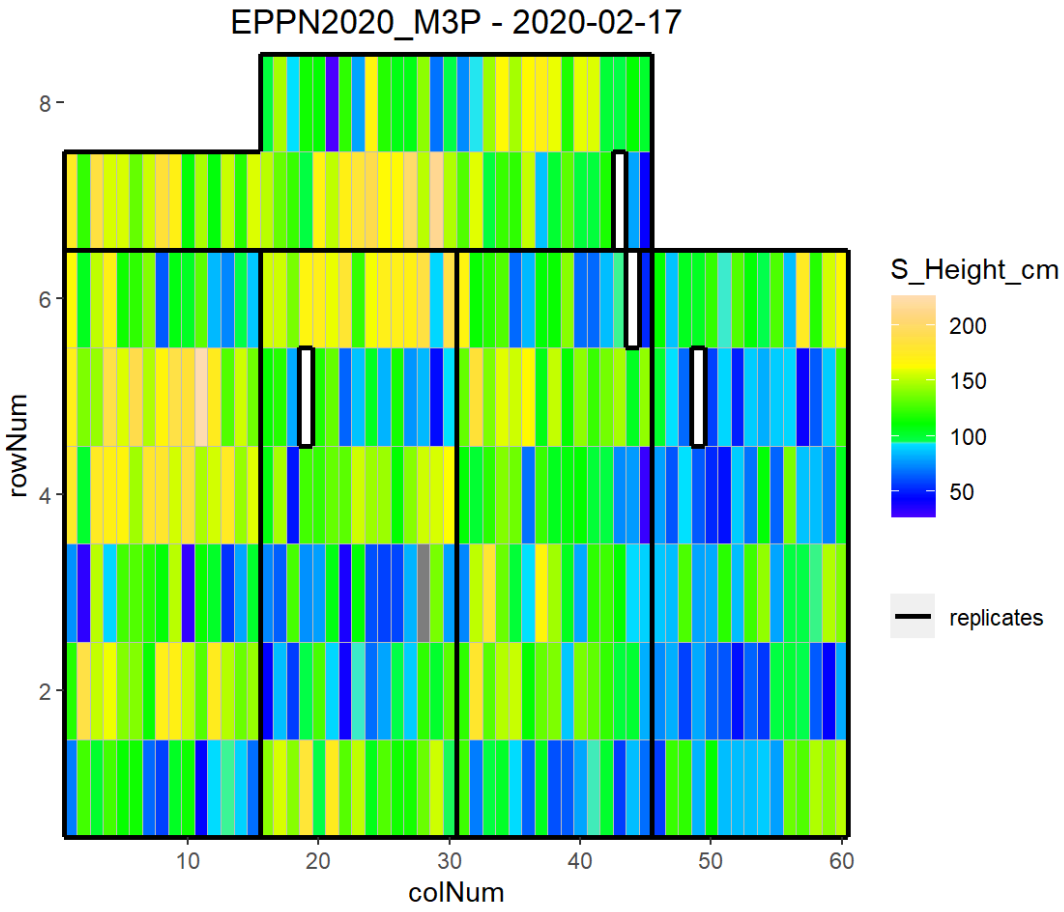
Check the heatmap of the raw data per time point

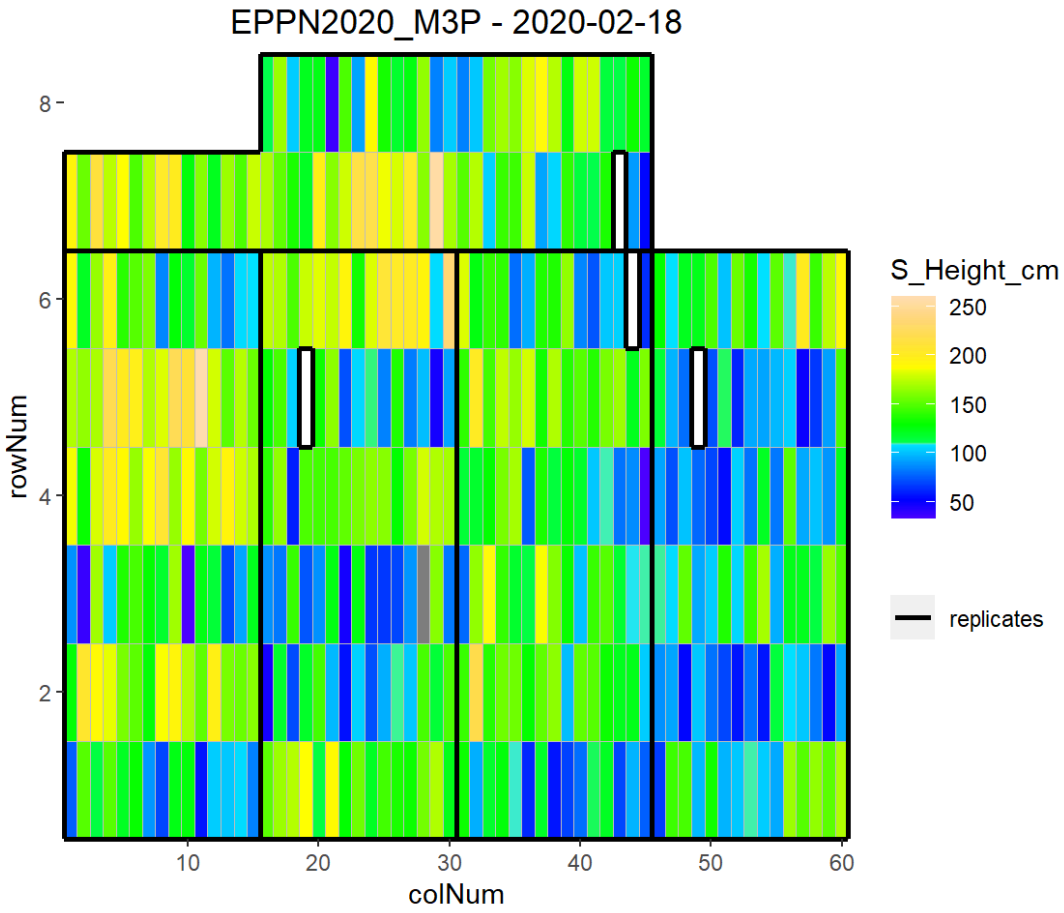
```
for (trait_name in traits) {
  for (tp in 1:length(num_timepoints$timeNumber)) {
    plot(timePoint_S,
         plotType = "layout",
         timePoints = tp,
         traits = trait_name)
  }
}
```

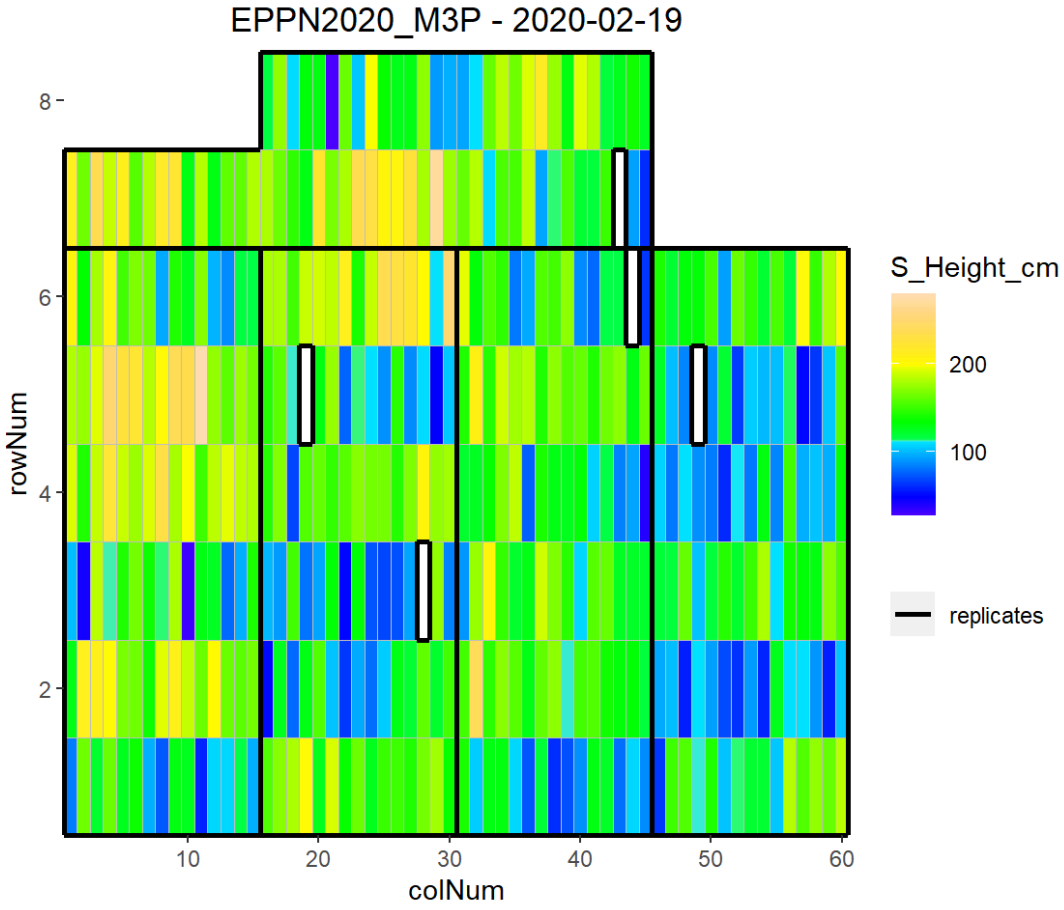

EPPN2020_M3P - 2020-02-14
rowNum
colNum
S_Height_cm
1500
1000
500
0
replicates

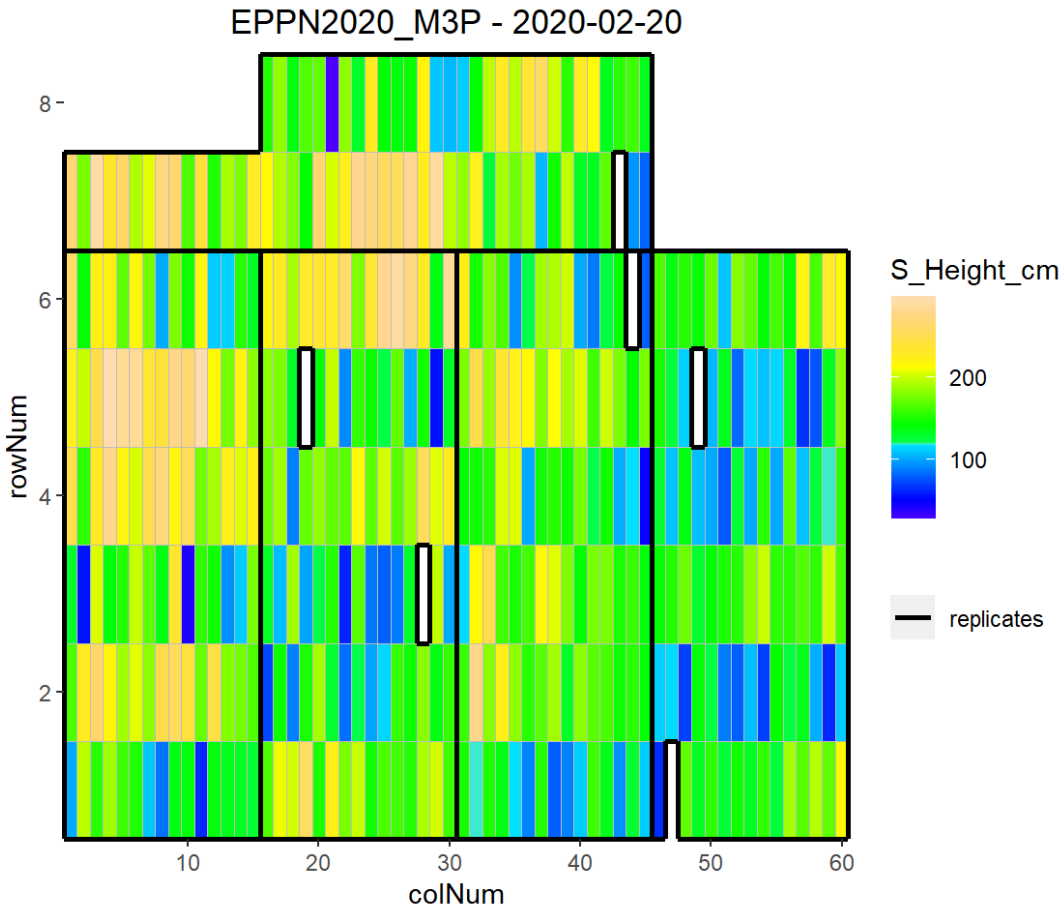

EPPN2020_M3P - 2020-02-15
rowNum
colNum
S_Height_cm
1000
500
0
replicates

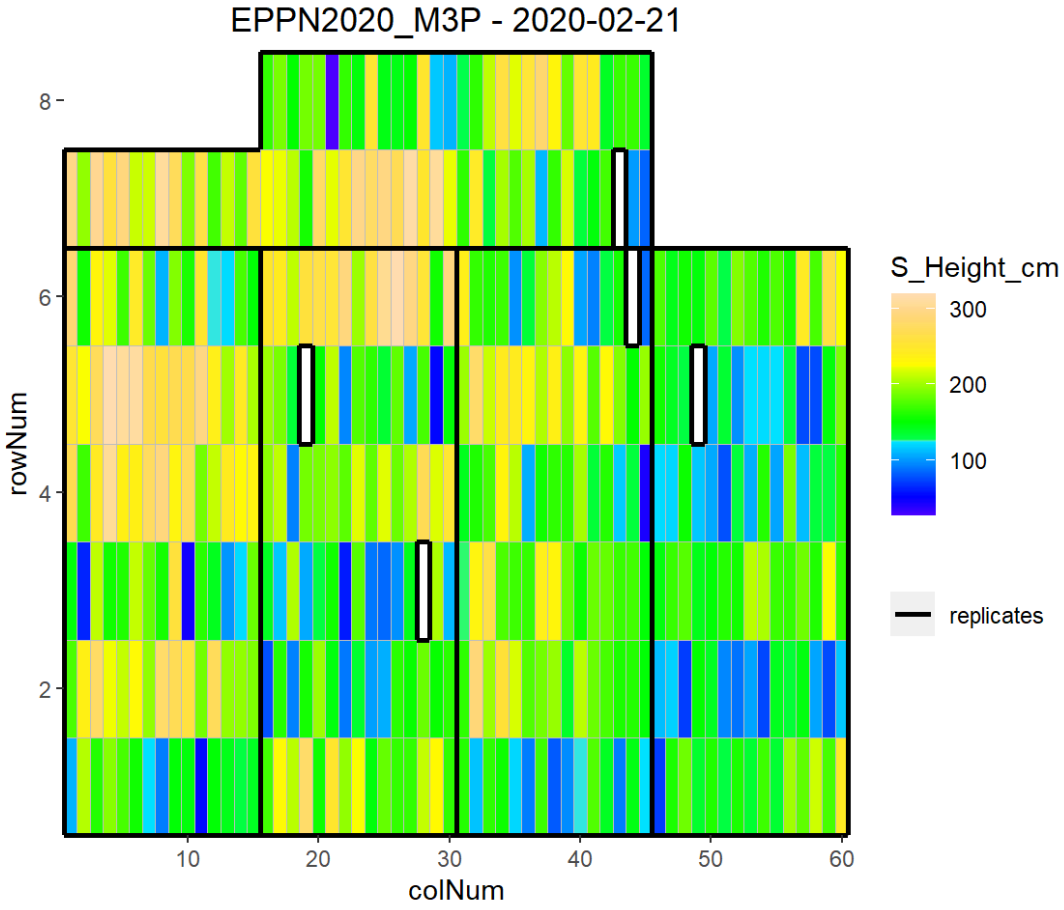

EPPN2020_M3P - 2020-02-16
rowNum
colNum
S_Height_cm
1500
1000
500
replicates

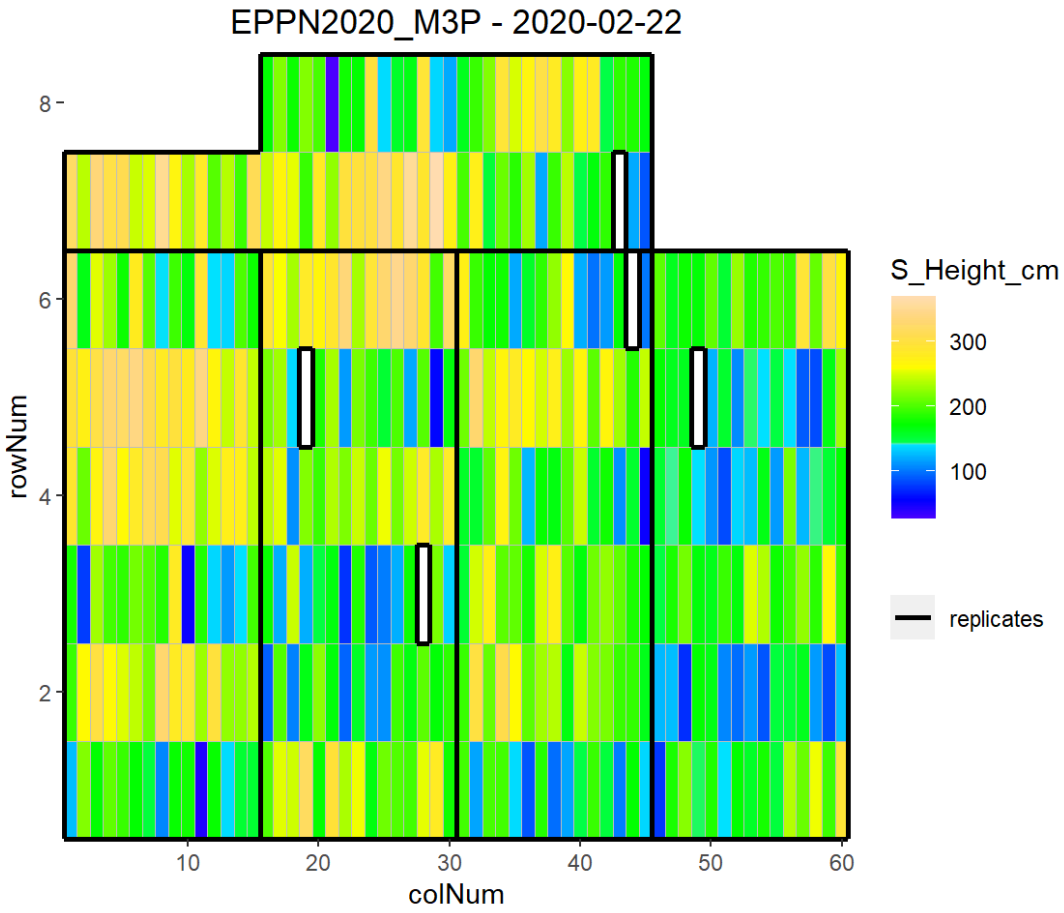

EPPN2020_M3P - 2020-02-17
rowNum
colNum
S_Height_cm
200
150
100
50
replicates

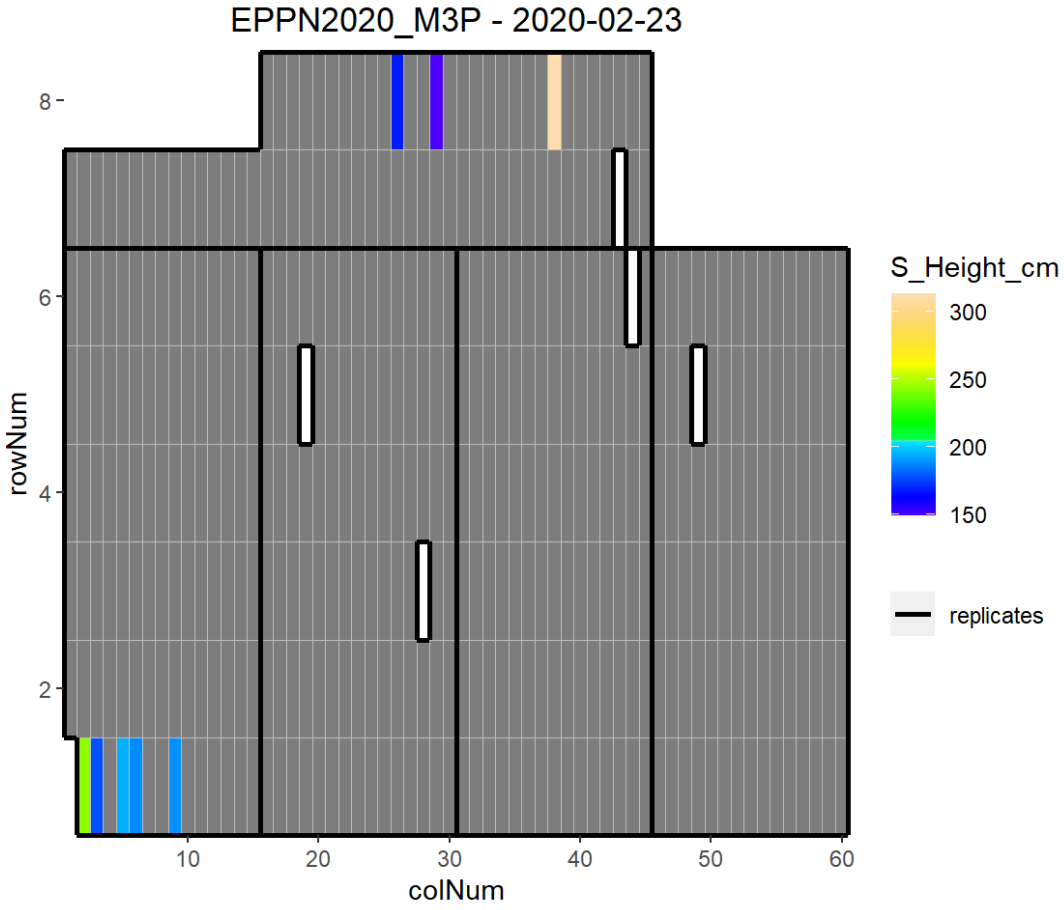

EPPN2020_M3P - 2020-02-18
rowNum
colNum
S_Height_cm
250
200
150
100
50
replicates

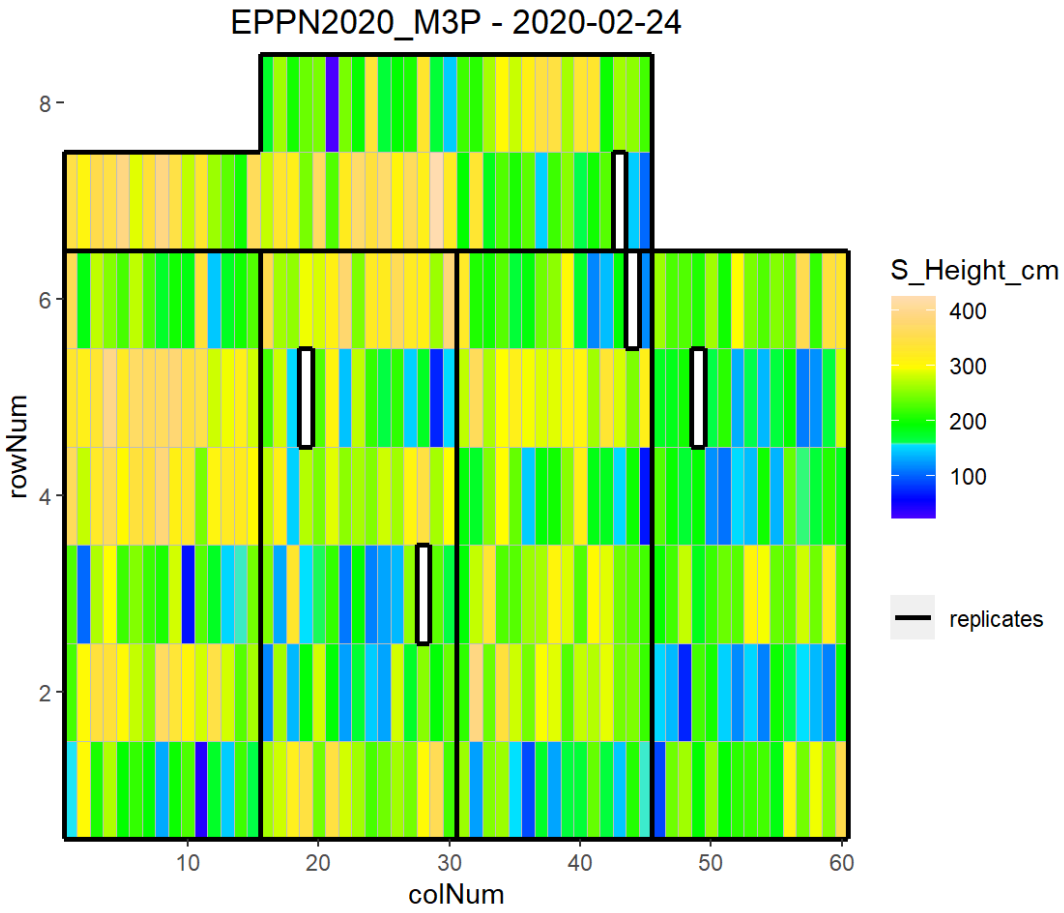

EPPN2020_M3P - 2020-02-19
rowNum
colNum
S_Height_cm
200
100
replicates

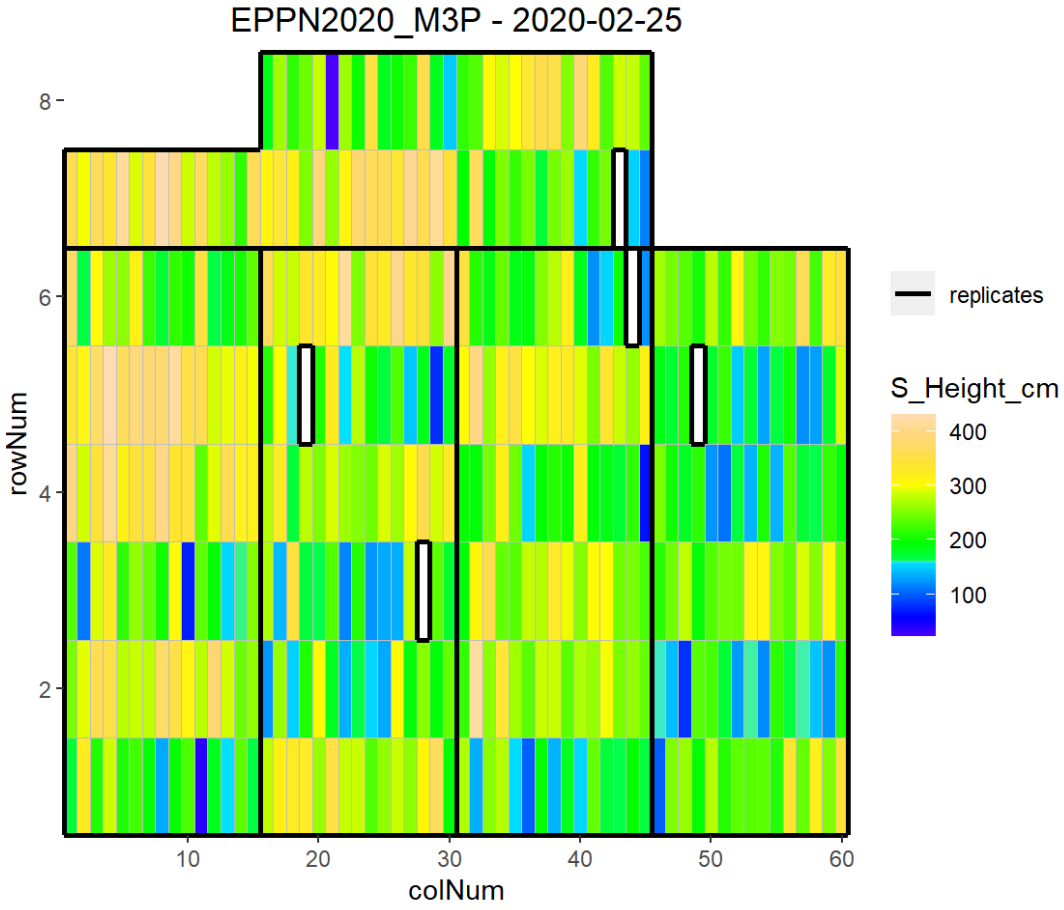

EPPN2020_M3P - 2020-02-20
rowNum
colNum
S_Height_cm
200
100
replicates

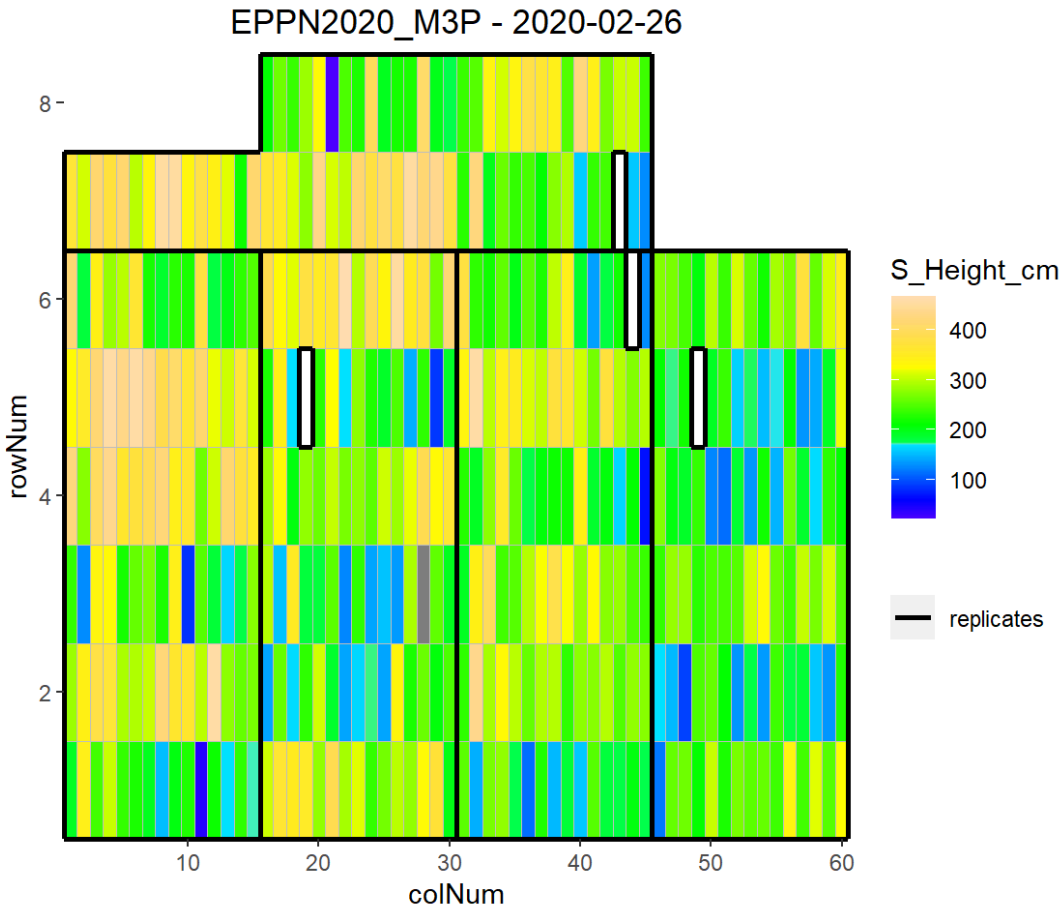

EPPN2020_M3P - 2020-02-21
rowNum
colNum
S_Height_cm
300
200
100
replicates

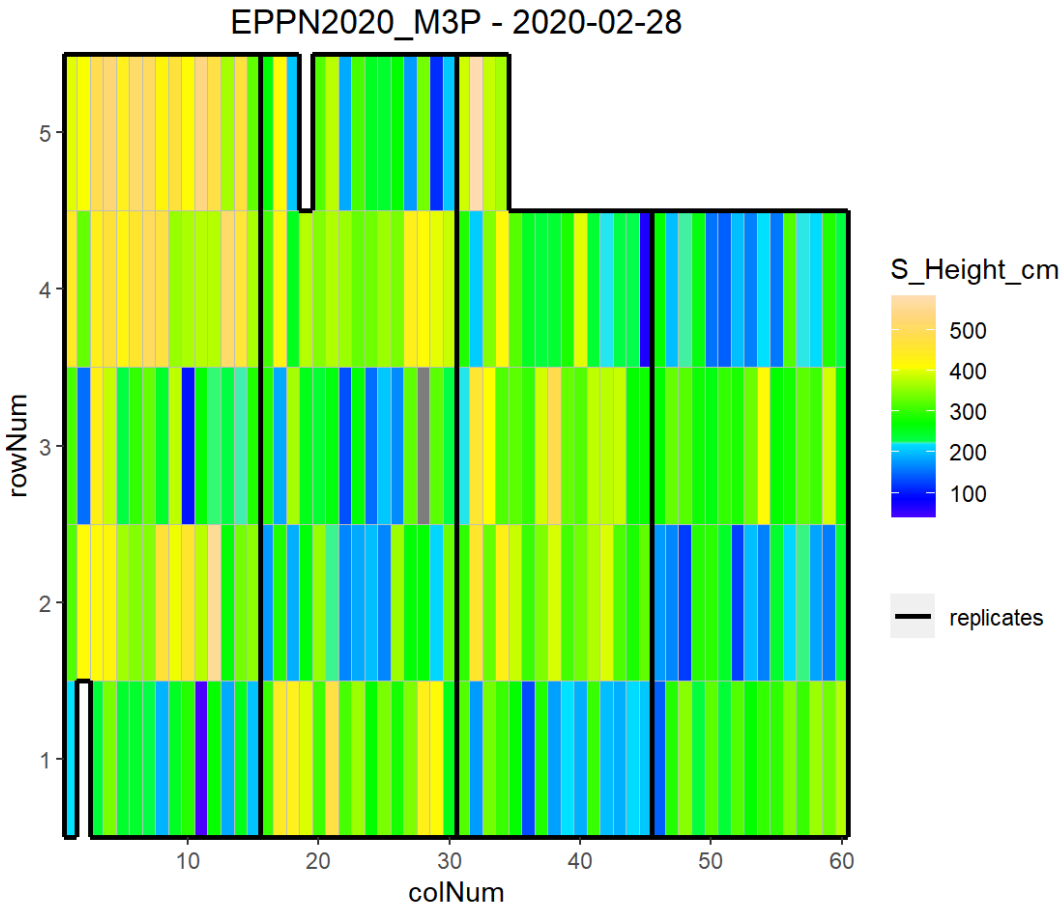

EPPN2020_M3P - 2020-02-22
rowNum
colNum
S_Height_cm
300
200
100
replicates

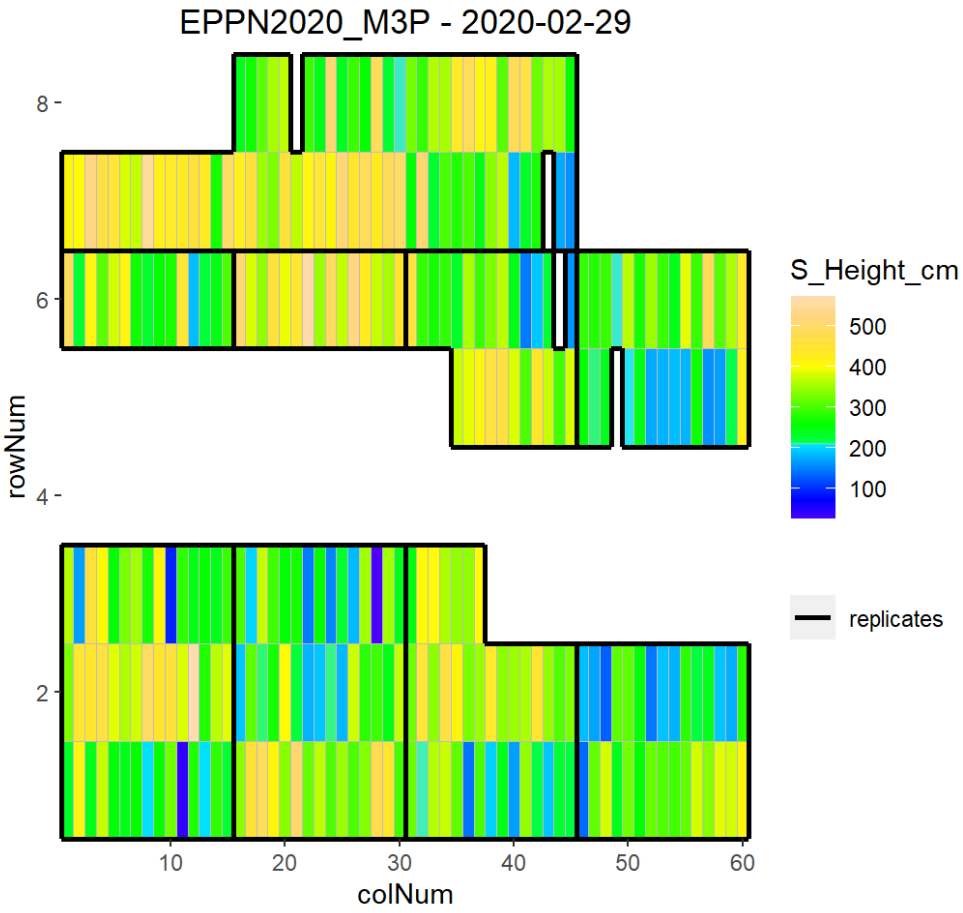

EPPN2020_M3P - 2020-02-23
rowNum
colNum
S_Height_cm
300
250
200
150
replicates

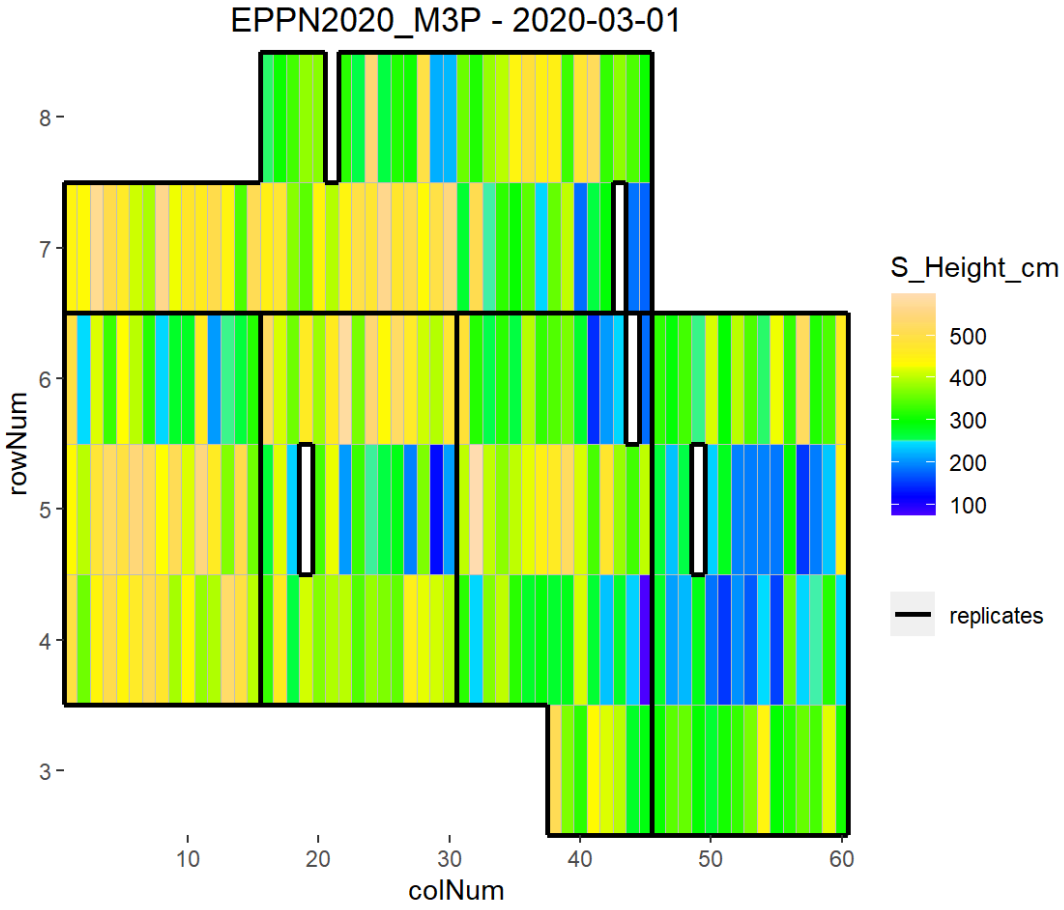

EPPN2020_M3P - 2020-02-24
rowNum
colNum
S_Height_cm
400
300
200
100
replicates

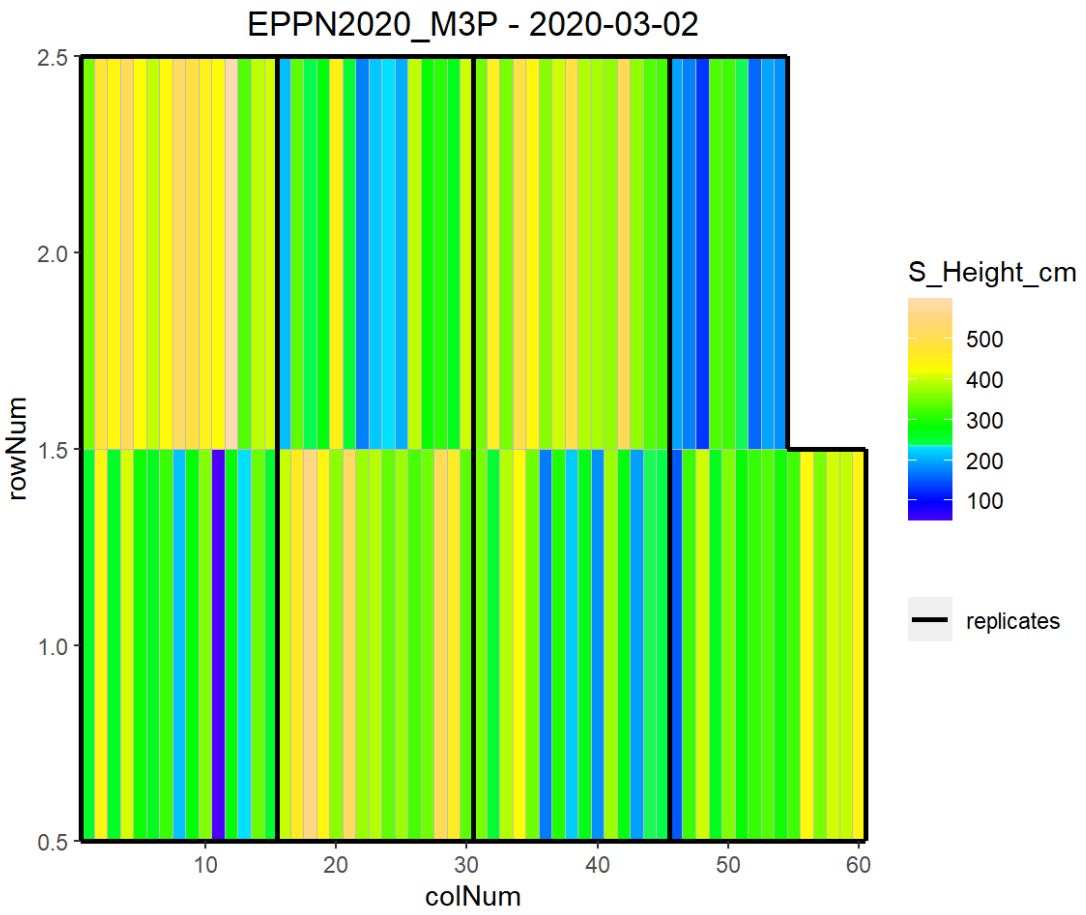

EPPN2020_M3P - 2020-02-25
rowNum
colNum
replicates
S_Height_cm
400
300
200
100

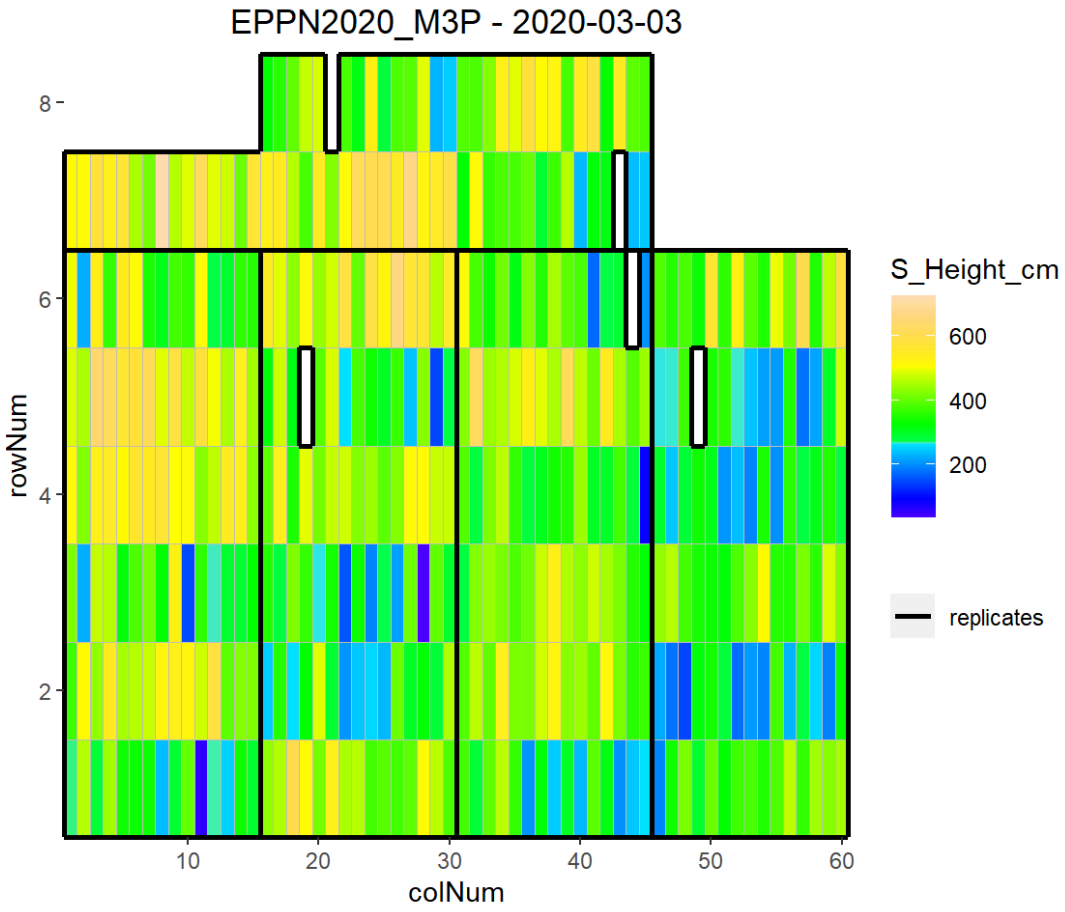

EPPN2020_M3P - 2020-02-26
rowNum
colNum
S_Height_cm
400
300
200
100
replicates

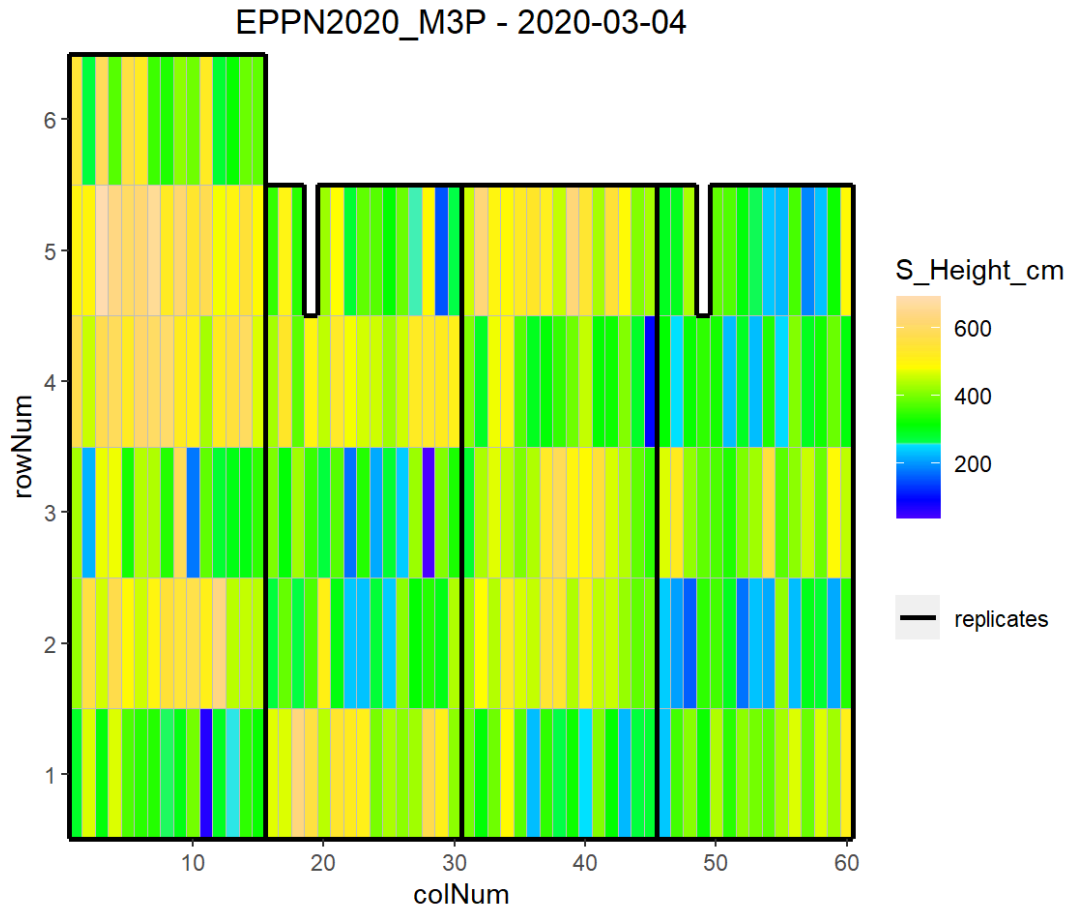

EPPN2020_M3P - 2020-02-28
rowNum
colNum
S_Height_cm
500
400
300
200
100
replicates

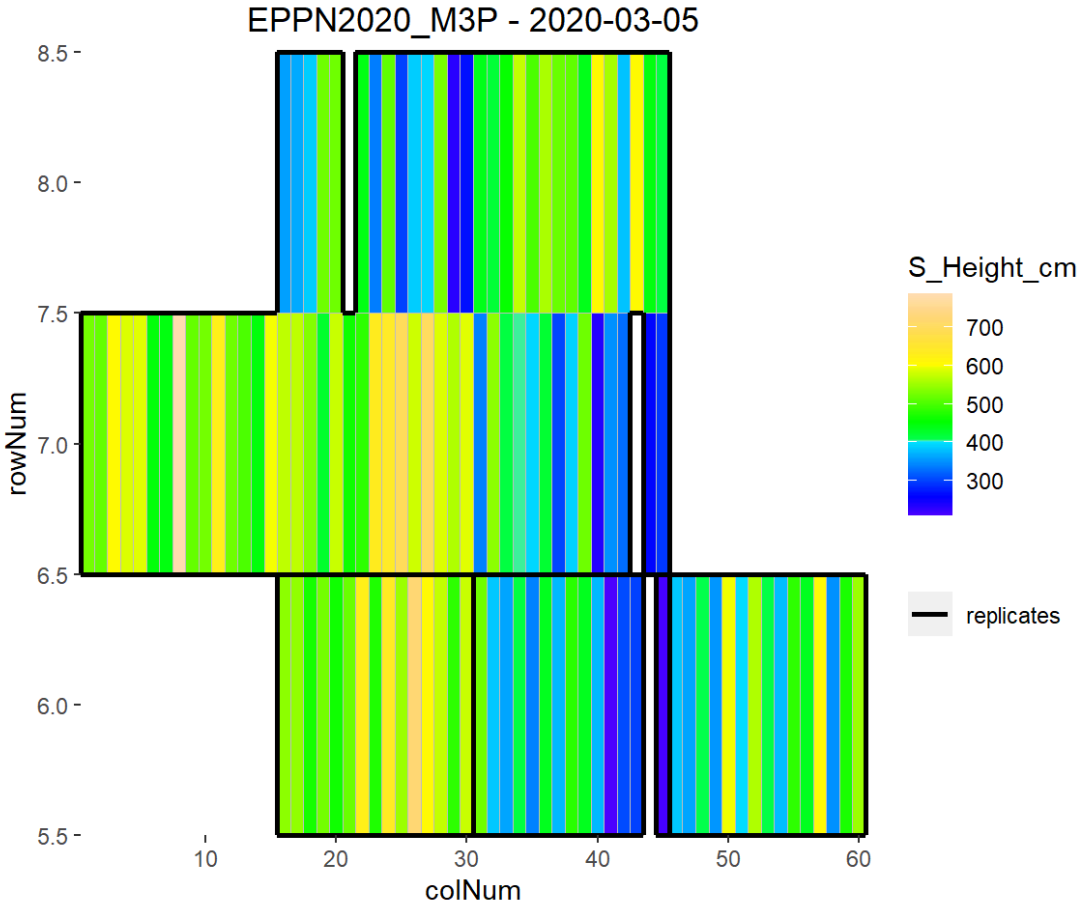

EPPN2020_M3P - 2020-02-29
rowNum
colNum
S_Height_cm
500
400
300
200
100
replicates

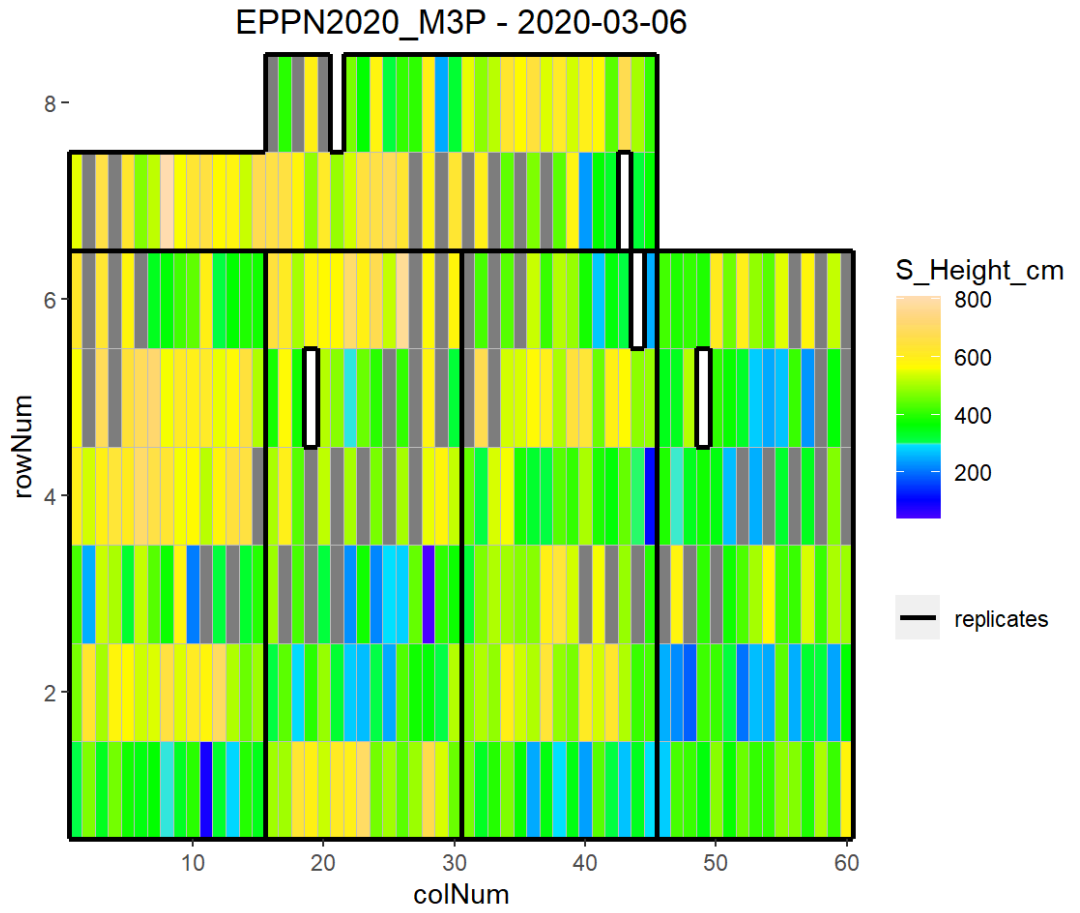

EPPN2020_M3P - 2020-03-01
rowNum
colNum
S_Height_cm
500
400
300
200
100
replicates

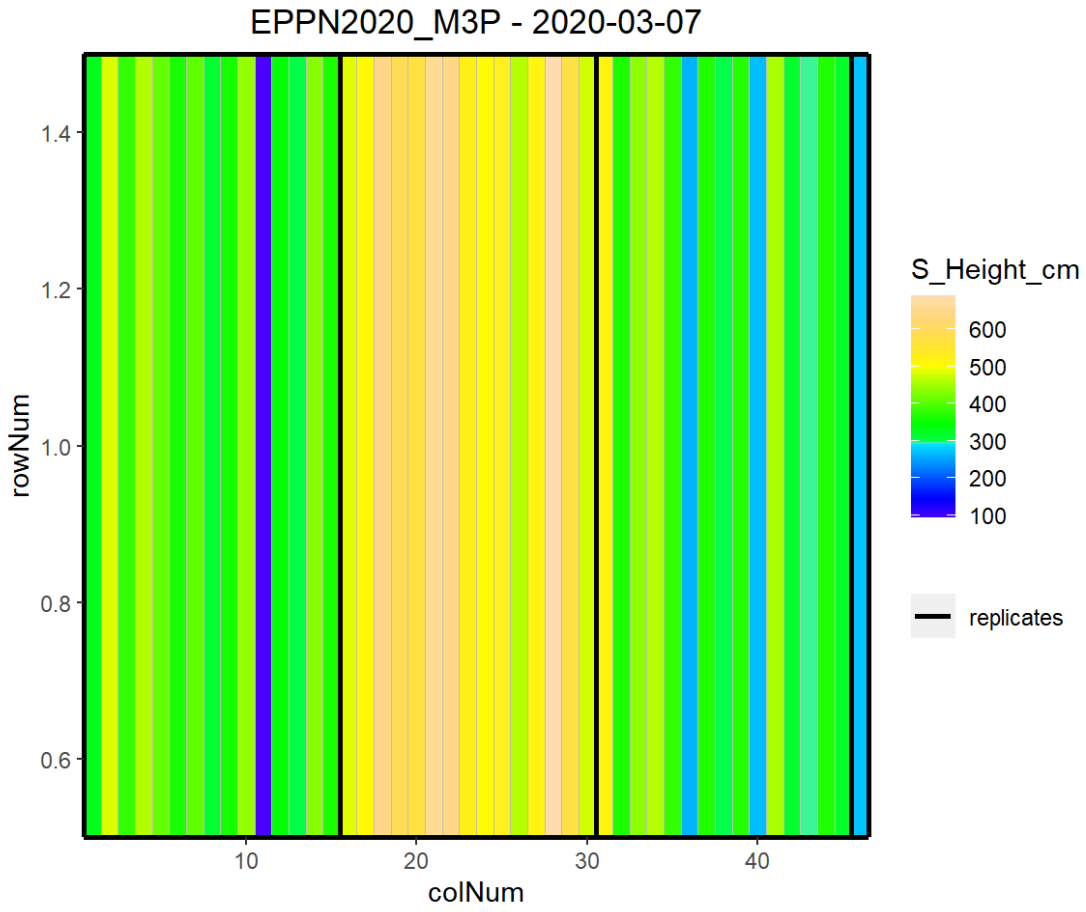

EPPN2020_M3P - 2020-03-02
rowNum
colNum
S_Height_cm
500
400
300
200
100
replicates

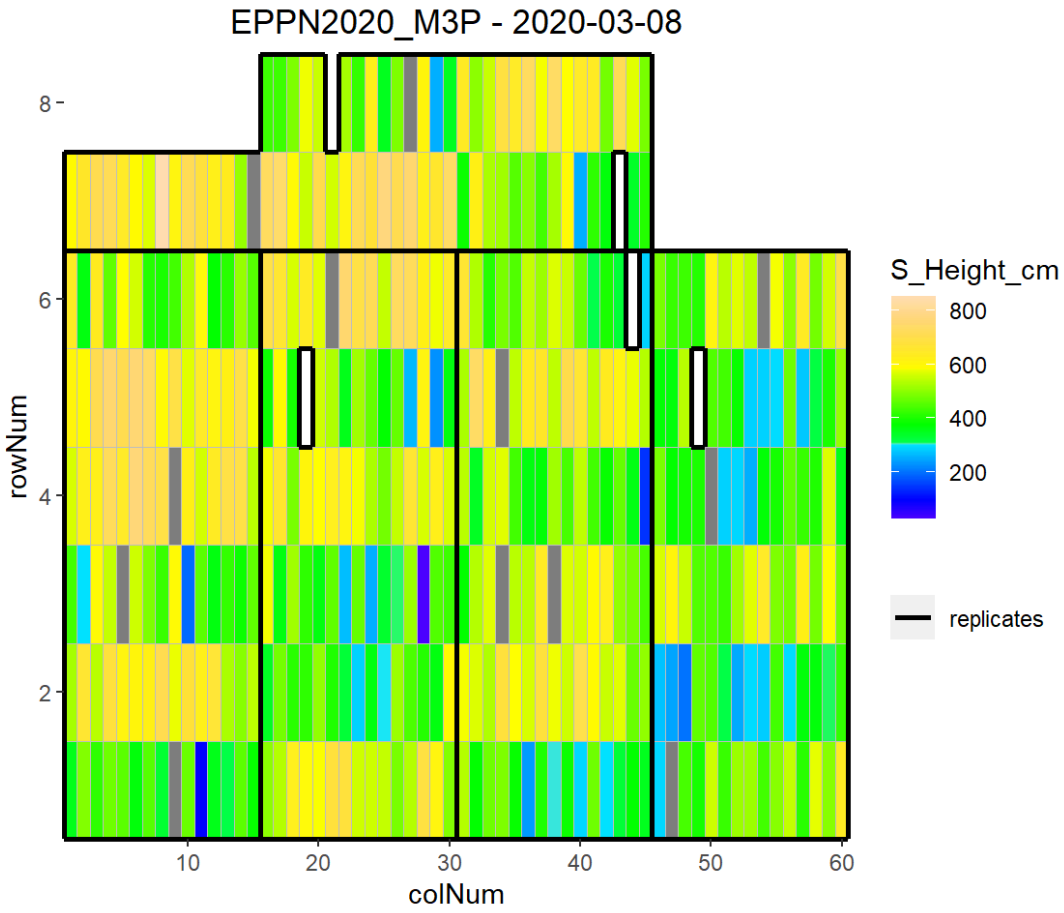

EPPN2020_M3P - 2020-03-03
rowNum
colNum
S_Height_cm
600
400
200
replicates

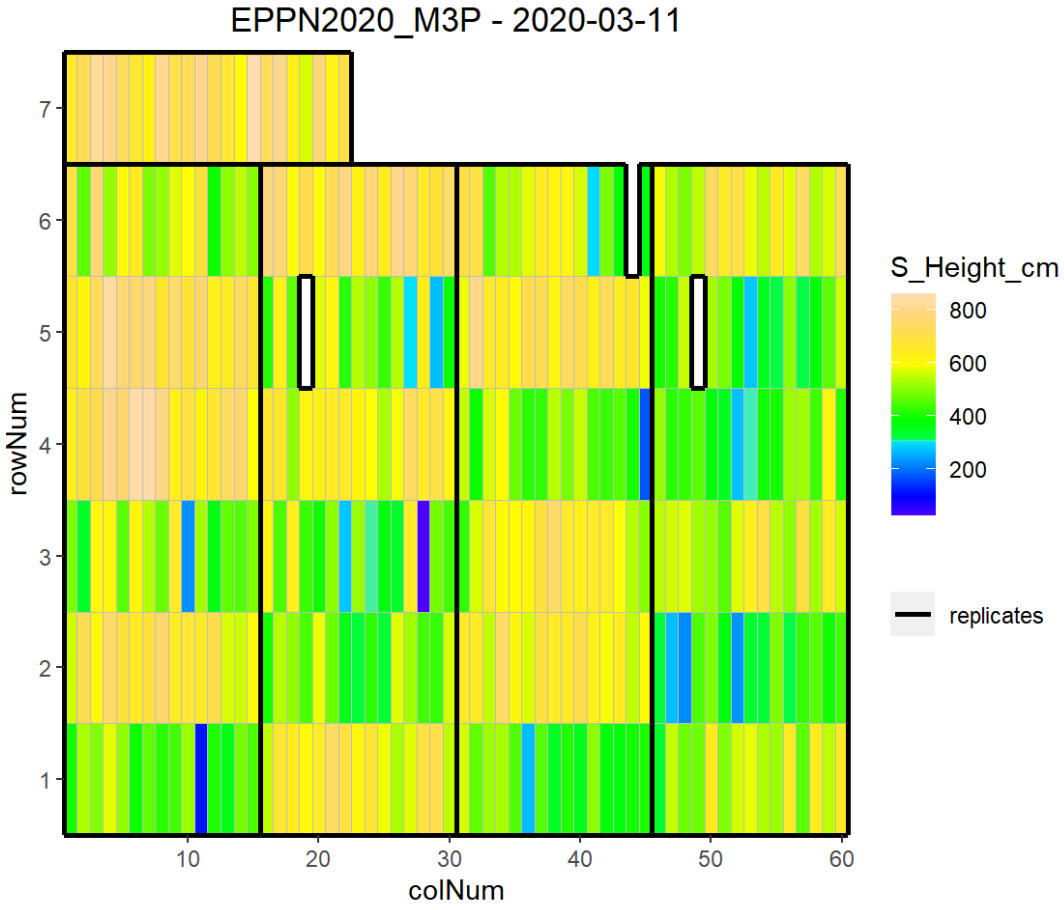

EPPN2020_M3P - 2020-03-04
rowNum
colNum
S_Height_cm
600
400
200
replicates

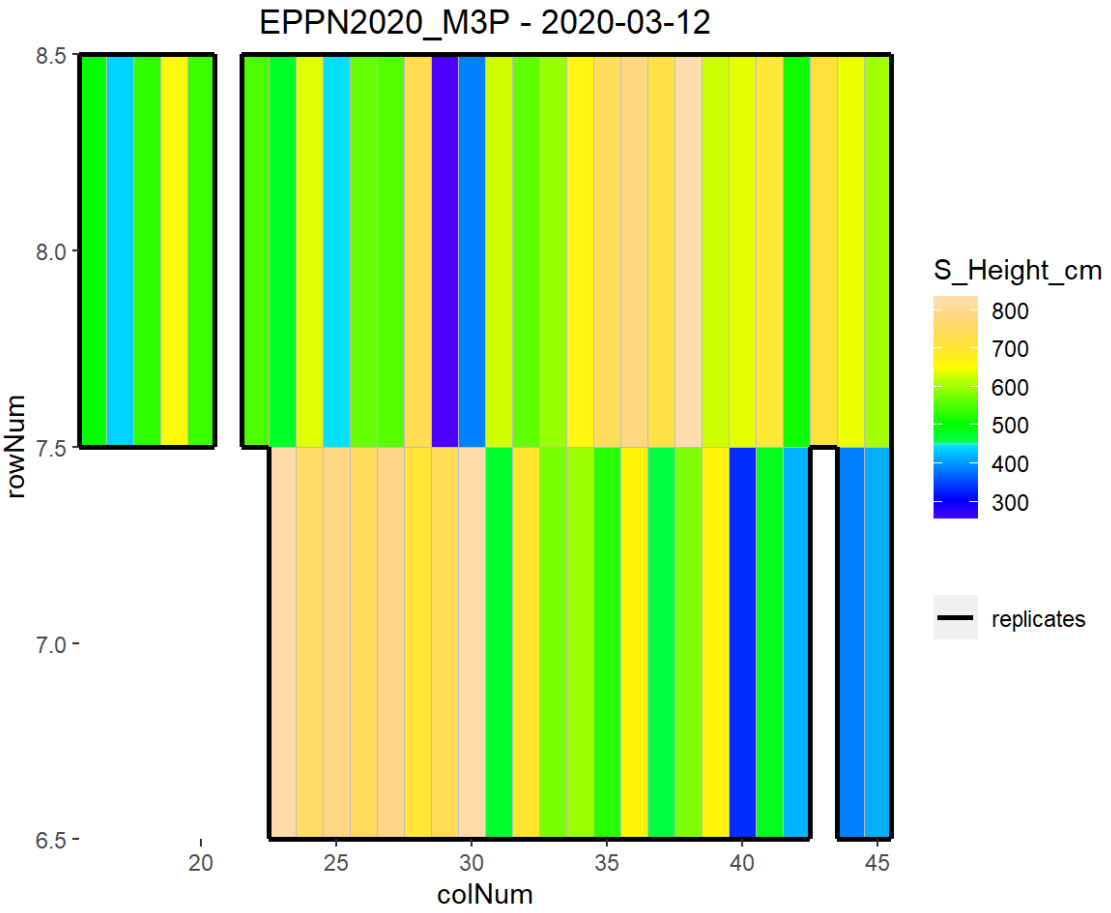

EPPN2020_M3P - 2020-03-05
rowNum
colNum
S_Height_cm
700
600
500
400
300
replicates

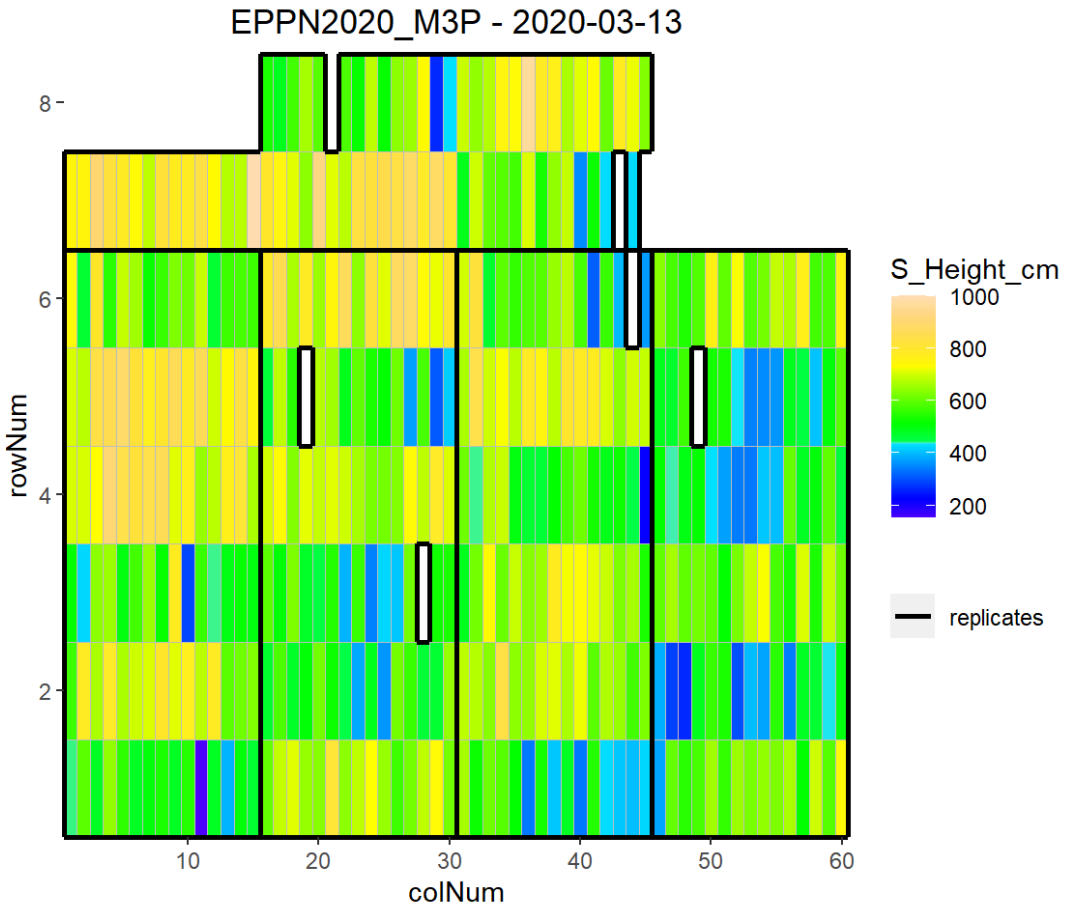

EPPN2020_M3P - 2020-03-06
rowNum
colNum
S_Height_cm
800
600
400
200
replicates

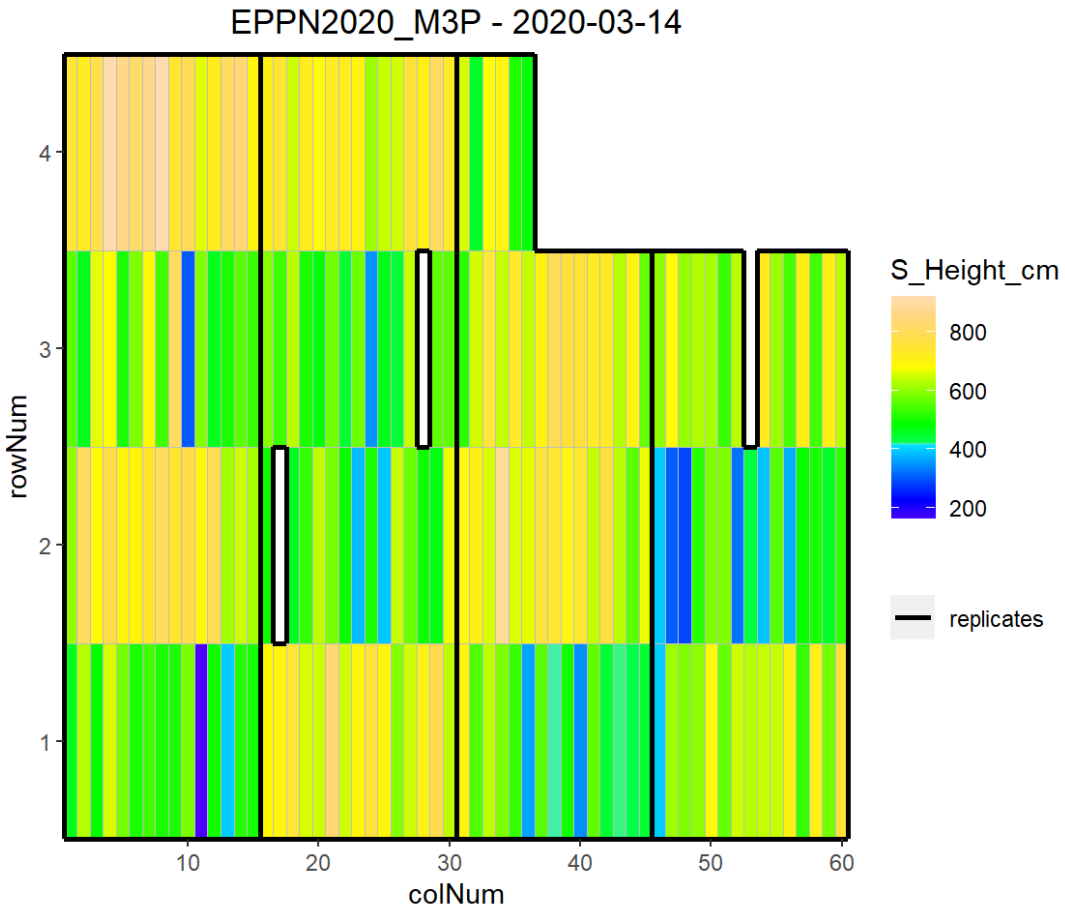

EPPN2020_M3P - 2020-03-07
rowNum
colNum
S_Height_cm
600
500
400
300
200
100
replicates

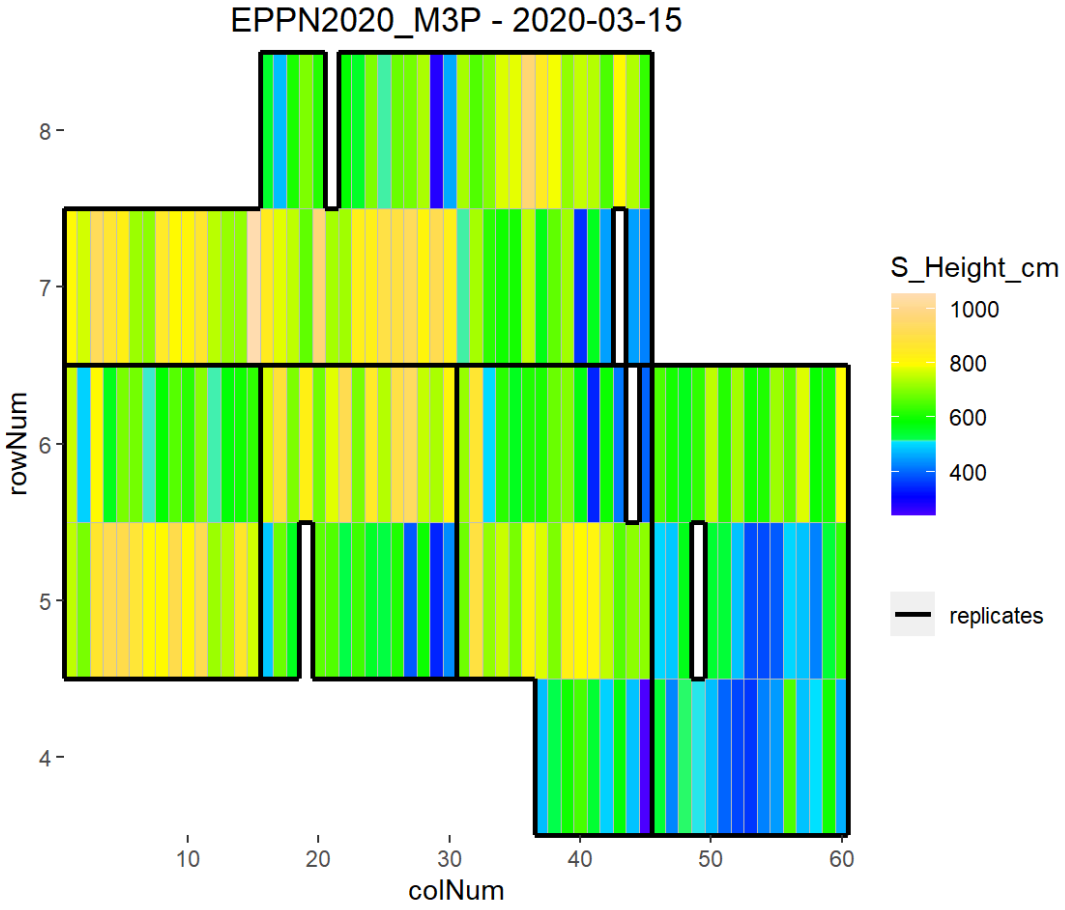

EPPN2020_M3P - 2020-03-08
rowNum
colNum
S_Height_cm
800
600
400
200
replicates

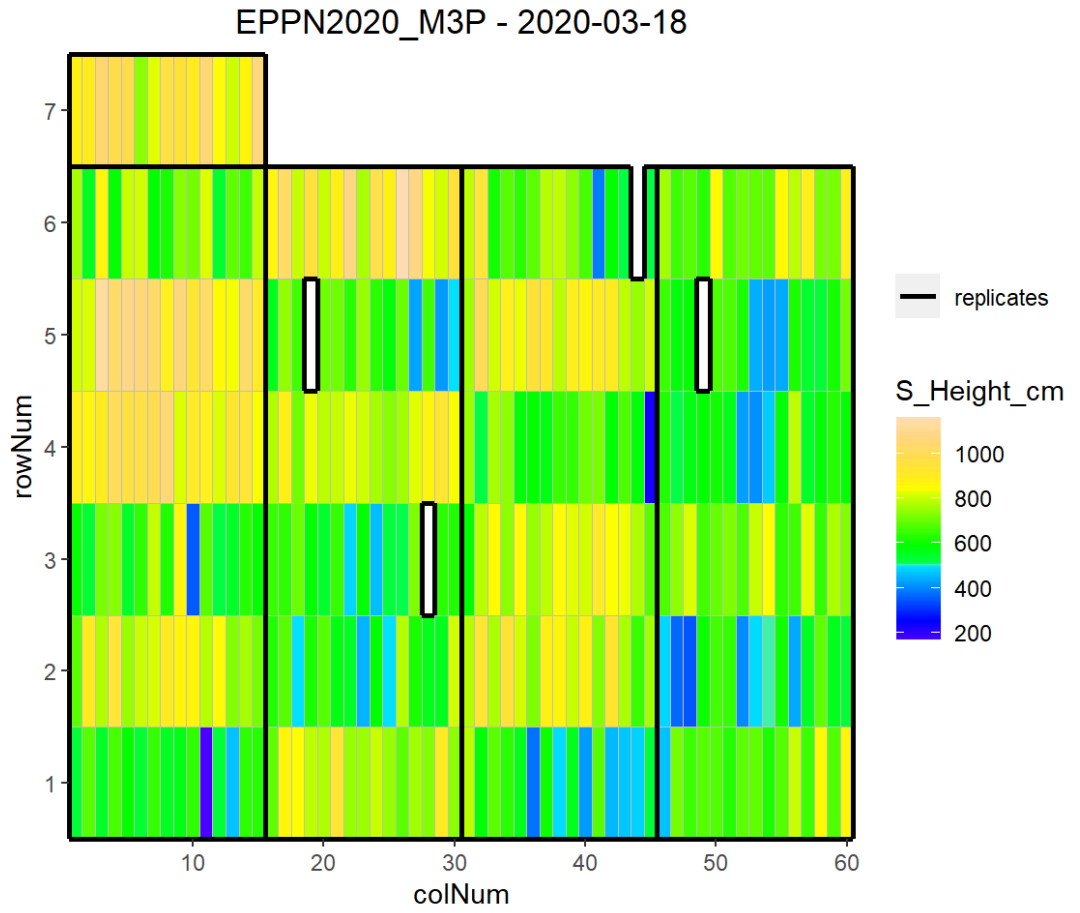

EPPN2020_M3P - 2020-03-11
rowNum
colNum
S_Height_cm
800
600
400
200
replicates

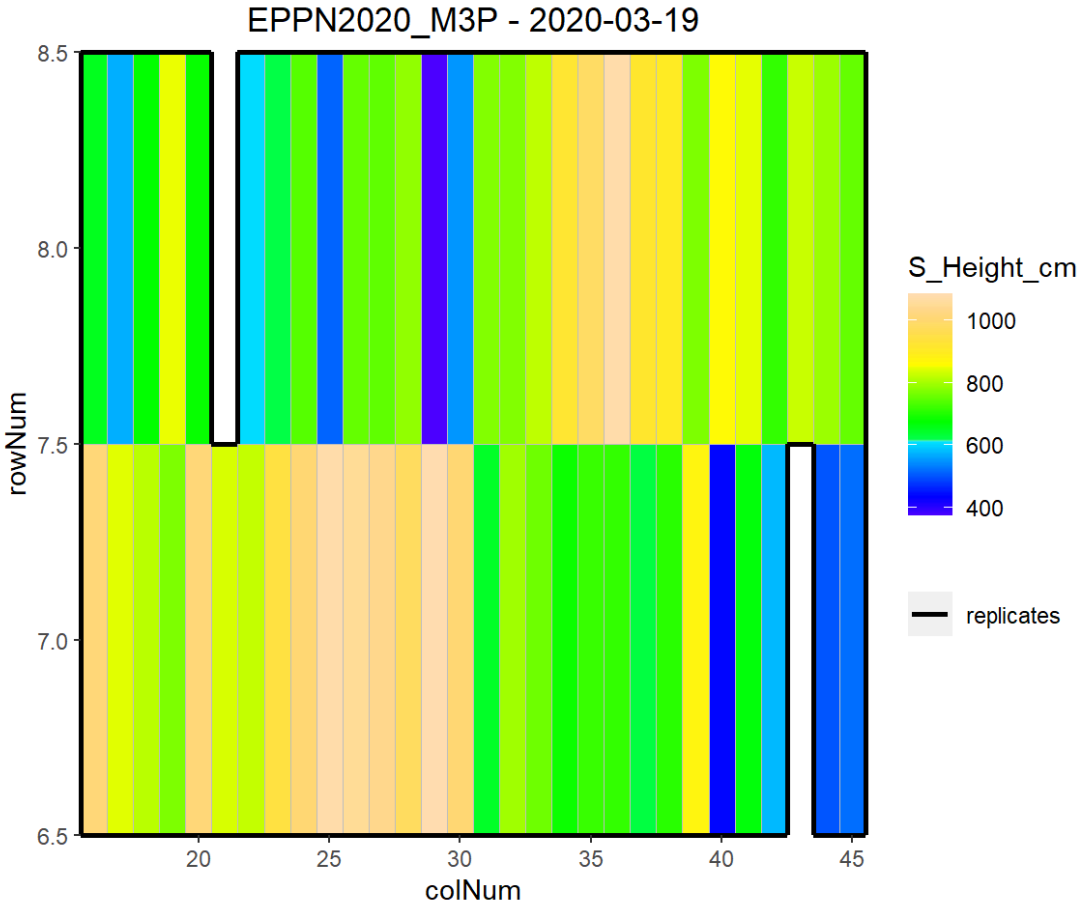

EPPN2020_M3P - 2020-03-12
rowNum
8.5
8.0
7.5
7.0
6.5
20
25
30
35
40
45
colNum
S_Height_cm
800
700
600
500
400
300
replicates

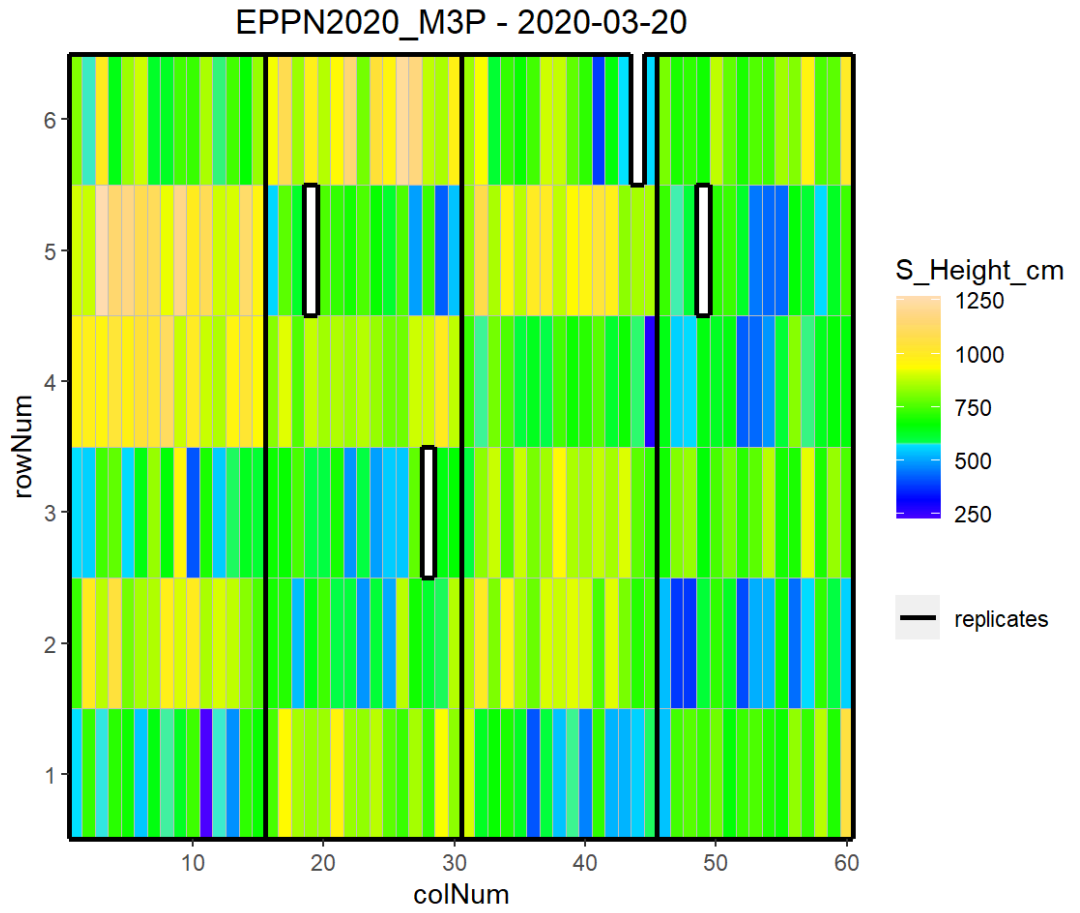

EPPN2020_M3P - 2020-03-13
rowNum
8
6
4
2
10
20
30
40
50
60
colNum
S_Height_cm
1000
800
600
400
200
replicates

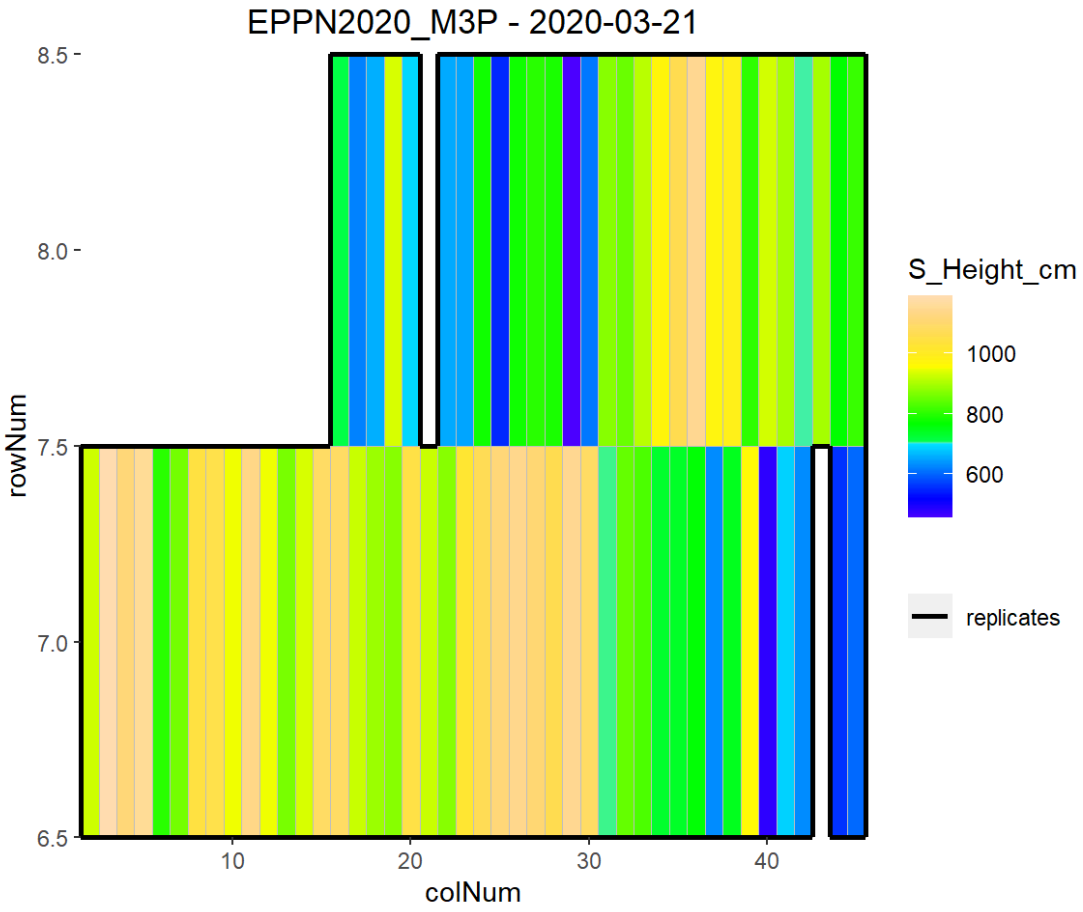

EPPN2020_M3P - 2020-03-14
rowNum
colNum
S_Height_cm
800
600
400
200
replicates

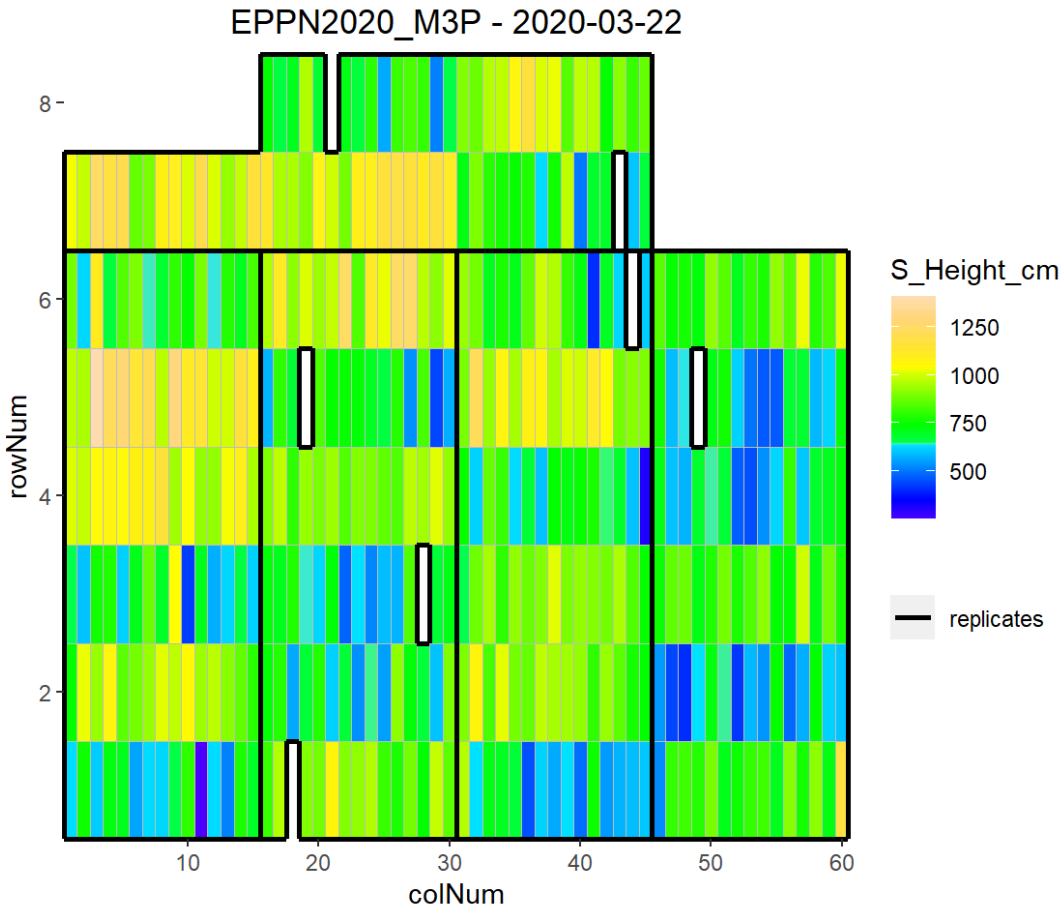

EPPN2020_M3P - 2020-03-15
rowNum
colNum
S_Height_cm
1000
800
600
400
replicates

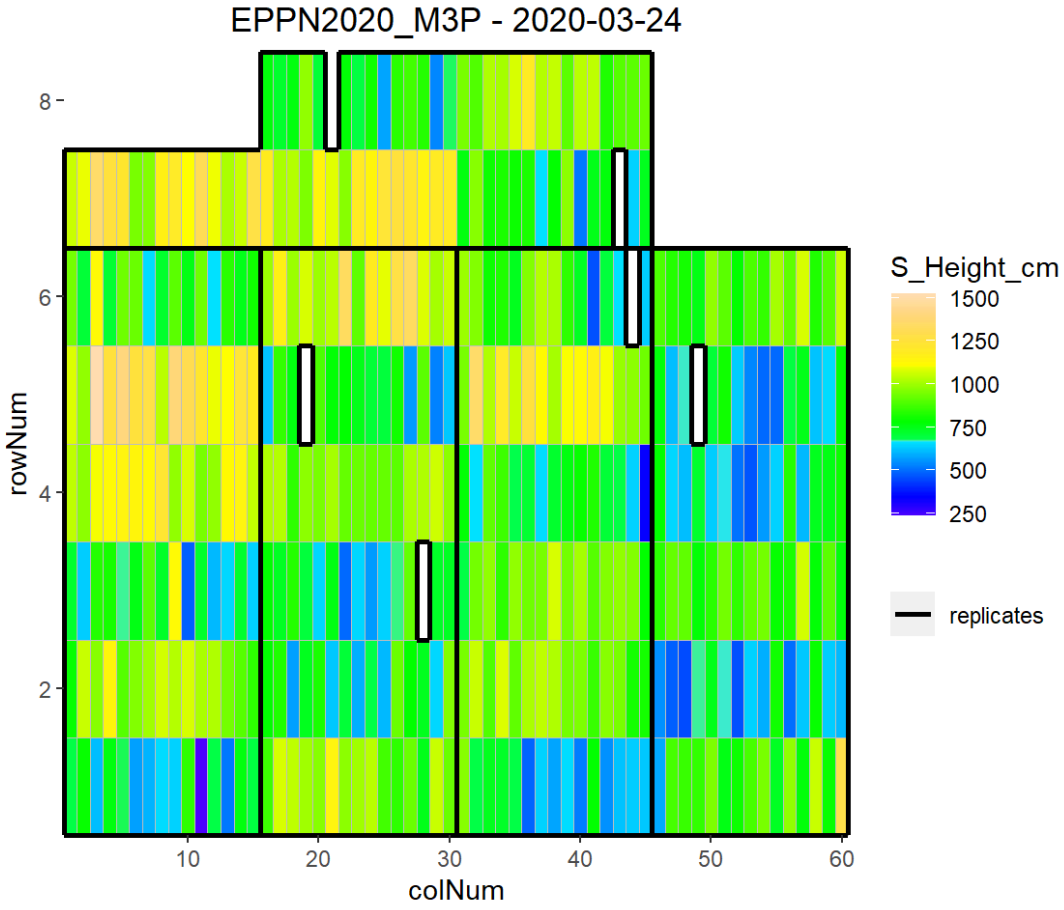

EPPN2020_M3P - 2020-03-18
rowNum
colNum
replicates
S_Height_cm
1000
800
600
400
200

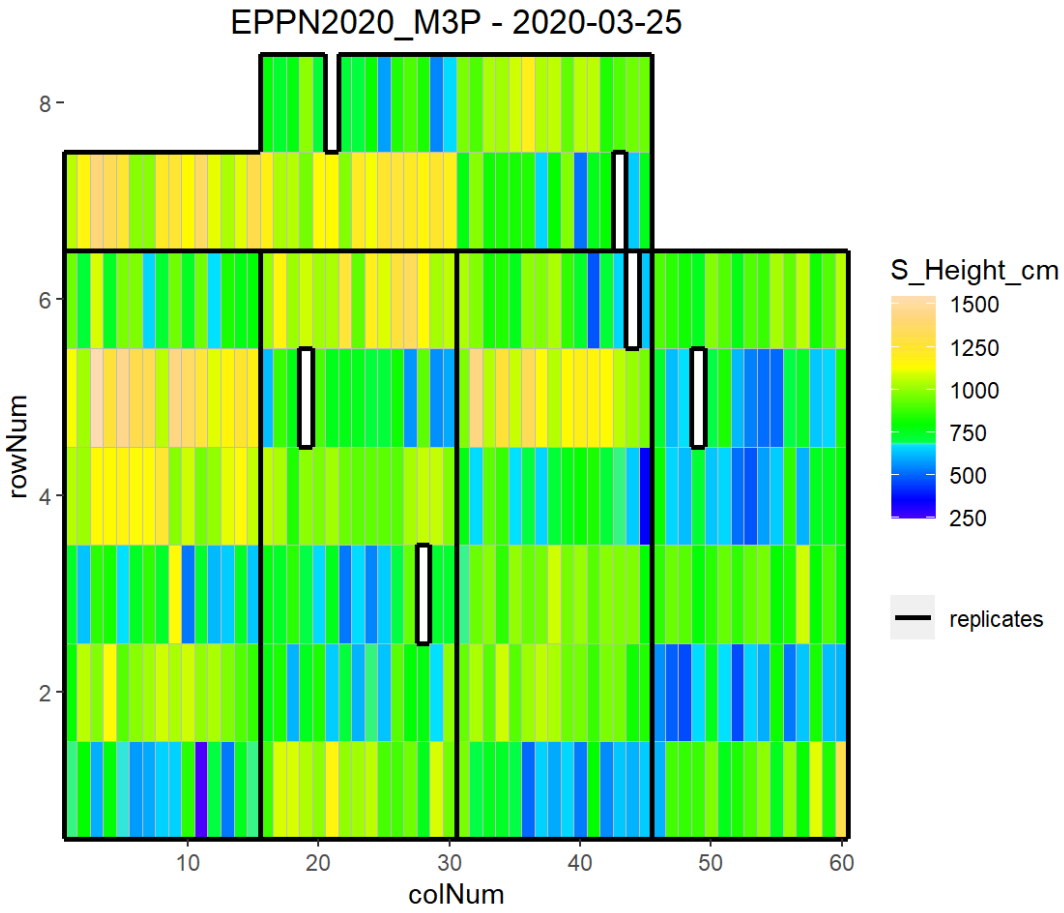

EPPN2020_M3P - 2020-03-19
rowNum
colNum
S_Height_cm
1000
800
600
400
replicates

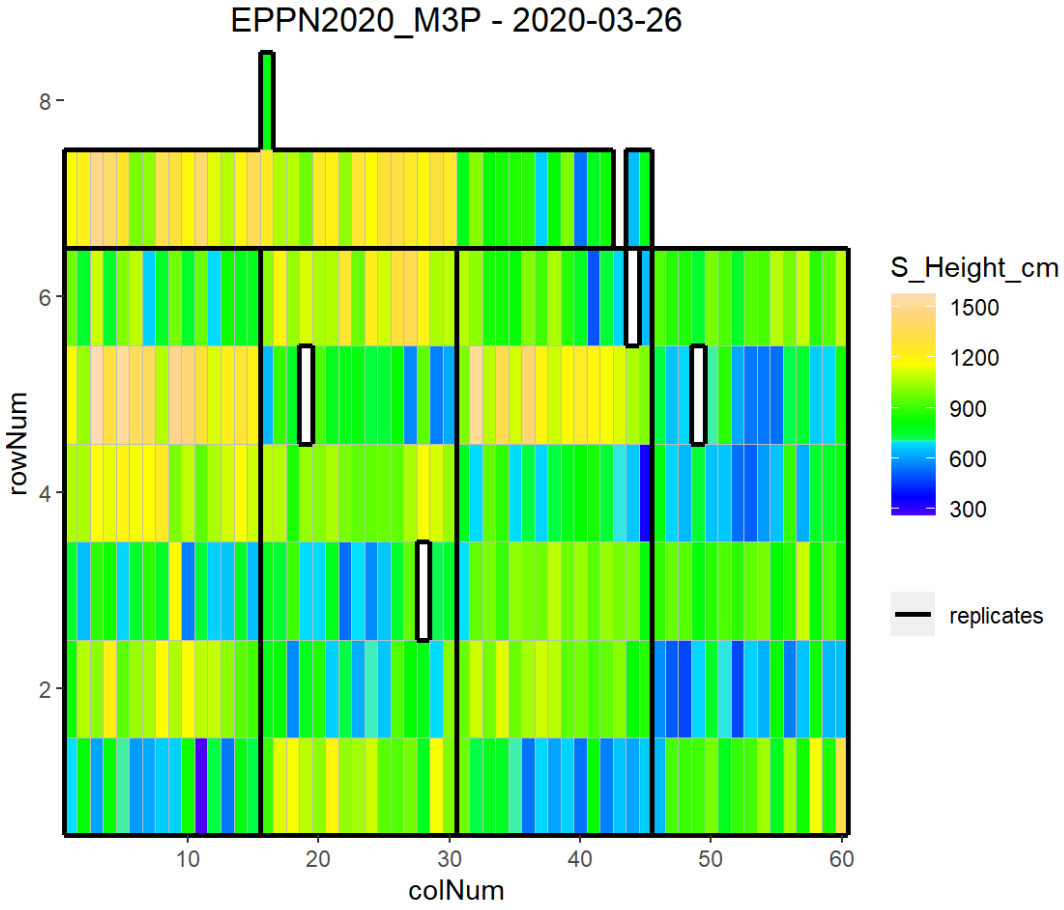

EPPN2020_M3P - 2020-03-20
rowNum
colNum
S_Height_cm
1250
1000
750
500
250
replicates

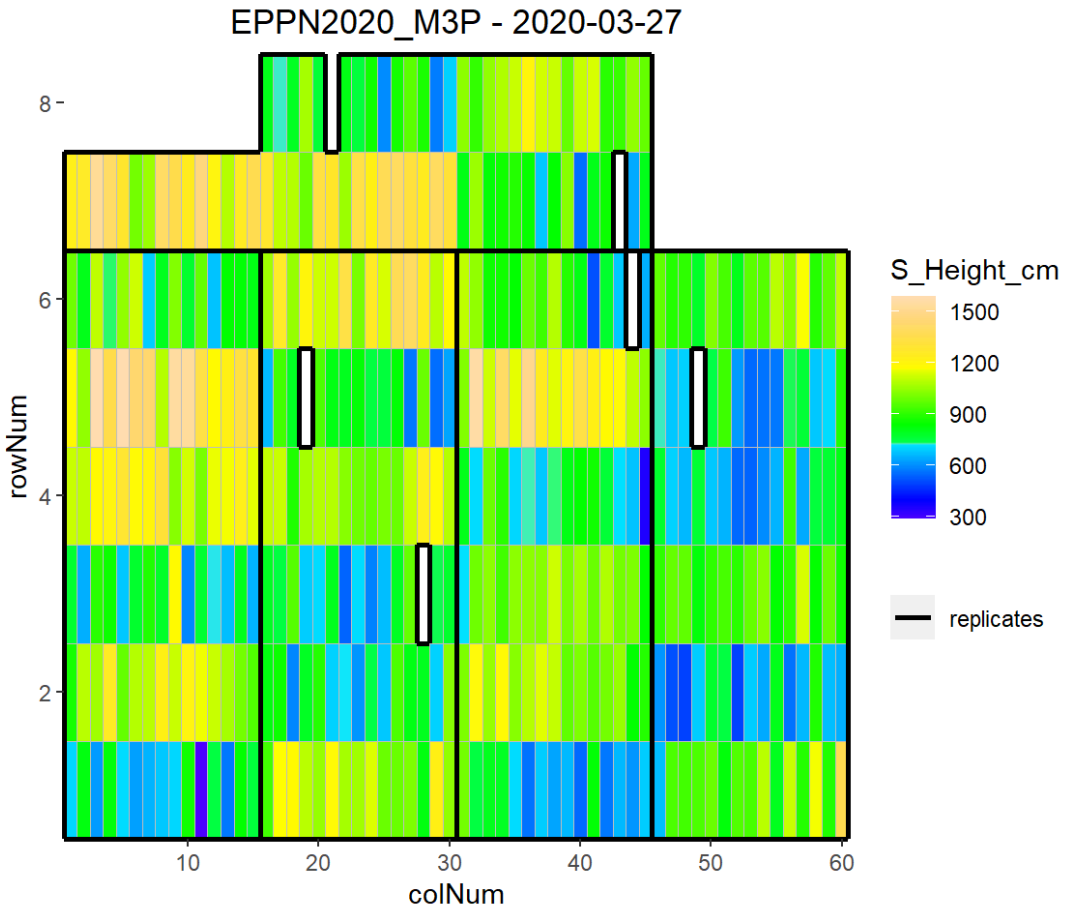

EPPN2020_M3P - 2020-03-21
rowNum
colNum
S_Height_cm
1000
800
600
replicates

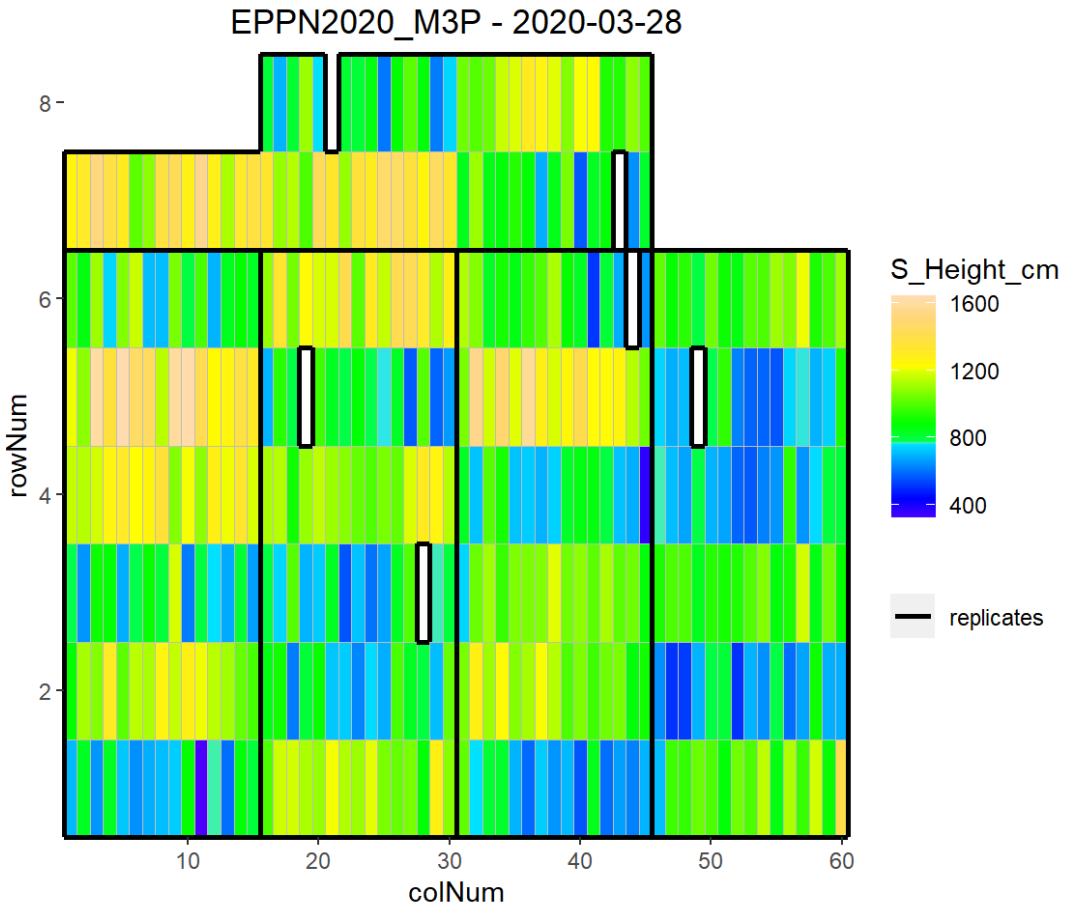

EPPN2020_M3P - 2020-03-22
rowNum
colNum
S_Height_cm
1250
1000
750
500
replicates

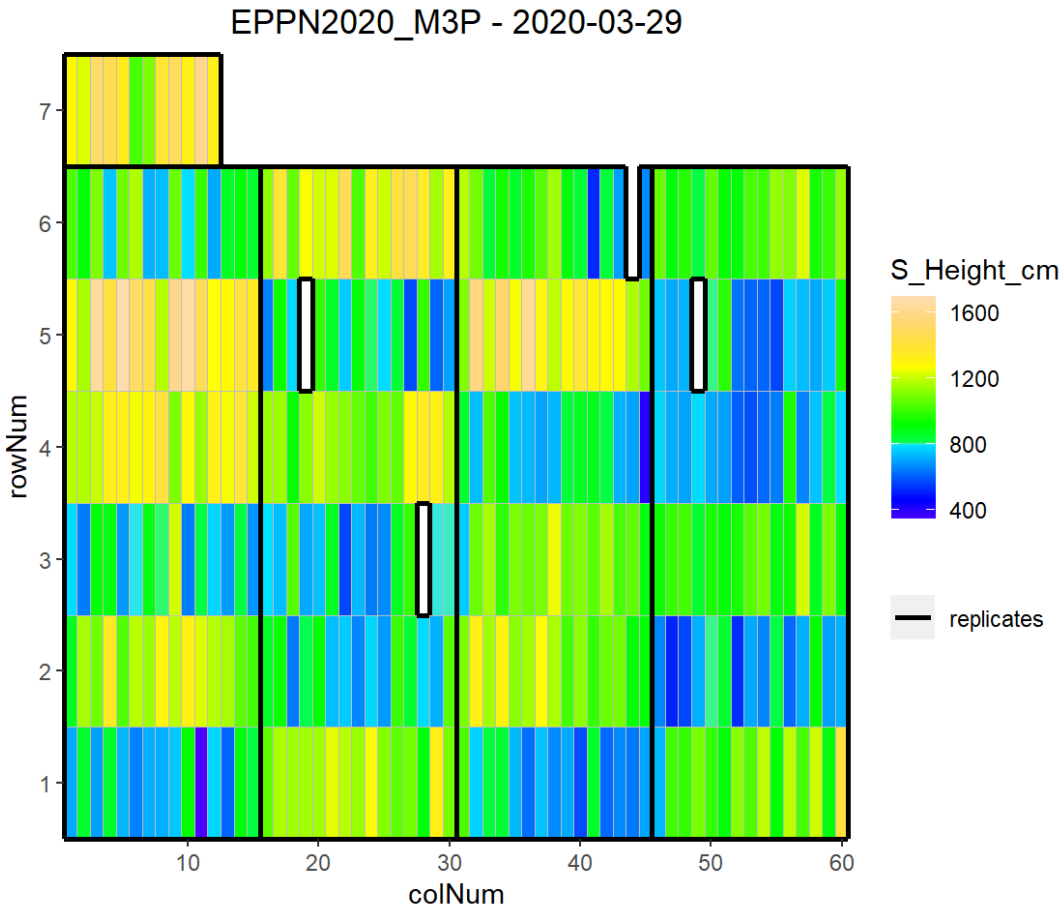

EPPN2020_M3P - 2020-03-24
rowNum
colNum
S_Height_cm
1500
1250
1000
750
500
250
replicates

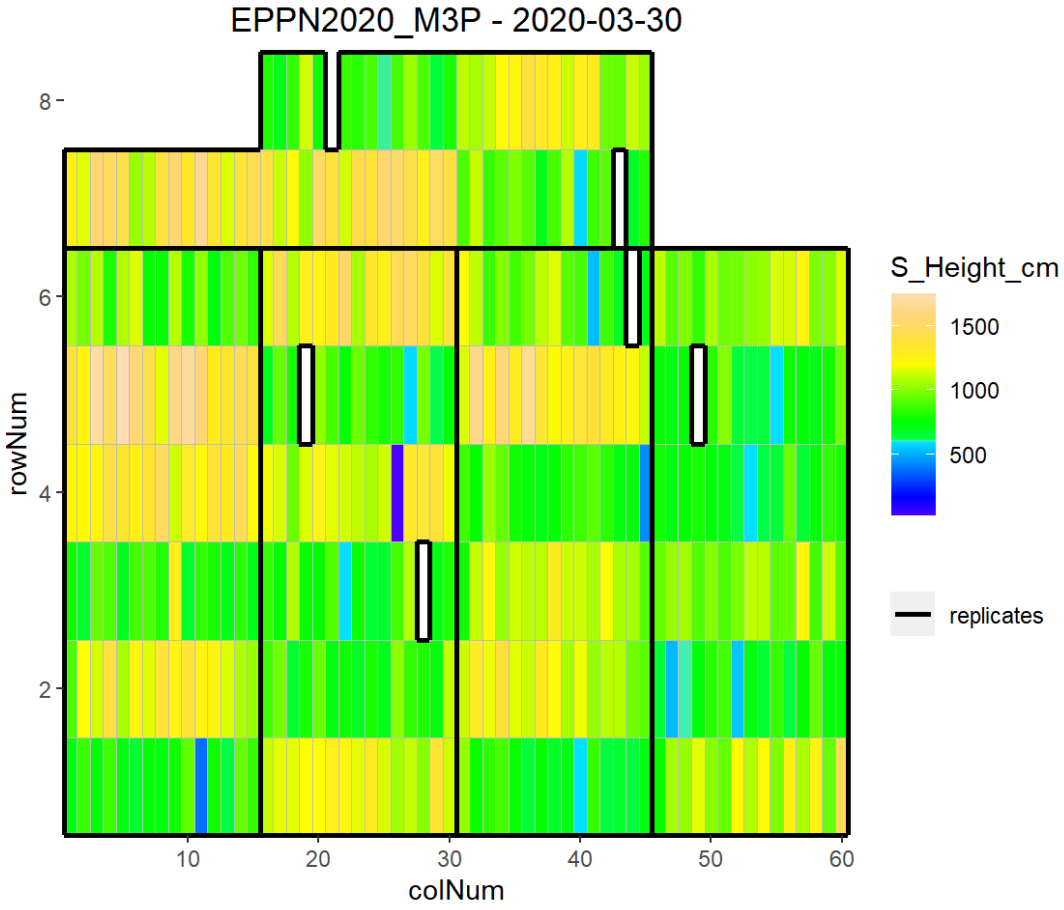

EPPN2020_M3P - 2020-03-25
rowNum
colNum
S_Height_cm
1500
1250
1000
750
500
250
replicates

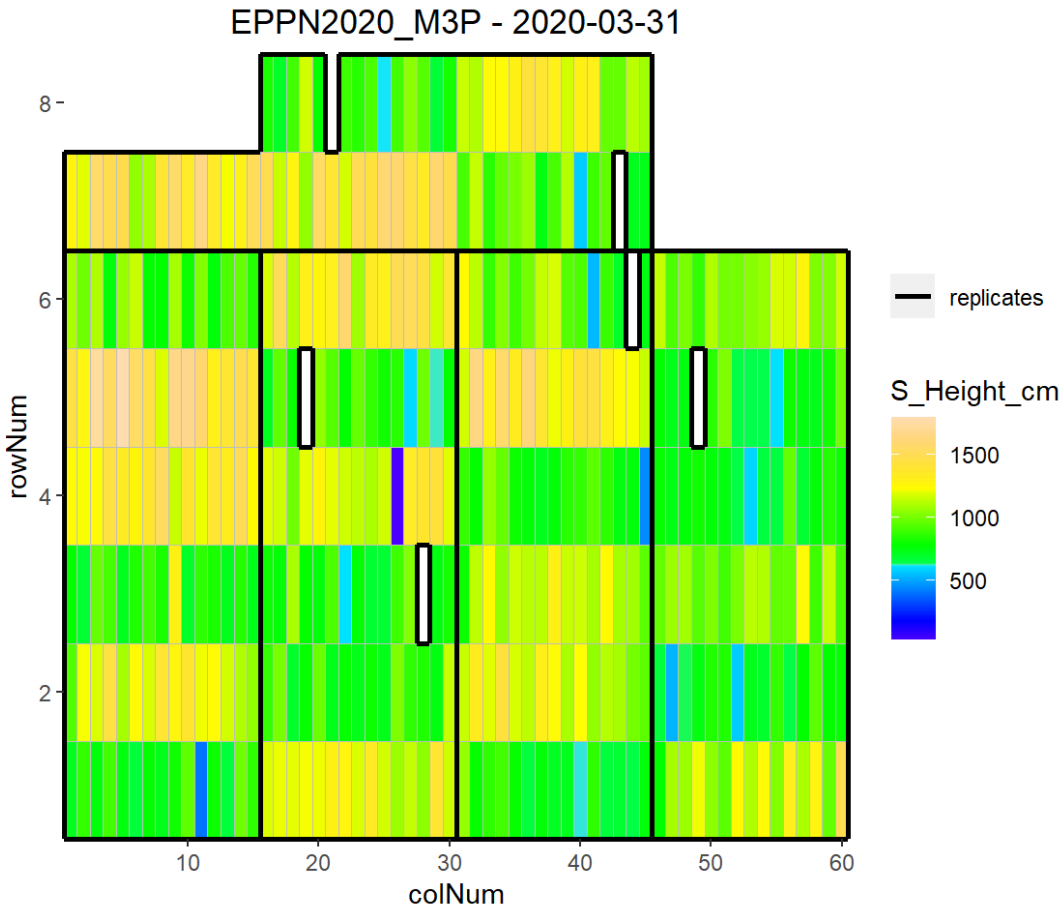

EPPN2020_M3P - 2020-03-26
rowNum
colNum
S_Height_cm
1500
1200
900
600
300
replicates

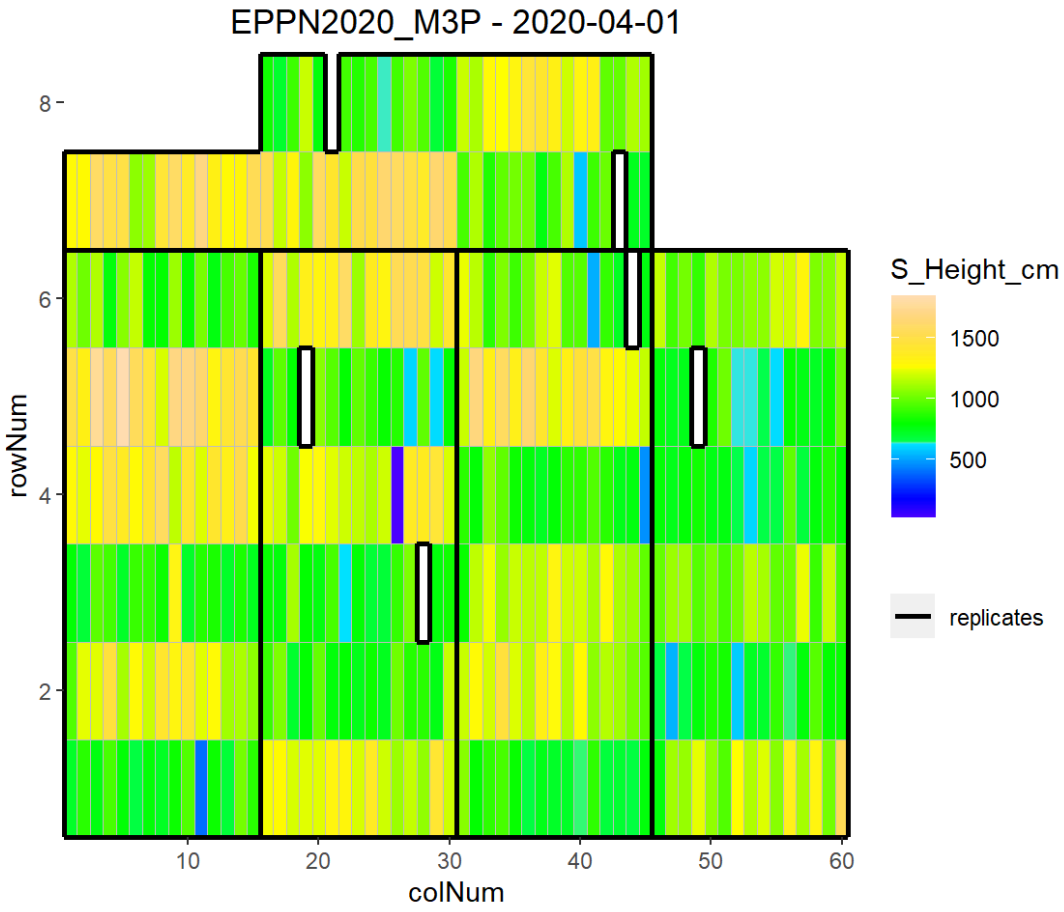

EPPN2020_M3P - 2020-03-27
rowNum
colNum
S_Height_cm
1500
1200
900
600
300
replicates

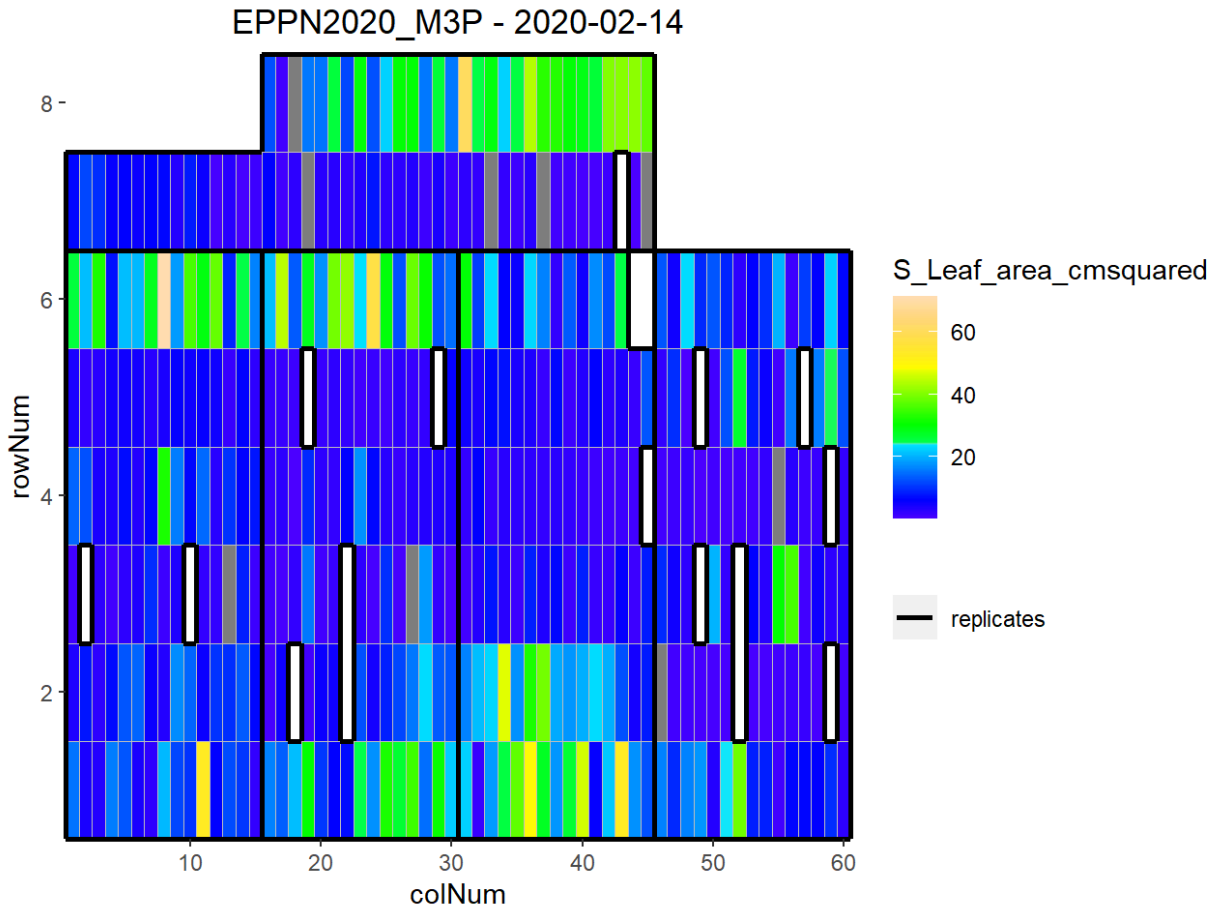

EPPN2020_M3P - 2020-03-28
rowNum
colNum
S_Height_cm
1600
1200
800
400
replicates

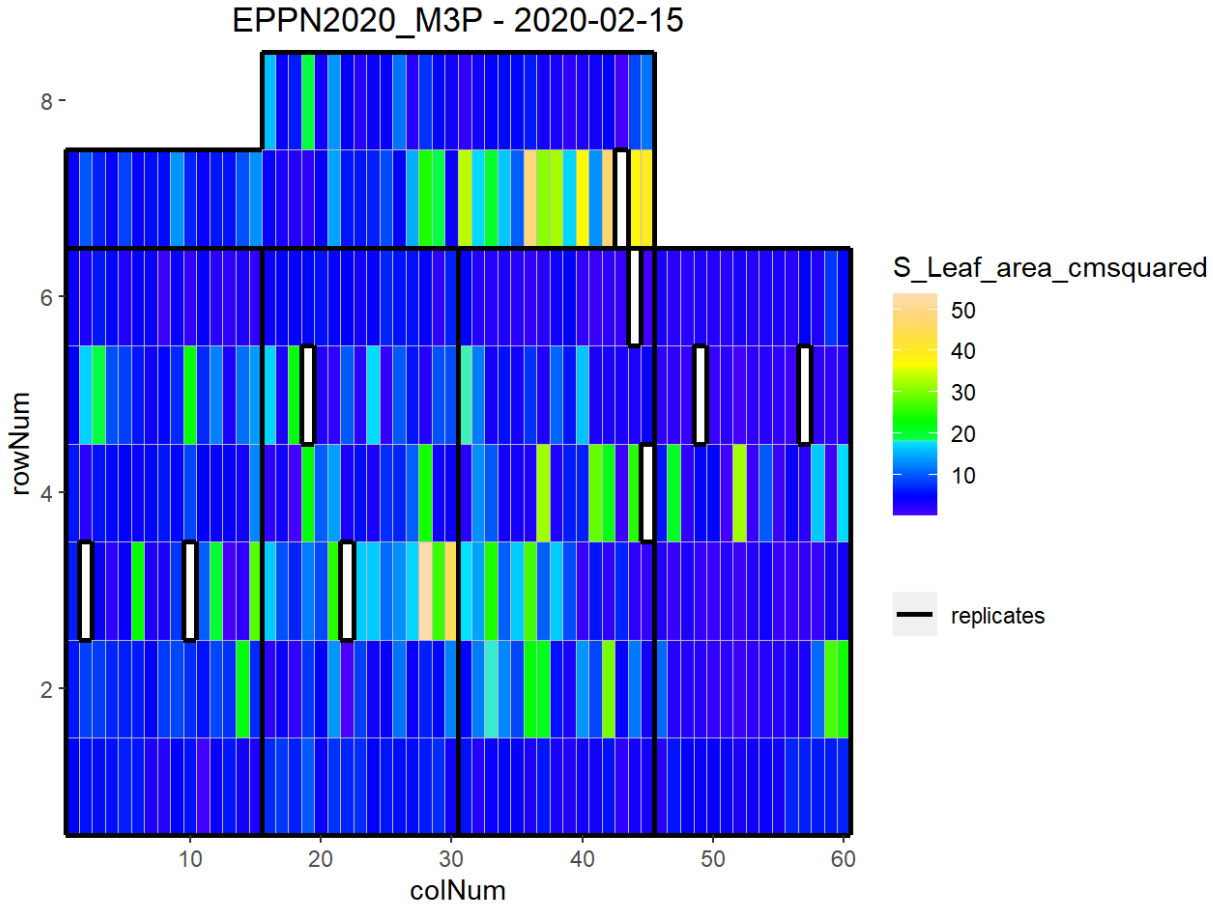

EPPN2020_M3P - 2020-03-29
rowNum
colNum
S_Height_cm
1600
1200
800
400
replicates

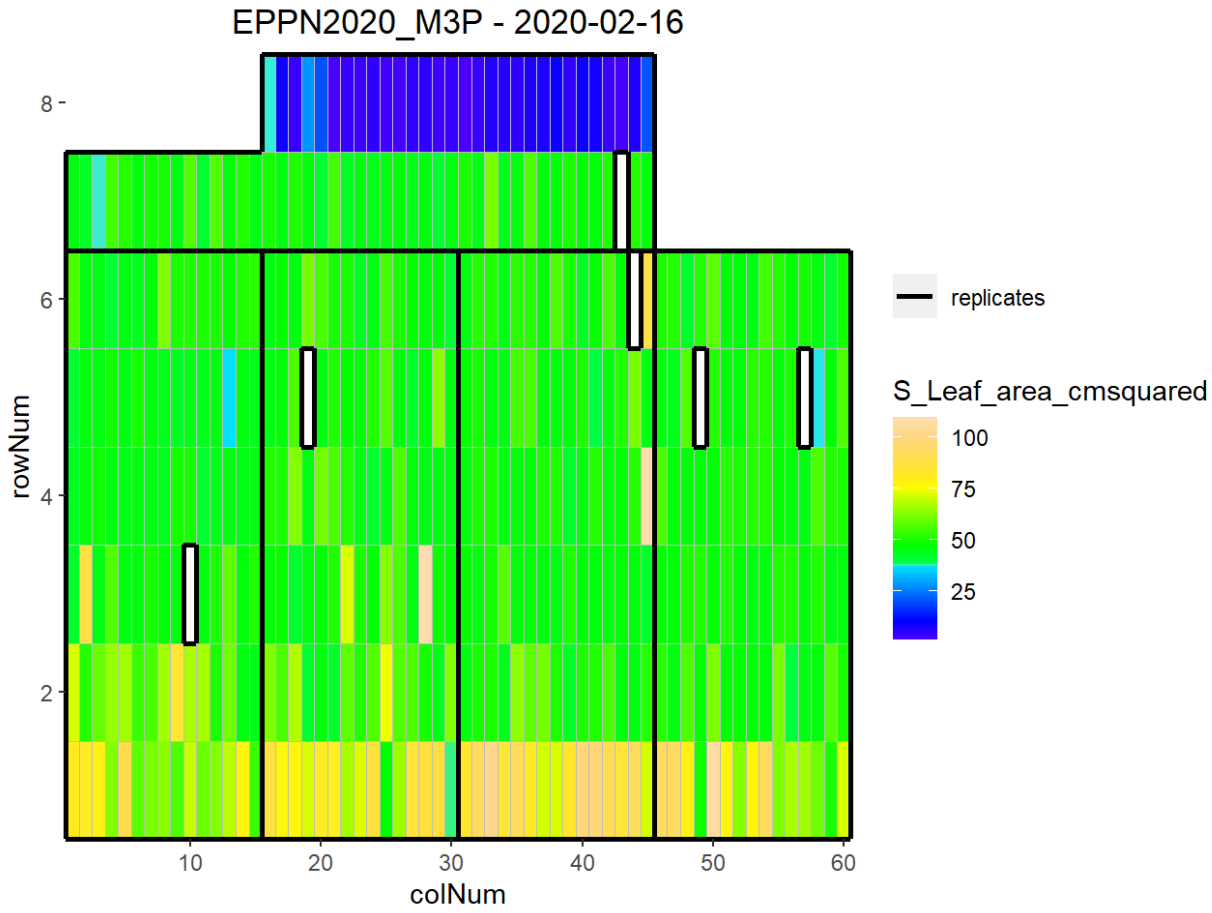

EPPN2020_M3P - 2020-03-30
rowNum
colNum
S_Height_cm
1500
1000
500
replicates

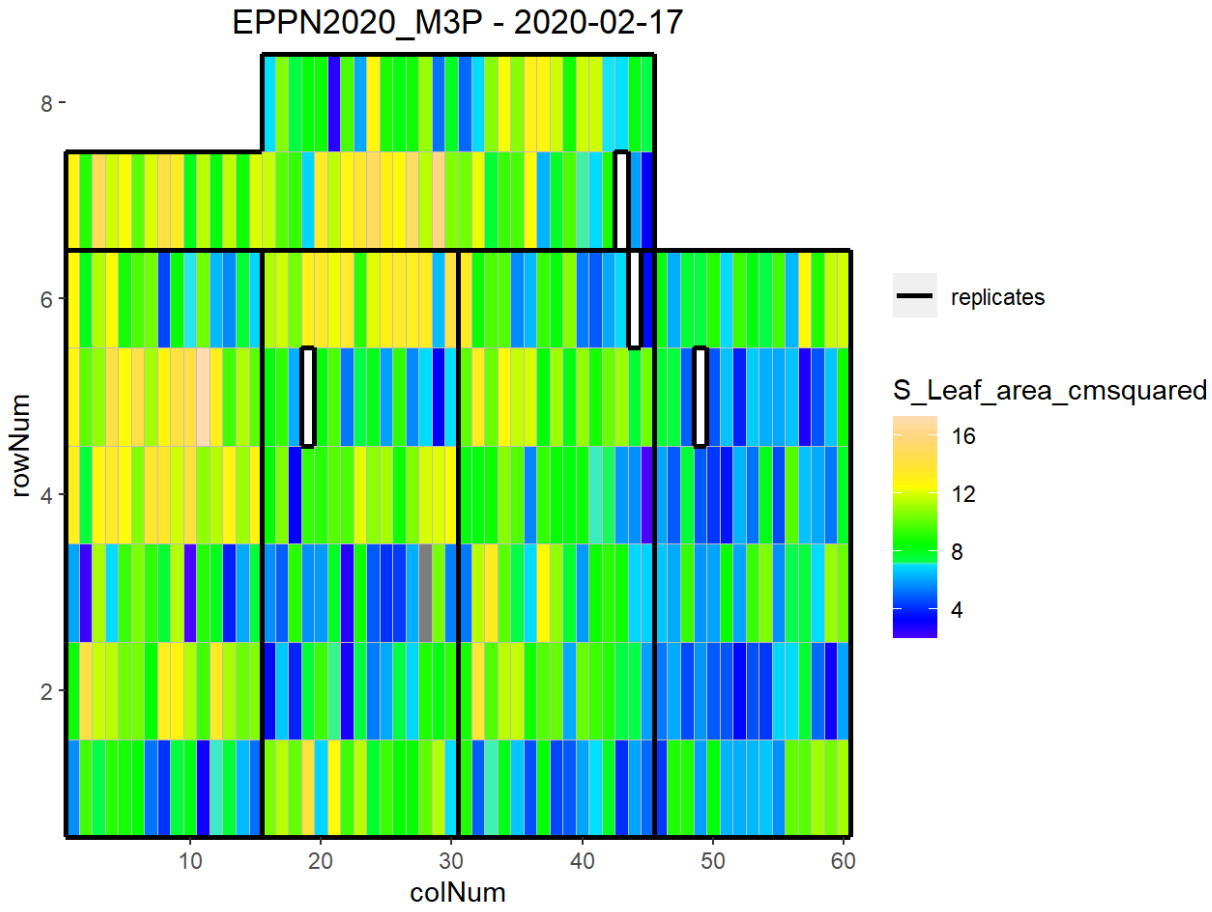

EPPN2020_M3P - 2020-03-31
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Height_cm
1500
1000
500

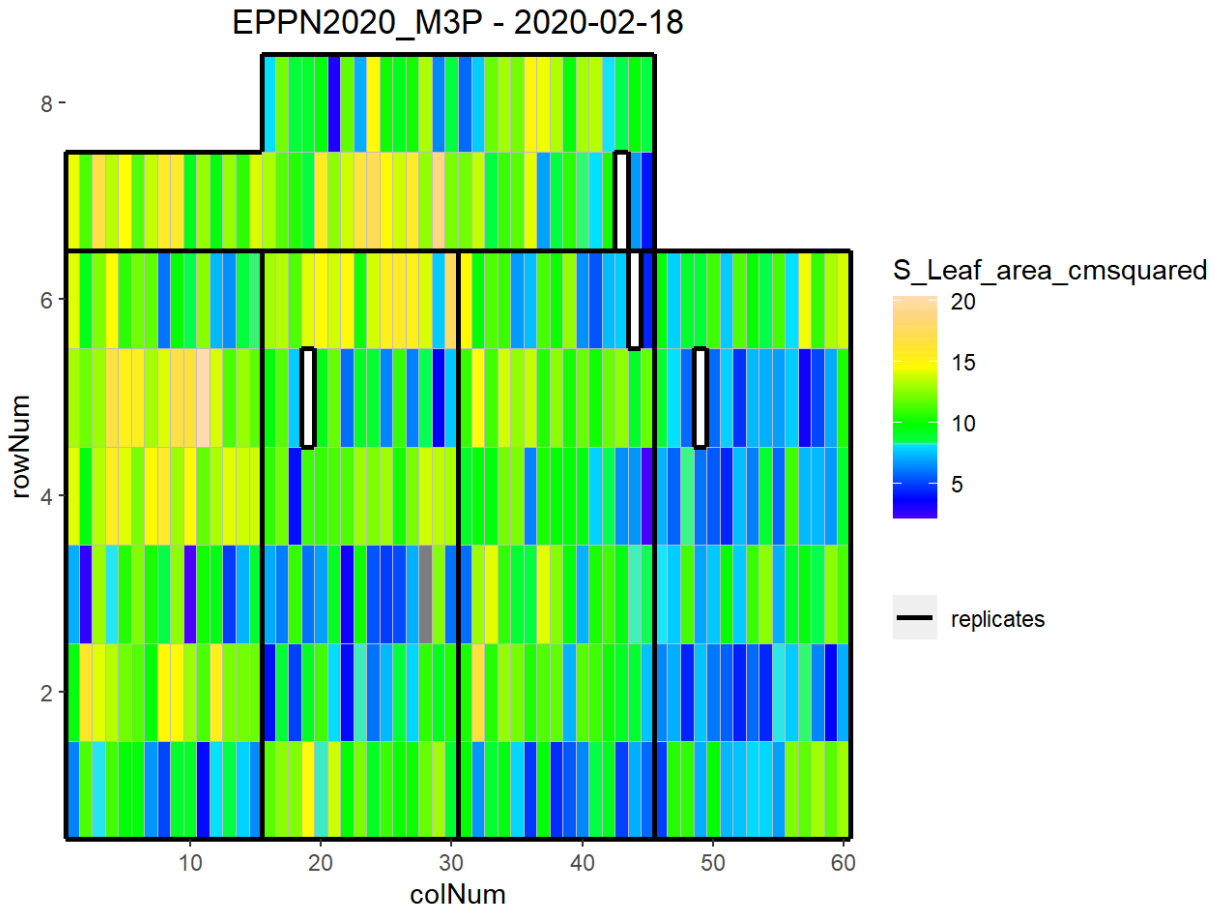

EPPN2020_M3P - 2020-04-01
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Height_cm
1500
1000
500
replicates

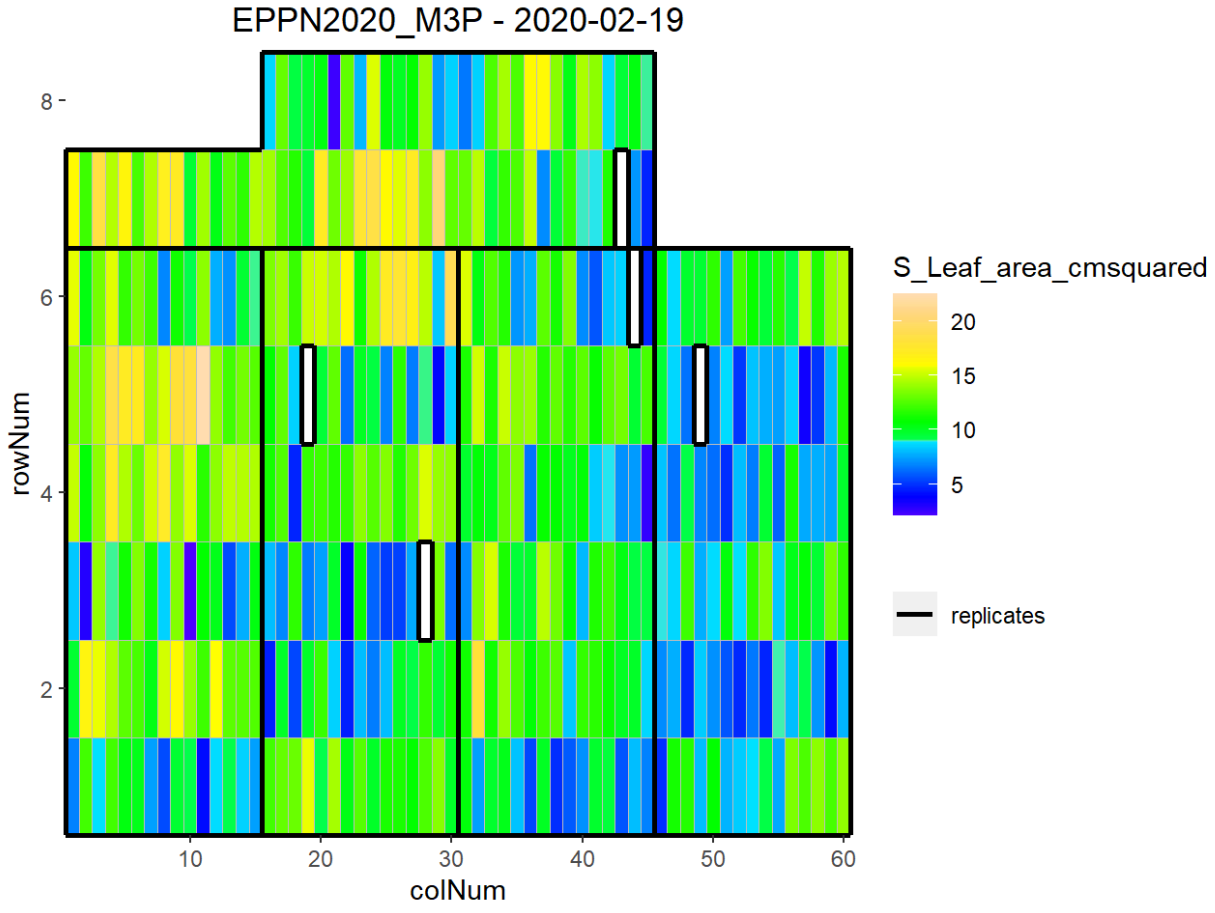

EPPN2020_M3P - 2020-02-14
rowNum
colNum
S_Leaf_area_cmsquared
60
40
20
replicates

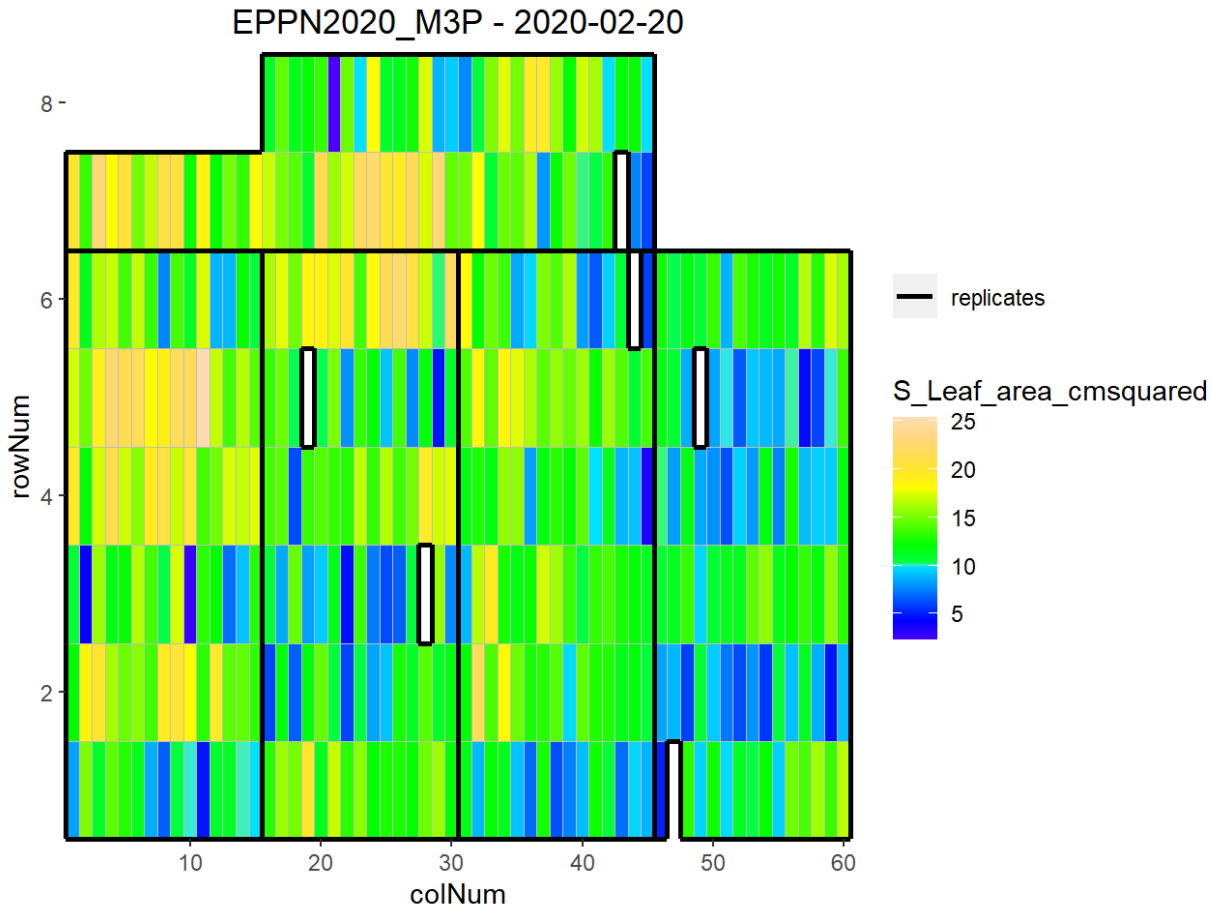

EPPN2020_M3P - 2020-02-15
rowNum
colNum
S_Leaf_area_cmsquared
50
40
30
20
10
replicates

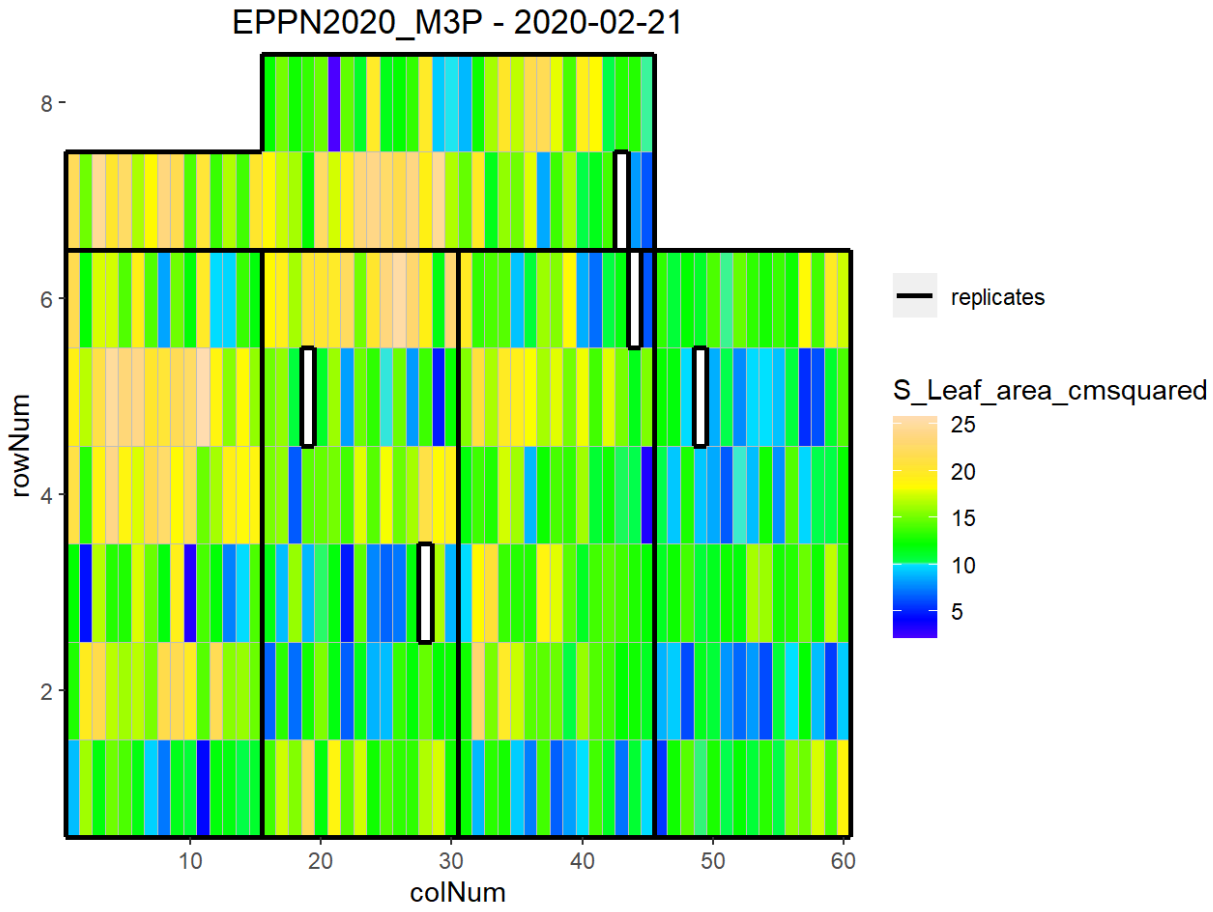

EPPN2020_M3P - 2020-02-16
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
100
75
50
25

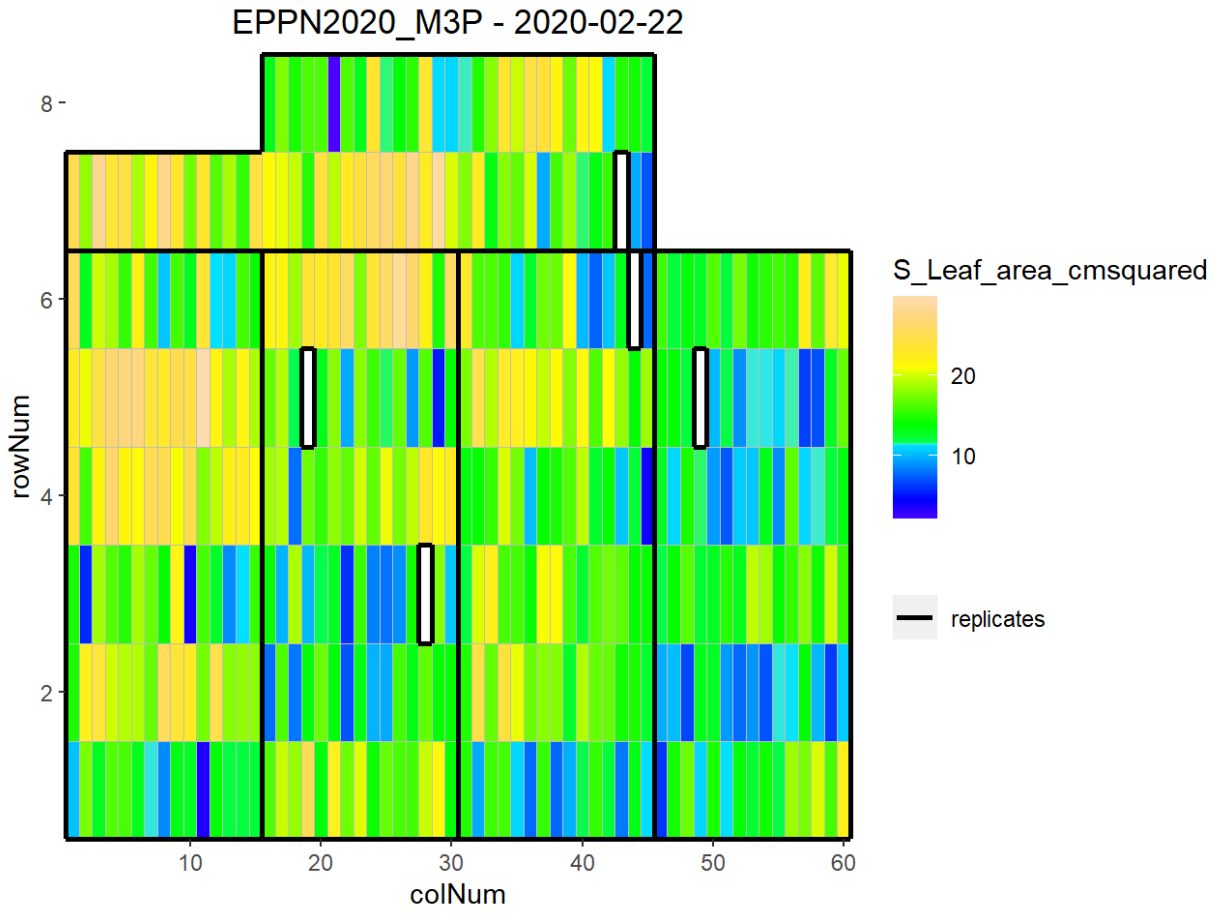

EPPN2020_M3P - 2020-02-17
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
16
12
8
4

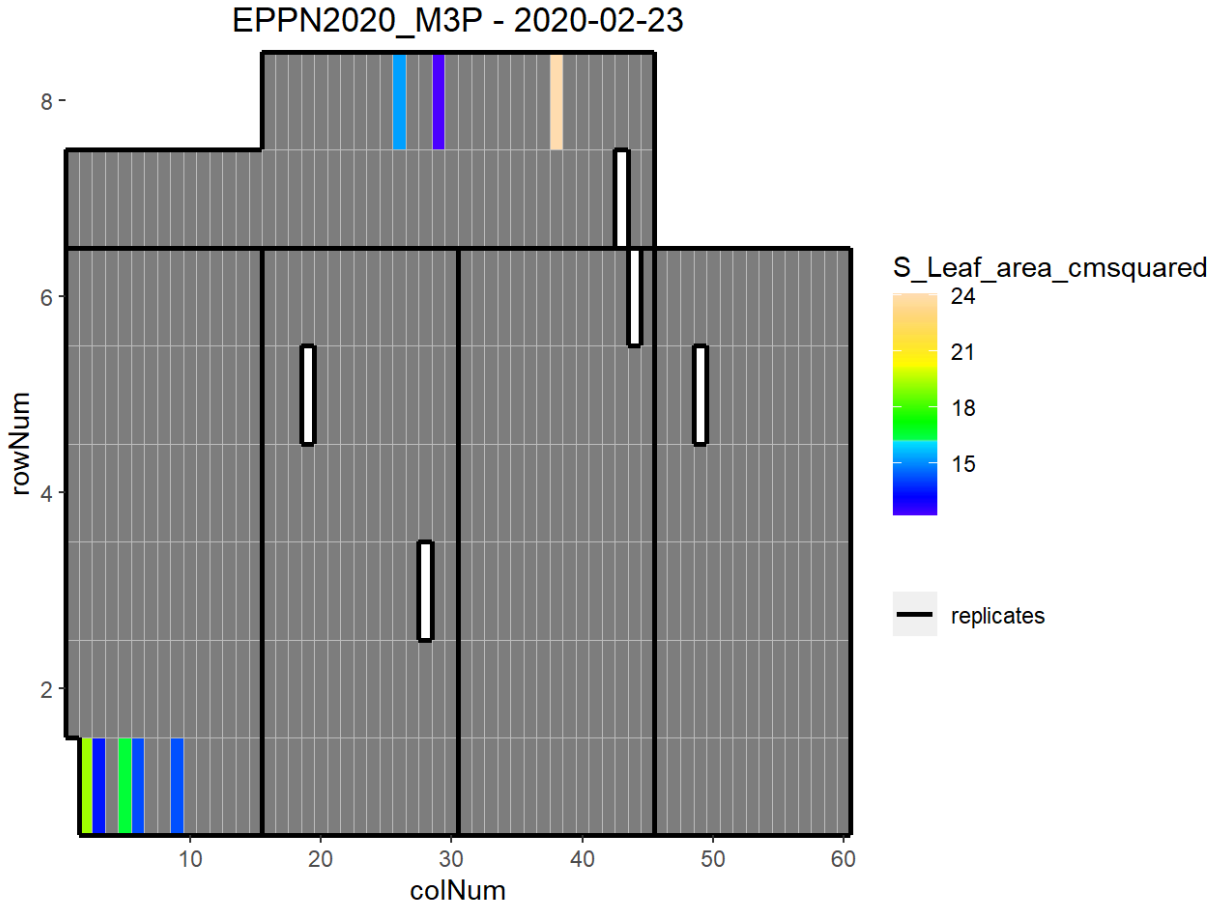

EPPN2020_M3P - 2020-02-18
rowNum
colNum
S_Leaf_area_cmsquared
20
15
10
5
replicates

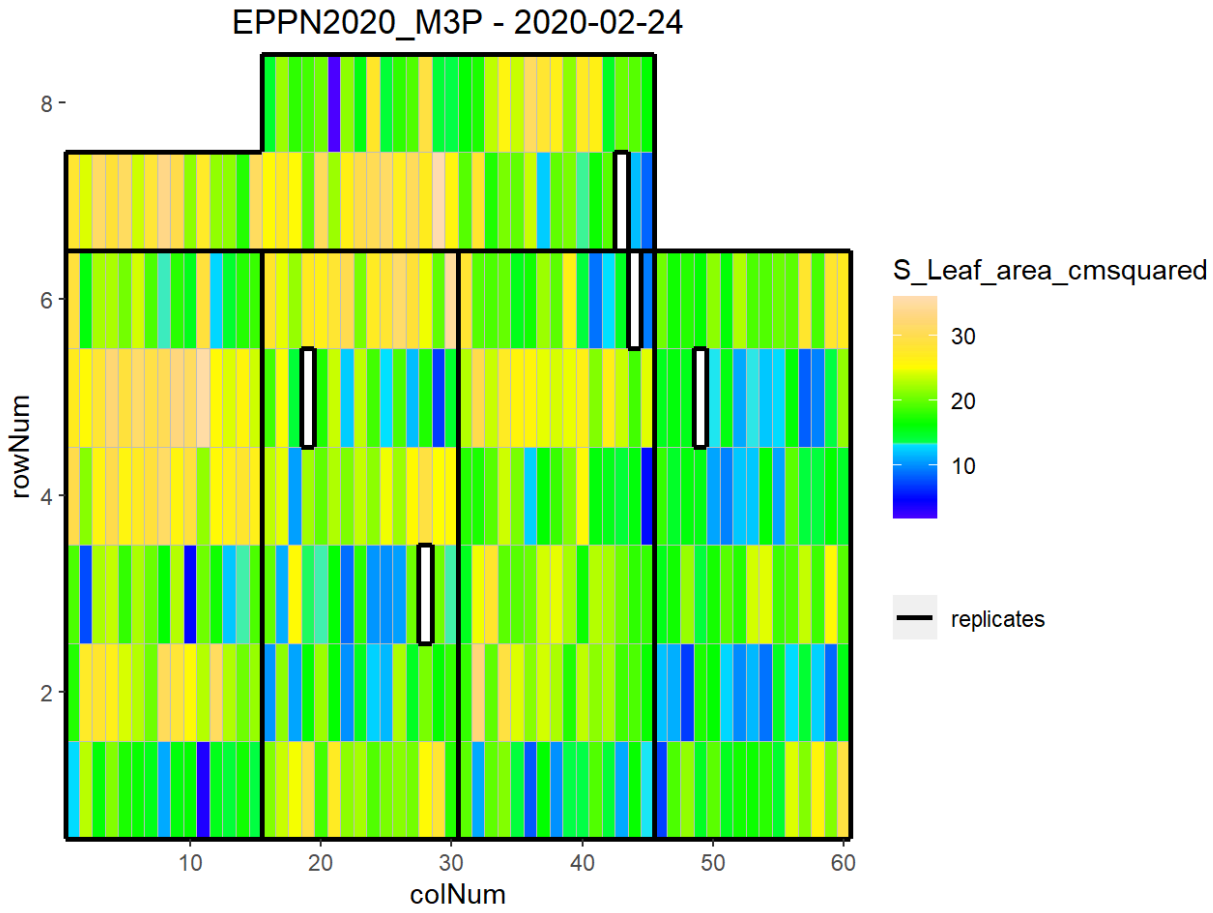

EPPN2020_M3P - 2020-02-19
rowNum
colNum
S_Leaf_area_cmsquared
20
15
10
5
replicates

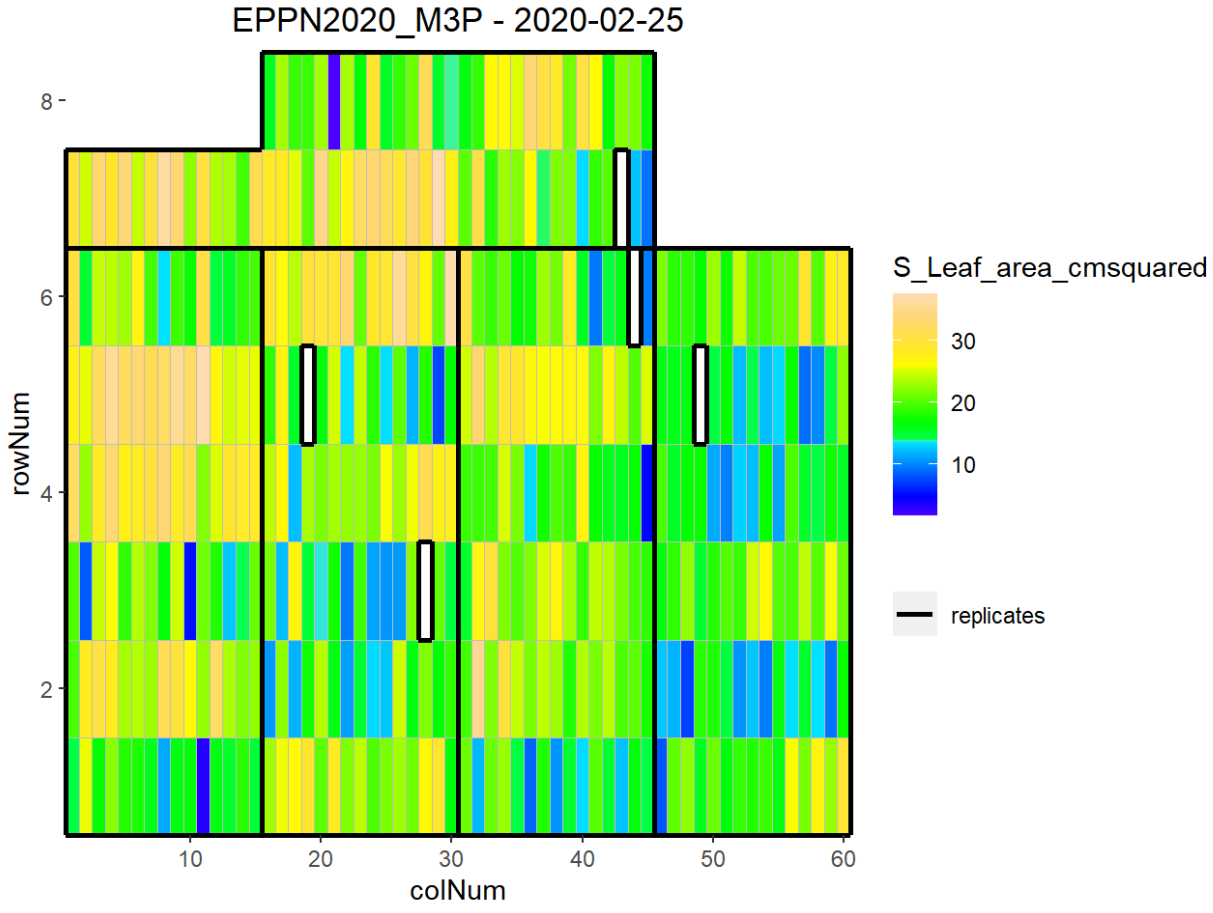

EPPN2020_M3P - 2020-02-20
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
25
20
15
10
5

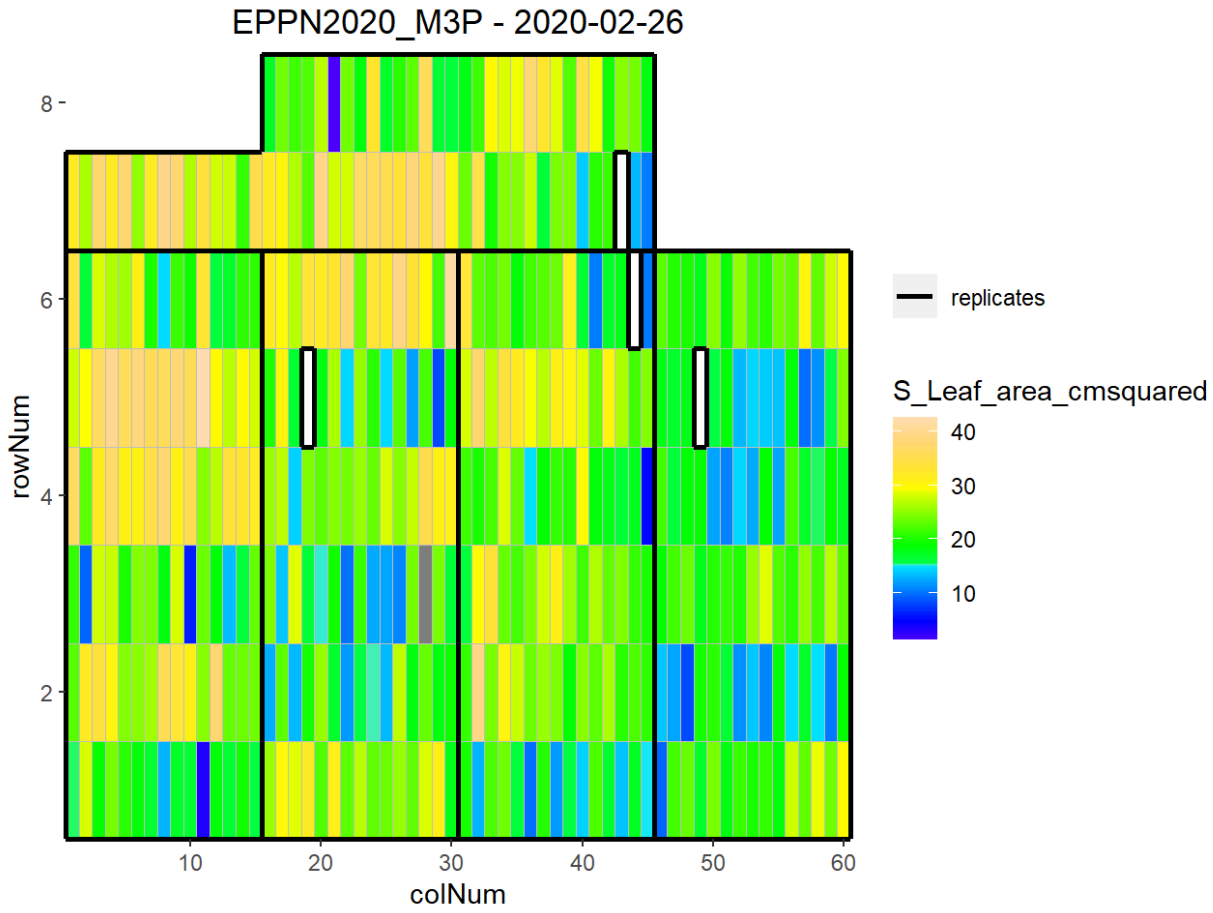

EPPN2020_M3P - 2020-02-21
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
25
20
15
10
5

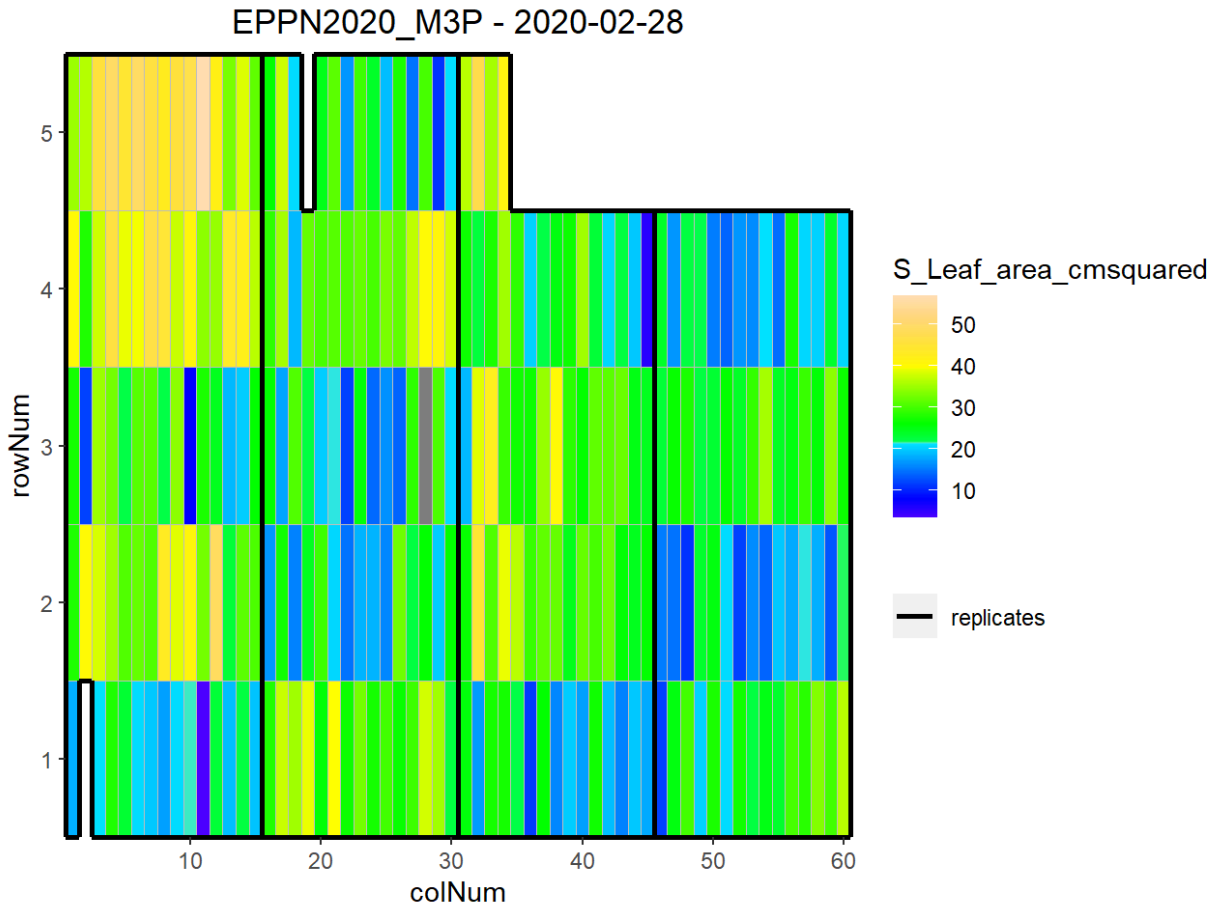

EPPN2020_M3P - 2020-02-22
rowNum
colNum
S_Leaf_area_cmsquared
20
10
replicates

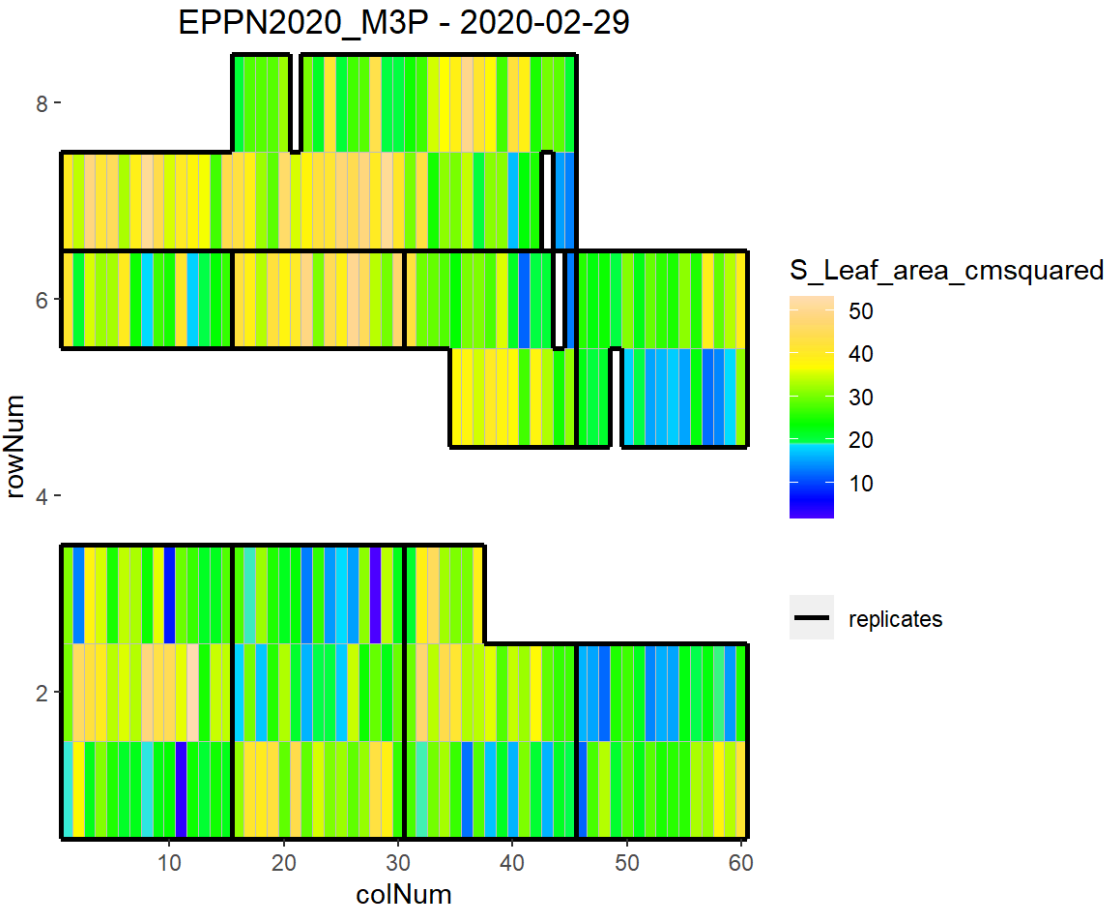

EPPN2020_M3P - 2020-02-23
rowNum
colNum
S_Leaf_area_cmsquared
24
21
18
15
replicates

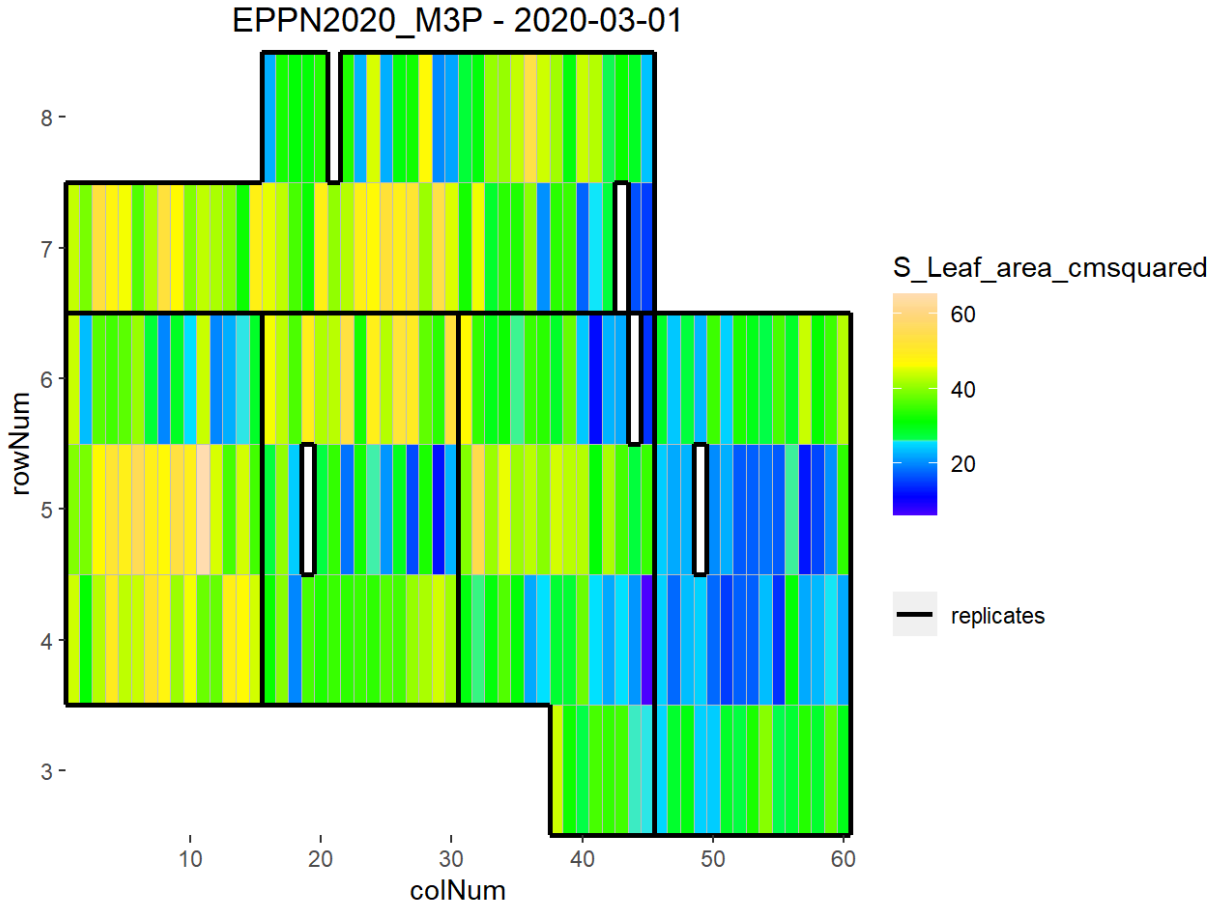

EPPN2020_M3P - 2020-02-24
rowNum
colNum
S_Leaf_area_cmsquared
30
20
10
replicates

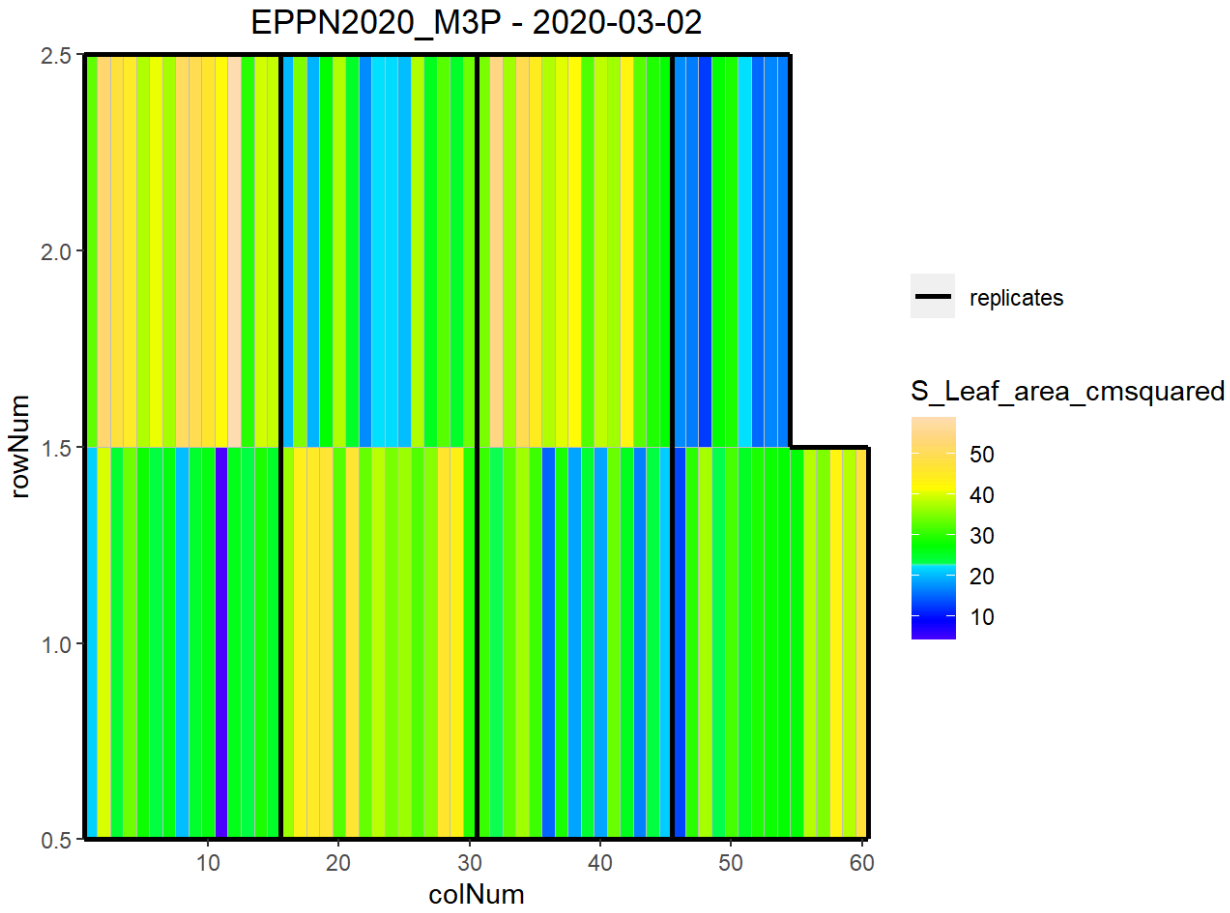

EPPN2020_M3P - 2020-02-25
rowNum
colNum
S_Leaf_area_cmsquared
30
20
10
replicates

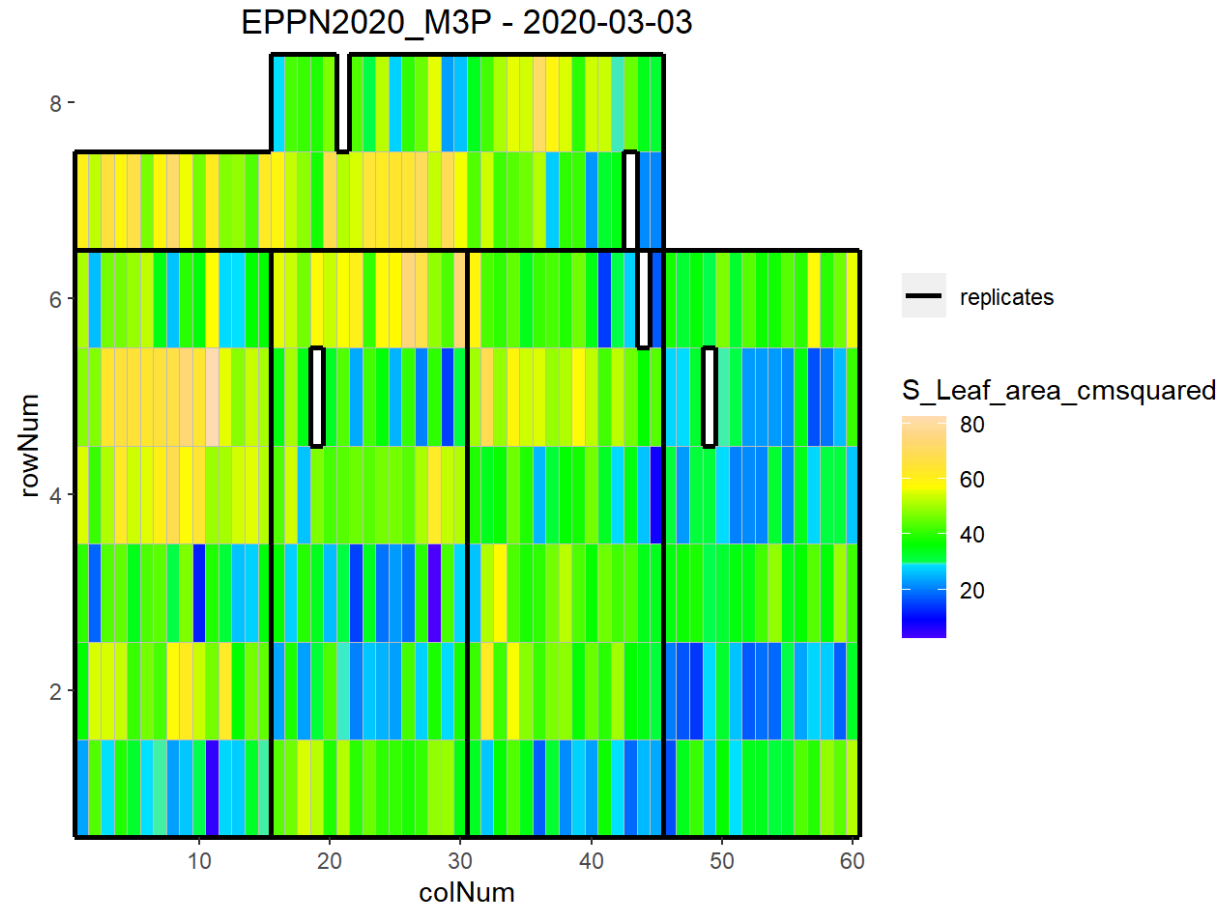

EPPN2020_M3P - 2020-02-26
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
40
30
20
10

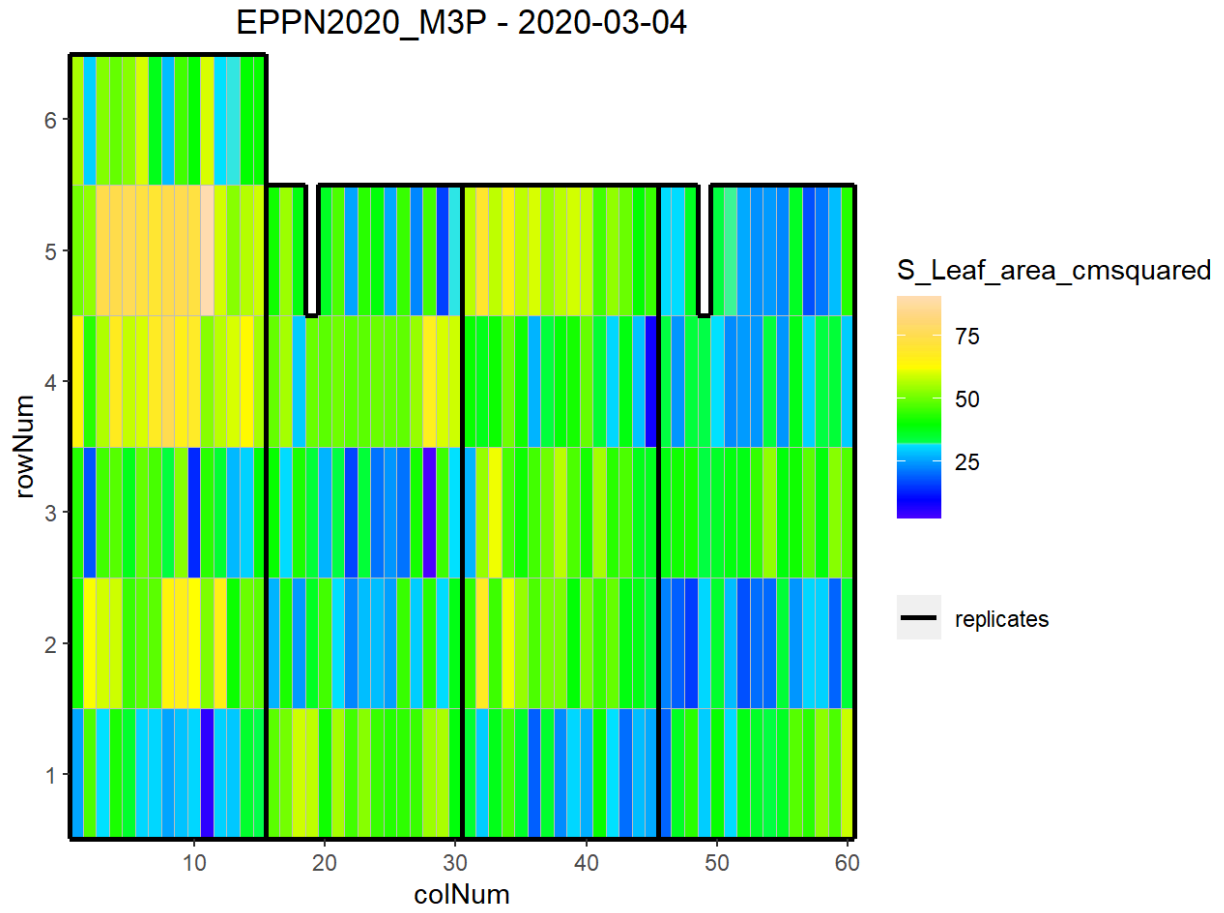

EPPN2020_M3P - 2020-02-28
rowNum
colNum
5
4
3
2
1
10
20
30
40
50
60
S_Leaf_area_cmsquared
50
40
30
20
10
replicates

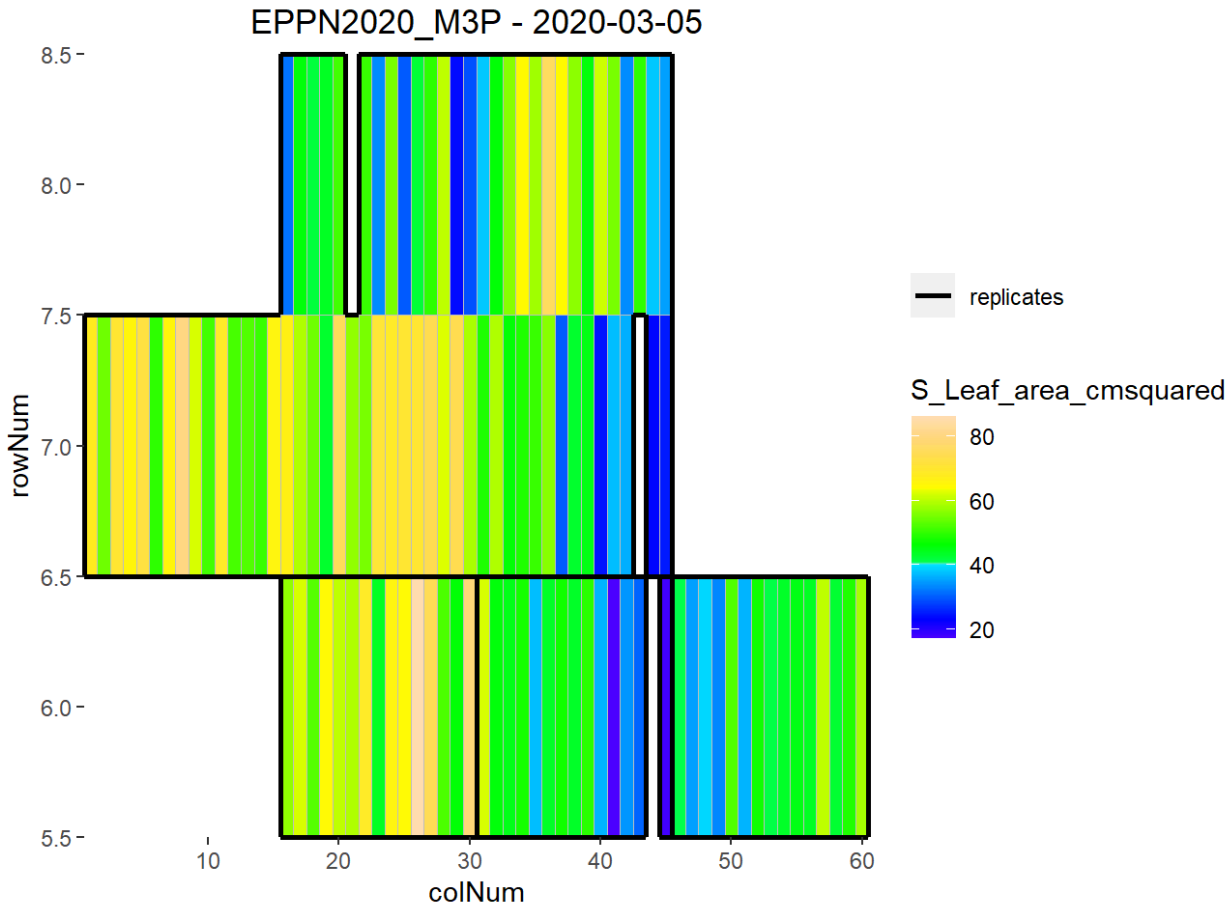

EPPN2020_M3P - 2020-02-29
rowNum
colNum
S_Leaf_area_cmsquared
50
40
30
20
10
replicates

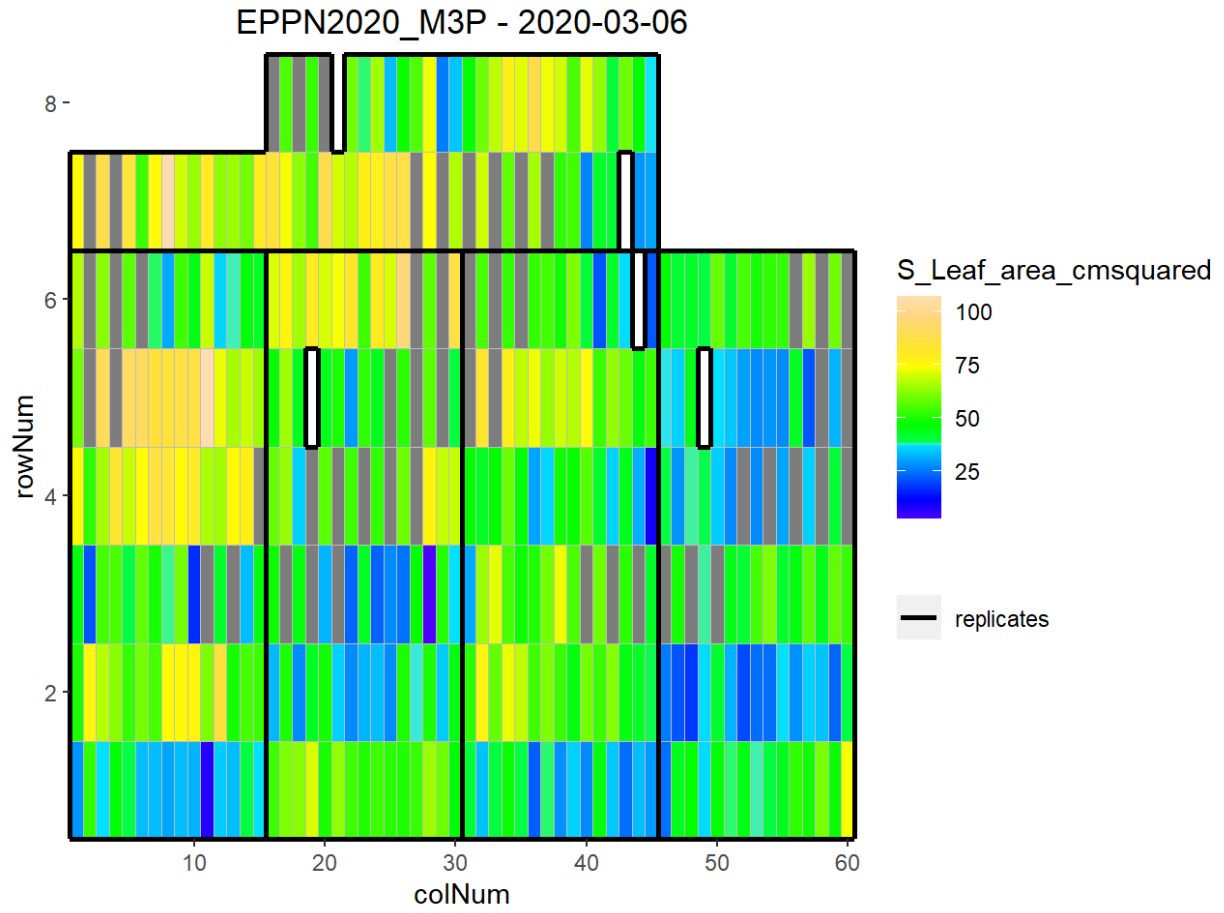

EPPN2020_M3P - 2020-03-01
rowNum
colNum
S_Leaf_area_cmsquared
60
40
20
replicates

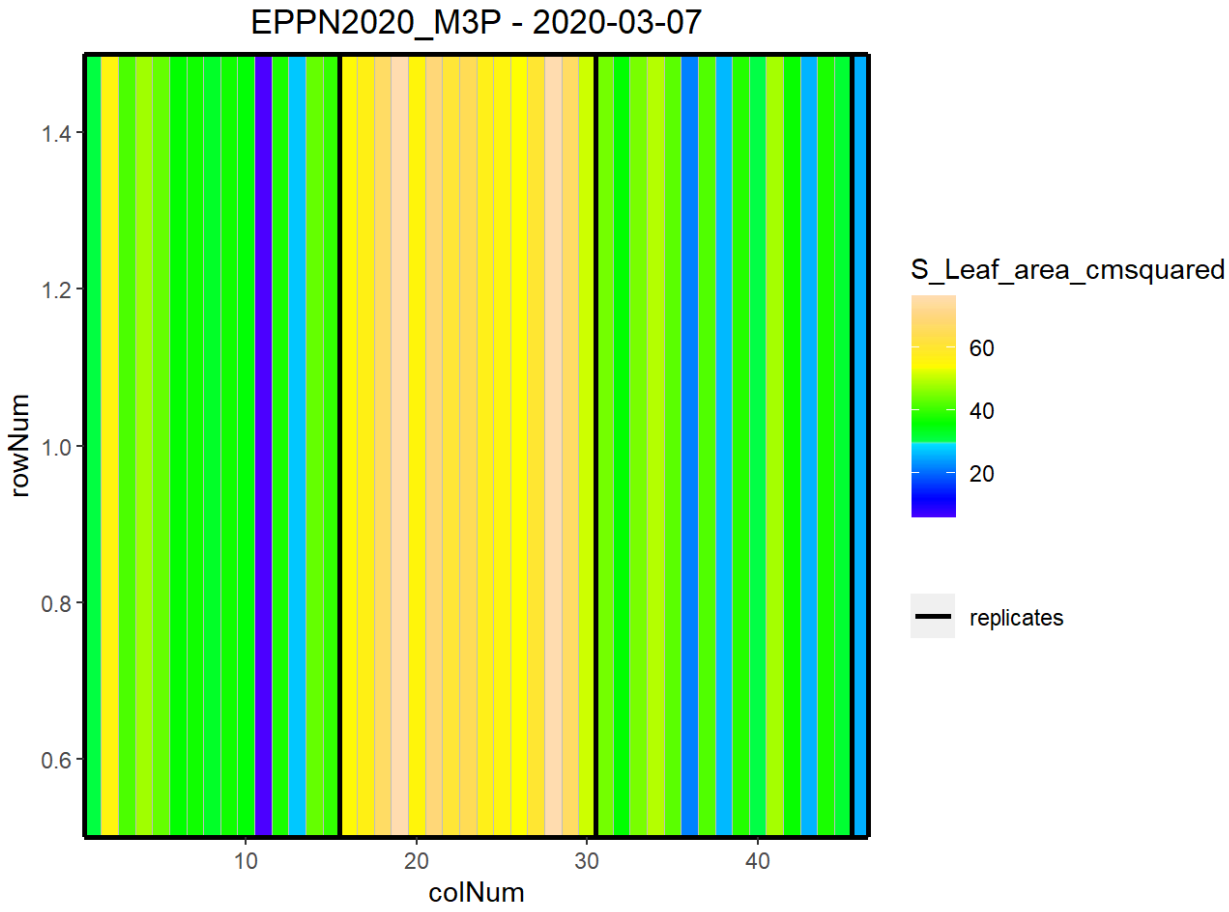

EPPN2020_M3P - 2020-03-02
rowNum
colNum
2.5
2.0
1.5
1.0
0.5
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
50
40
30
20
10

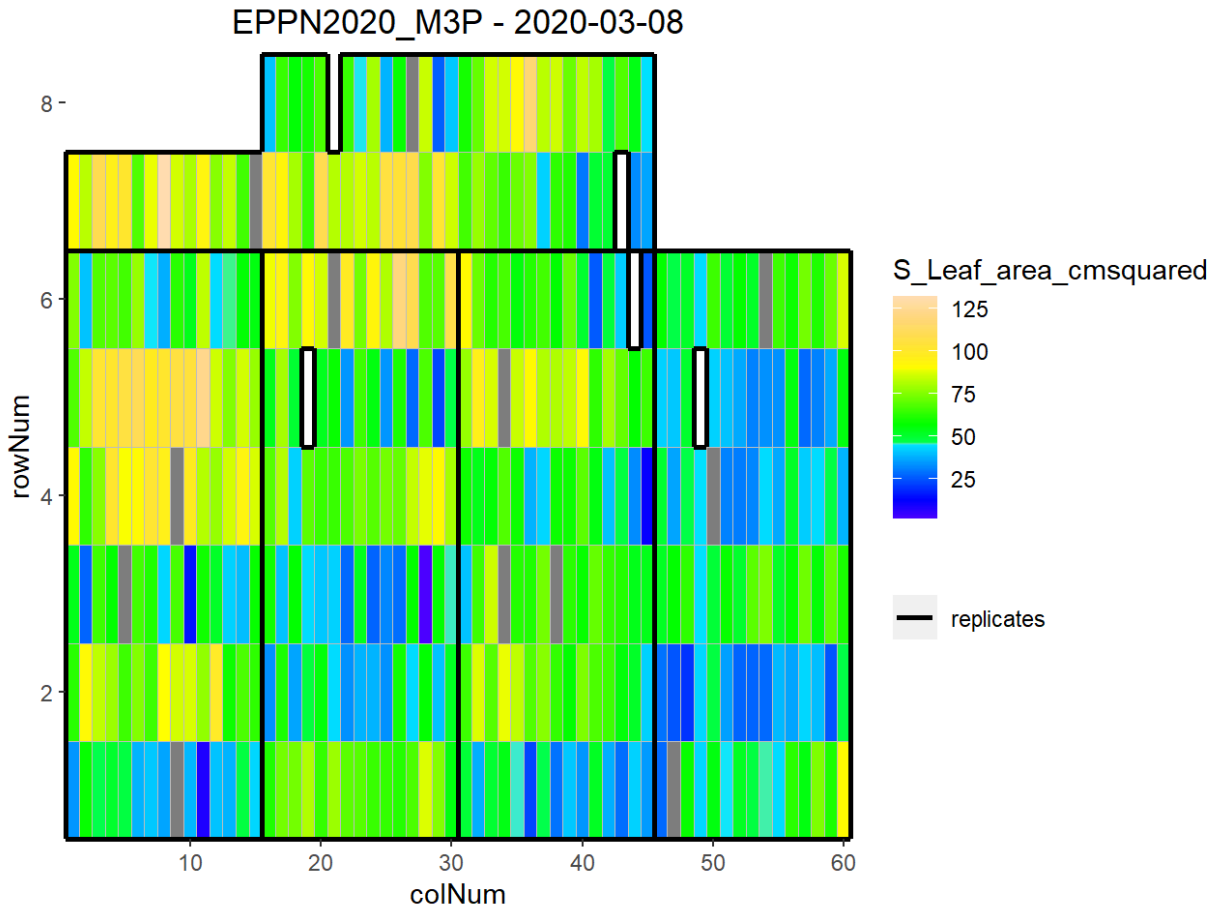

EPPN2020_M3P - 2020-03-03
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
80
60
40
20

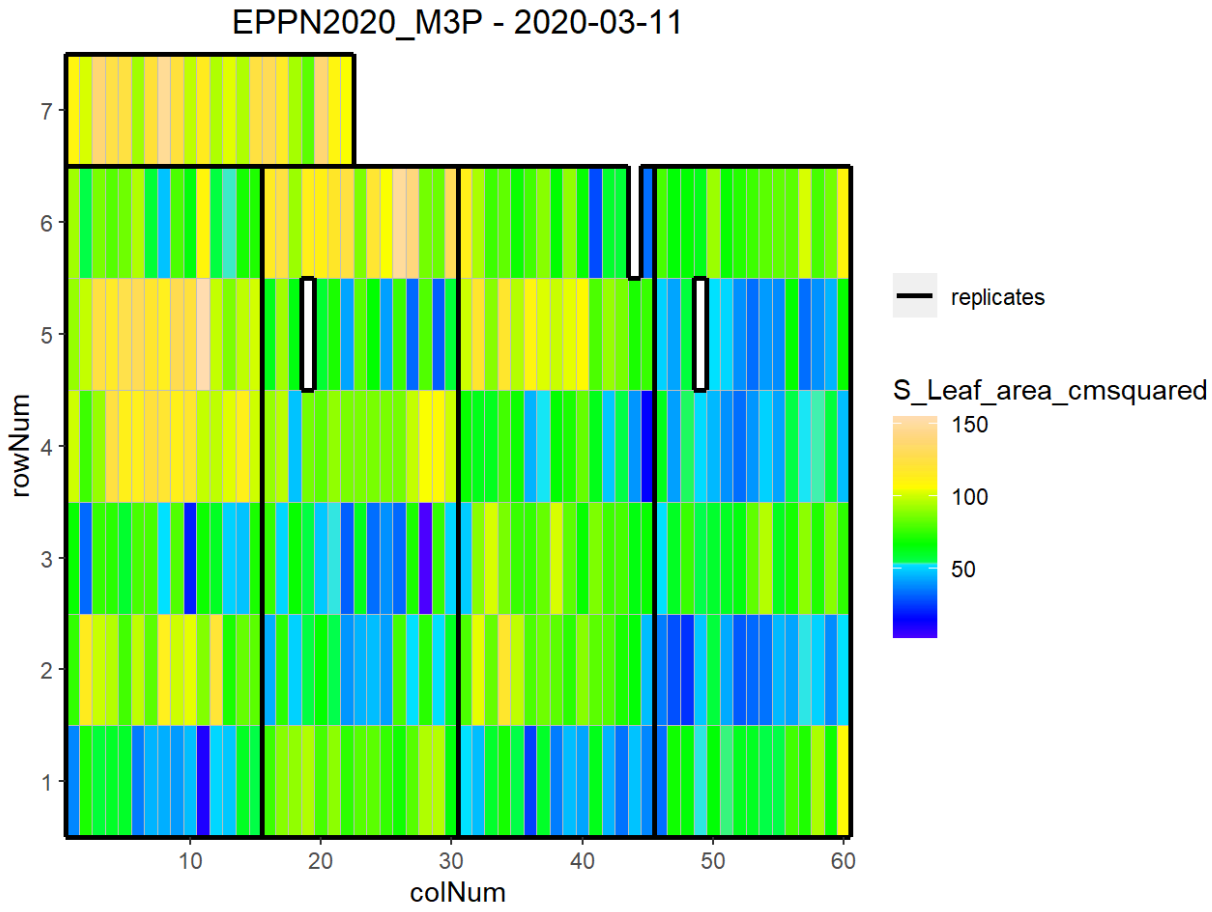

EPPN2020_M3P - 2020-03-04
rowNum
colNum
S_Leaf_area_cmsquared
75
50
25
replicates

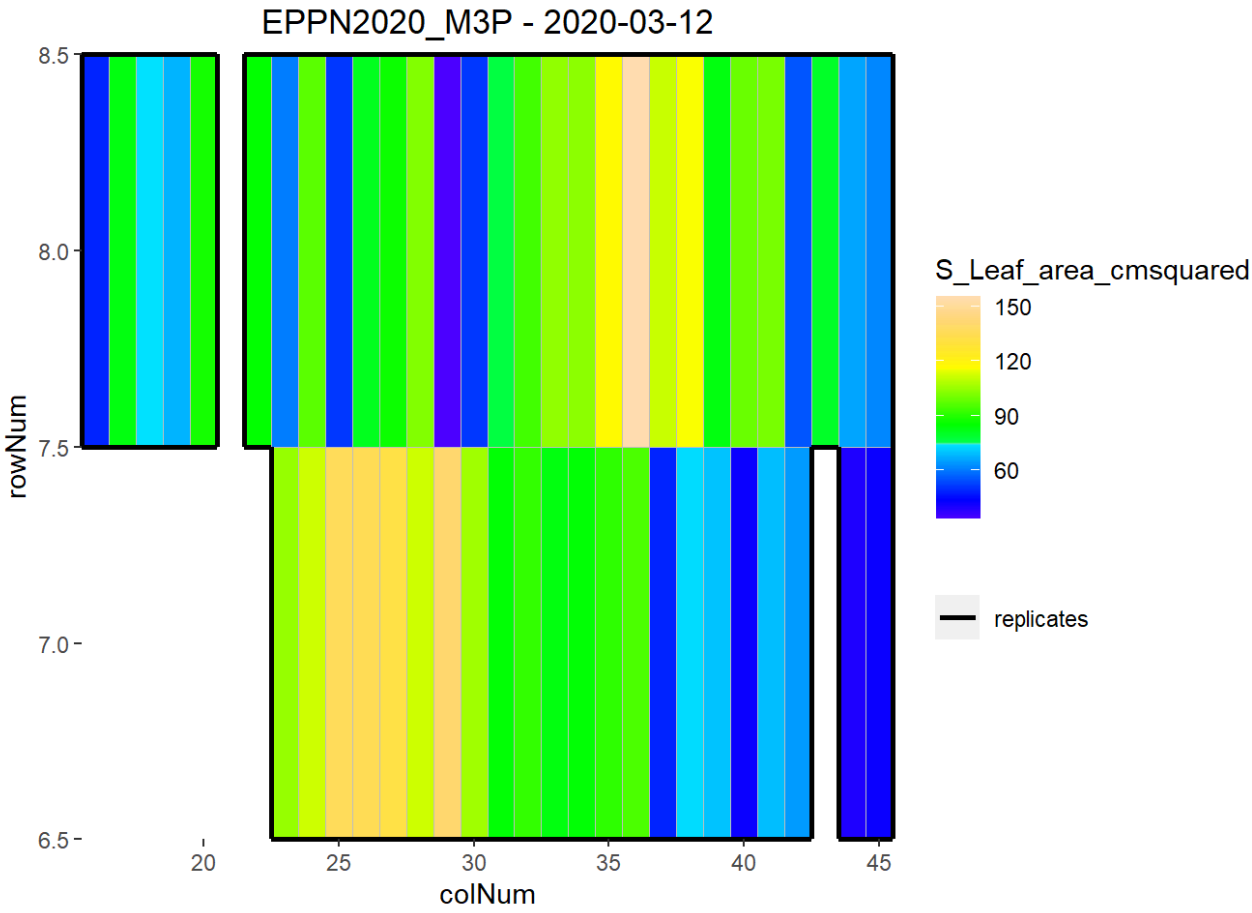

EPPN2020_M3P - 2020-03-05
rowNum
colNum
replicates
S_Leaf_area_cmsquared
80
60
40
20

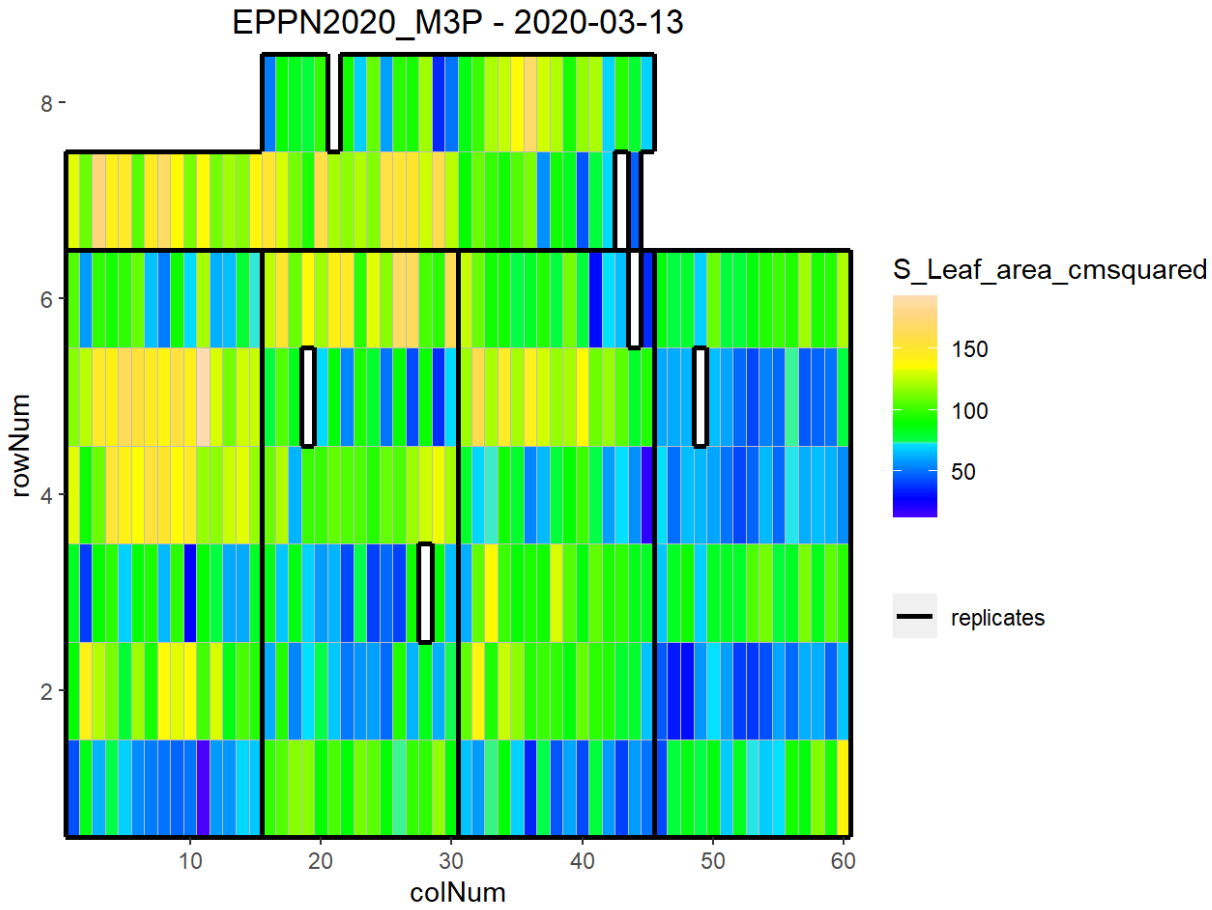

EPPN2020_M3P - 2020-03-06
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Leaf_area_cmsquared
100
75
50
25
replicates

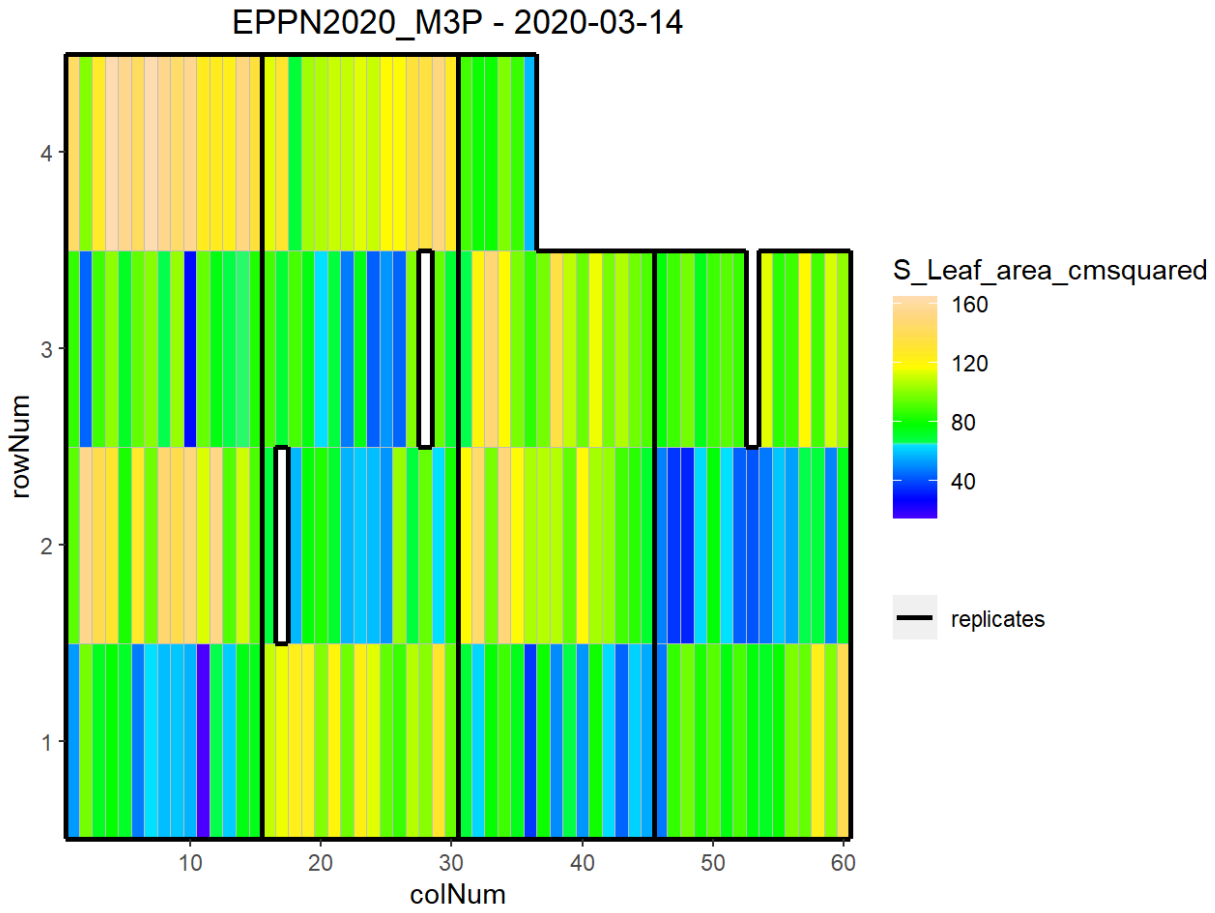

EPPN2020_M3P - 2020-03-07
rowNum
colNum
1.4
1.2
1.0
0.8
0.6
10
20
30
40
S_Leaf_area_cmsquared
60
40
20
replicates

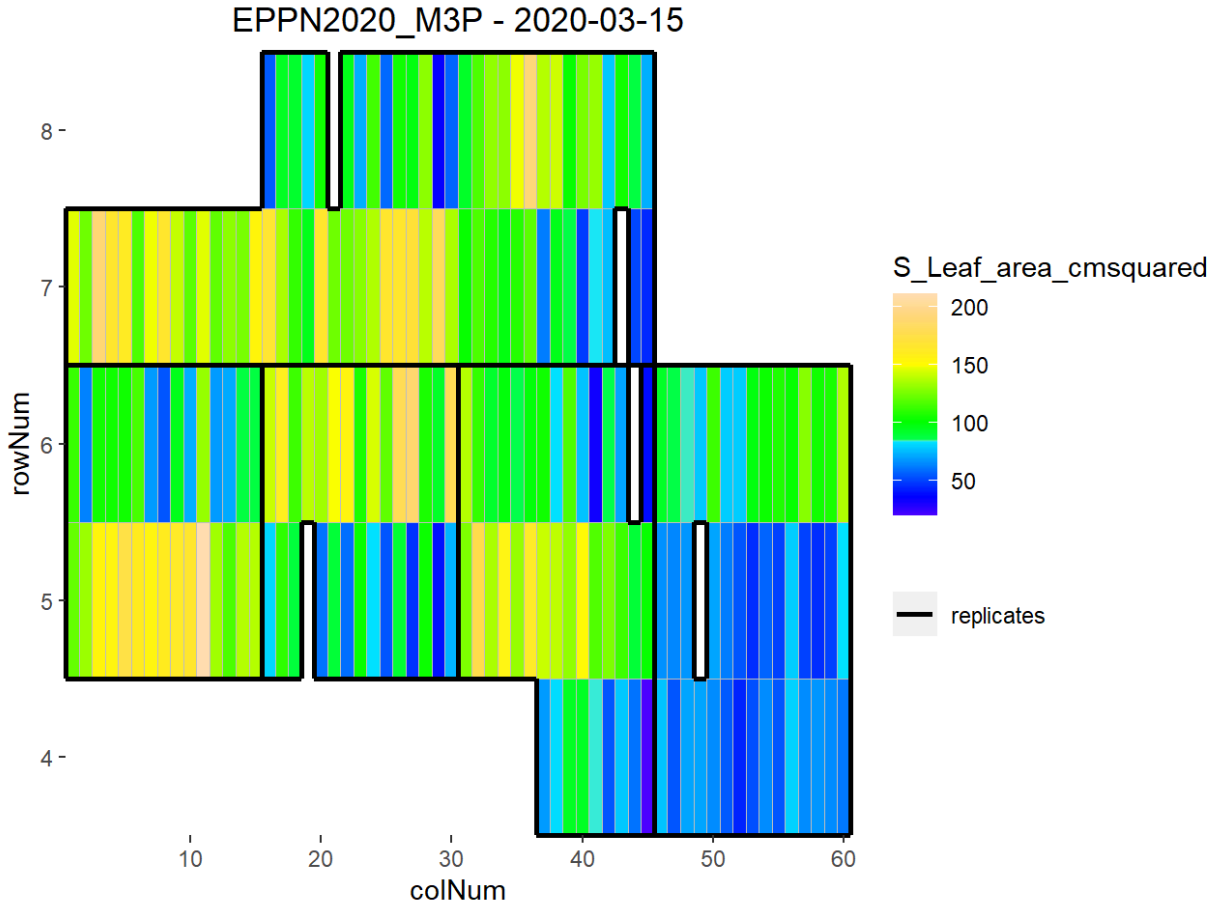

EPPN2020_M3P - 2020-03-08
rowNum
colNum
S_Leaf_area_cmsquared
125
100
75
50
25
replicates

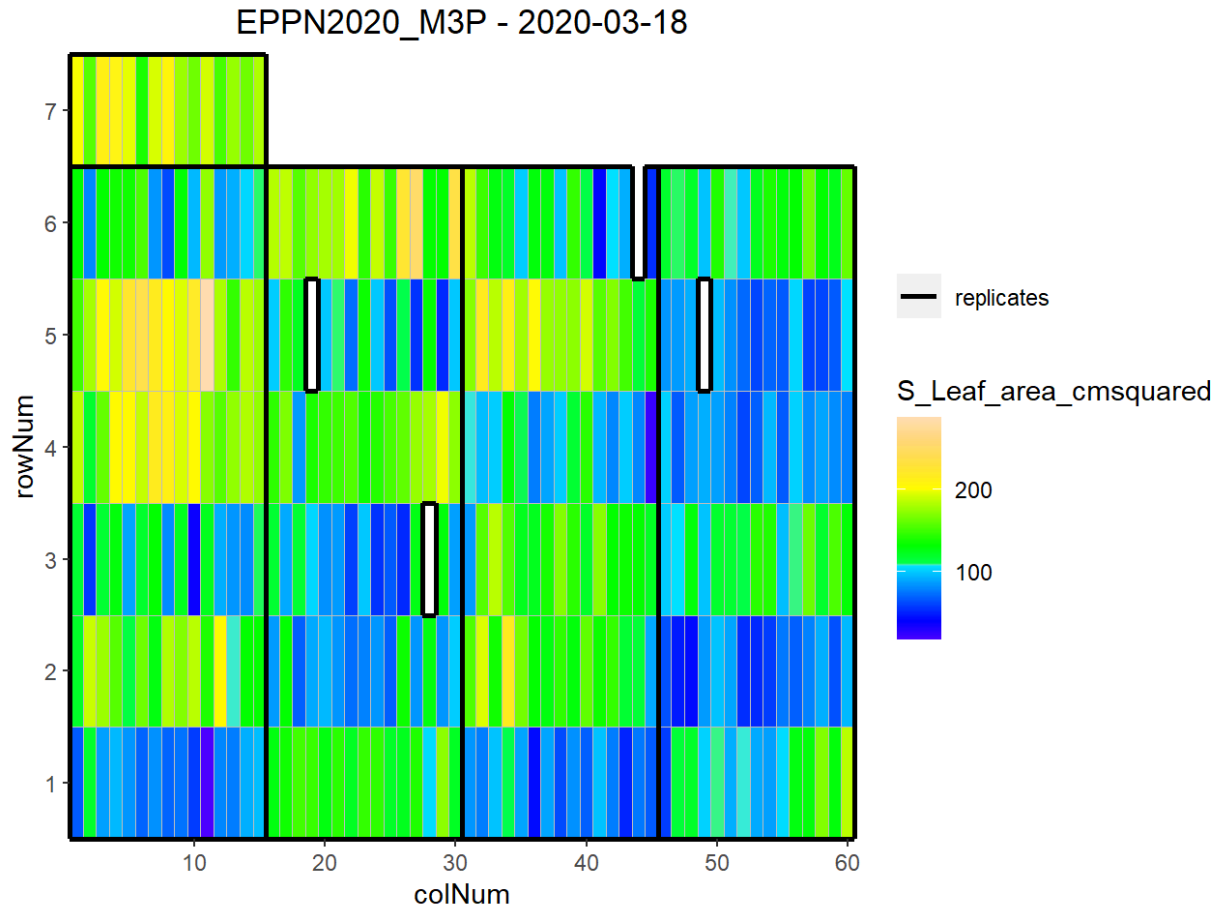

EPPN2020_M3P - 2020-03-11
rowNum
colNum
replicates
S_Leaf_area_cmsquared
150
100
50

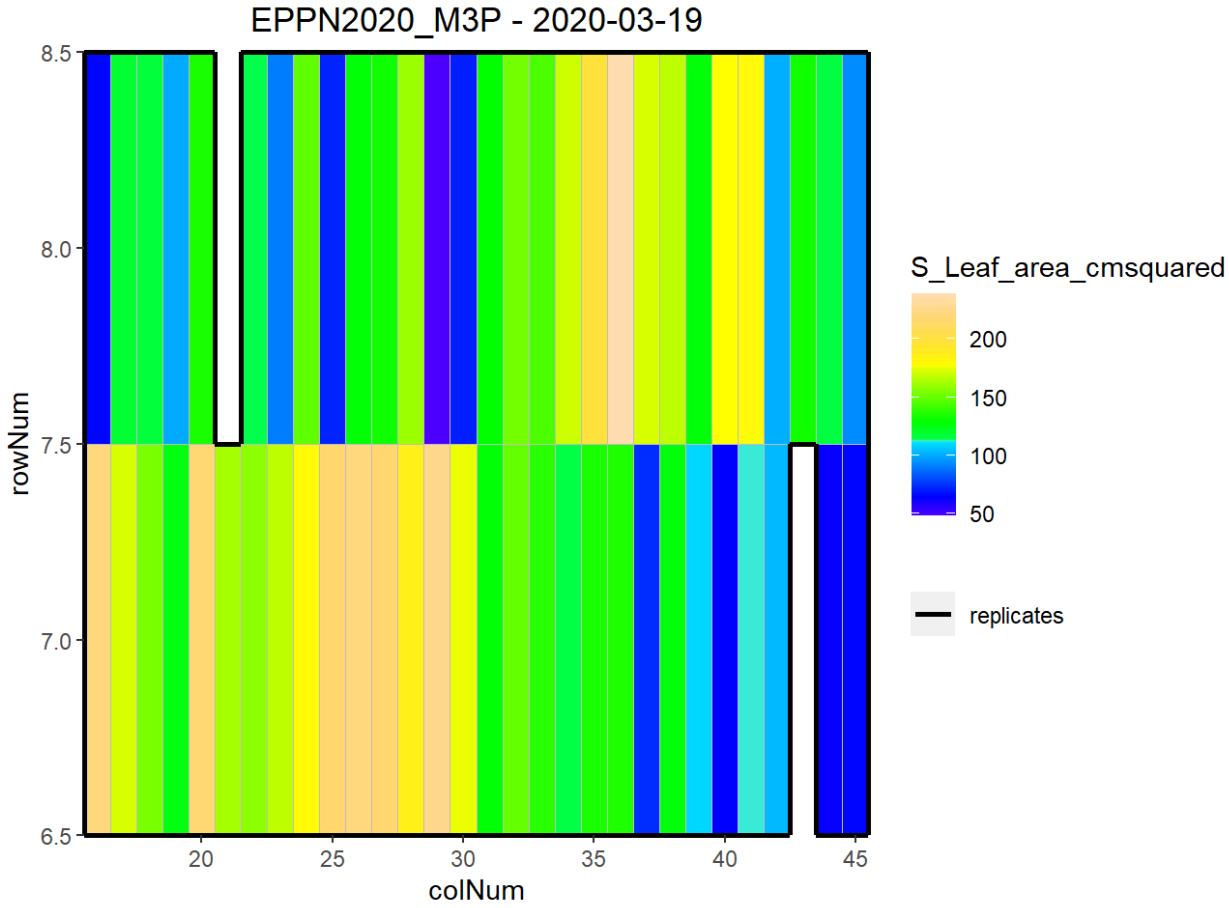

EPPN2020_M3P - 2020-03-12
rowNum
colNum
8.5
8.0
7.5
7.0
6.5
20
25
30
35
40
45
S_Leaf_area_cmsquared
150
120
90
60
replicates

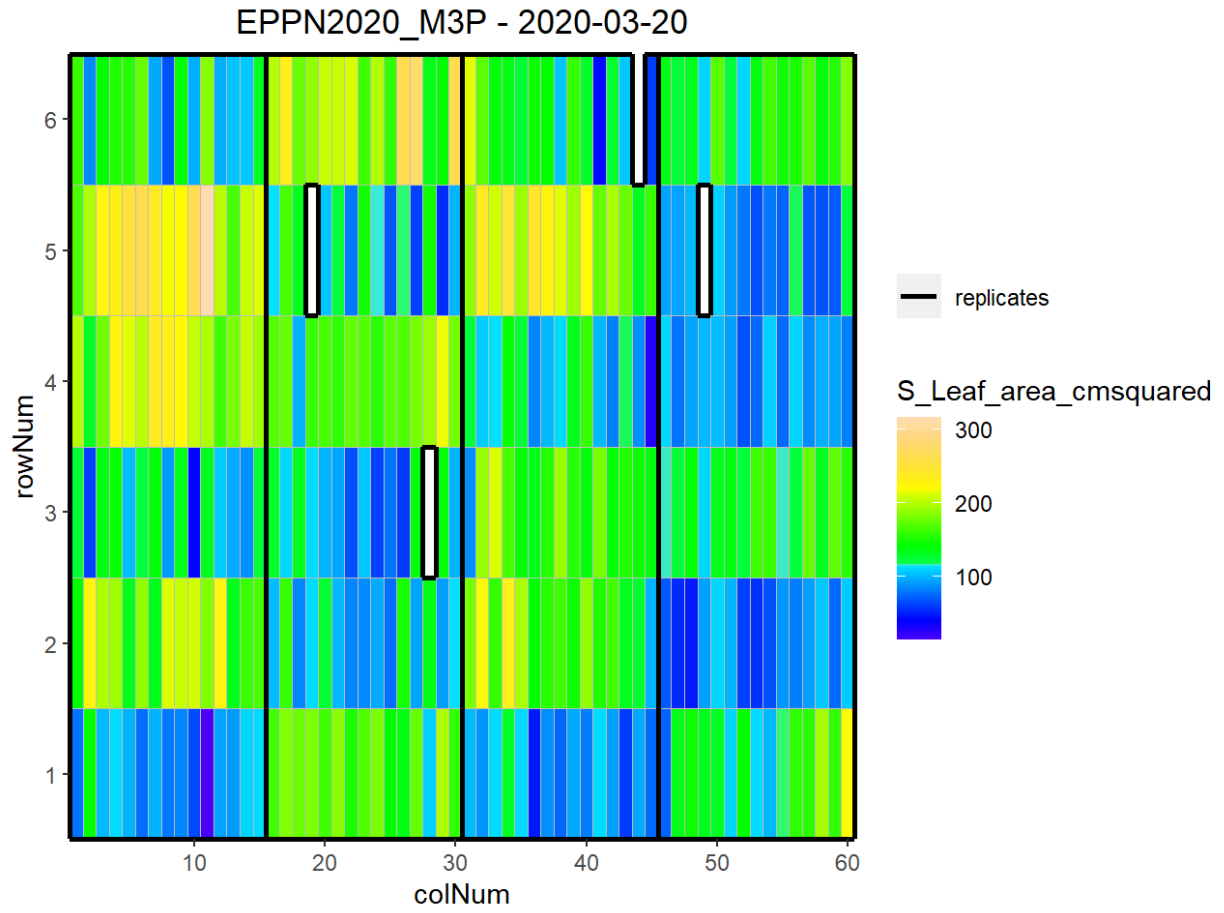

EPPN2020_M3P - 2020-03-13
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Leaf_area_cmsquared
150
100
50
replicates

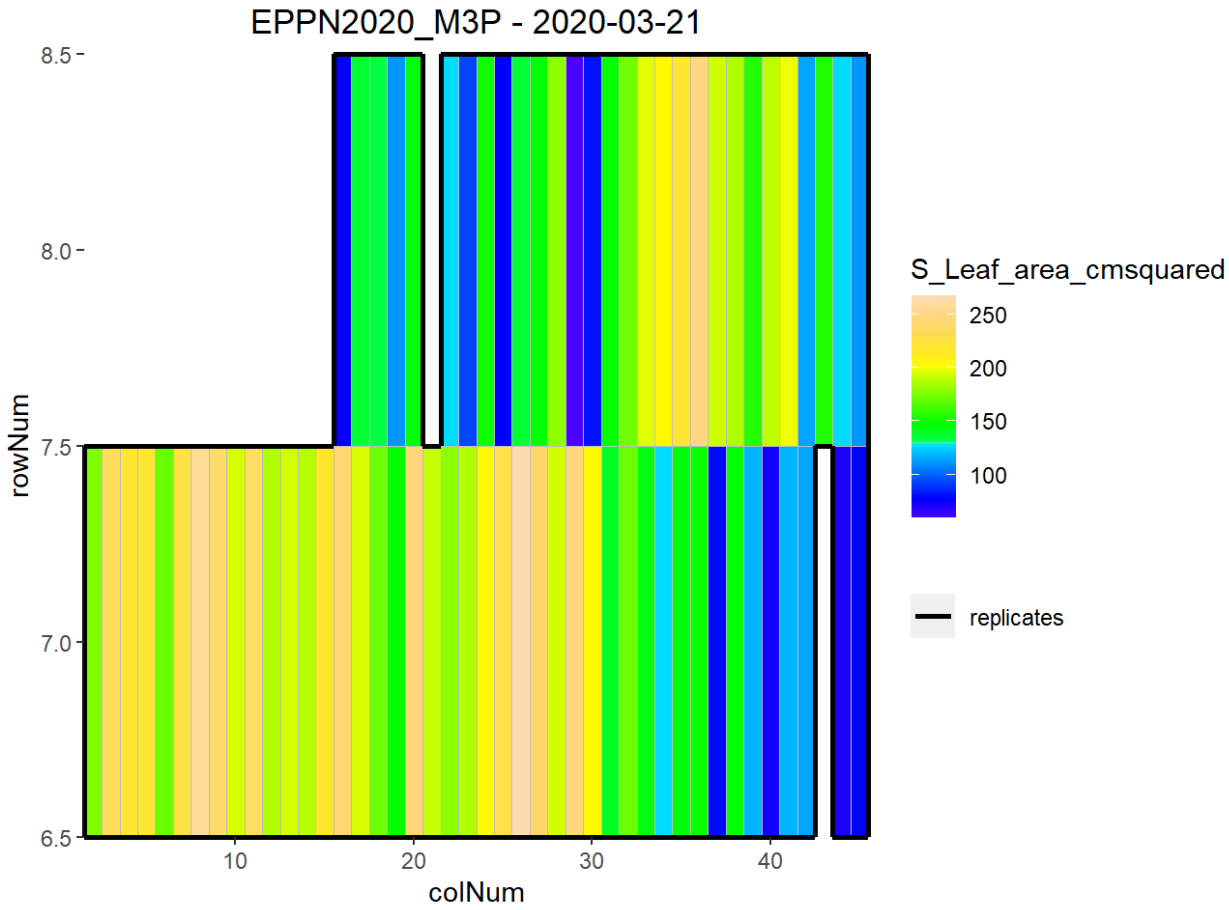

EPPN2020_M3P - 2020-03-14
rowNum
colNum
S_Leaf_area_cmsquared
160
120
80
40
replicates

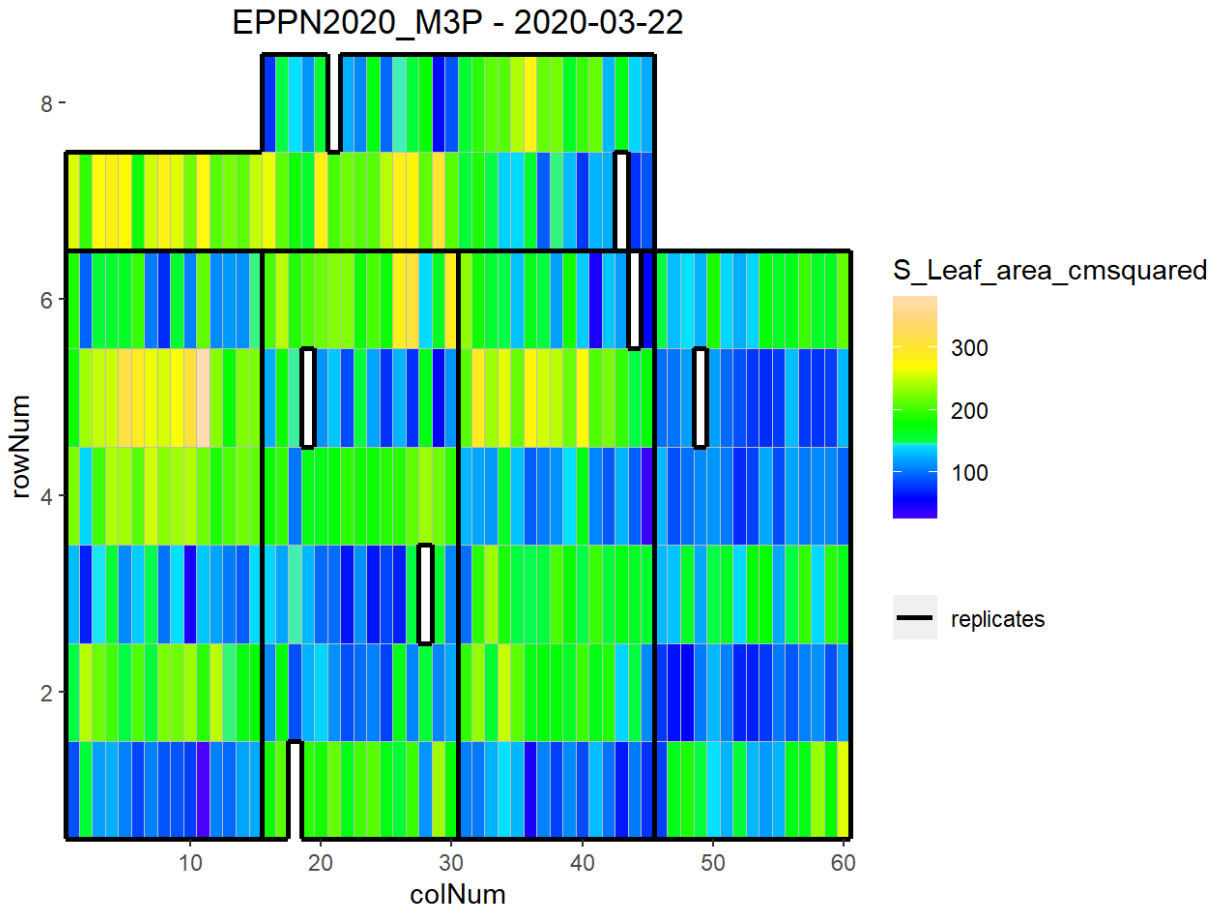

EPPN2020_M3P - 2020-03-15
rowNum
colNum
S_Leaf_area_cmsquared
200
150
100
50
replicates

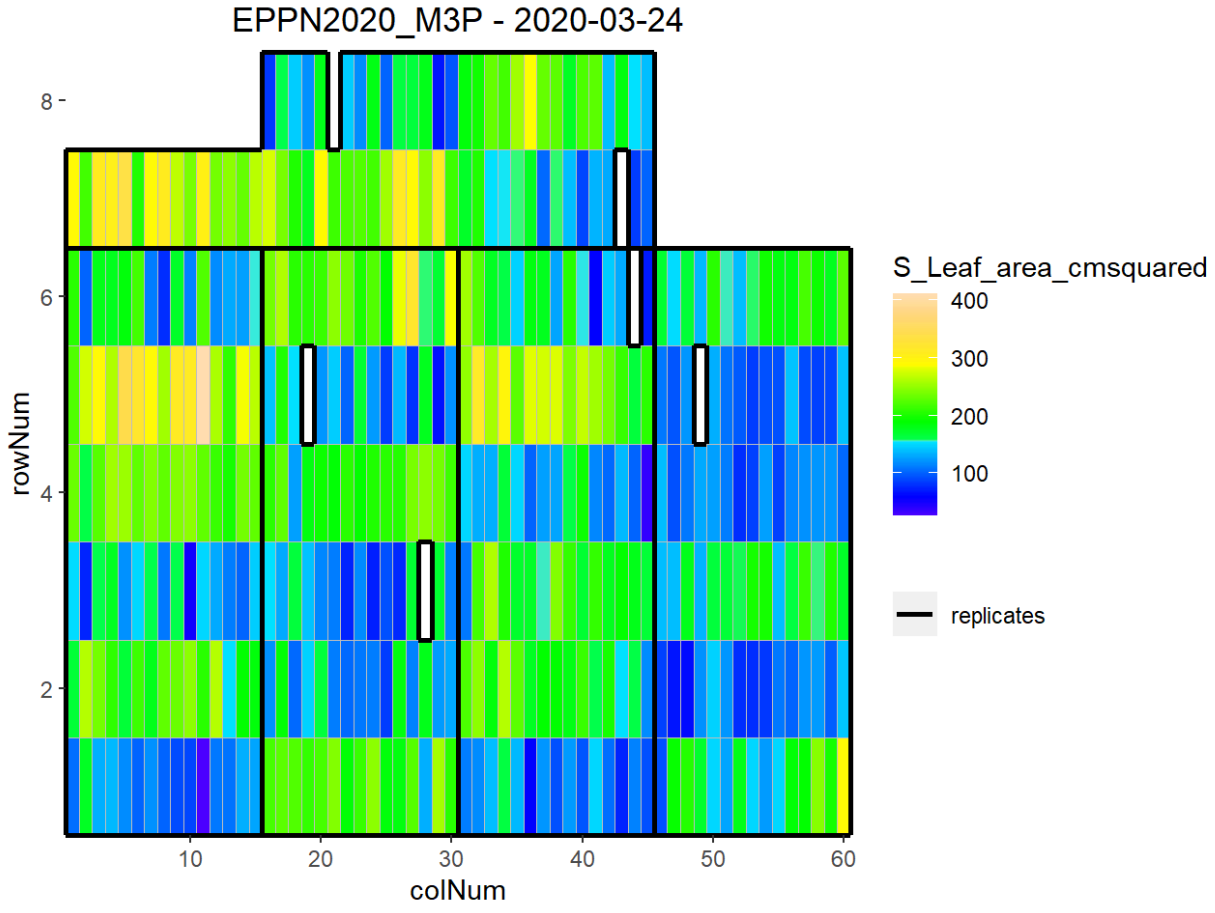

EPPN2020_M3P - 2020-03-18
rowNum
colNum
replicates
S_Leaf_area_cmsquared
200
100

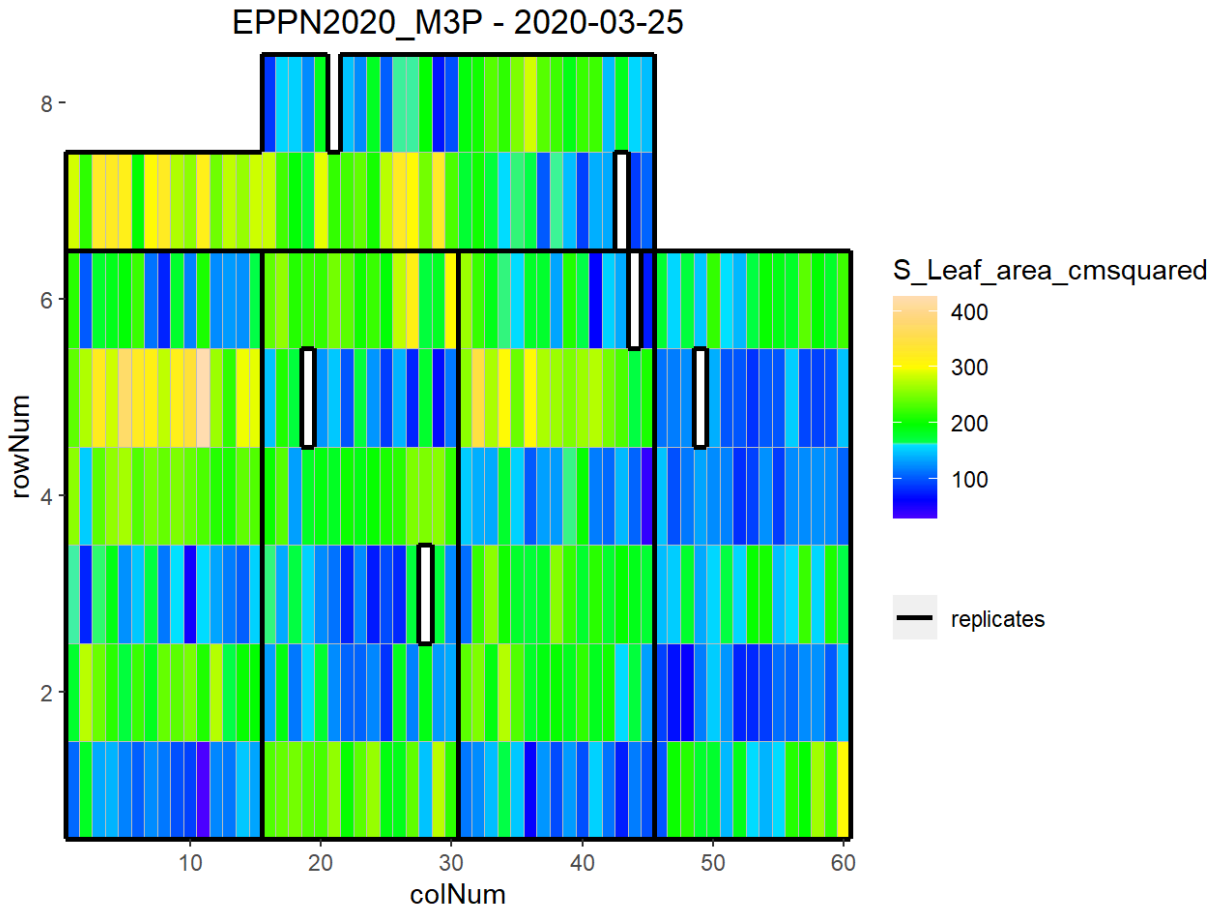

EPPN2020_M3P - 2020-03-19
rowNum
colNum
S_Leaf_area_cmsquared
200
150
100
50
replicates

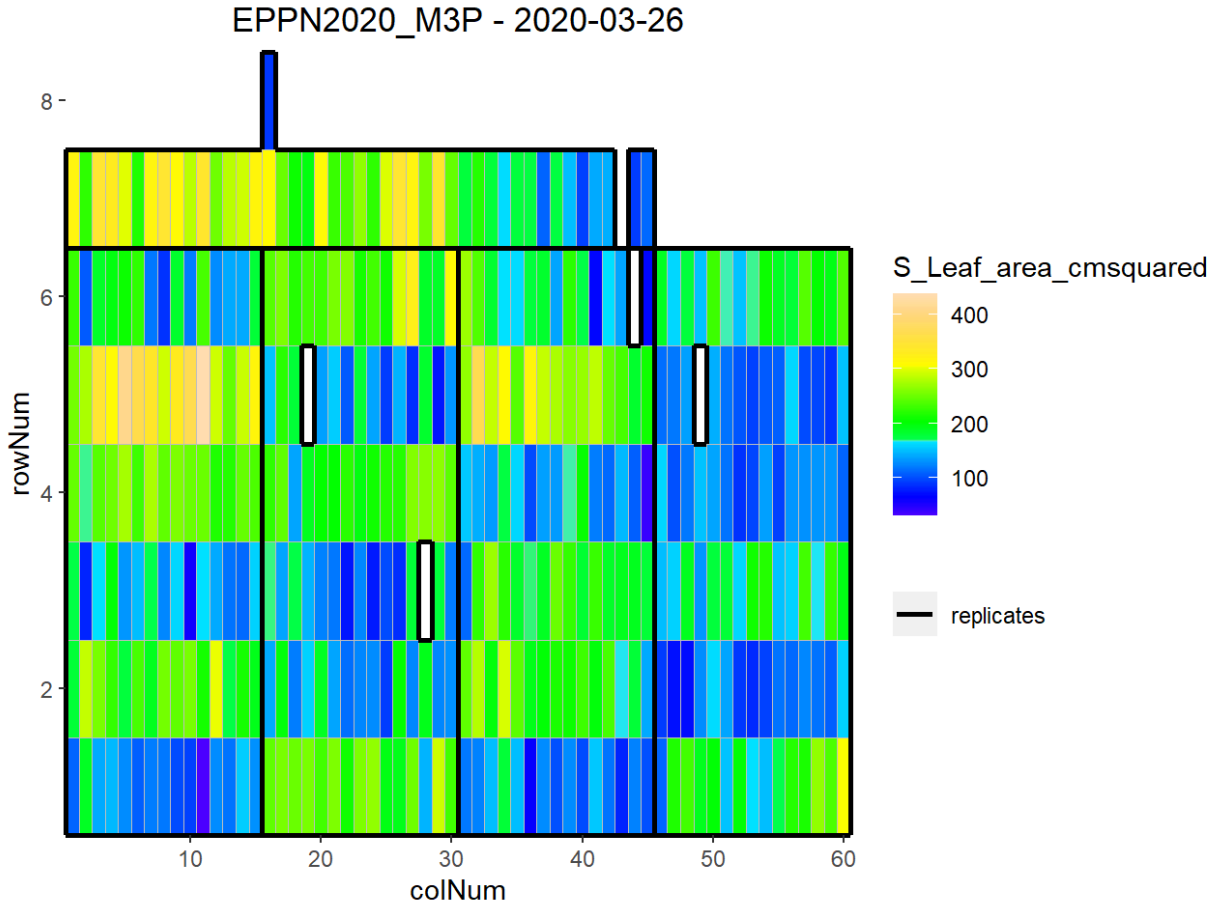

EPPN2020_M3P - 2020-03-20
rowNum
colNum
1
2
3
4
5
6
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
300
200
100

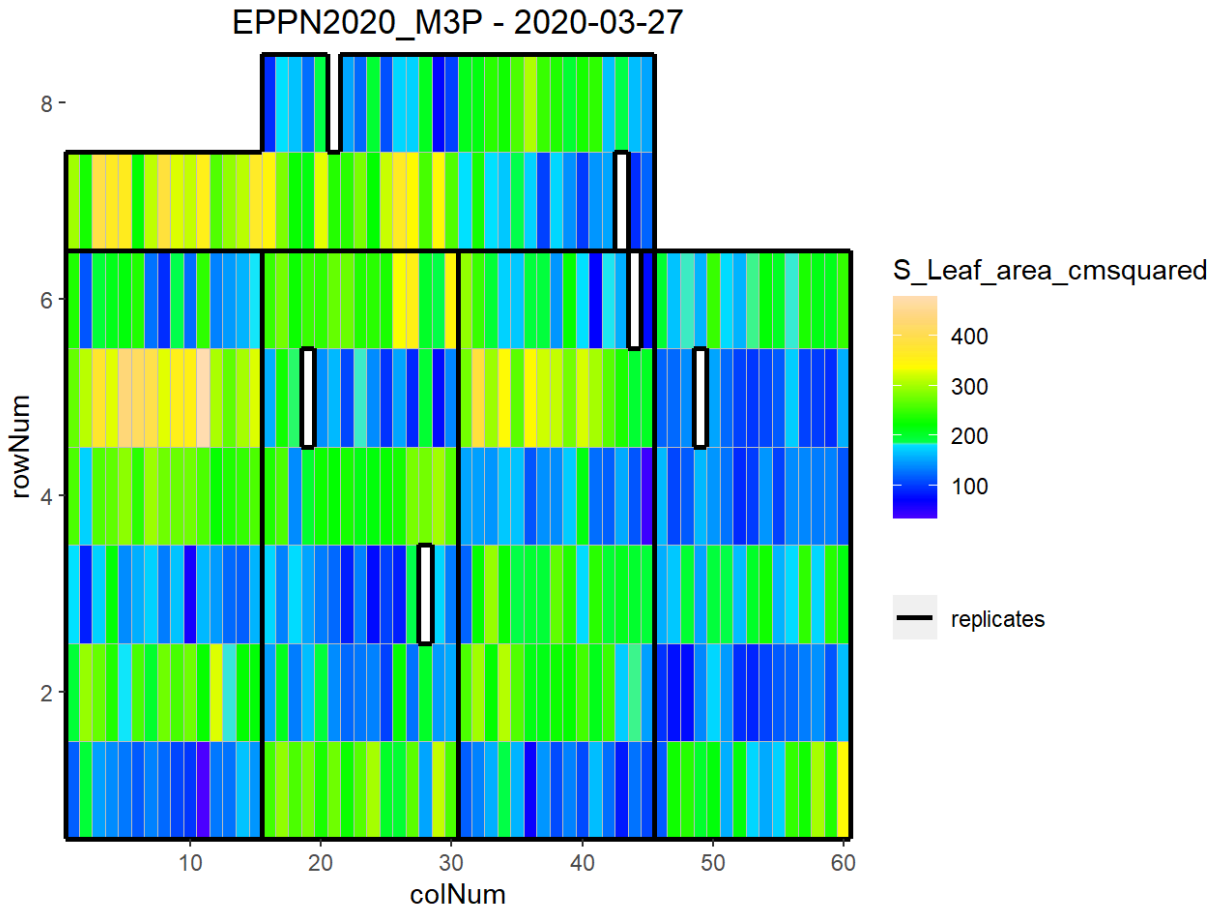

EPPN2020_M3P - 2020-03-21
rowNum
colNum
6.5
7.0
7.5
8.0
8.5
10
20
30
40
S_Leaf_area_cmsquared
250
200
150
100
replicates

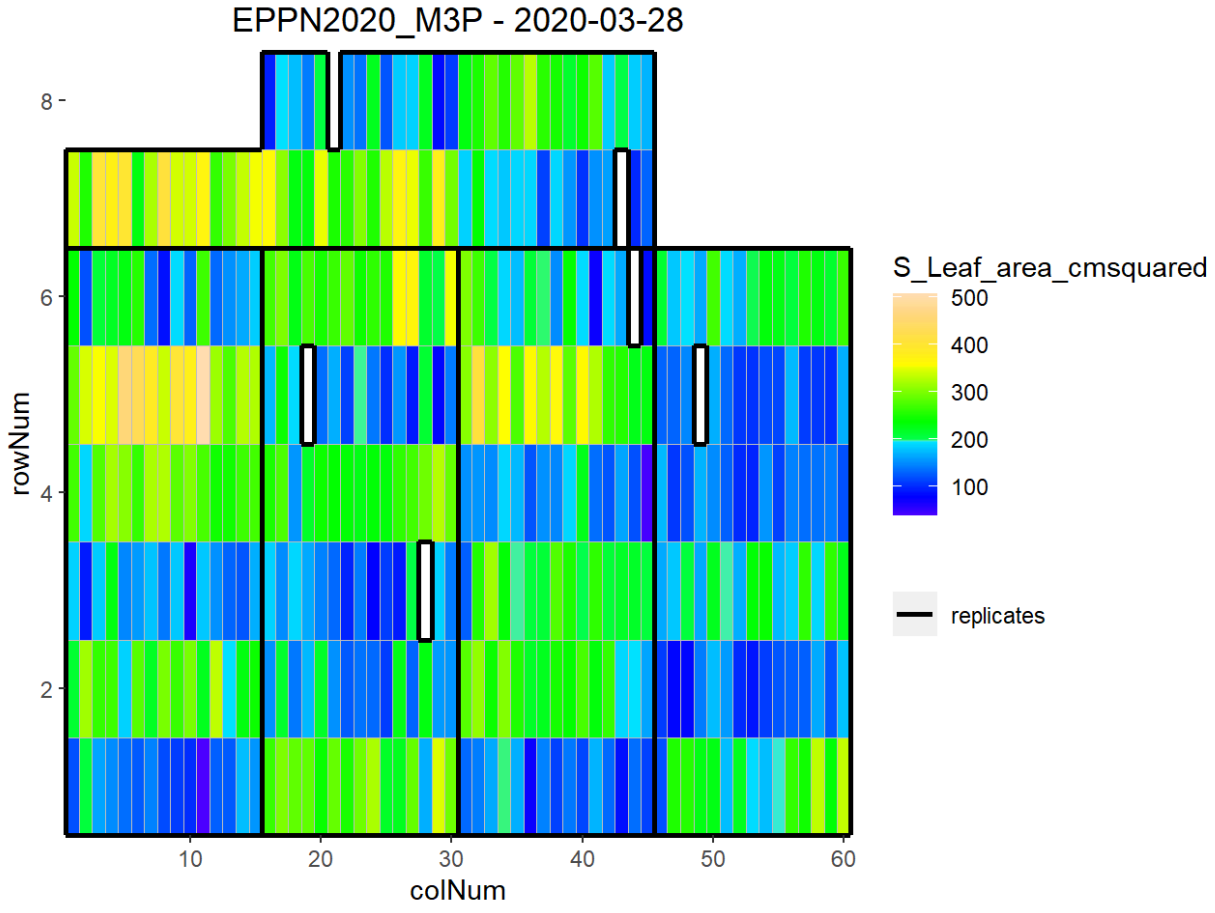

EPPN2020_M3P - 2020-03-22
rowNum
colNum
S_Leaf_area_cmsquared
300
200
100
replicates

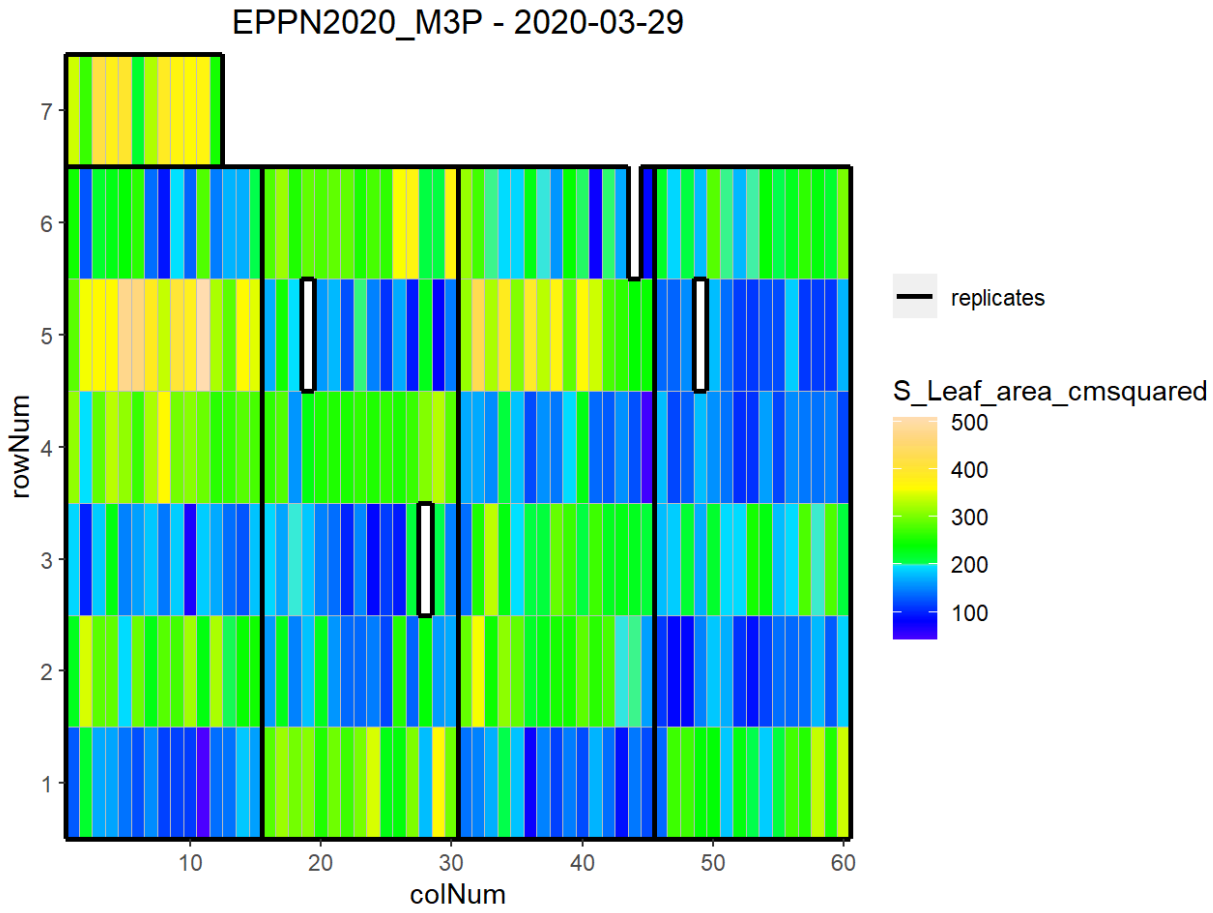

EPPN2020_M3P - 2020-03-24
rowNum
colNum
S_Leaf_area_cmsquared
400
300
200
100
replicates

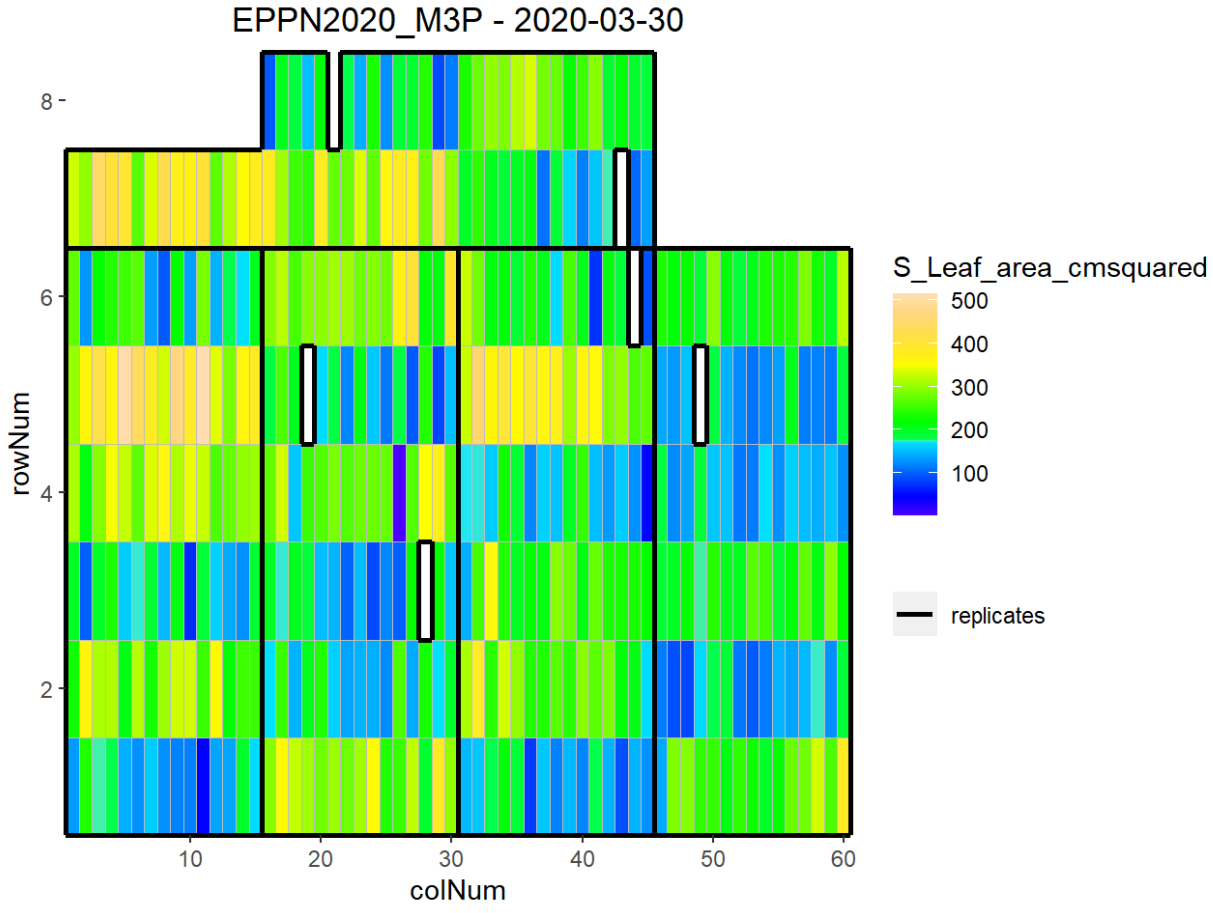

EPPN2020_M3P - 2020-03-25
rowNum
colNum
S_Leaf_area_cmsquared
400
300
200
100
replicates

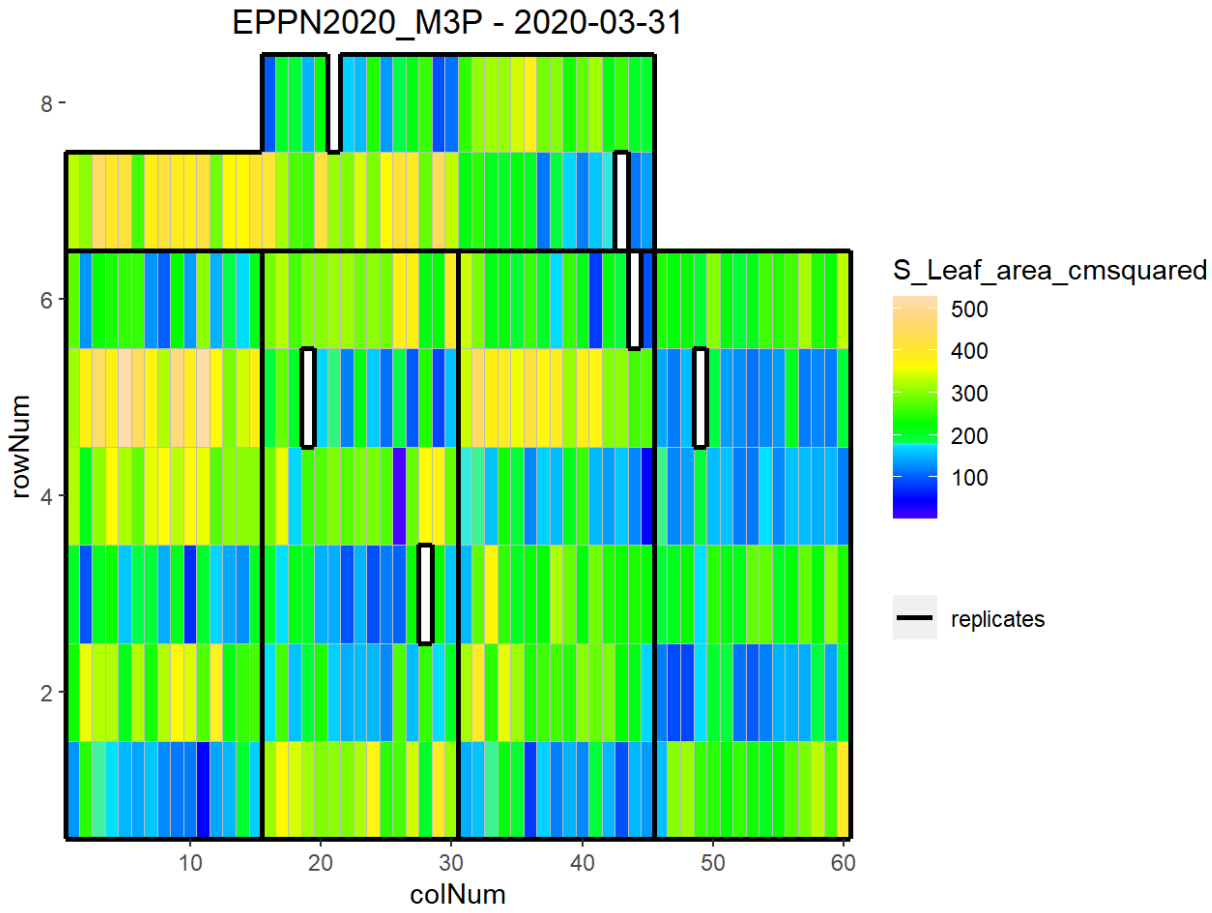

EPPN2020_M3P - 2020-03-26
rowNum
colNum
S_Leaf_area_cmsquared
400
300
200
100
replicates

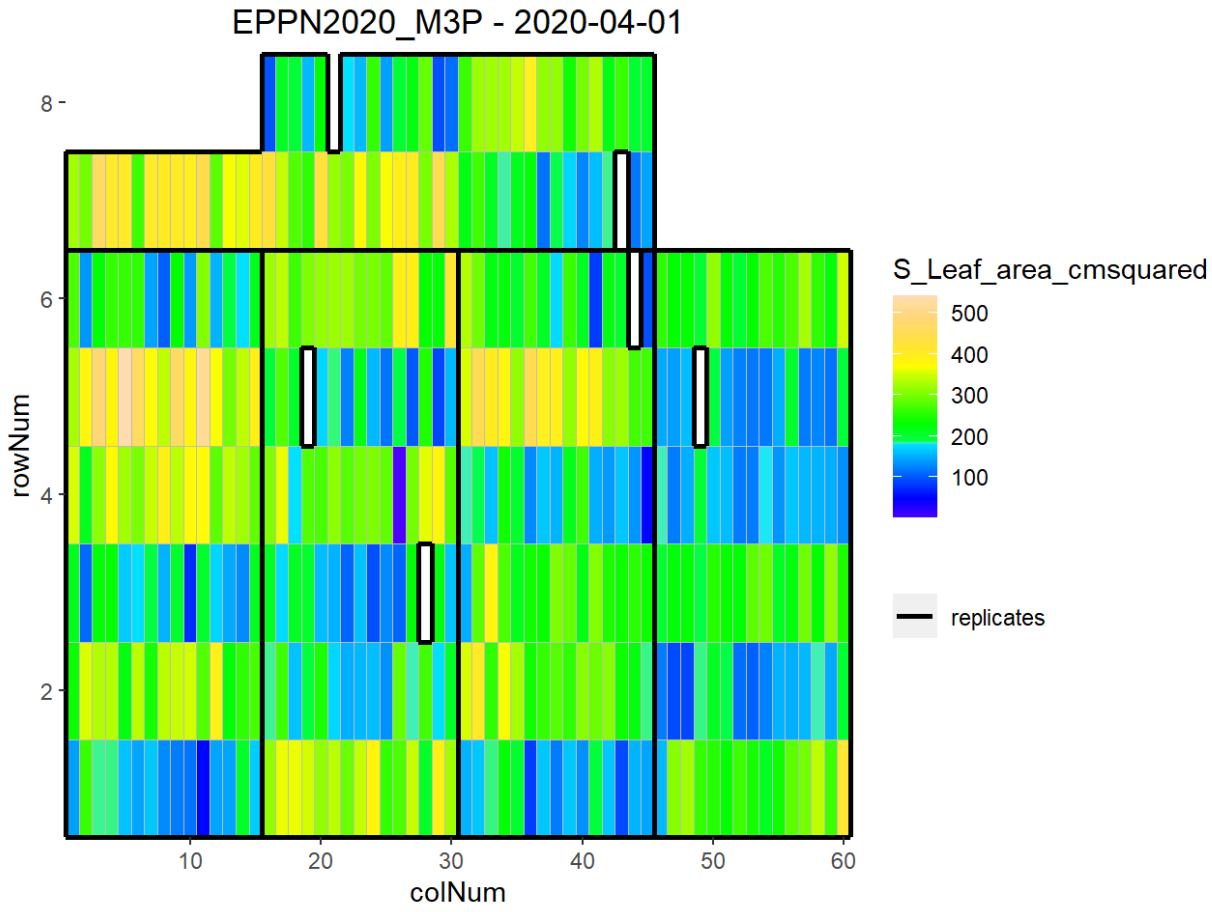

EPPN2020_M3P - 2020-03-27
rowNum
colNum
S_Leaf_area_cmsquared
400
300
200
100
replicates


EPPN2020_M3P - 2020-03-28
rowNum
colNum
S_Leaf_area_cmsquared
500
400
300
200
100
replicates


EPPN2020_M3P - 2020-03-29
rowNum
colNum
1
2
3
4
5
6
7
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
500
400
300
200
100

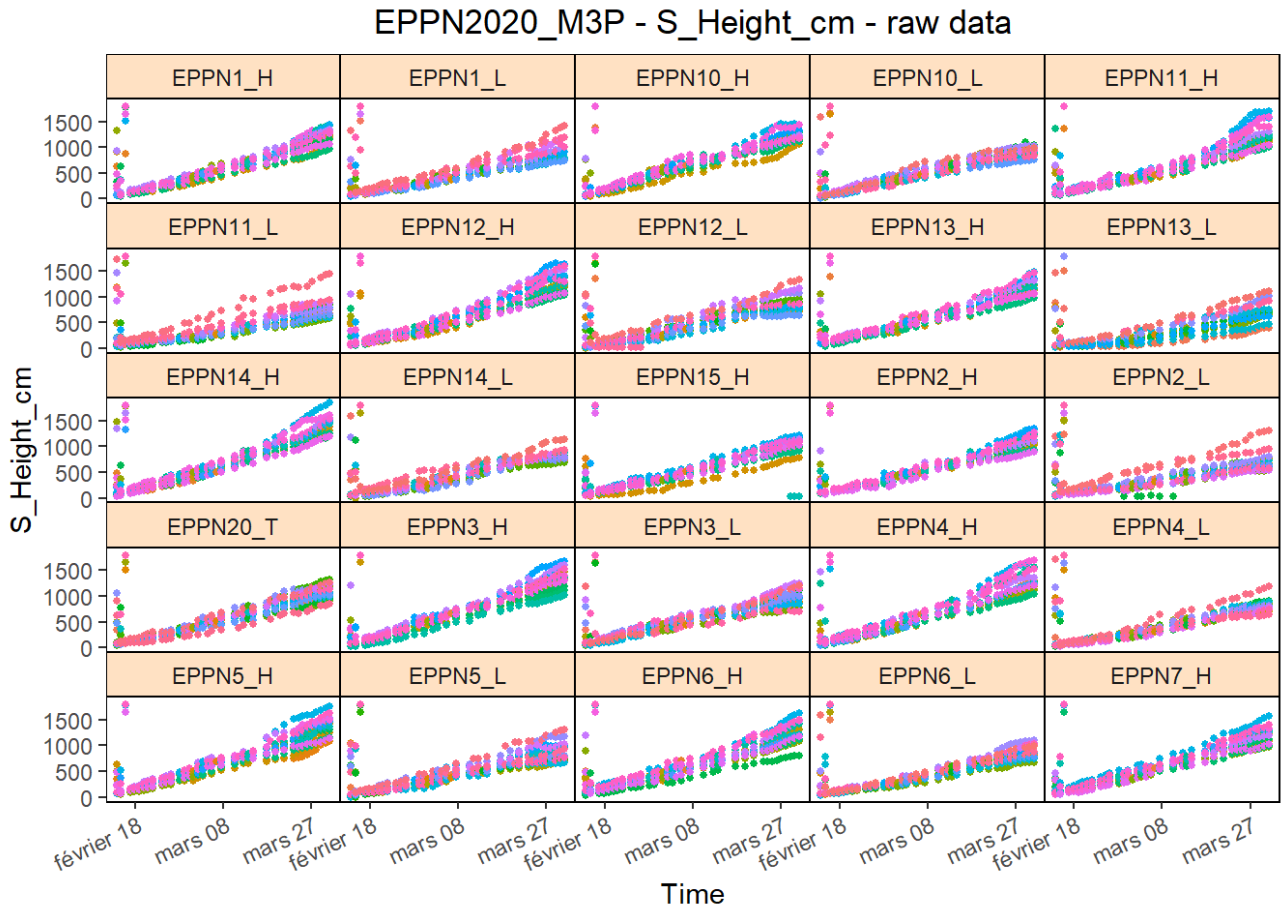

EPPN2020_M3P - 2020-03-30
rowNum
colNum
2
4
6
8
10
20
30
40
50
60
S_Leaf_area_cmsquared
500
400
300
200
100
replicates

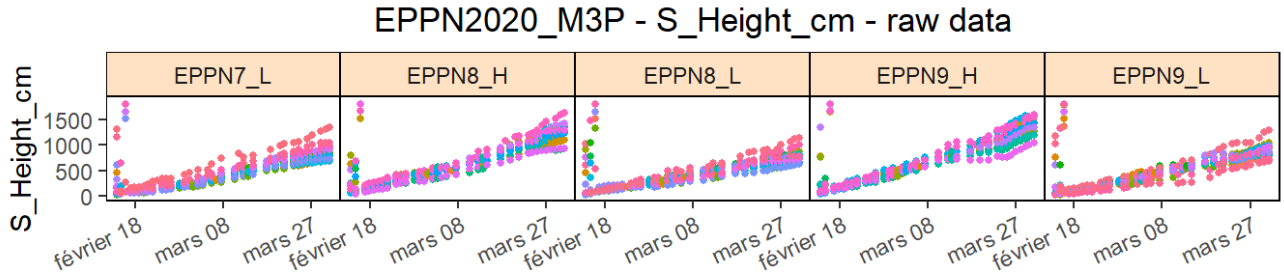




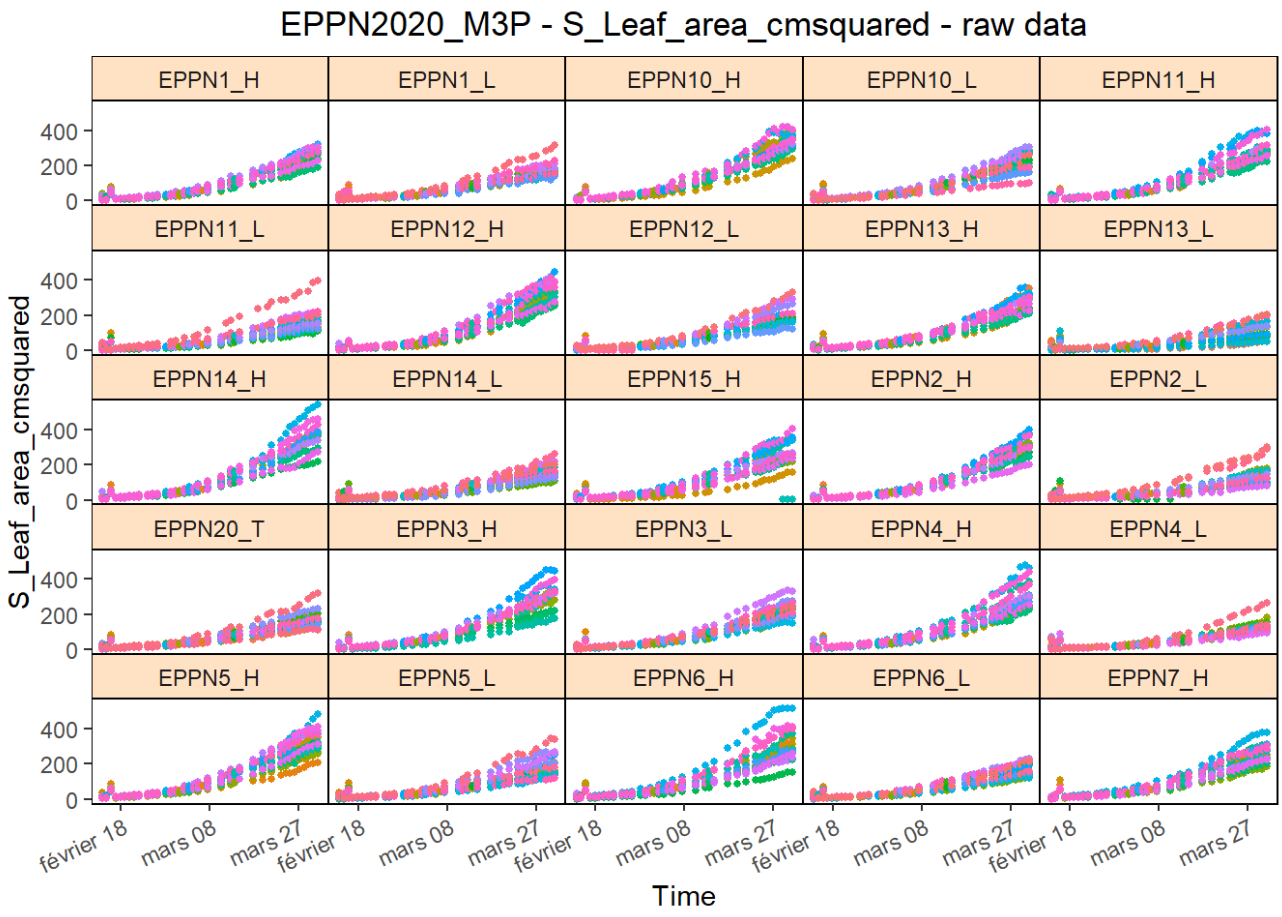
Check time course of raw data per time point

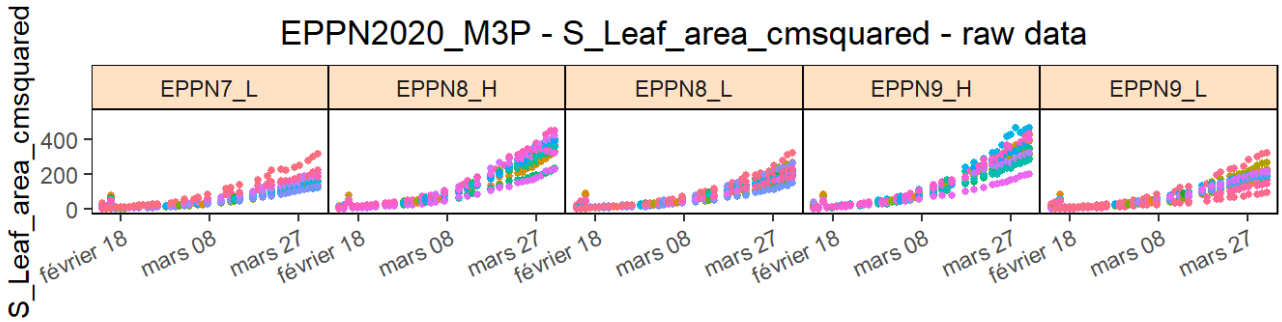
```
for (trait_name in traits) {
  plot(timePoint_S,
     traits = trait_name,
     plotType = "raw")
}
```


EPPN2020_M3P - S_Height_cm - raw data
EPPN1_H
EPPN1_L
EPPN10_H
EPPN10_L
EPPN11_H
EPPN11_L
EPPN12_H
EPPN12_L
EPPN13_H
EPPN13_L
EPPN14_H
EPPN14_L
EPPN15_H
EPPN2_H
EPPN2_L
EPPN20_T
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
EPPN5_H
EPPN5_L
EPPN6_H
EPPN6_L
EPPN7_H
S_Height_cm
0
500
1000
1500
février 18
mars 08
mars 27
Time


EPPN2020_M3P - S_Height_cm - raw data
EPPN7_L
EPPN8_H
EPPN8_L
EPPN9_H
EPPN9_L
S_Height_cm
0
500
1000
1500
février 18
mars 08
mars 27

Time

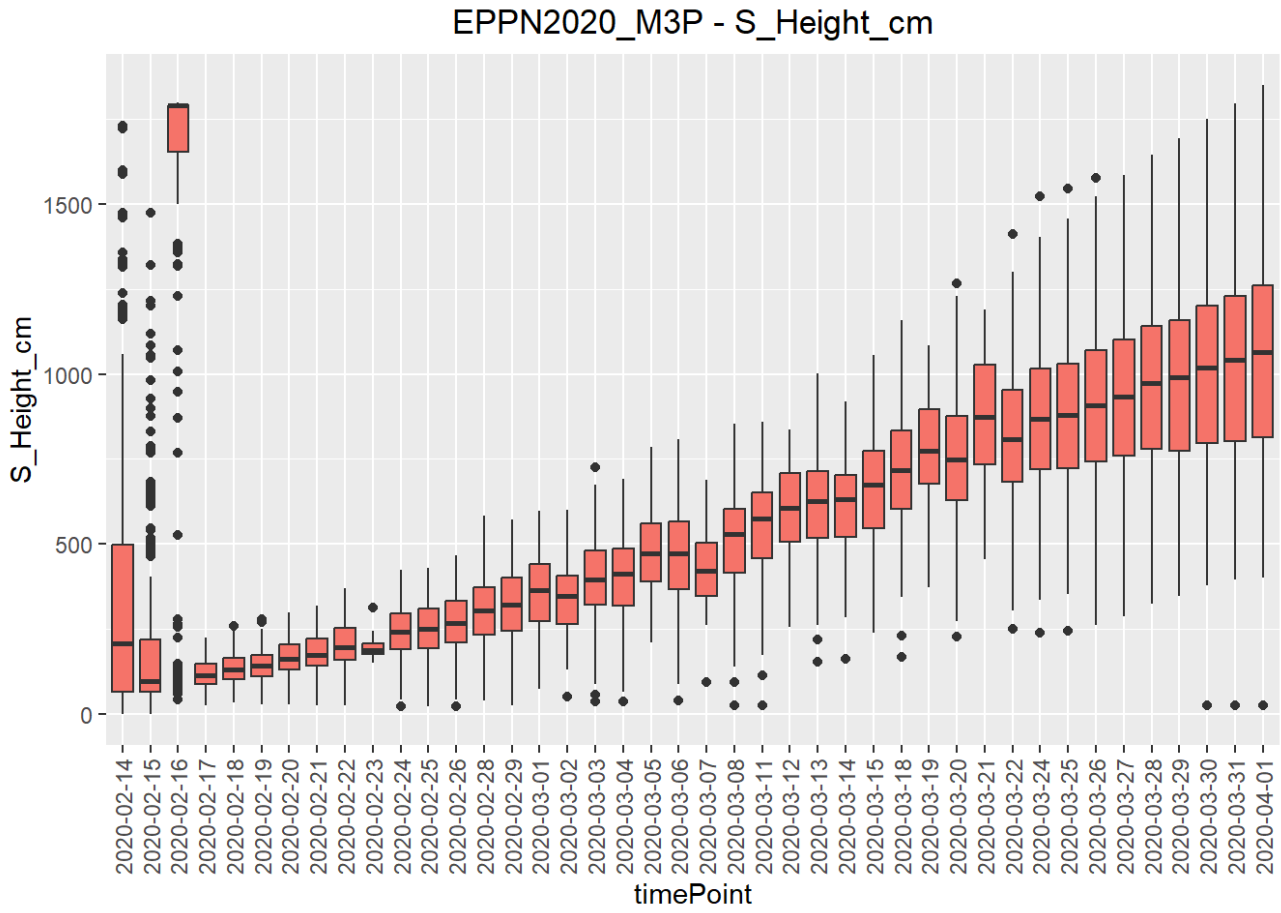


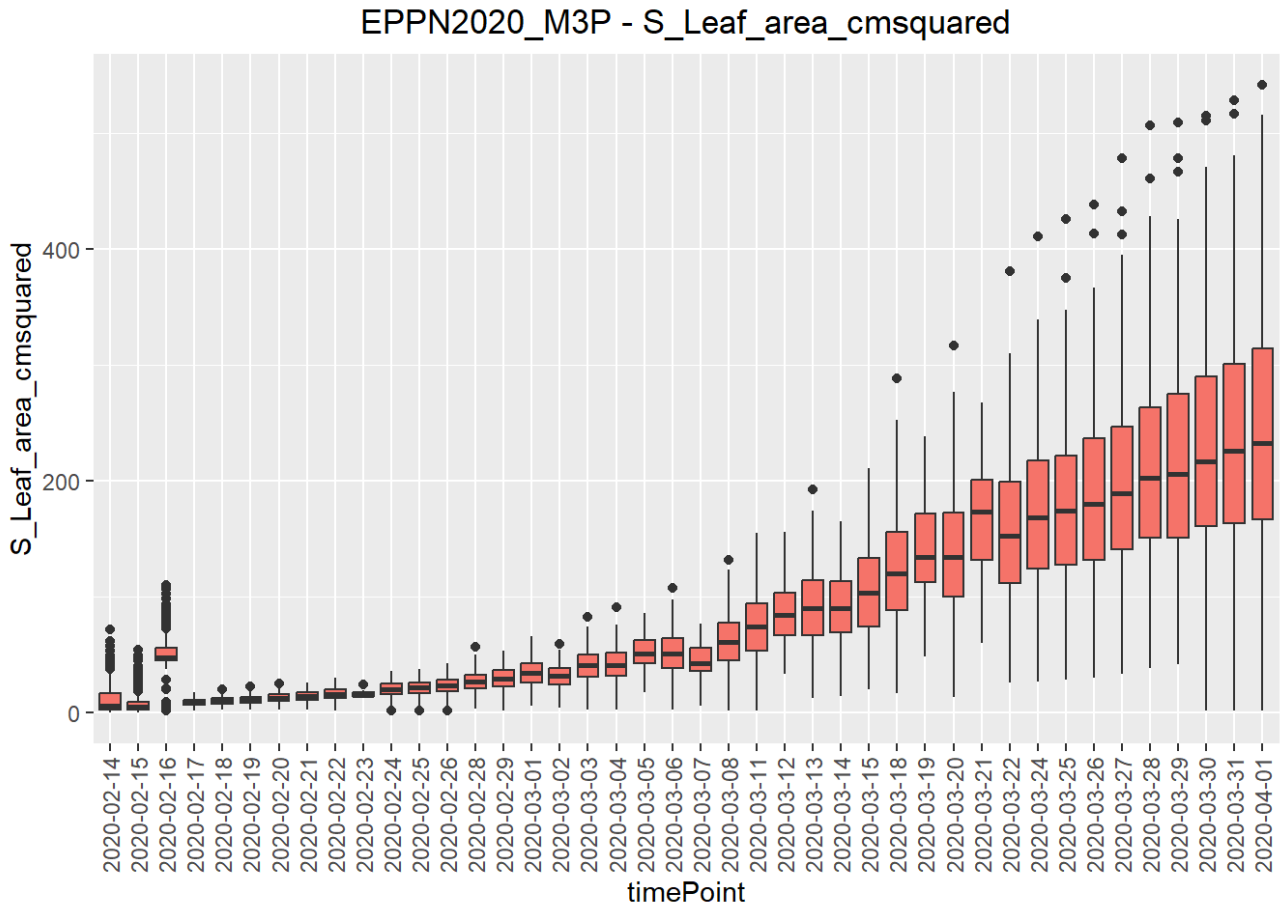


Time

Check the boxplots of raw data per time point

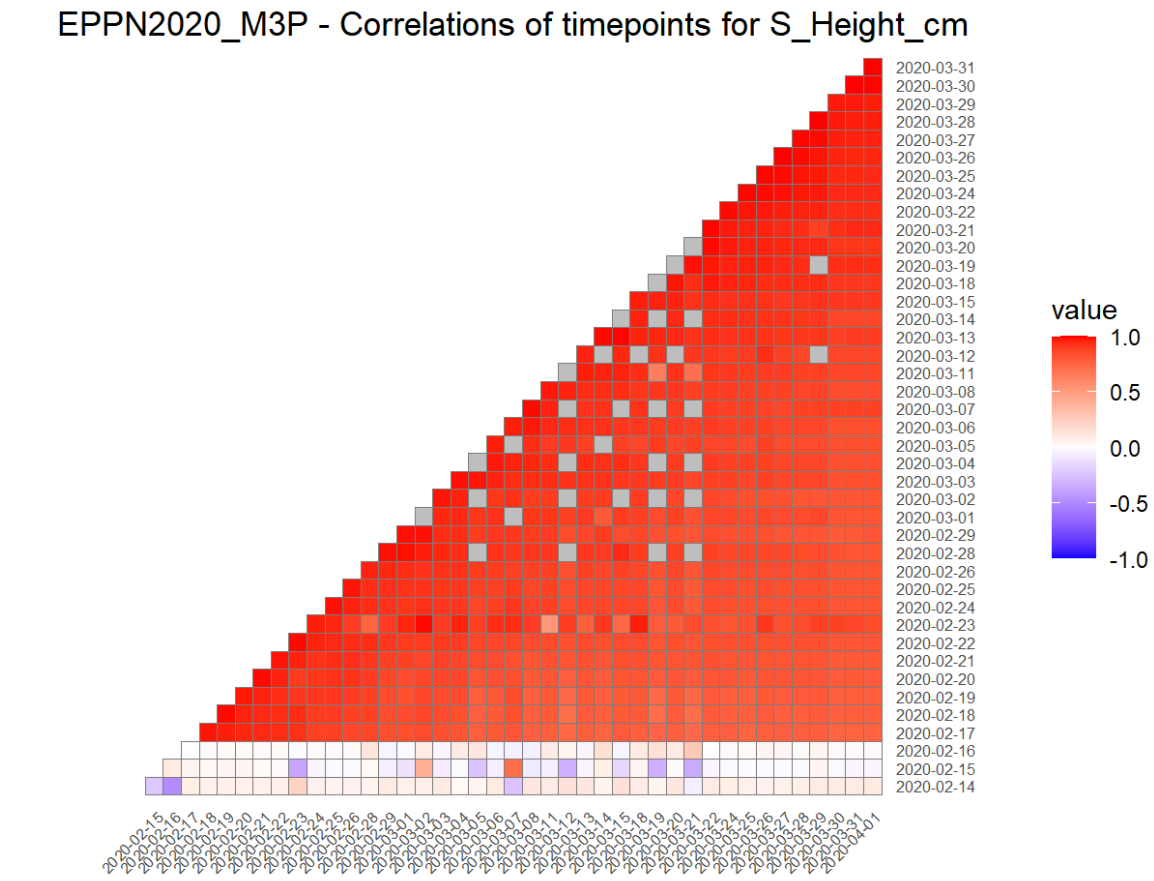
```
for (trait_name in traits) {
  plot(timePoint_S,
     plotType = "box",
     traits = trait_name)
}
```

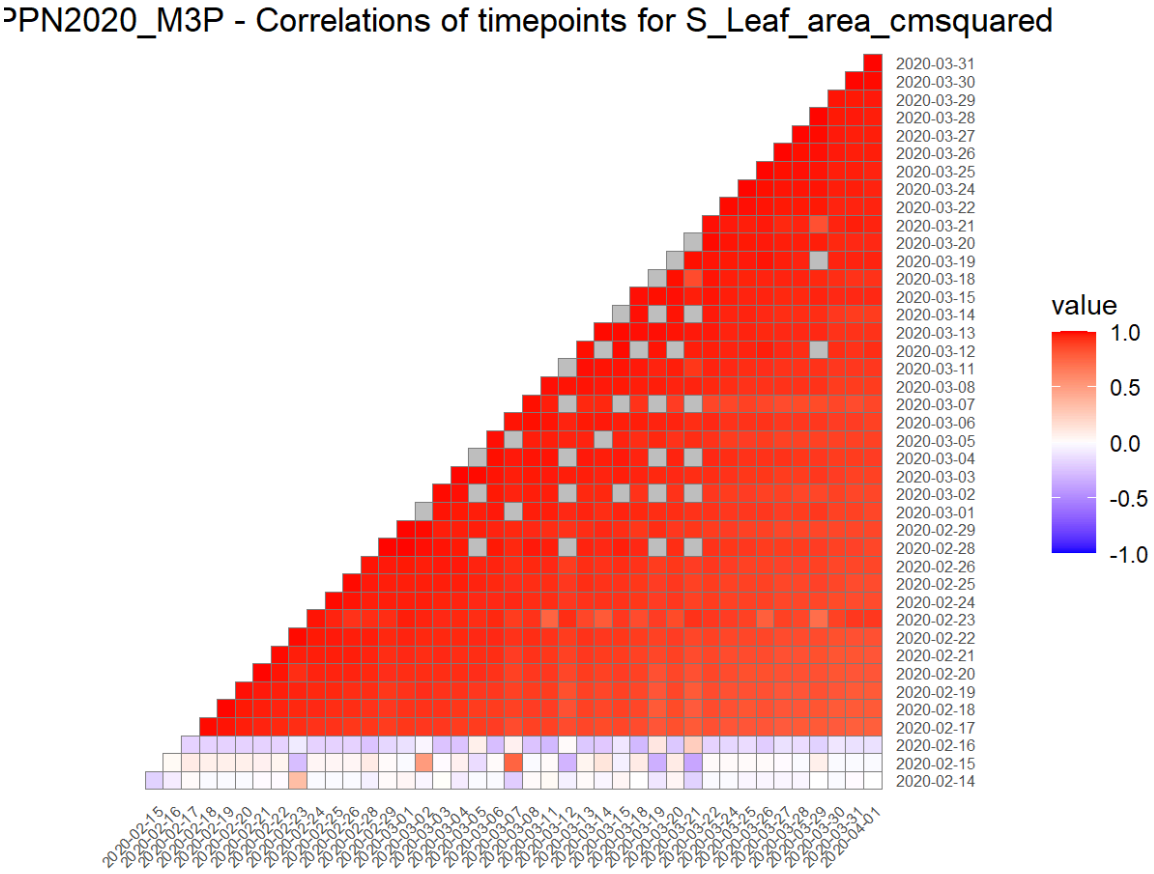




Check the correlation plots of raw data per time point

```
for (trait_name in traits) {
  plot(timePoint_S,
     plotType = "cor",
     traits = trait_name)
}
```



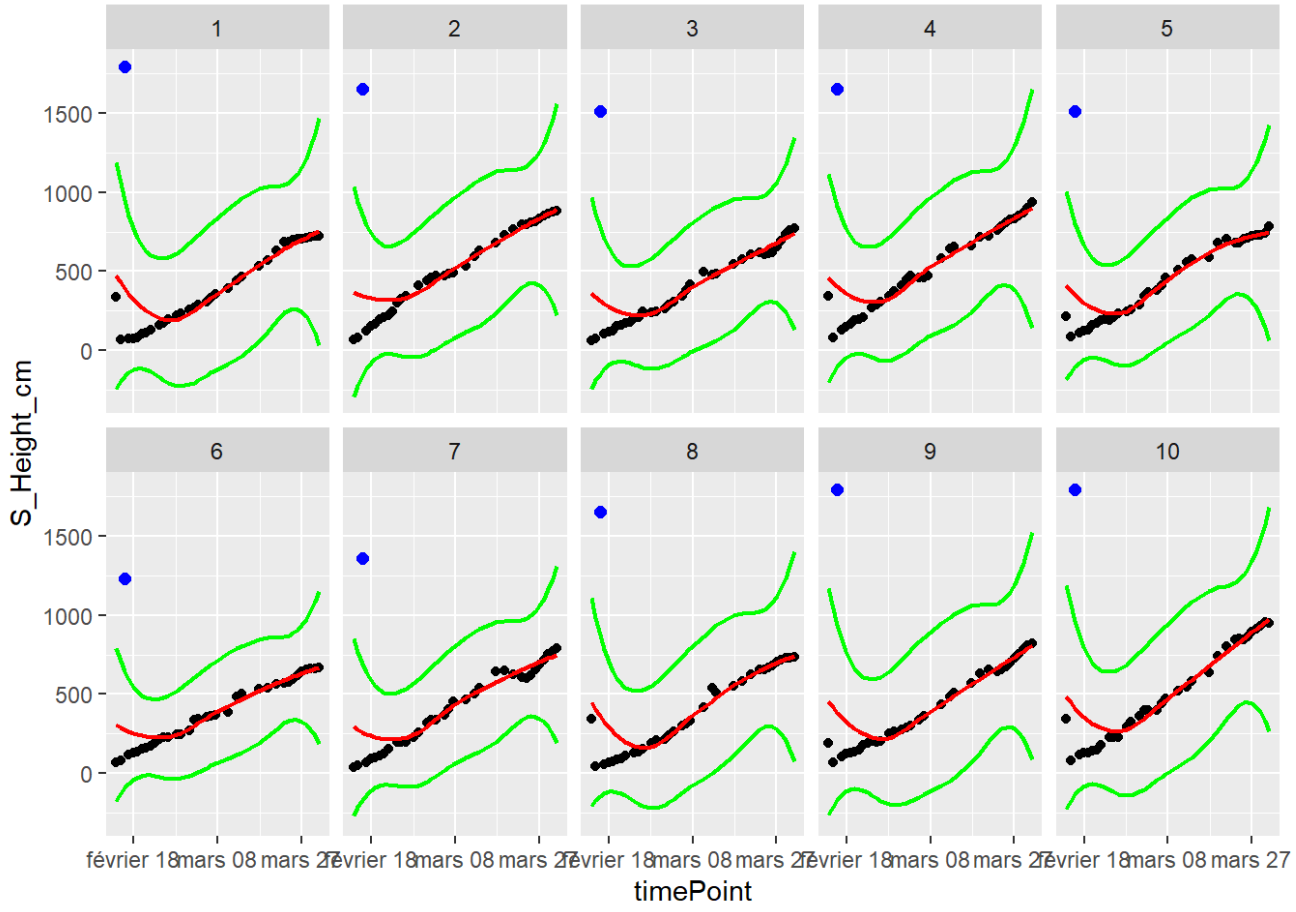


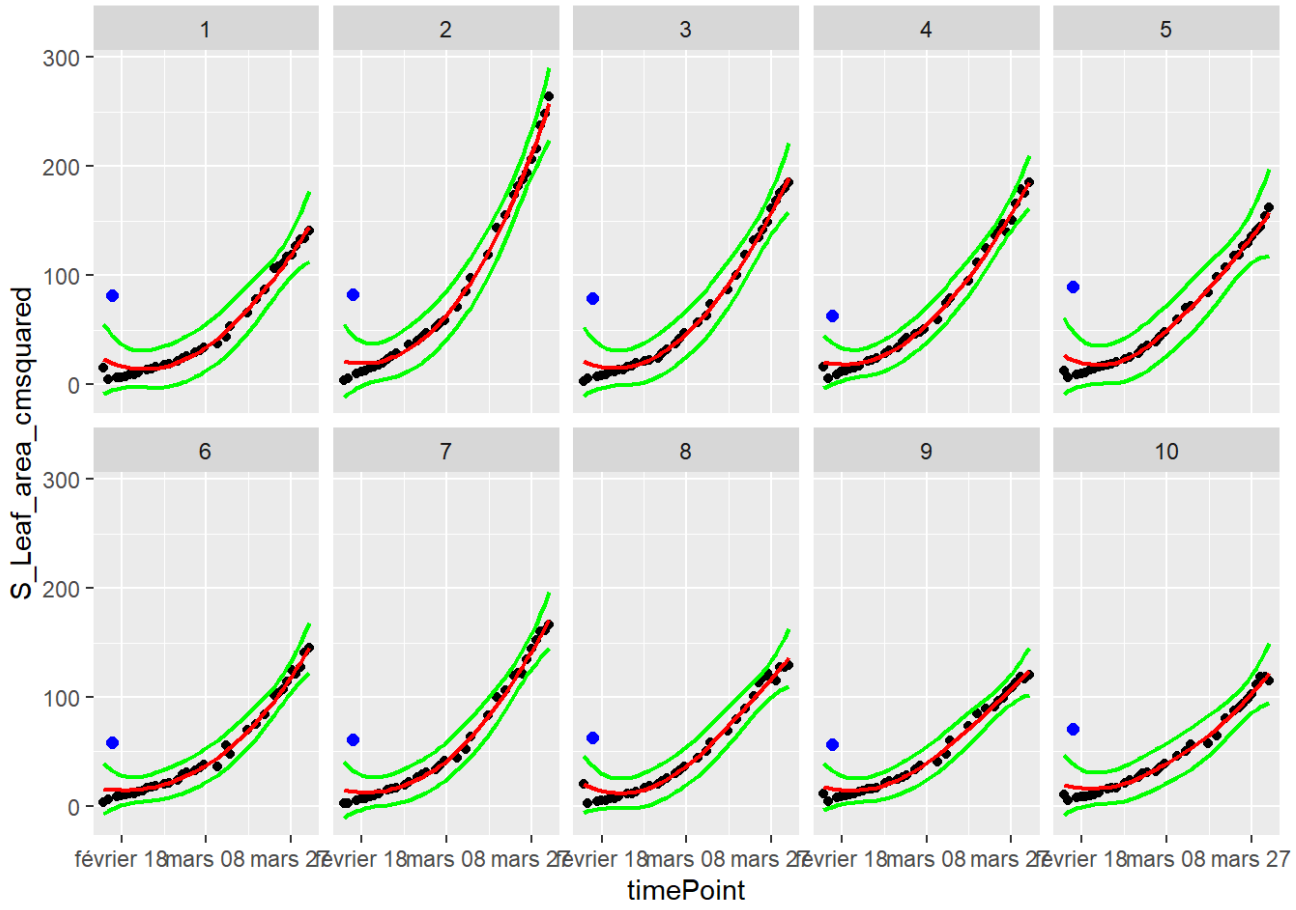
1. Detection of outliers for single observations

Using the SingleOut detect and single functions. We select a subset of plants to adjust the settings for the confIntSize and nnLocfit.

```
plantSel<- c(1,2,3,4,5,6,7,8,9,10)

ci <- 5 # confidence interval
nn <- 0.8 # nearest neighbor
ce <- FALSE
```

```
for (trait_name in traits) {
  variable_name <- paste0("Single_test_", trait_name)

  single_test <- detectSingleOut(
    TP = timePoint_S,
    trait = trait_name,
    plotIds = plantSel,
    confIntSize = ci,
    nnLocfit = nn,
    checkEdges = TRUE # check for outlier values in start and end of experiment
  )

  assign(variable_name, single_test)

  plot(single_test, outOnly = FALSE)
}
```





We can then run on all plants of the data set.

```
for (trait_name in traits) {
  single_test_object_name <- paste0("Single_test_", trait_name)
  Single_test <- get(single_test_object_name)
    if (any(Single_test$outlier == 1)) {
    outliers_count <- with(Single_test[Single_test$outlier == 1,], table(timePoint))
    print(trait_name)
    print(outliers_count)

    Single_outliers <- removeSingleOut(timePoint_S, Single_test)
    assign(paste0("Single_outliers_", trait_name), Single_outliers)

    readr::write_tsv(Single_test, sprintf("%s/single_outliers_%s.tsv", datadir, trait_n
ame))
  } else {
    cat("No outlier for", trait_name, "\n")
  }
}
```

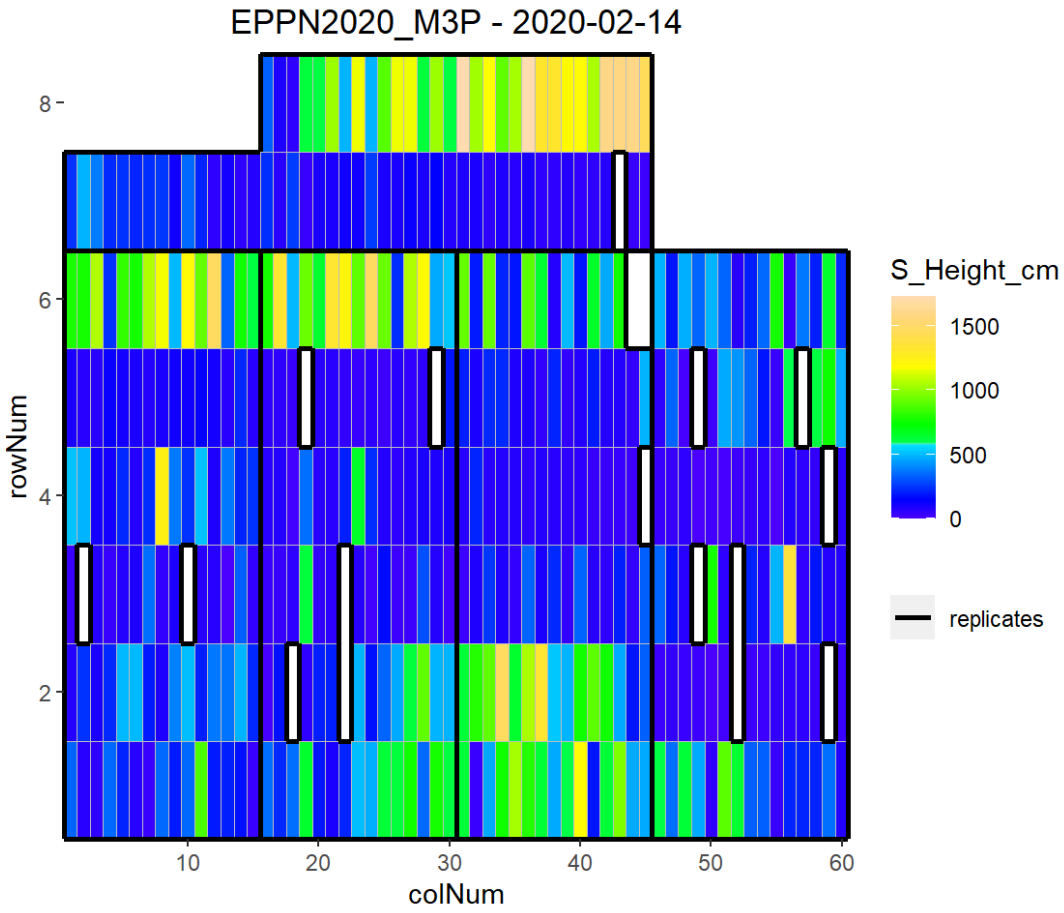
```
## [1] "S_Height_cm"
## timePoint
## 2020-02-16
##         10
## [1] "S_Leaf_area_cmsquared"
## timePoint
## 2020-02-16
##         10
```

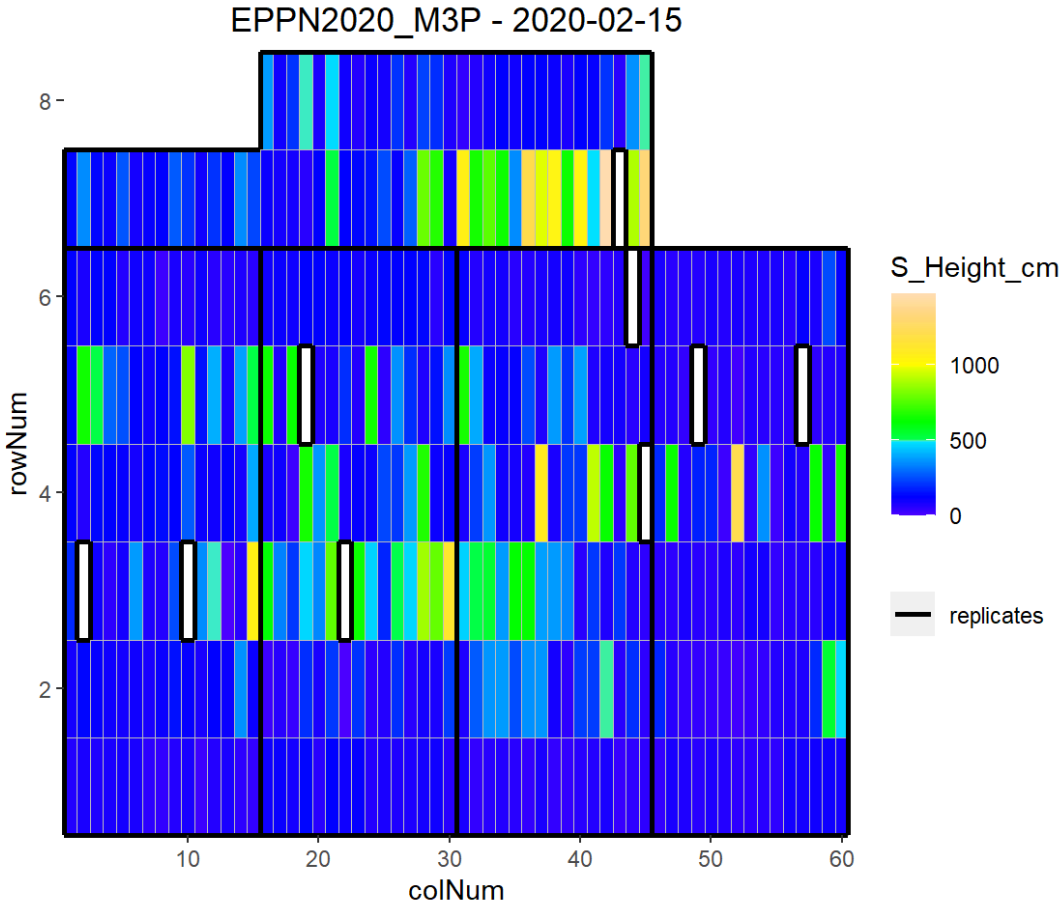
Data visualisation after single observations outliers removal

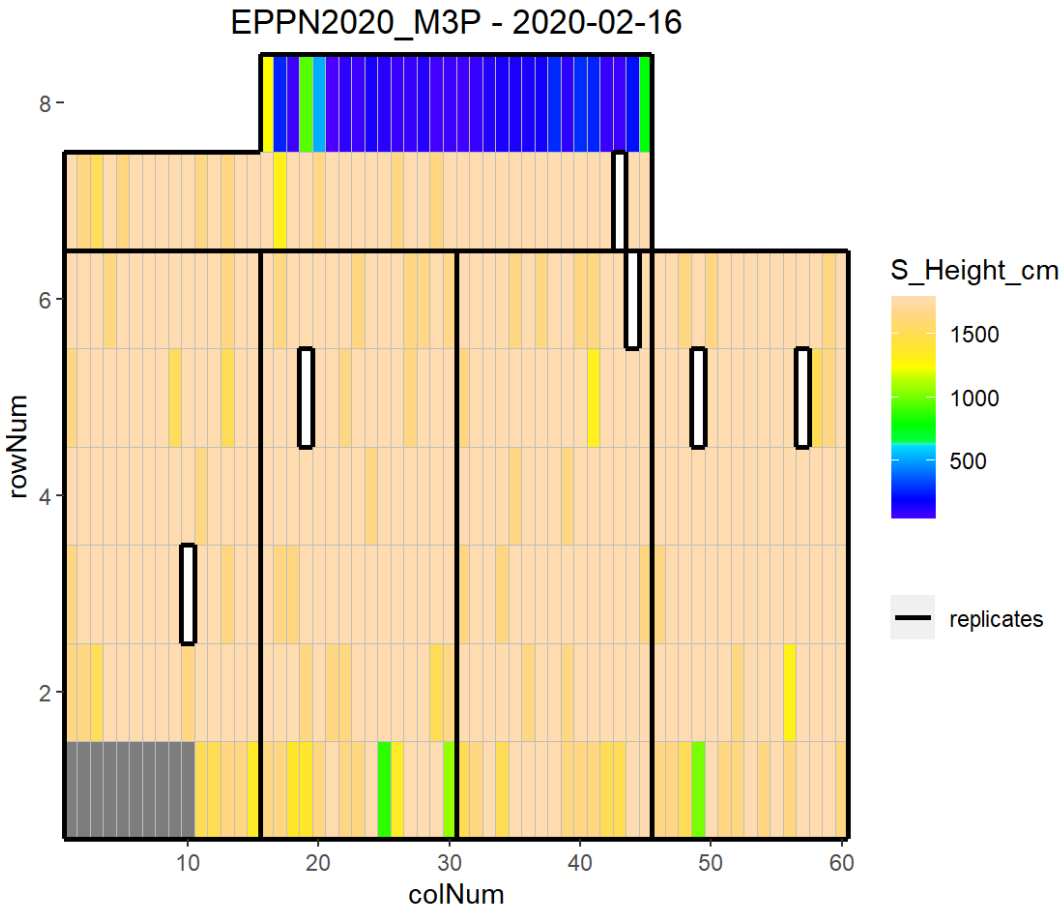
Heatmap of data

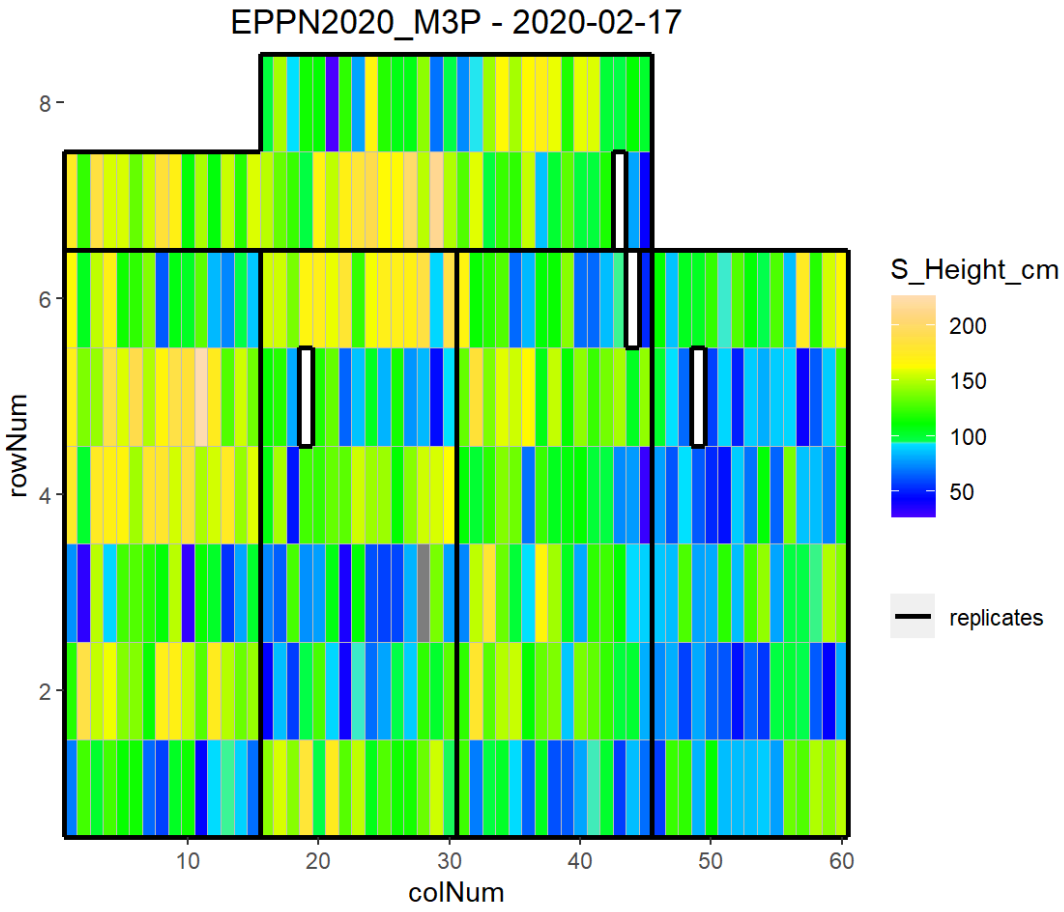
Check the heatmap of the data with outliers detection at all the time points.

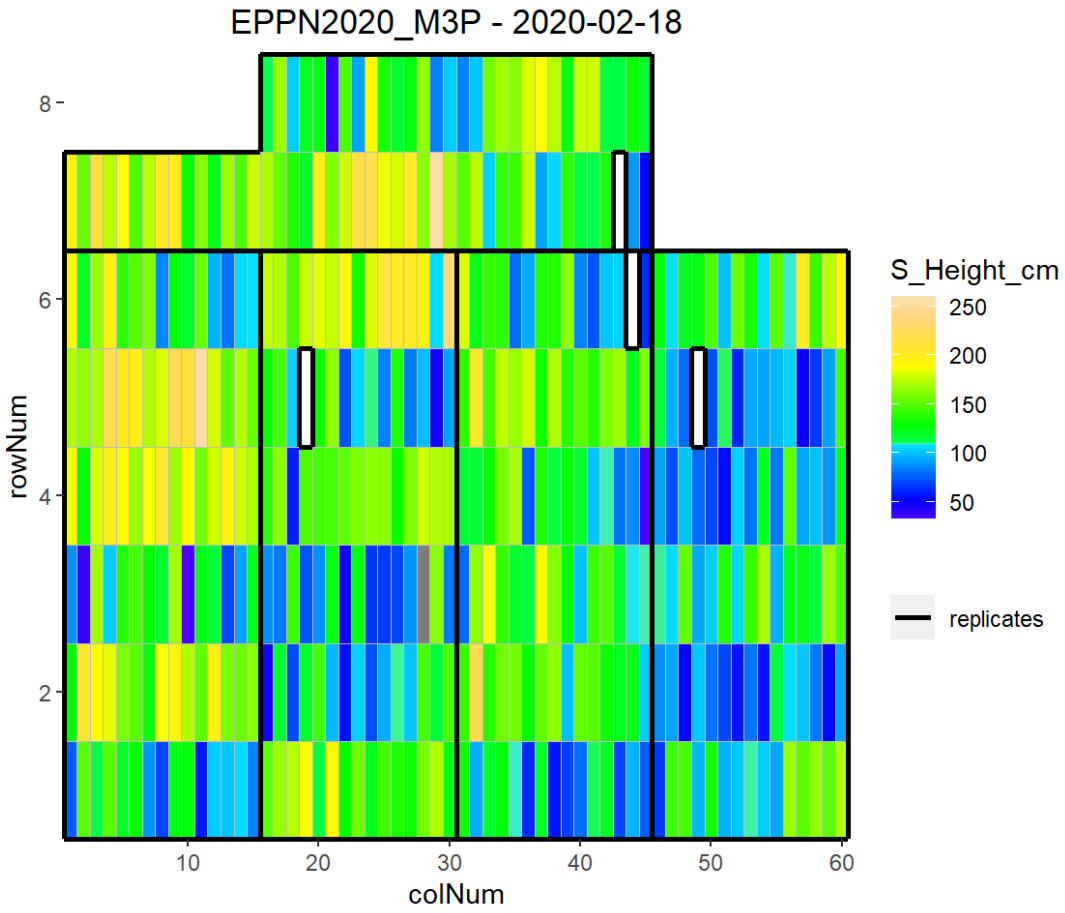
```
for (trait_name in traits) {
  single_outliers_name <- paste0("Single_outliers_", trait_name)

  if (exists(single_outliers_name)) {
    Single_outliers <- get(single_outliers_name)

    for (tp in 1:length(num_timepoints$timeNumber)) {
      plot(Single_outliers,
           plotType = "layout",
           timePoints = tp,
           traits = trait_name)
    }
  } else {
    cat("No object Single_outliers found for trait", trait_name, "\n")
  }
}
```

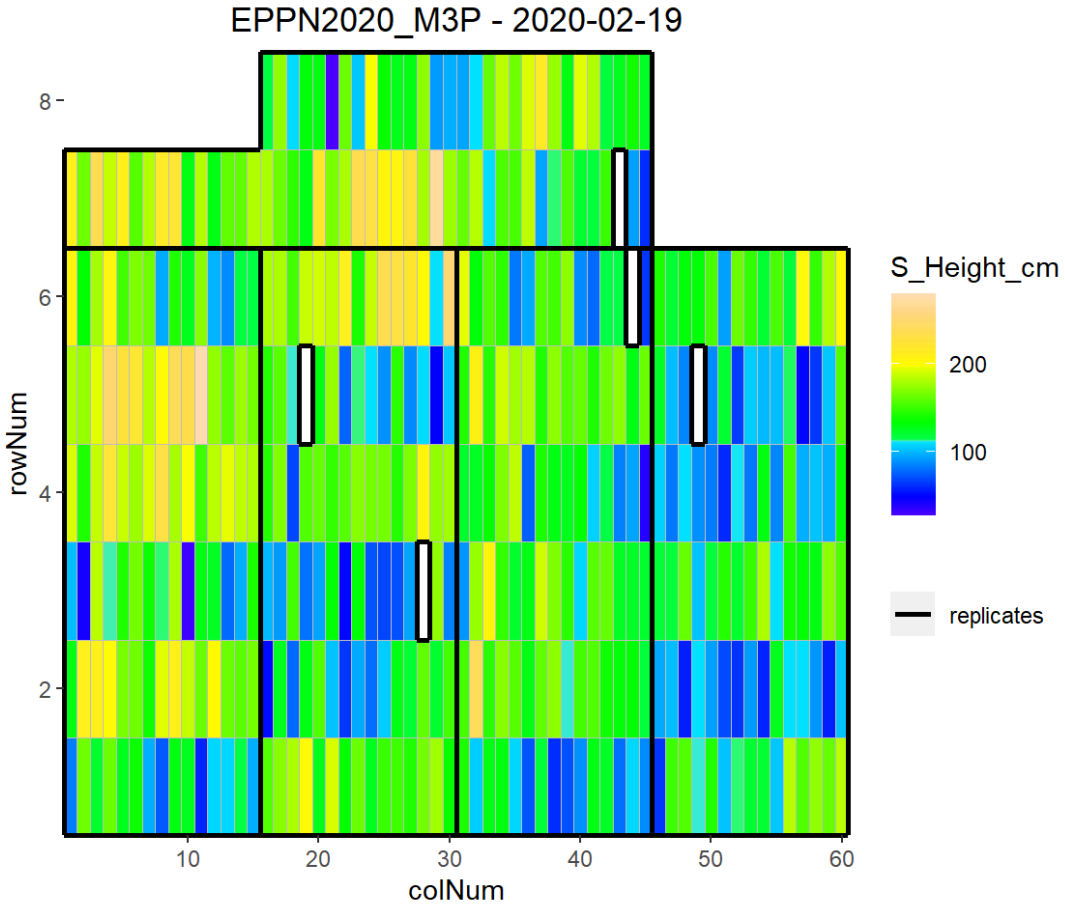

EPPN2020_M3P - 2020-02-14
rowNum
colNum
S_Height_cm
1500
1000
500
0
replicates

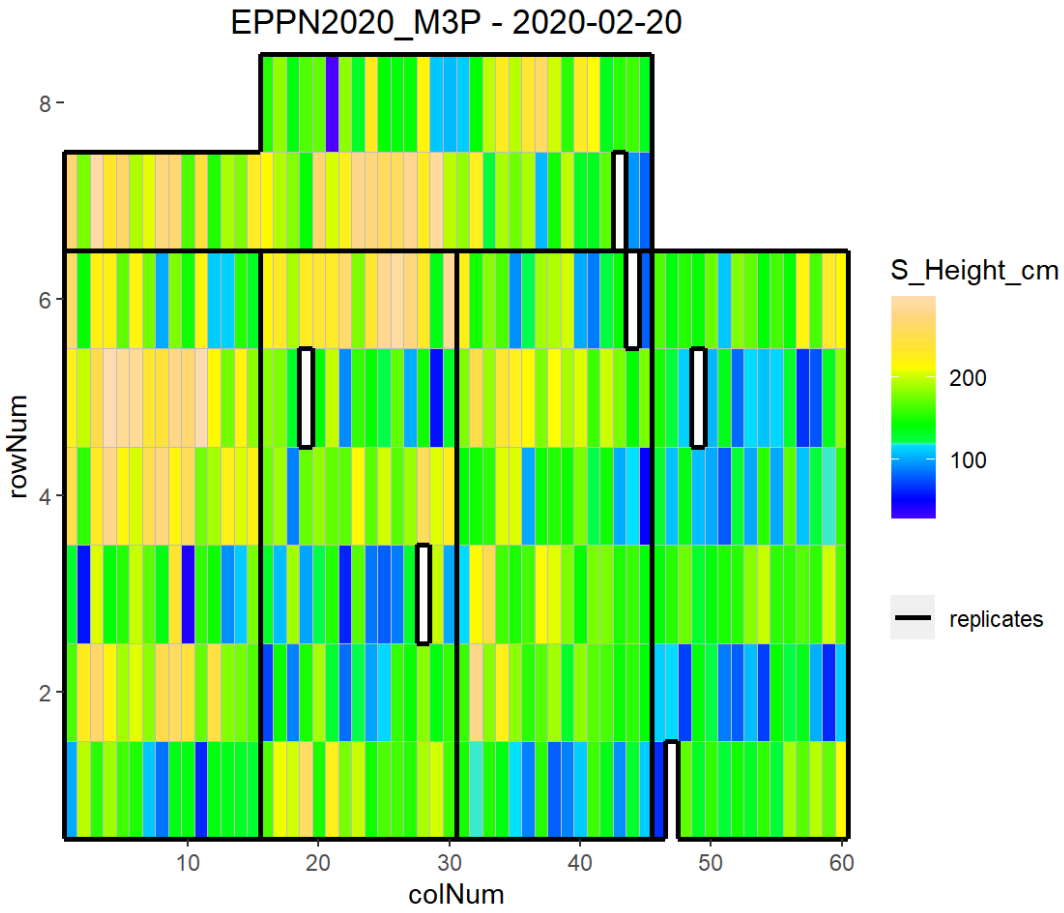

EPPN2020_M3P - 2020-02-15
rowNum
colNum
S_Height_cm
1000
500
0
replicates

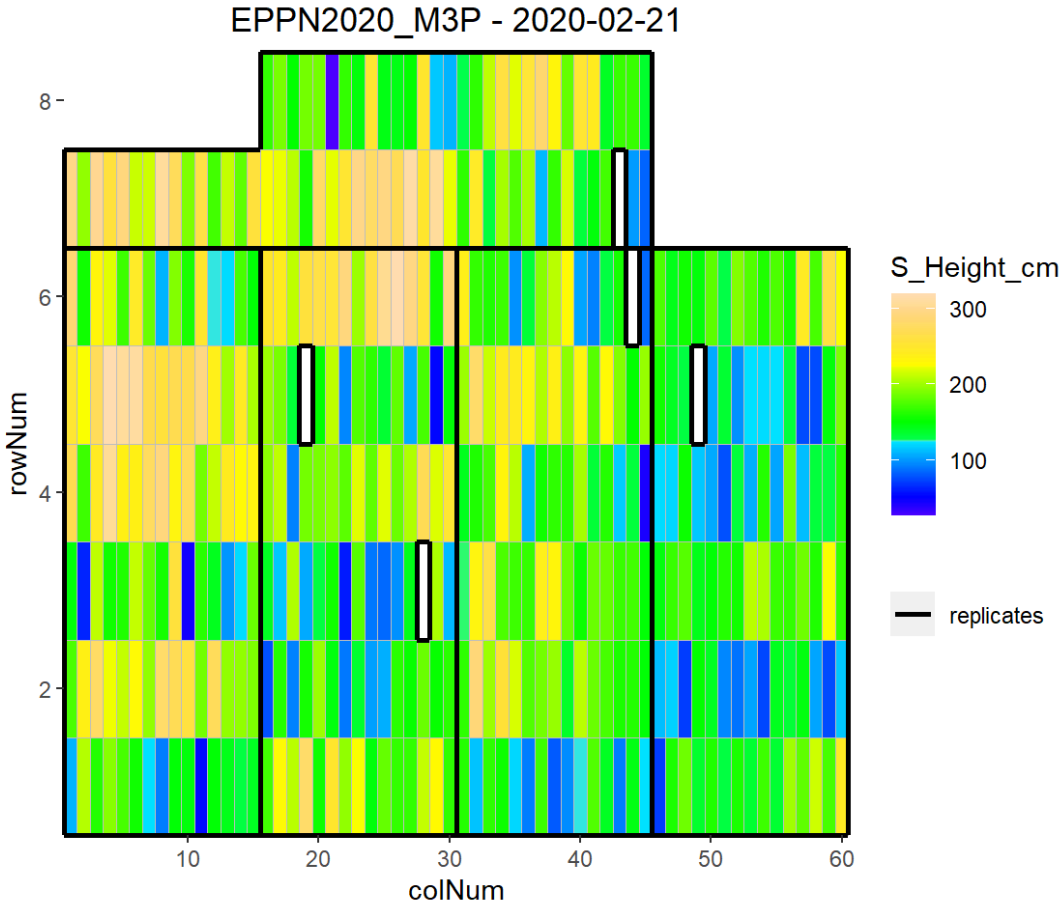

EPPN2020_M3P - 2020-02-16
rowNum
colNum
S_Height_cm
1500
1000
500
replicates

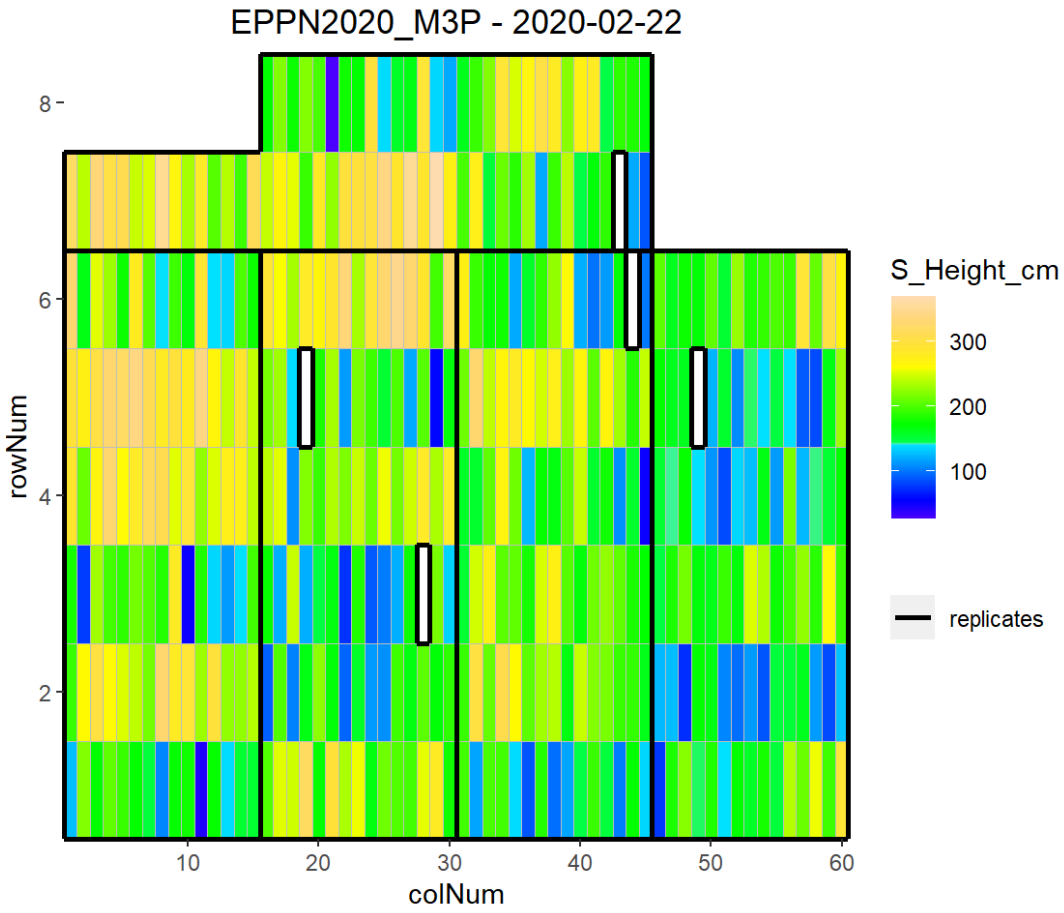

EPPN2020_M3P - 2020-02-17
rowNum
colNum
S_Height_cm
200
150
100
50
replicates

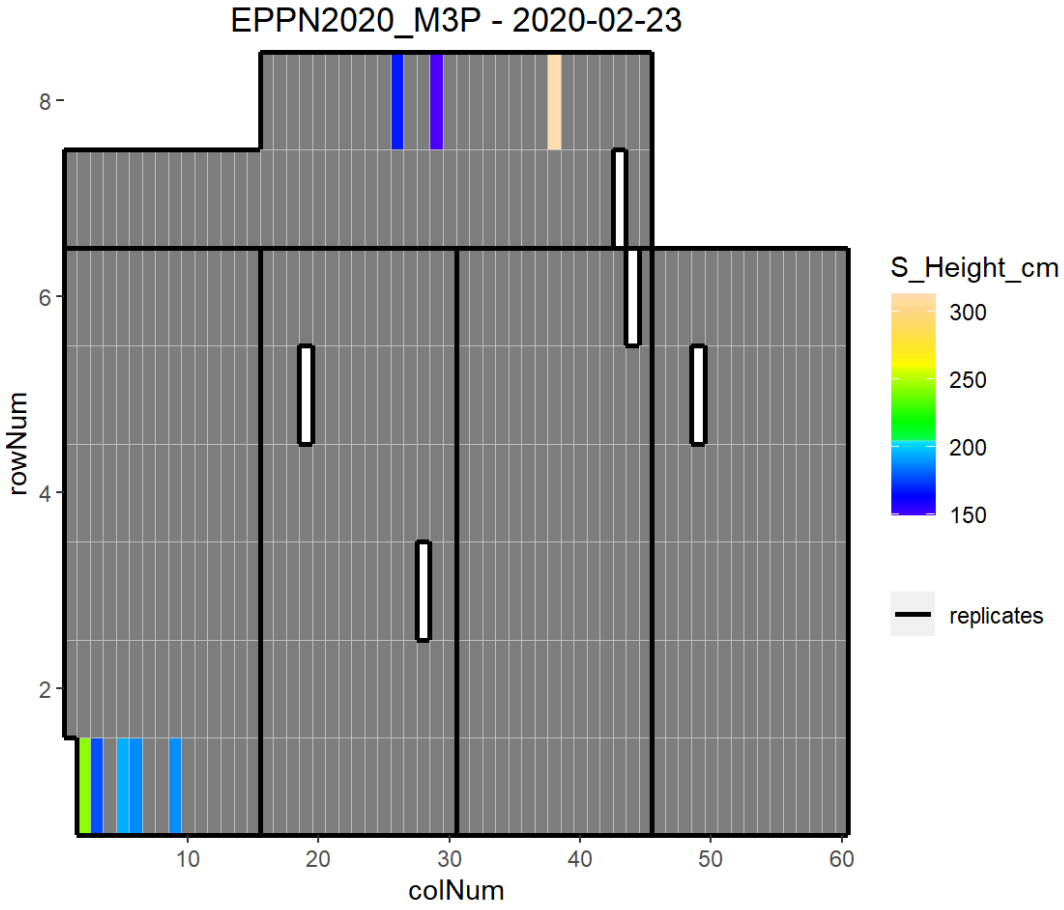

EPPN2020_M3P - 2020-02-18
rowNum
colNum
S_Height_cm
250
200
150
100
50
replicates

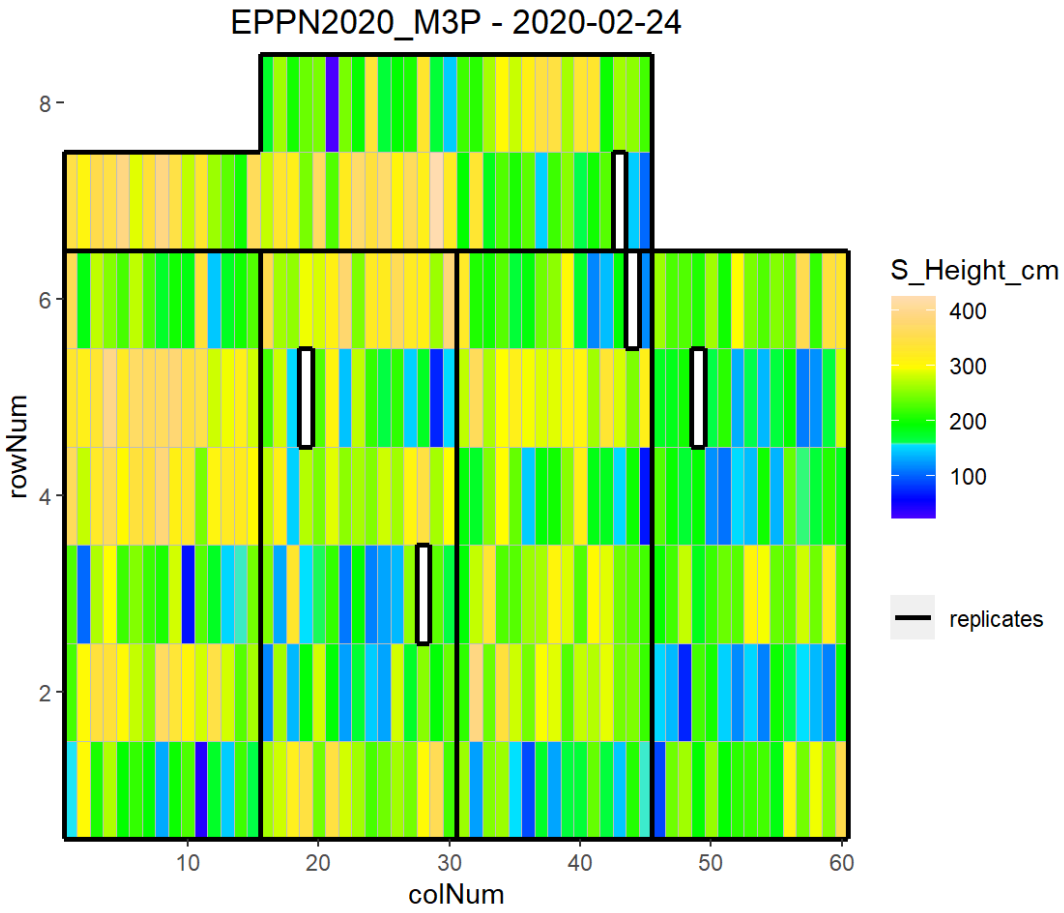

EPPN2020_M3P - 2020-02-19
rowNum
colNum
S_Height_cm
200
100
replicates

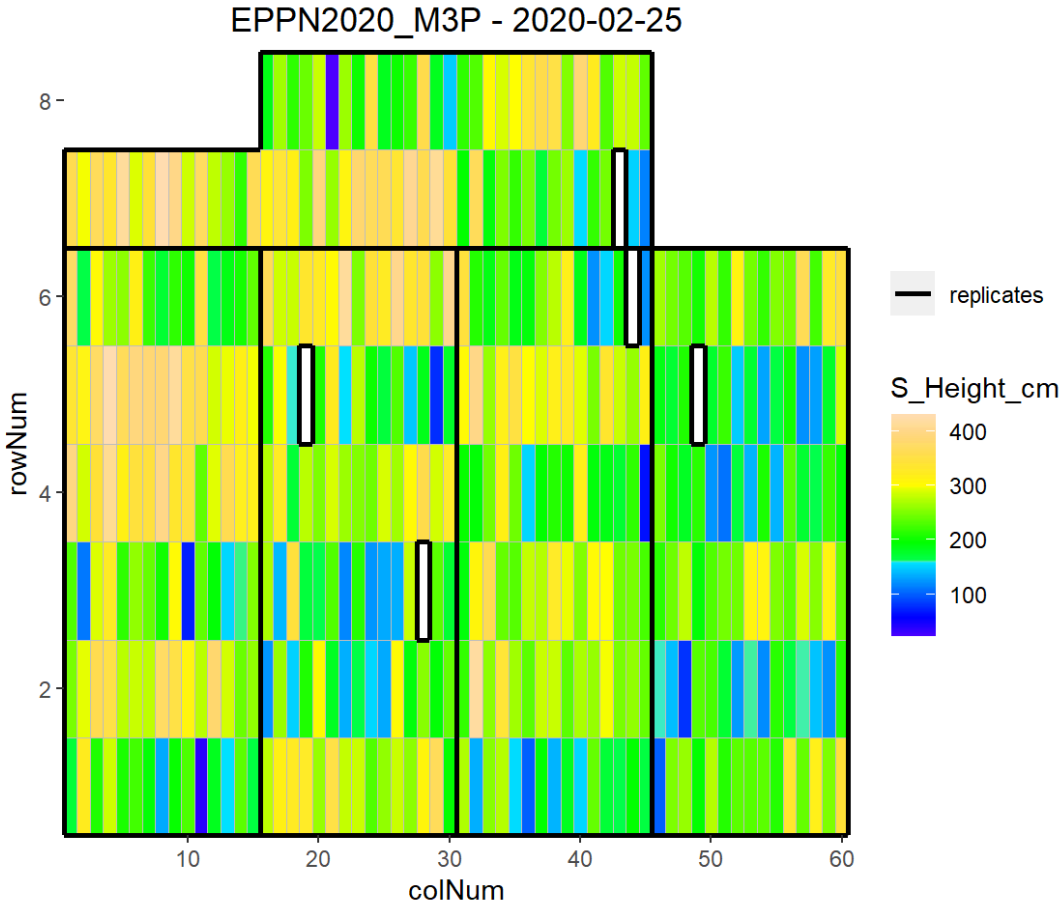

EPPN2020_M3P - 2020-02-20
rowNum
colNum
S_Height_cm
200
100
replicates

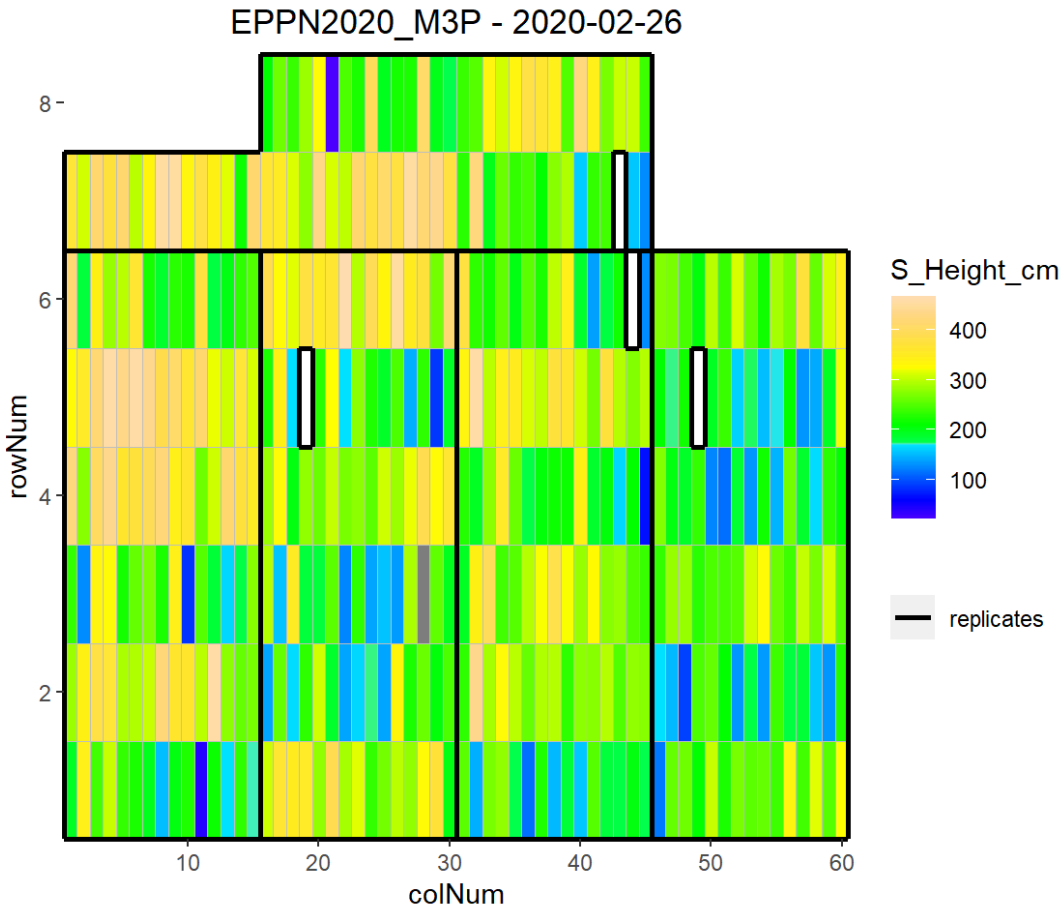

EPPN2020_M3P - 2020-02-21
rowNum
colNum
S_Height_cm
300
200
100
replicates

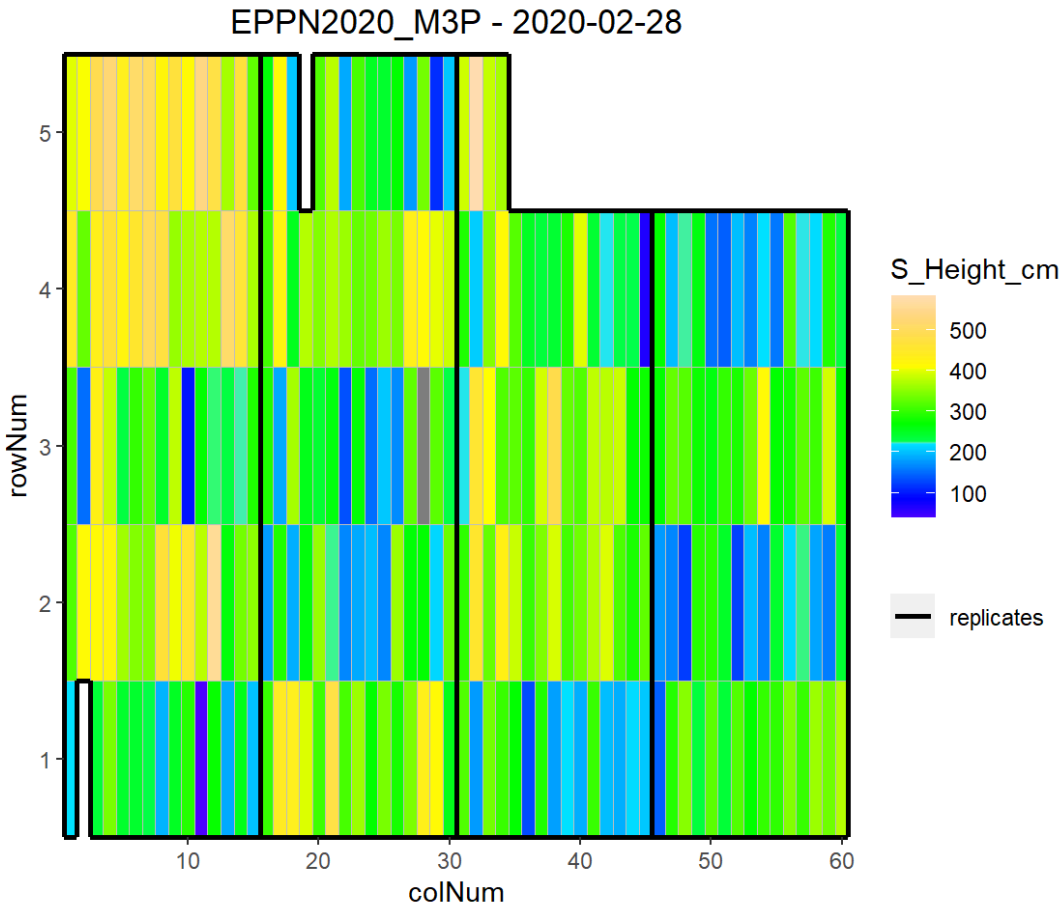

EPPN2020_M3P - 2020-02-22
rowNum
colNum
S_Height_cm
300
200
100
replicates

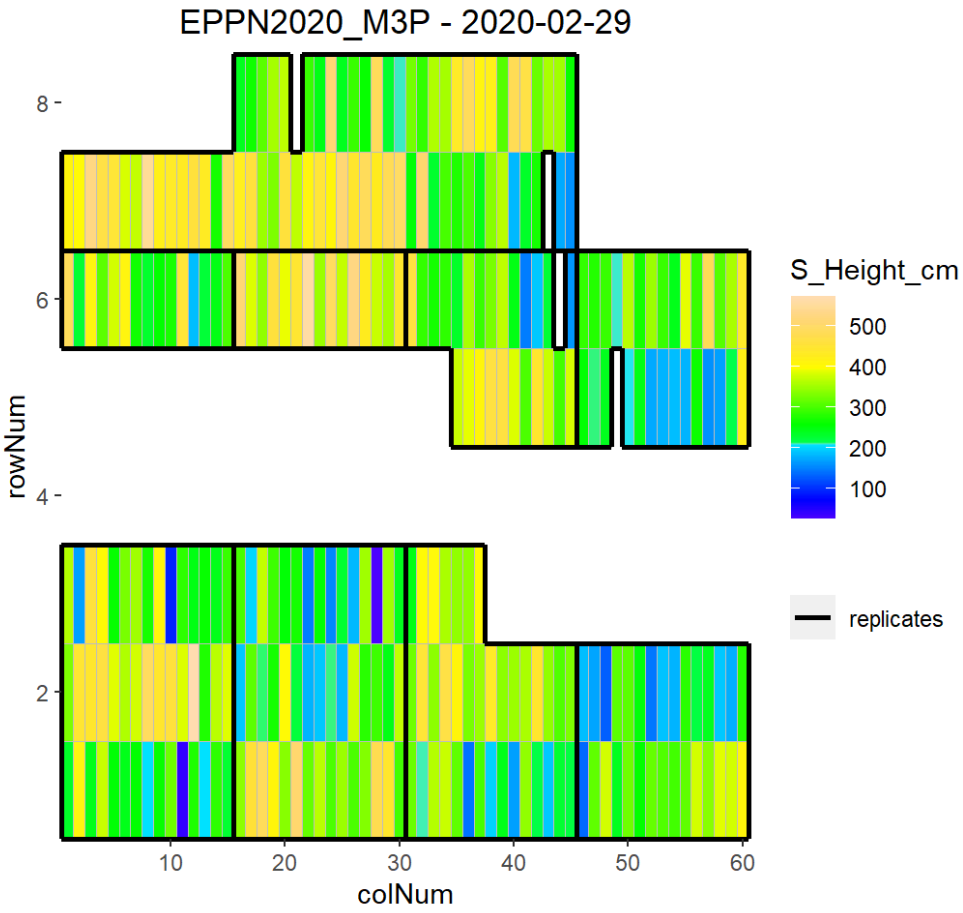

EPPN2020_M3P - 2020-02-23
rowNum
colNum
S_Height_cm
300
250
200
150
replicates

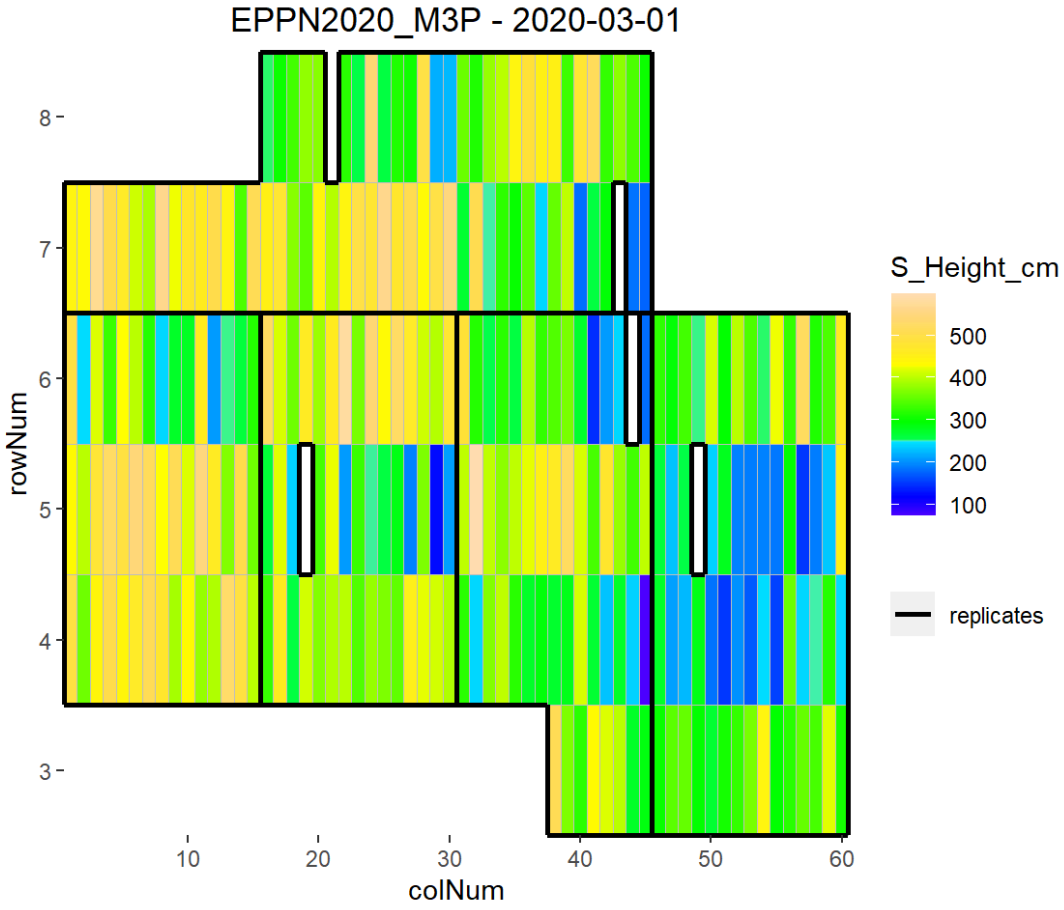

EPPN2020_M3P - 2020-02-24
rowNum
colNum
S_Height_cm
400
300
200
100
replicates

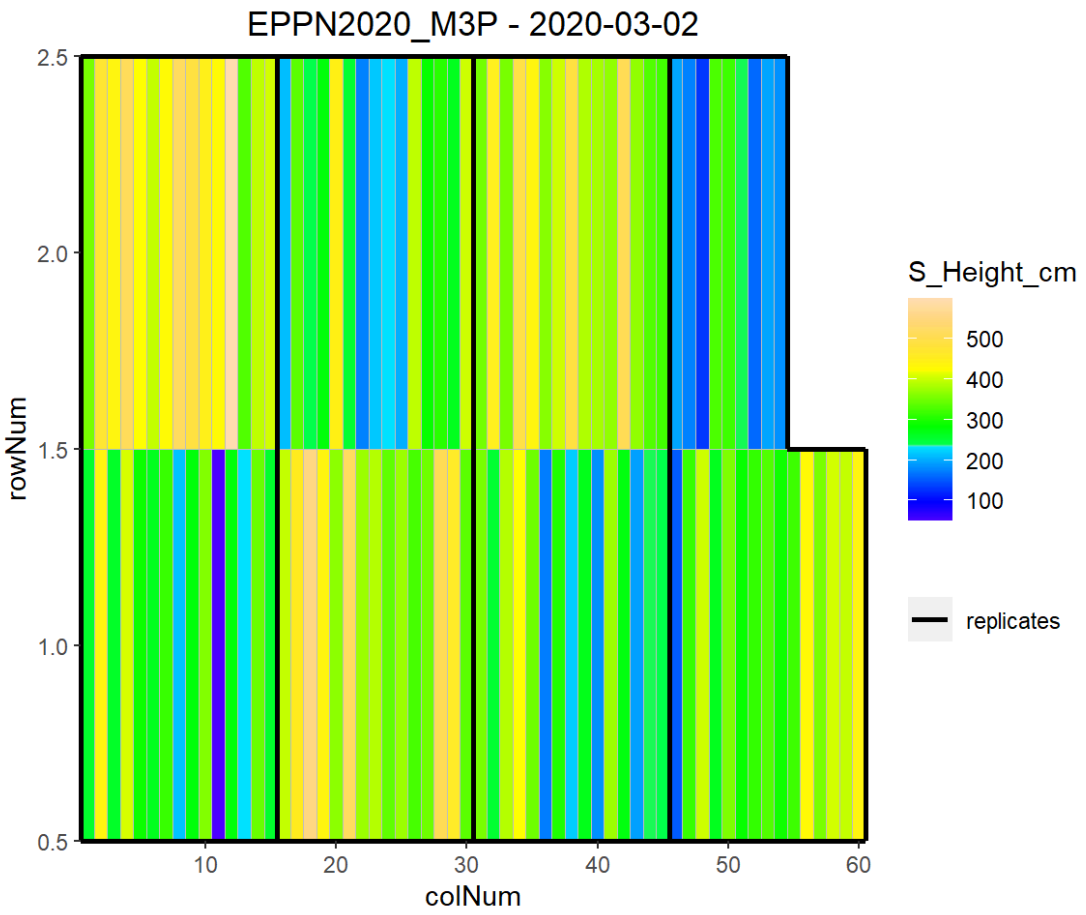

EPPN2020_M3P - 2020-02-25
rowNum
colNum
replicates
S_Height_cm
400
300
200
100

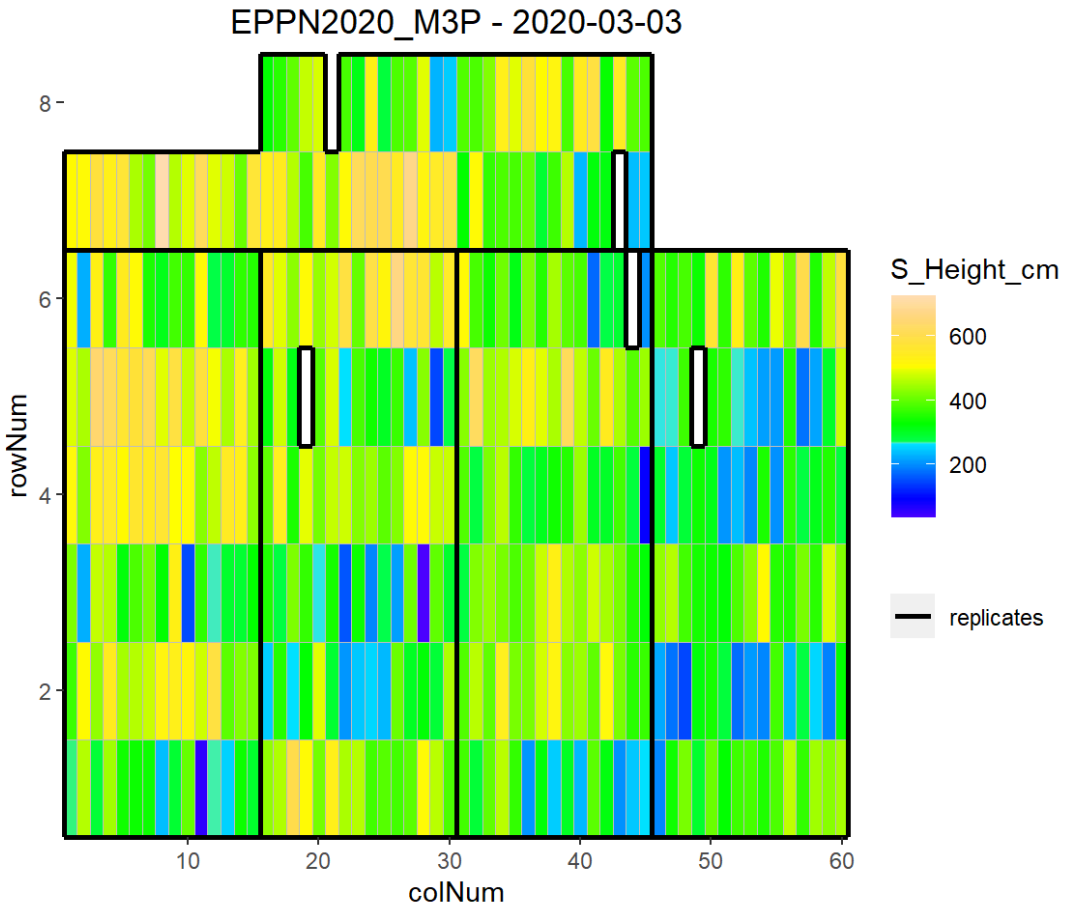

EPPN2020_M3P - 2020-02-26
rowNum
colNum
S_Height_cm
400
300
200
100
replicates

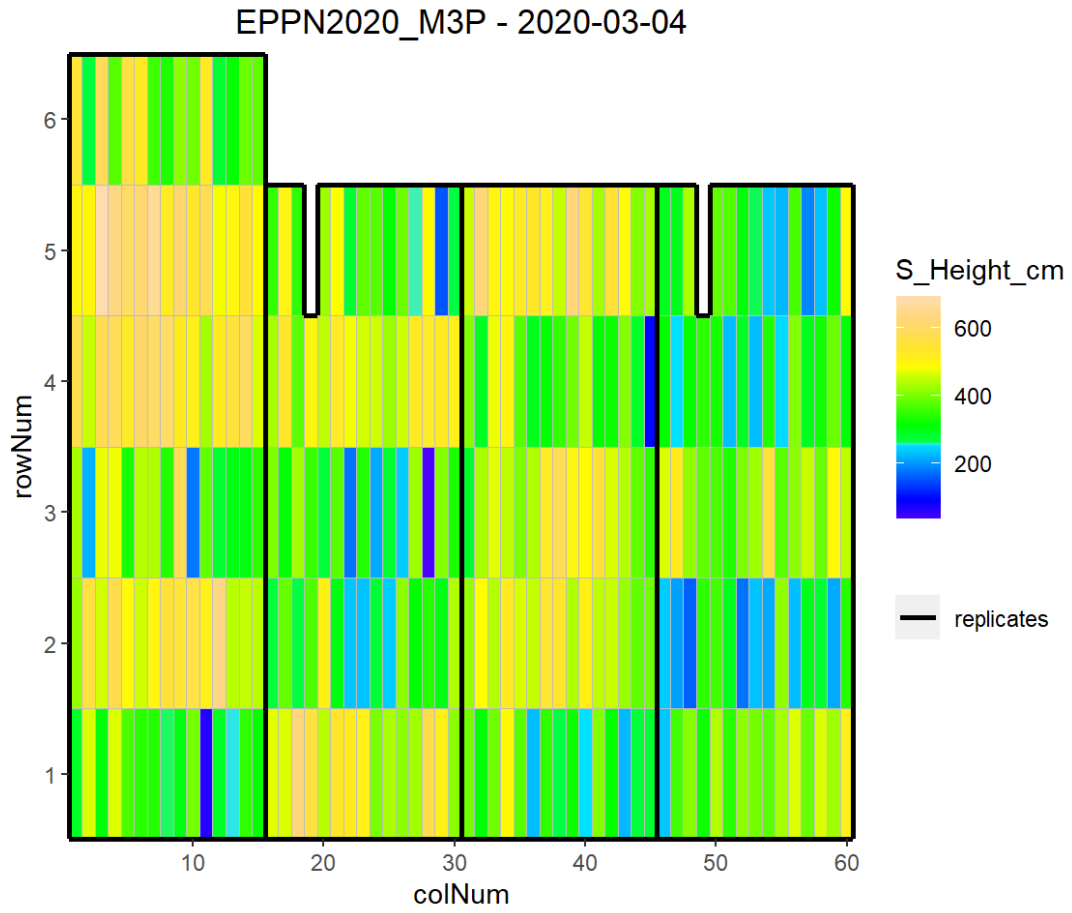

EPPN2020_M3P - 2020-02-28
rowNum
colNum
S_Height_cm
500
400
300
200
100
replicates

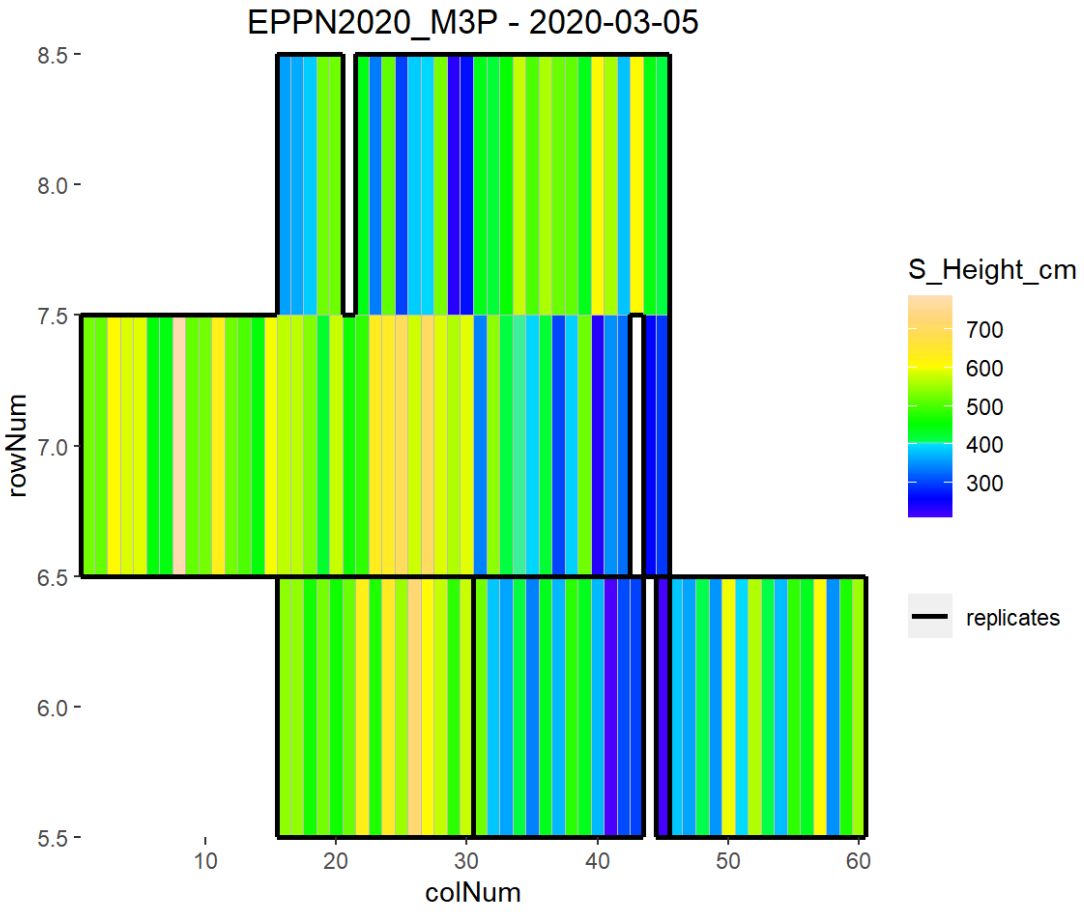

EPPN2020_M3P - 2020-02-29
rowNum
colNum
S_Height_cm
500
400
300
200
100
replicates

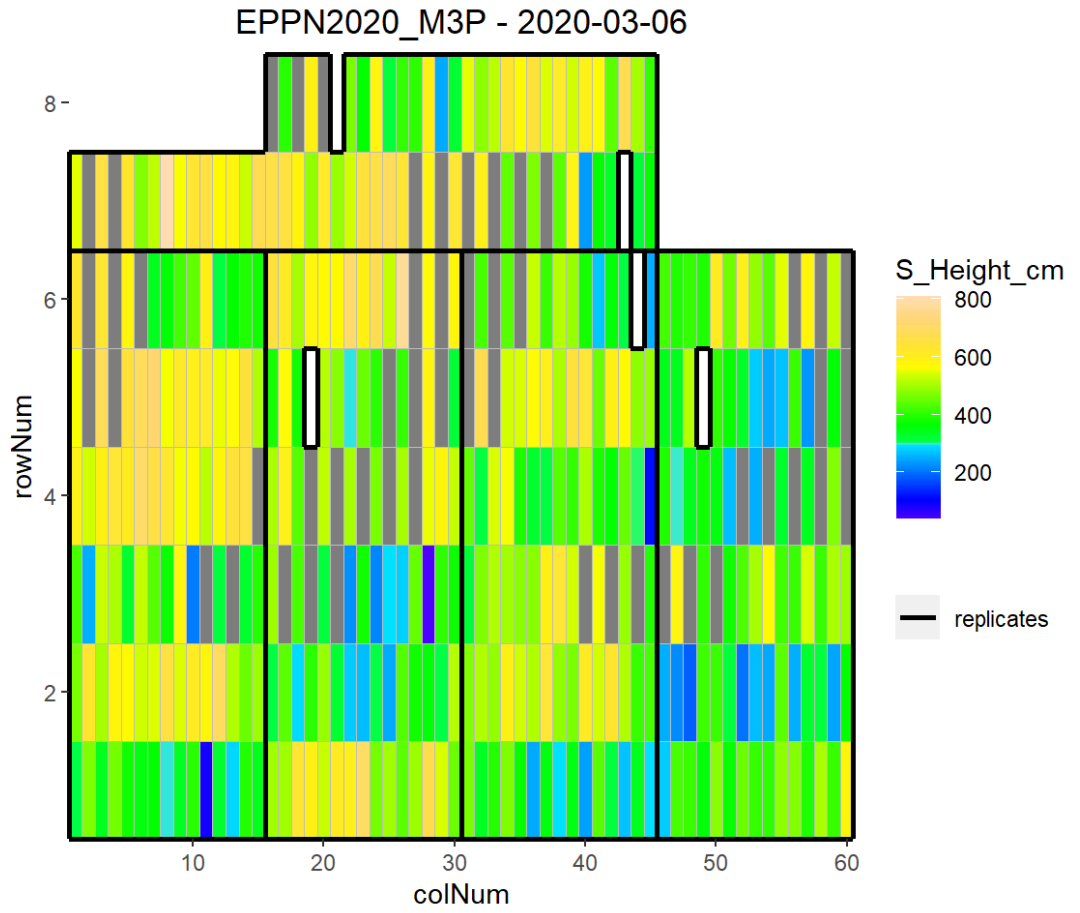

EPPN2020_M3P - 2020-03-01
rowNum
colNum
S_Height_cm
500
400
300
200
100
replicates

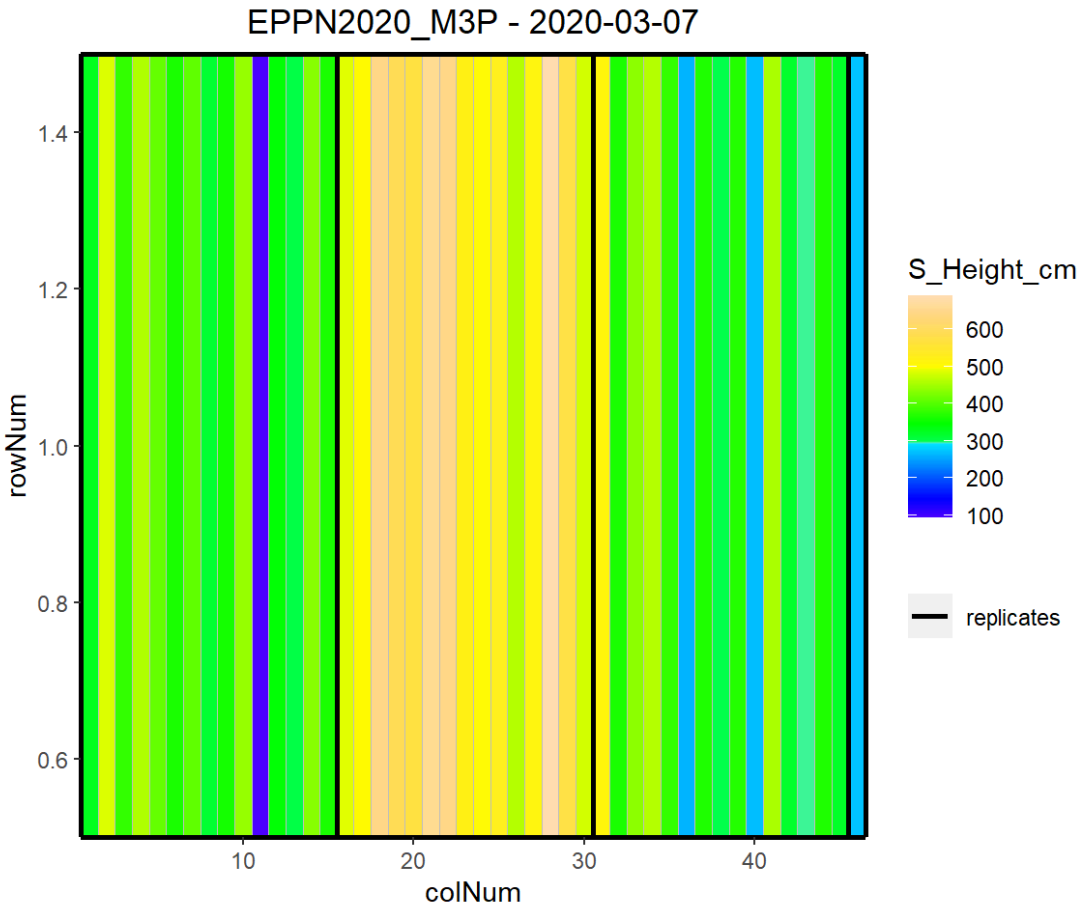

EPPN2020_M3P - 2020-03-02
rowNum
colNum
S_Height_cm
500
400
300
200
100
replicates

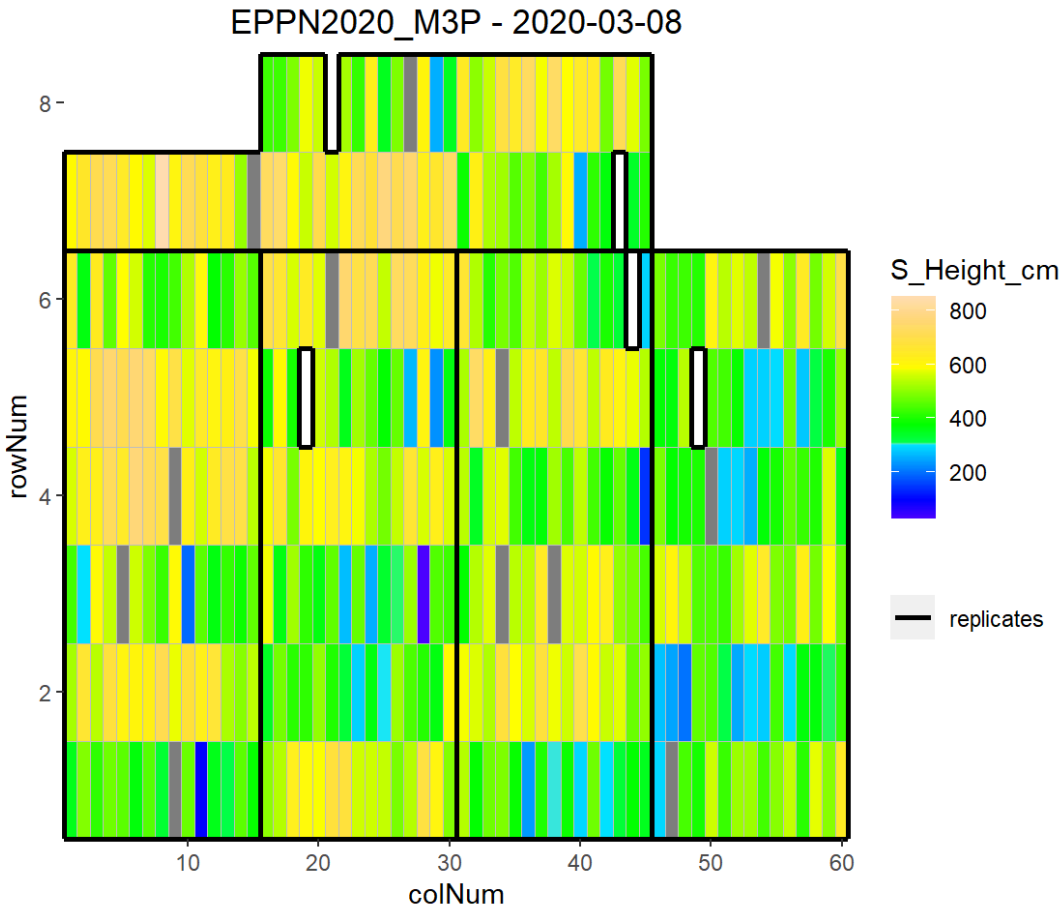

EPPN2020_M3P - 2020-03-03
rowNum
colNum
S_Height_cm
600
400
200
replicates

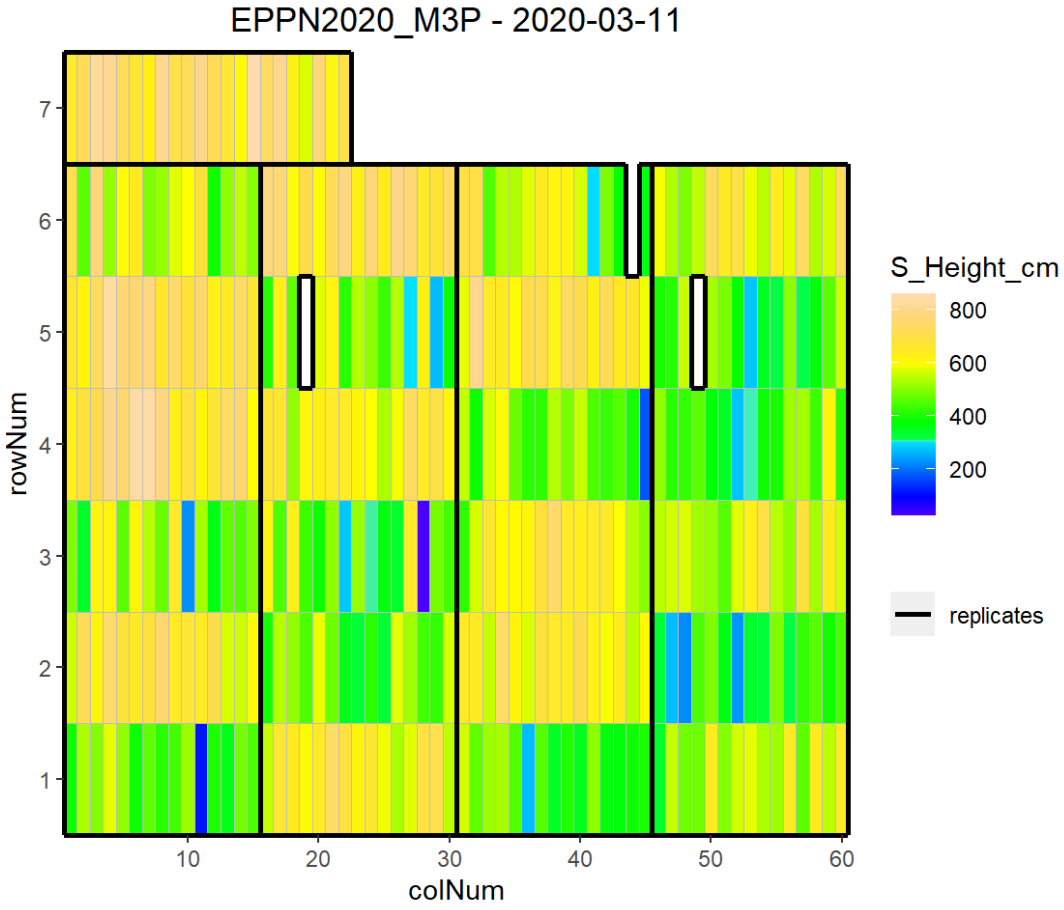

EPPN2020_M3P - 2020-03-04
rowNum
colNum
S_Height_cm
600
400
200
replicates

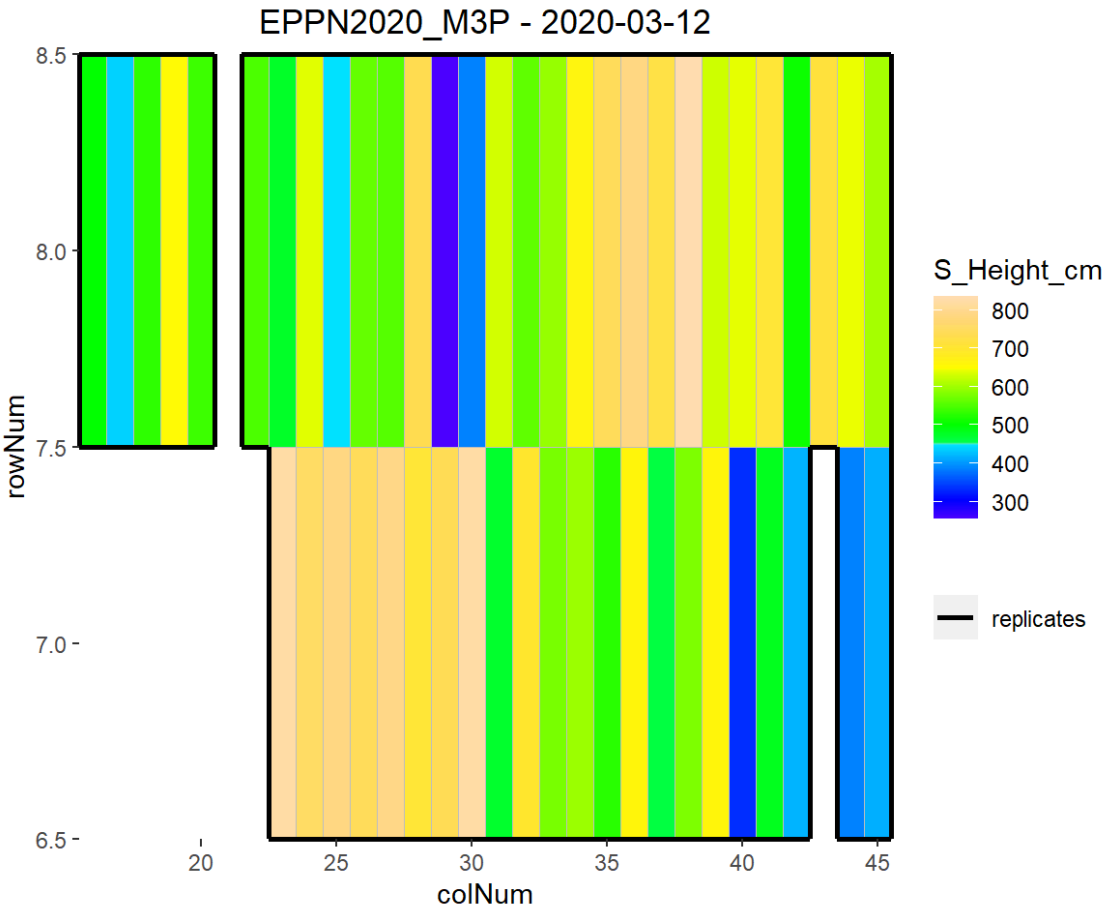

EPPN2020_M3P - 2020-03-05
rowNum
colNum
S_Height_cm
700
600
500
400
300
replicates

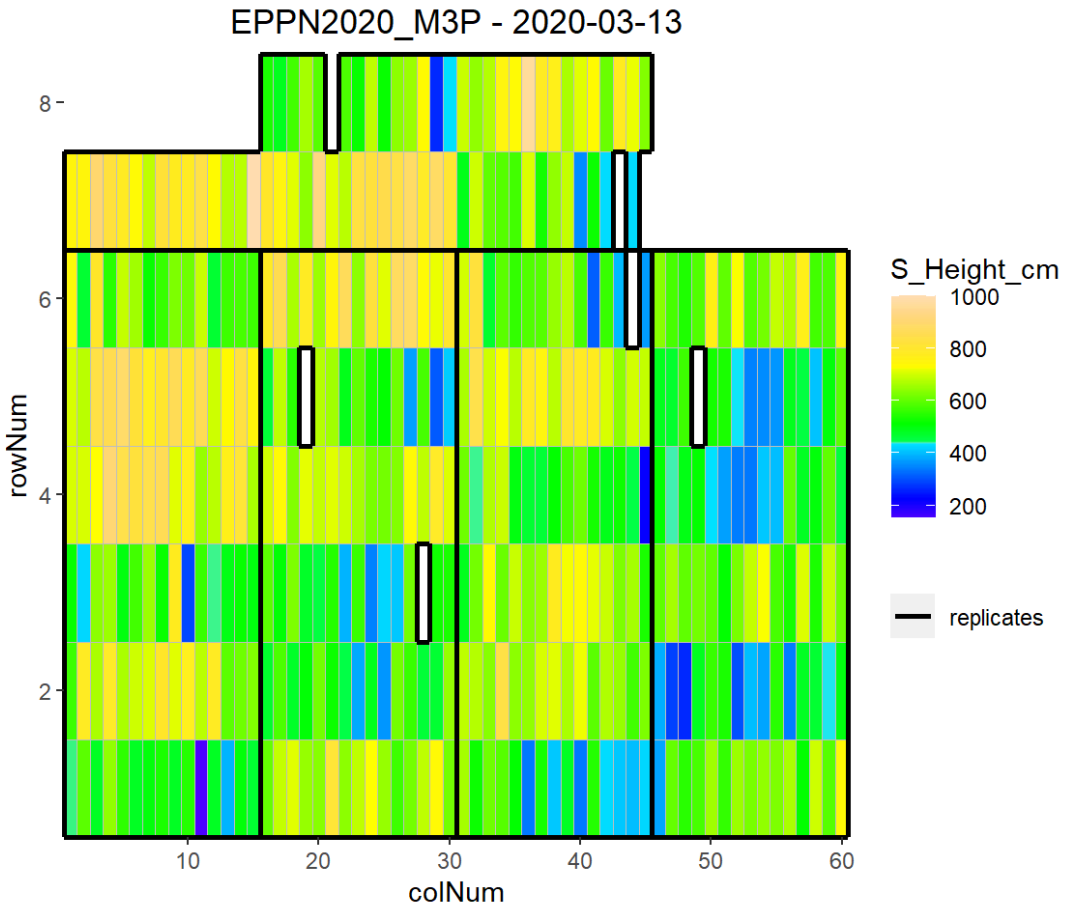

EPPN2020_M3P - 2020-03-06
rowNum
colNum
S_Height_cm
800
600
400
200
replicates

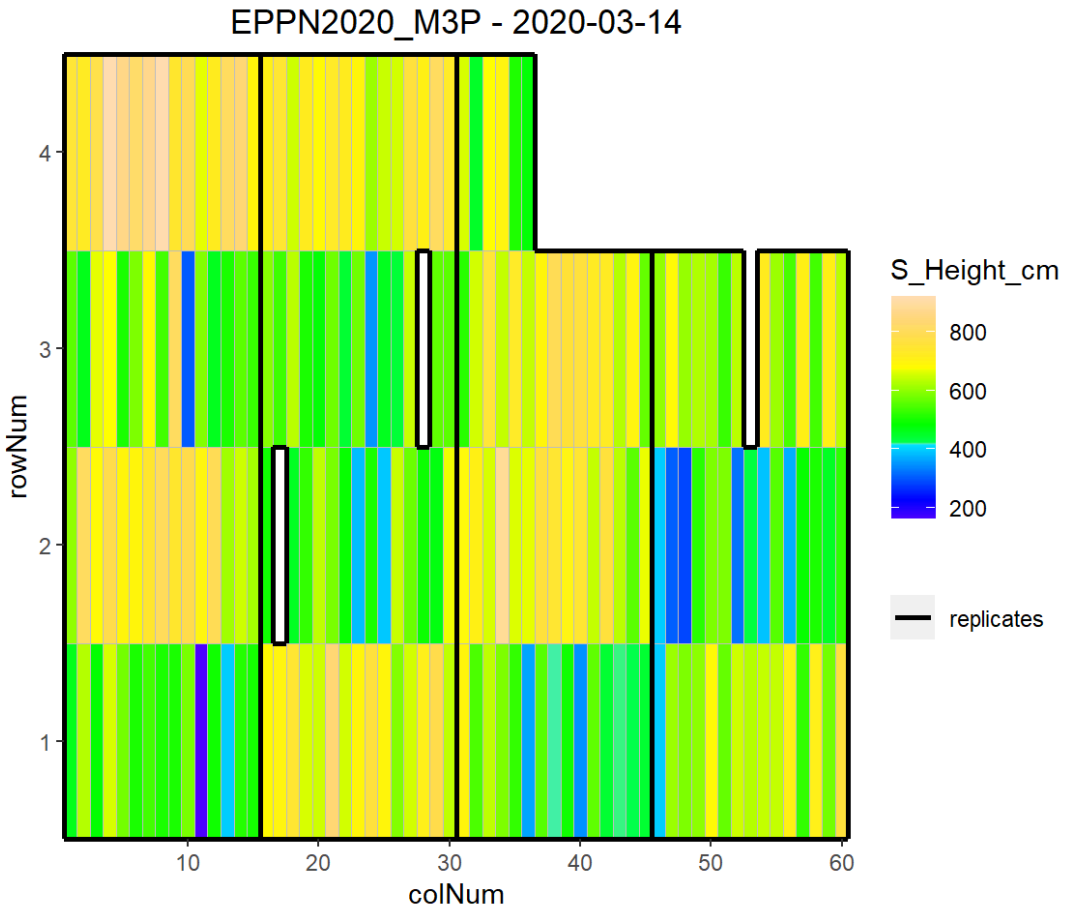

EPPN2020_M3P - 2020-03-07
rowNum
colNum
S_Height_cm
600
500
400
300
200
100
replicates

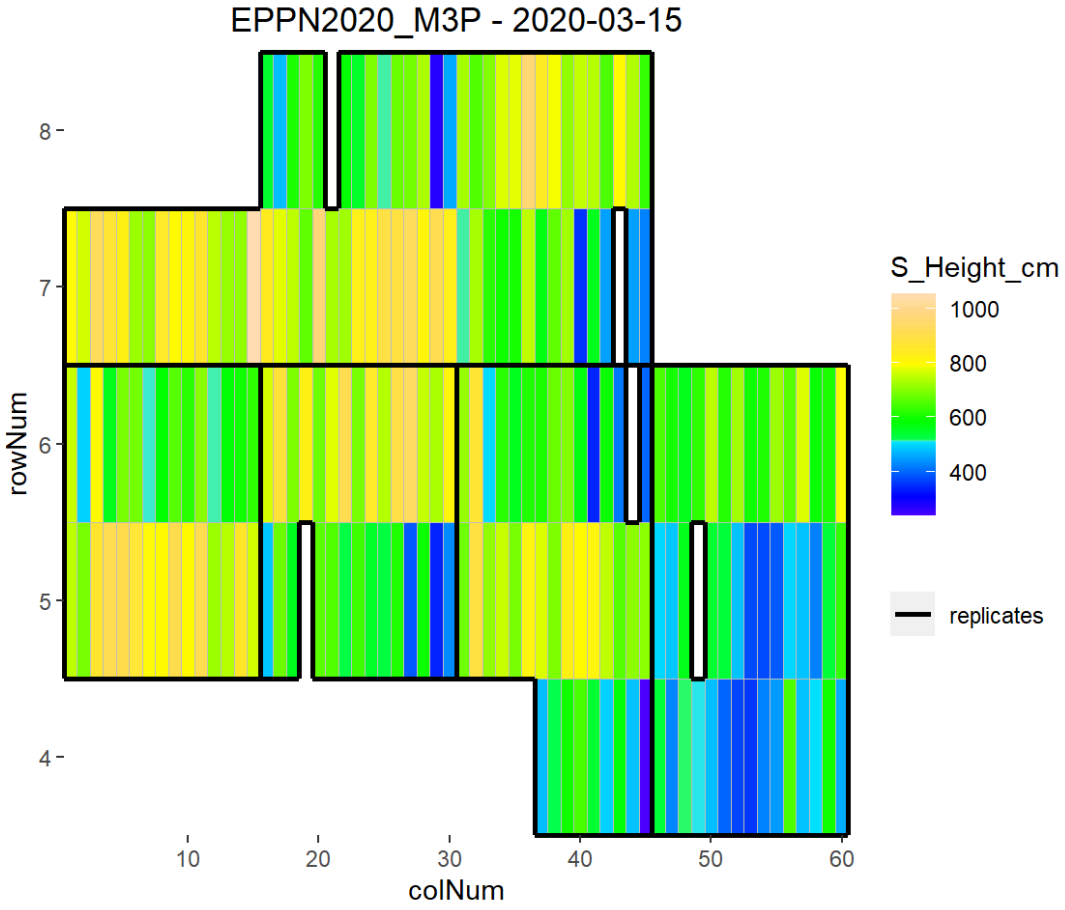

EPPN2020_M3P - 2020-03-08
rowNum
colNum
S_Height_cm
800
600
400
200
replicates

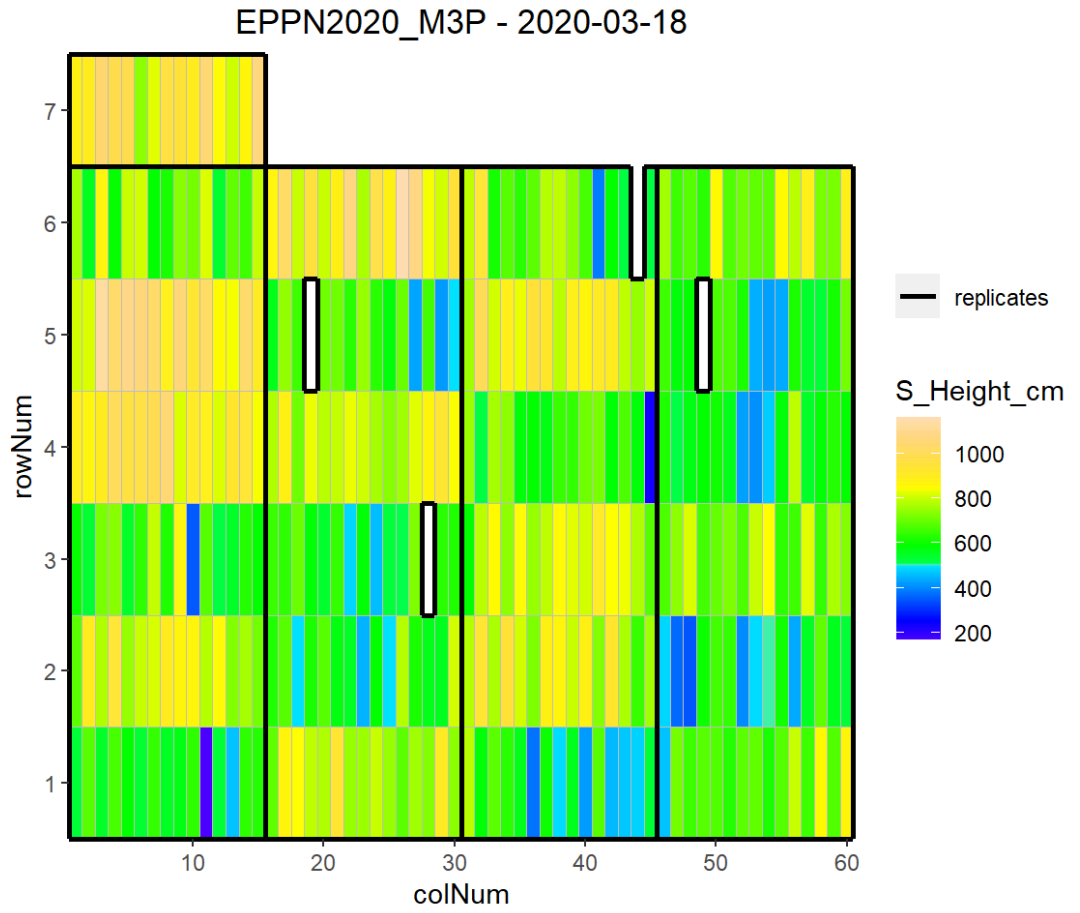

EPPN2020_M3P - 2020-03-11
rowNum
colNum
S_Height_cm
800
600
400
200
replicates

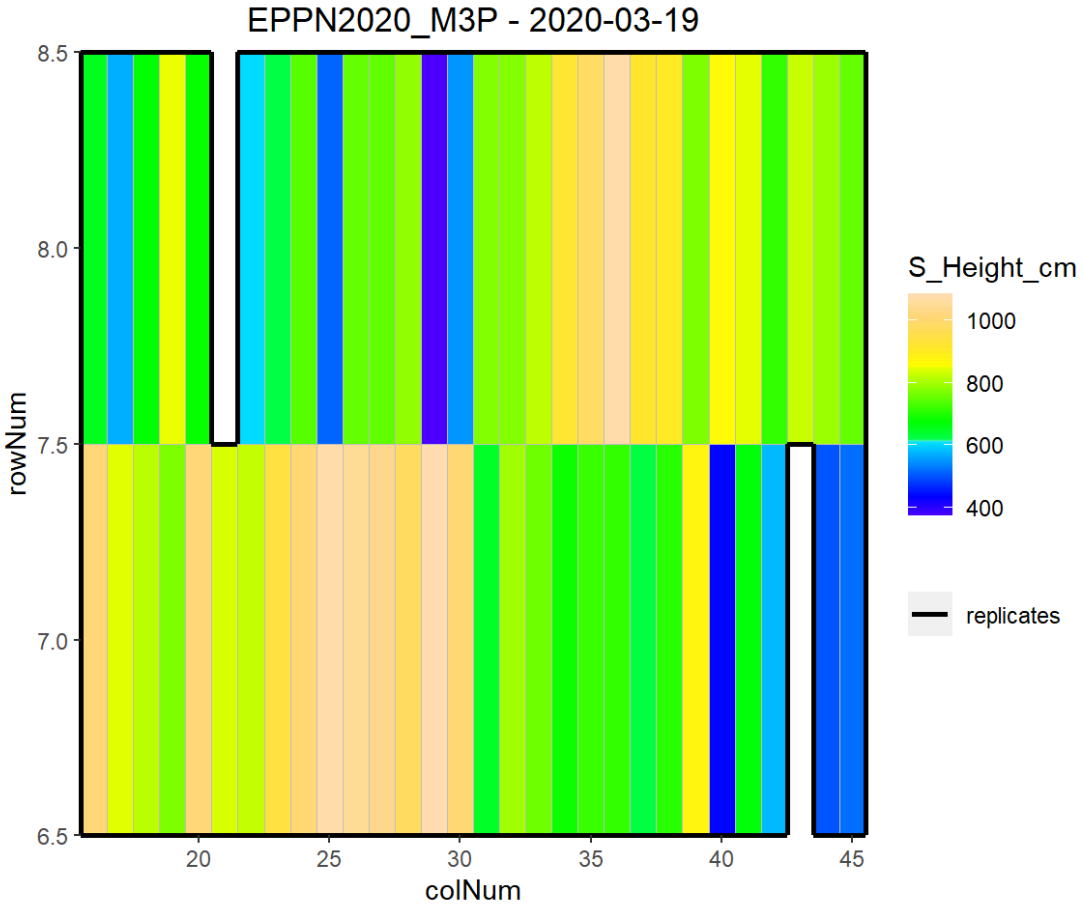

EPPN2020_M3P - 2020-03-12
rowNum
8.5
8.0
7.5
7.0
6.5
20
25
30
35
40
45
colNum
S_Height_cm
800
700
600
500
400
300
replicates

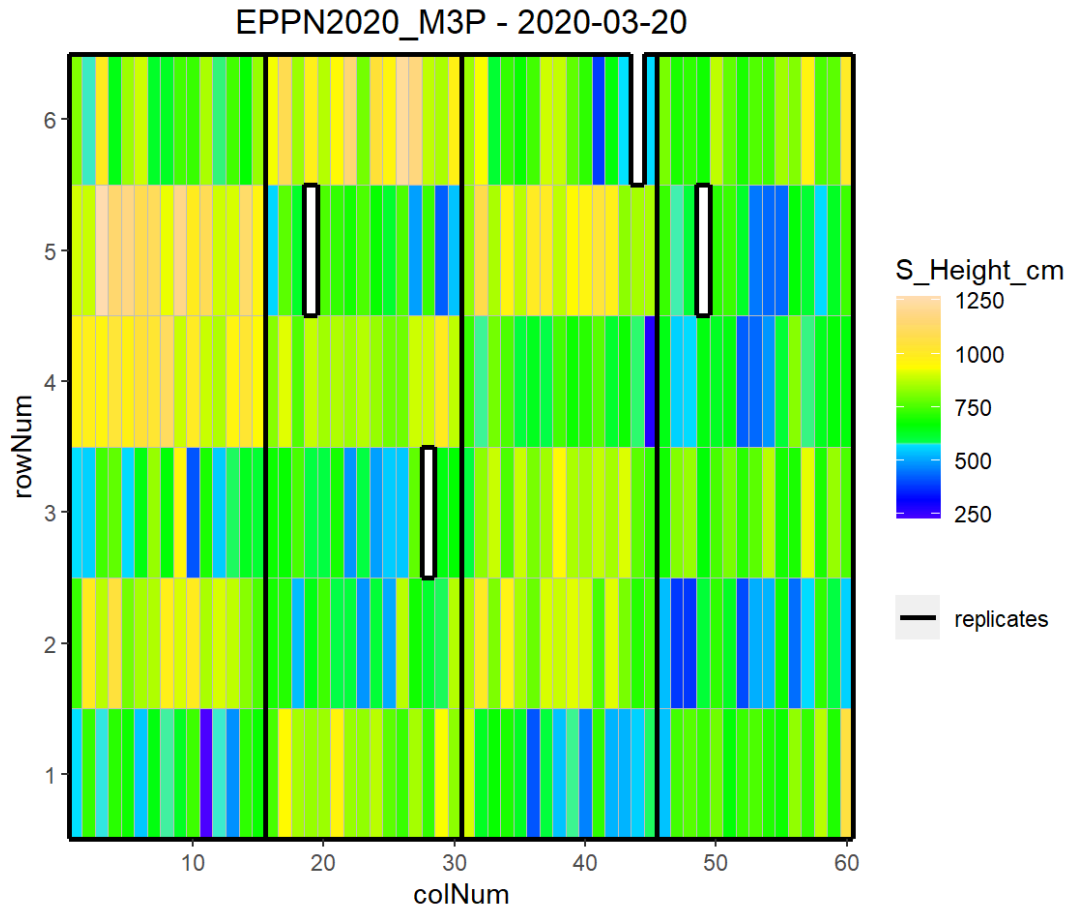

EPPN2020_M3P - 2020-03-13
rowNum
8
6
4
2
10
20
30
40
50
60
colNum
S_Height_cm
1000
800
600
400
200
replicates

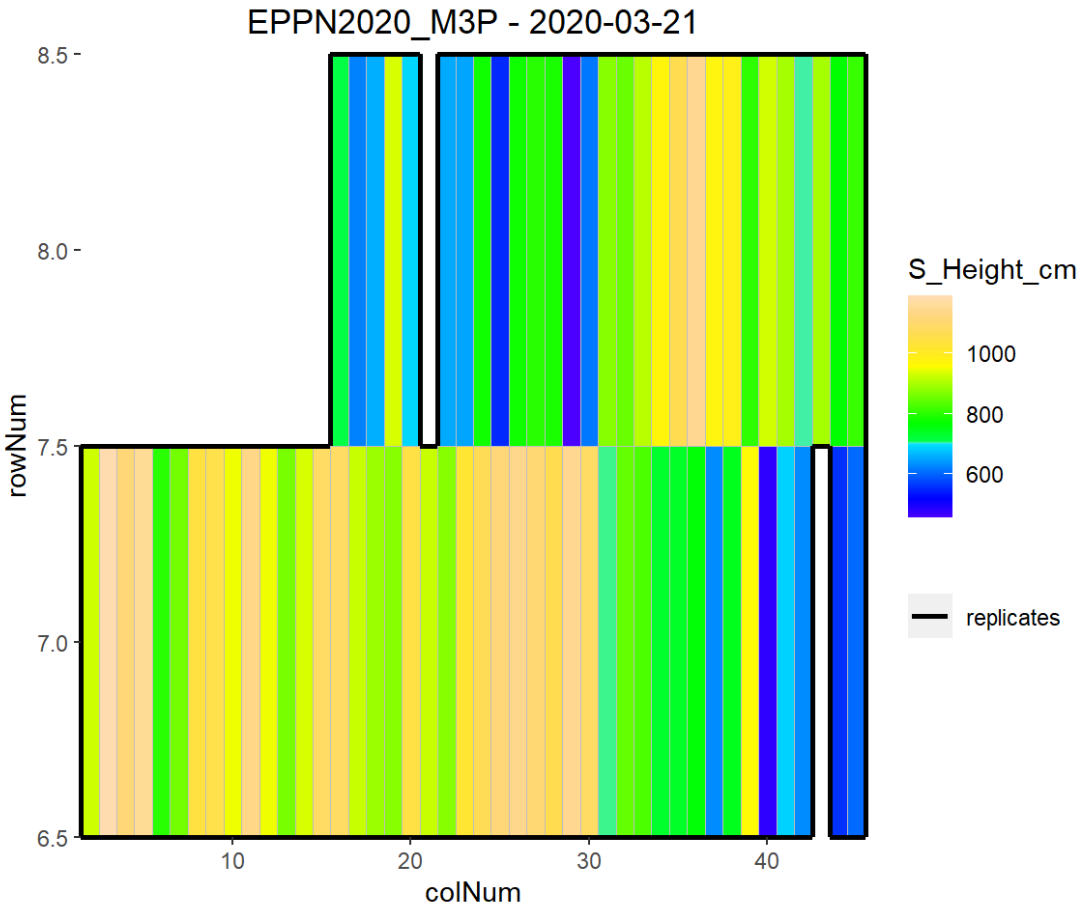

EPPN2020_M3P - 2020-03-14
rowNum
colNum
S_Height_cm
800
600
400
200
replicates

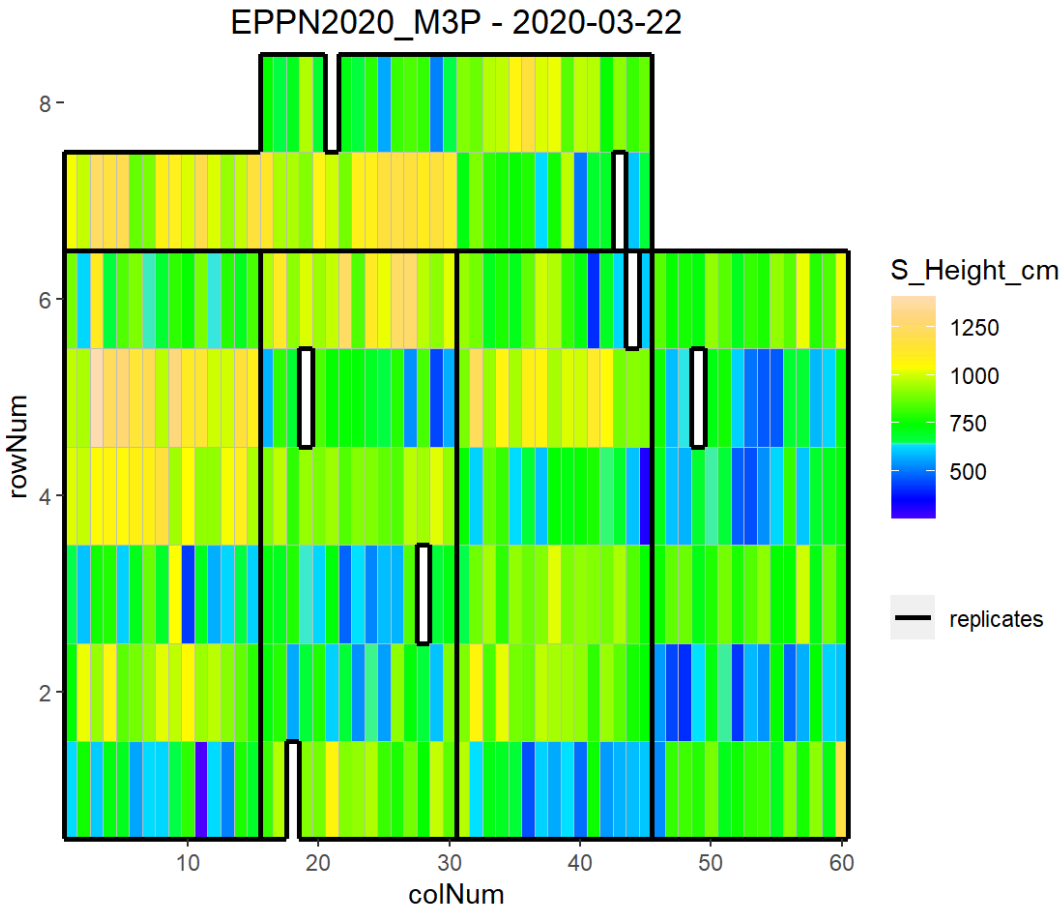

EPPN2020_M3P - 2020-03-15
rowNum
colNum
S_Height_cm
1000
800
600
400
replicates

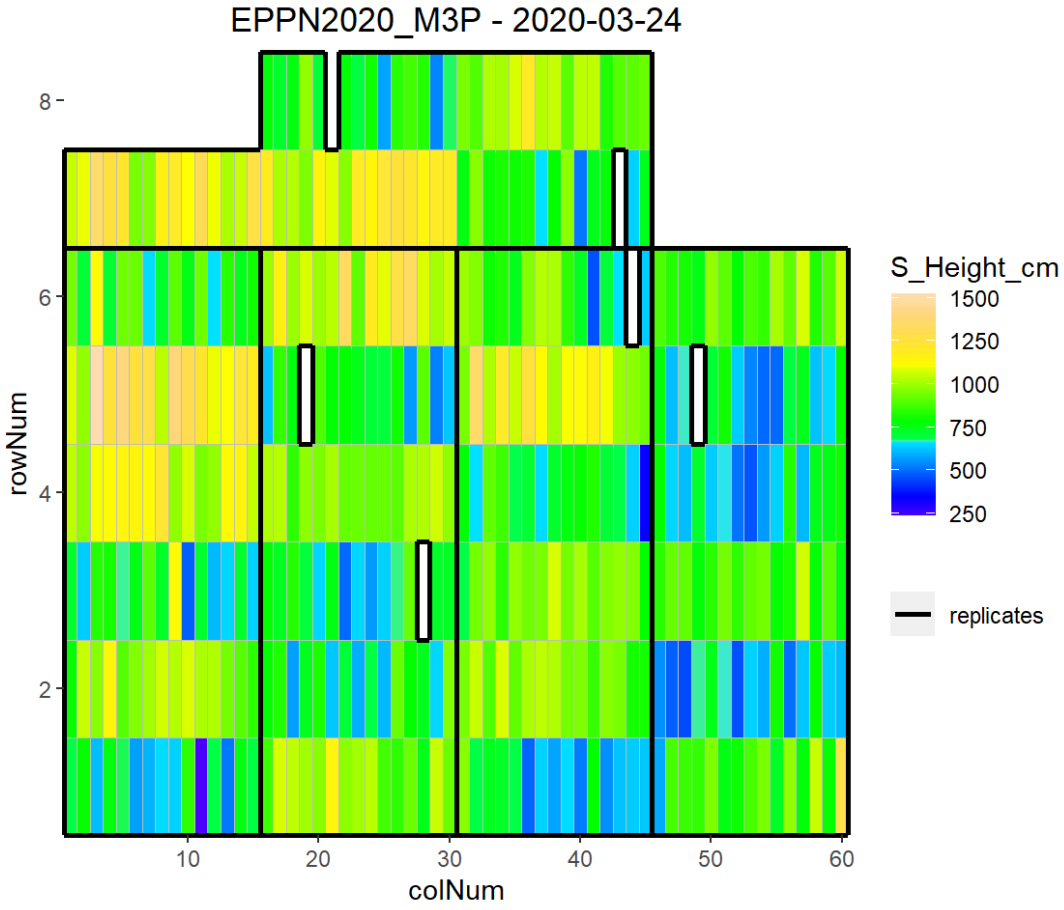

EPPN2020_M3P - 2020-03-18
rowNum
colNum
replicates
S_Height_cm
1000
800
600
400
200

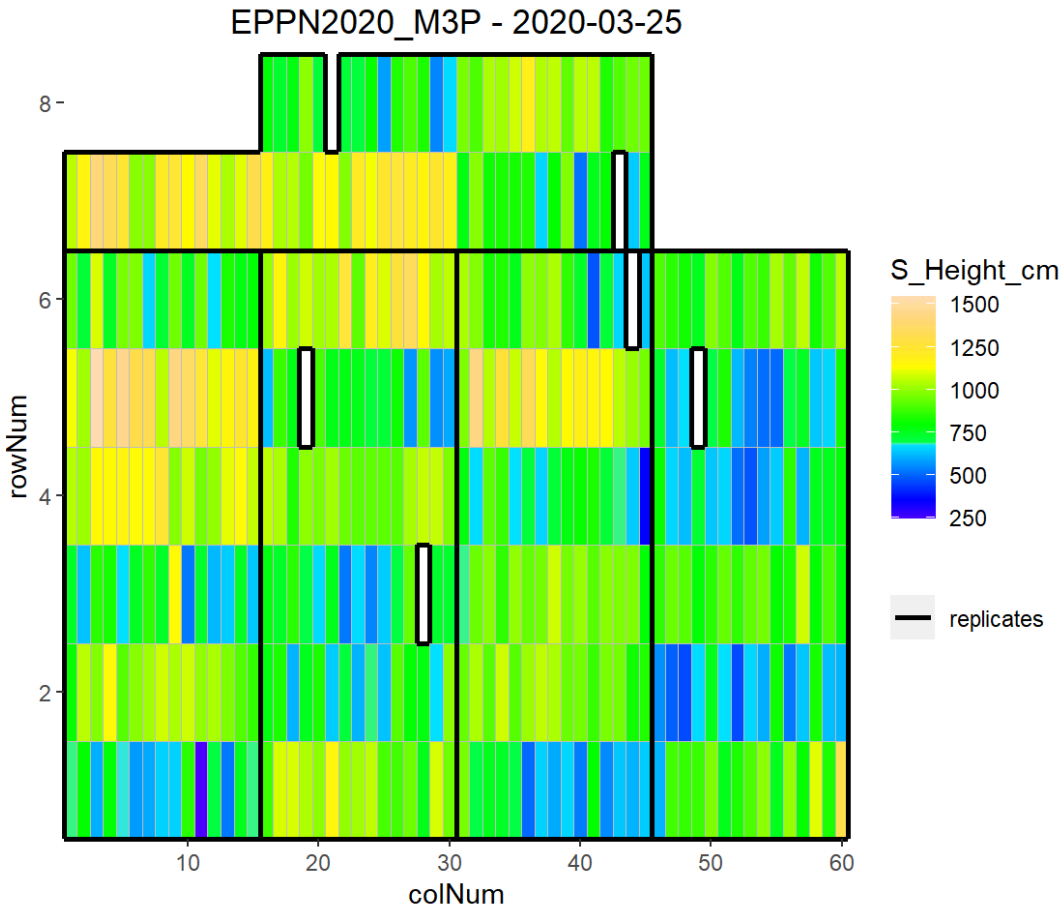

EPPN2020_M3P - 2020-03-19
rowNum
colNum
S_Height_cm
1000
800
600
400
replicates

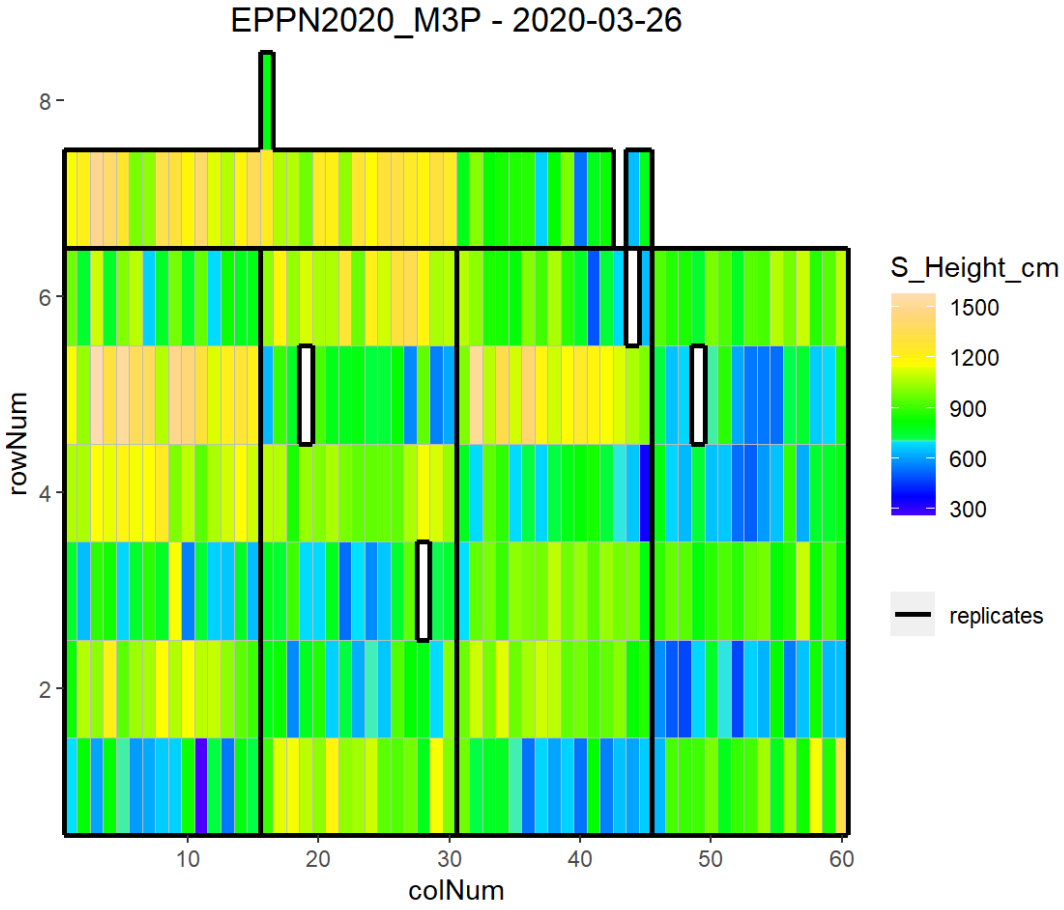

EPPN2020_M3P - 2020-03-20
rowNum
colNum
S_Height_cm
1250
1000
750
500
250
replicates

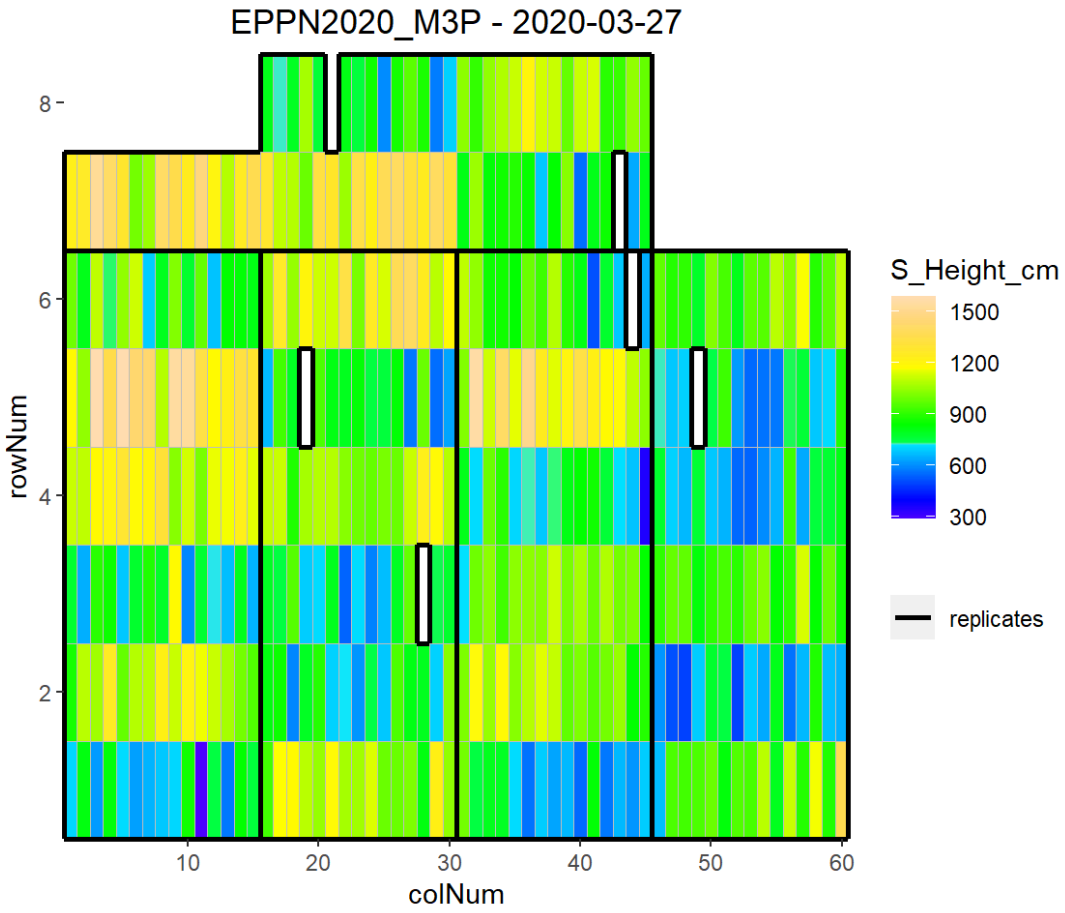

EPPN2020_M3P - 2020-03-21
rowNum
colNum
S_Height_cm
1000
800
600
replicates

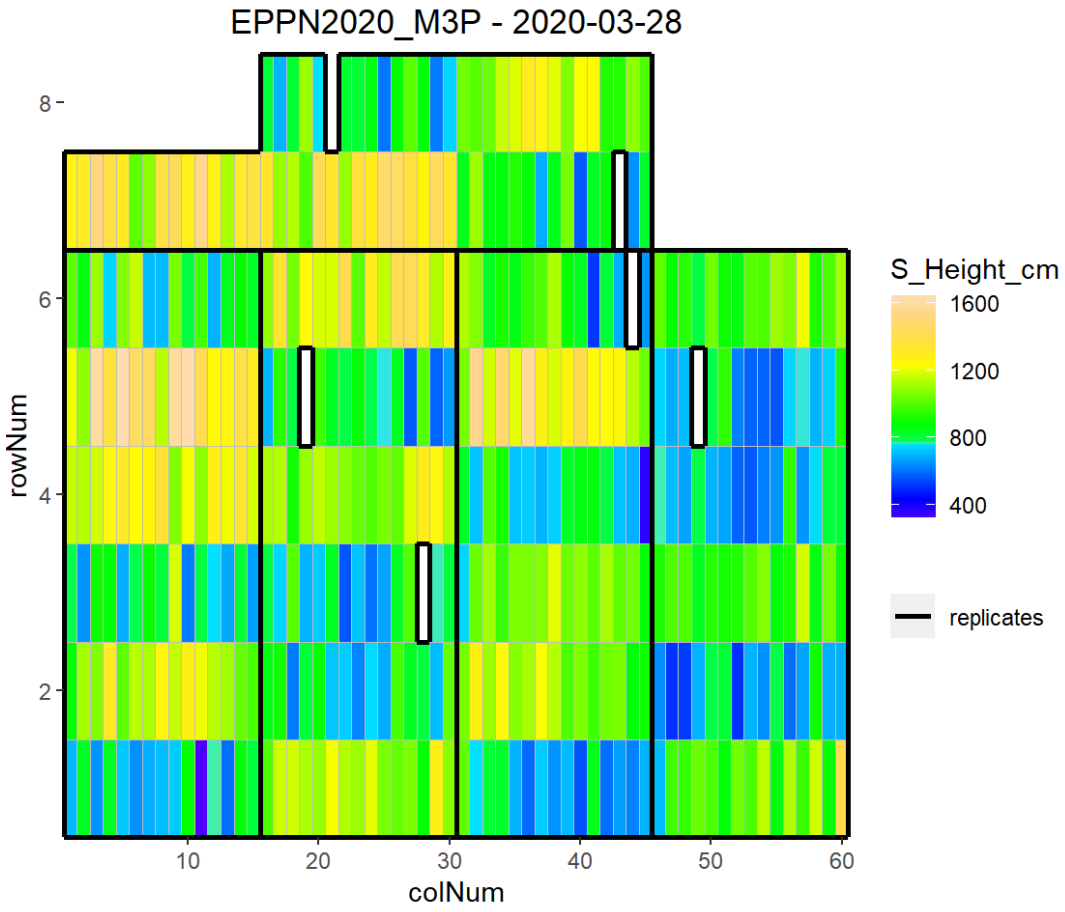

EPPN2020_M3P - 2020-03-22
rowNum
colNum
S_Height_cm
1250
1000
750
500
replicates

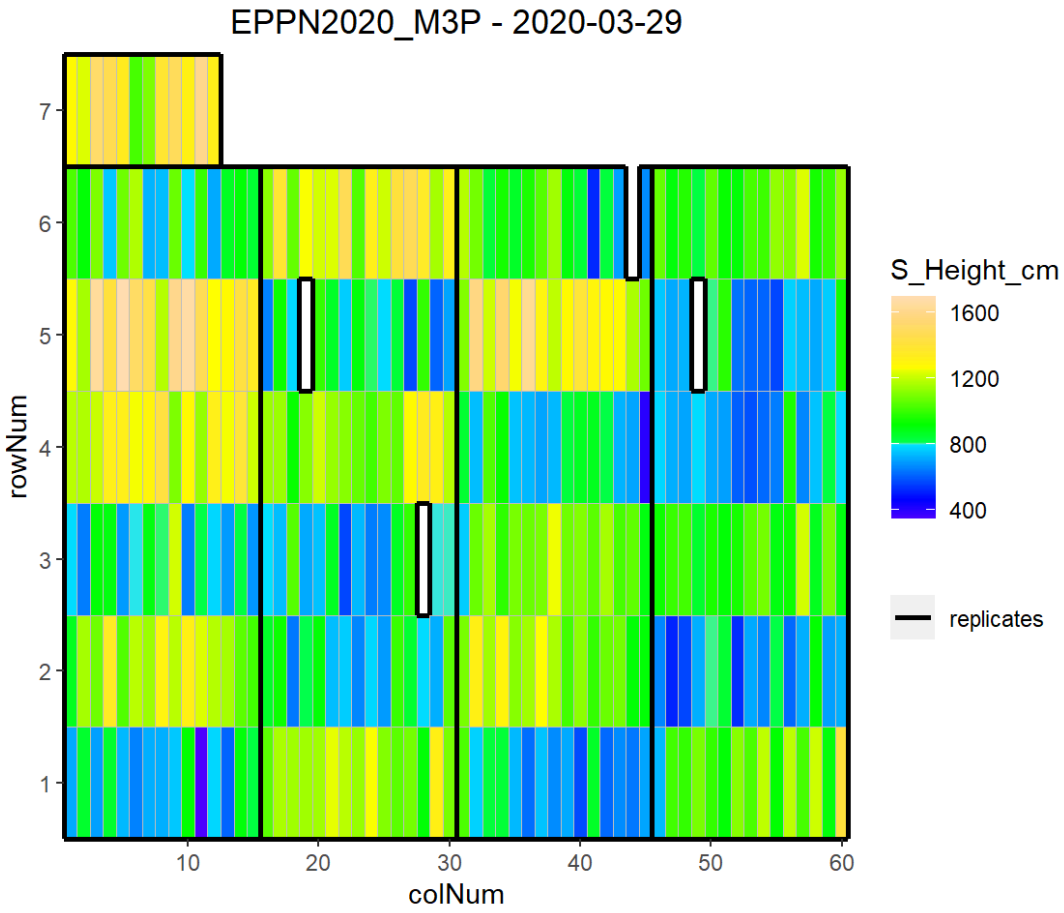

EPPN2020_M3P - 2020-03-24
rowNum
colNum
S_Height_cm
1500
1250
1000
750
500
250
replicates

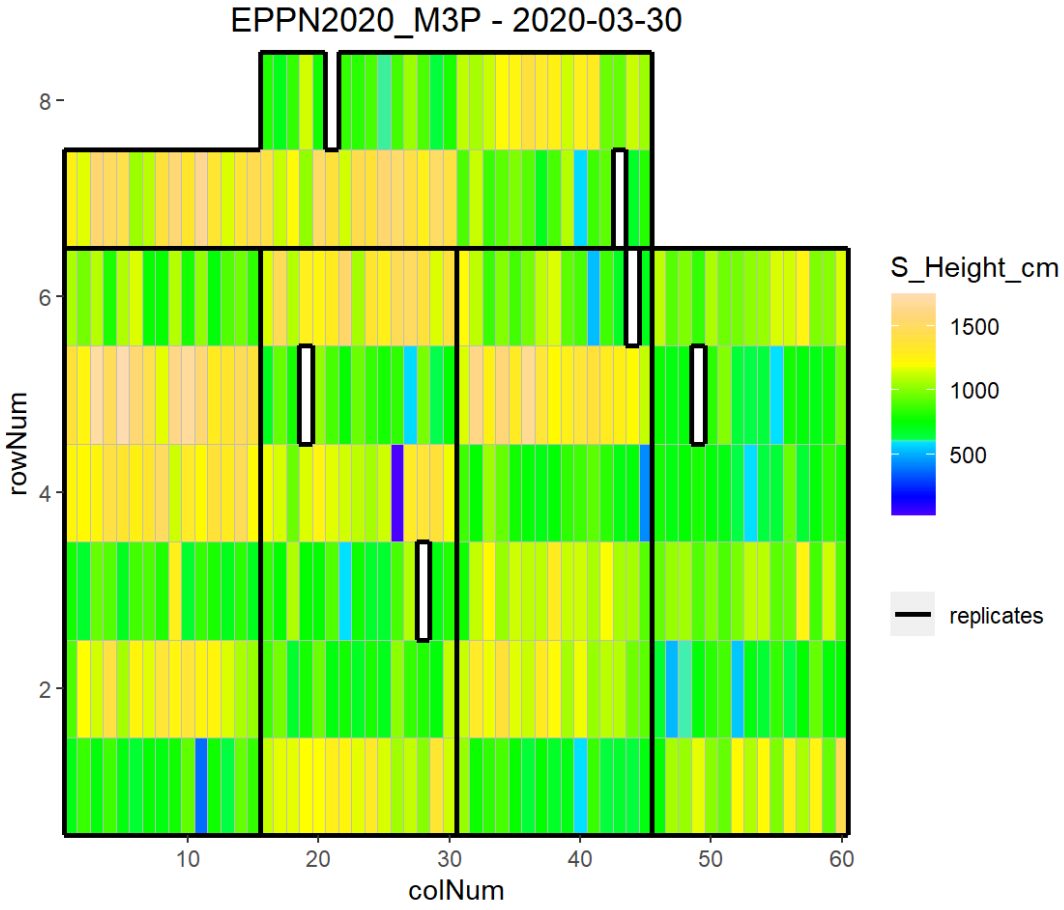

EPPN2020_M3P - 2020-03-25
rowNum
colNum
S_Height_cm
1500
1250
1000
750
500
250
replicates

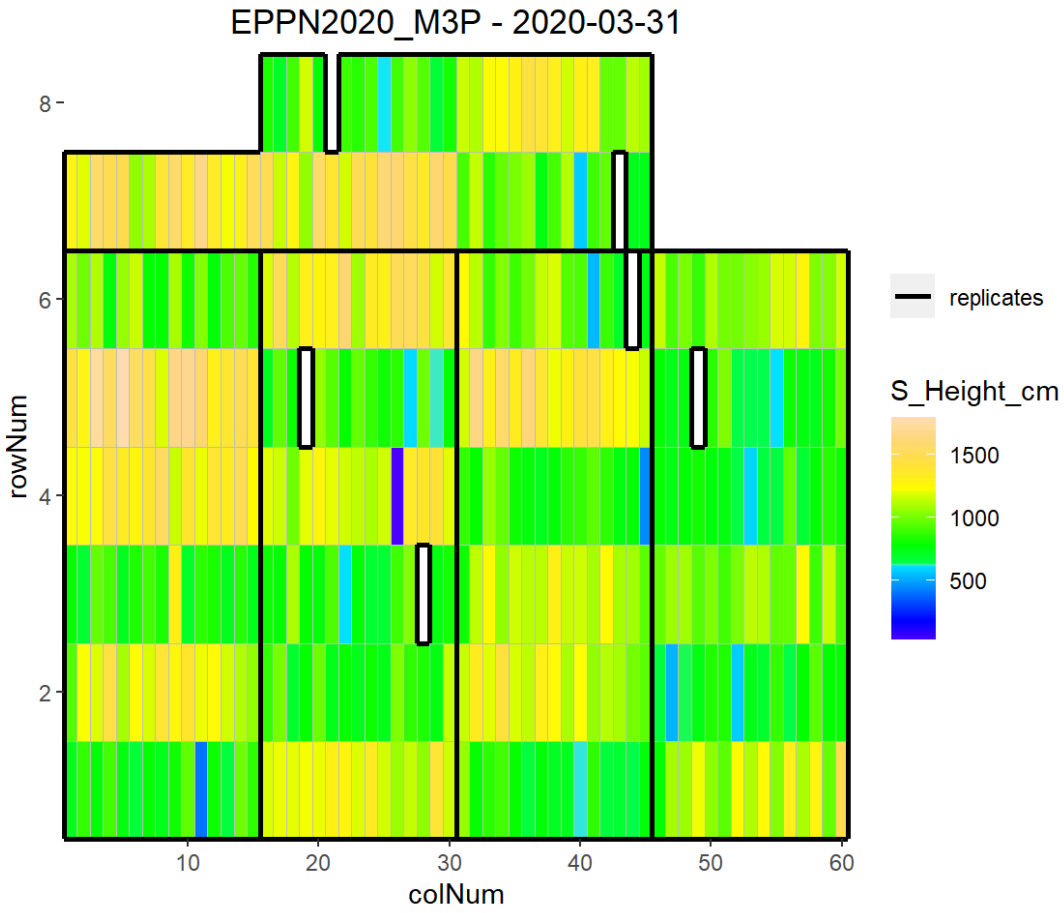

EPPN2020_M3P - 2020-03-26
rowNum
colNum
S_Height_cm
1500
1200
900
600
300
replicates

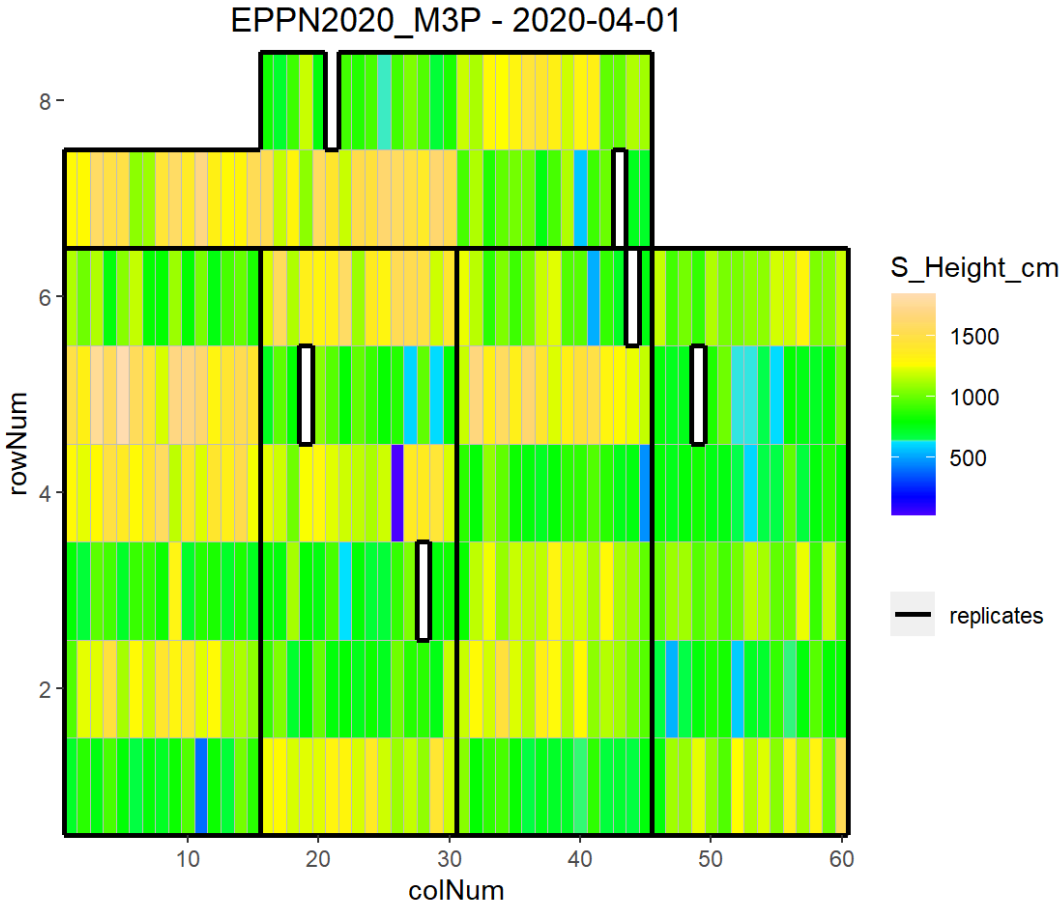

EPPN2020_M3P - 2020-03-27
rowNum
colNum
S_Height_cm
1500
1200
900
600
300
replicates

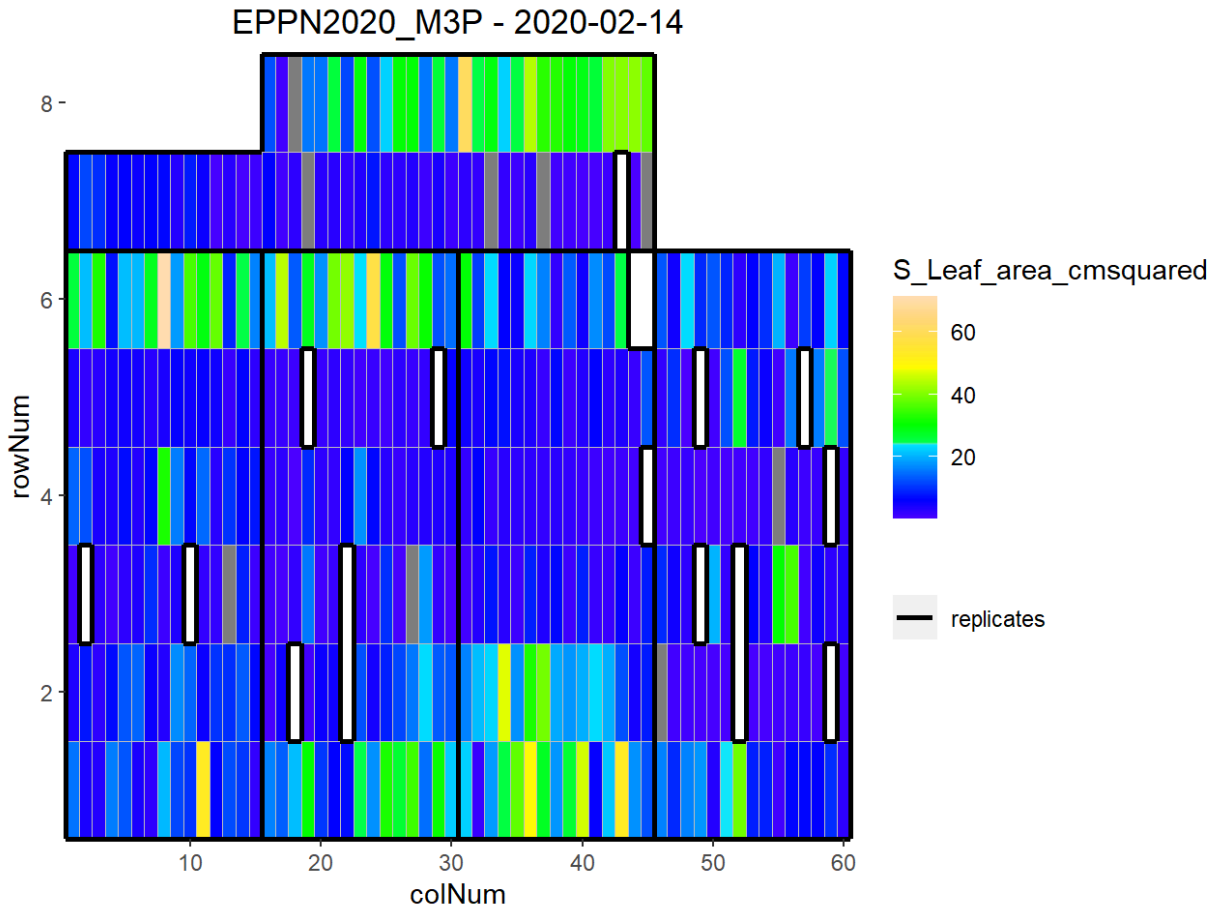

EPPN2020_M3P - 2020-03-28
rowNum
colNum
S_Height_cm
1600
1200
800
400
replicates

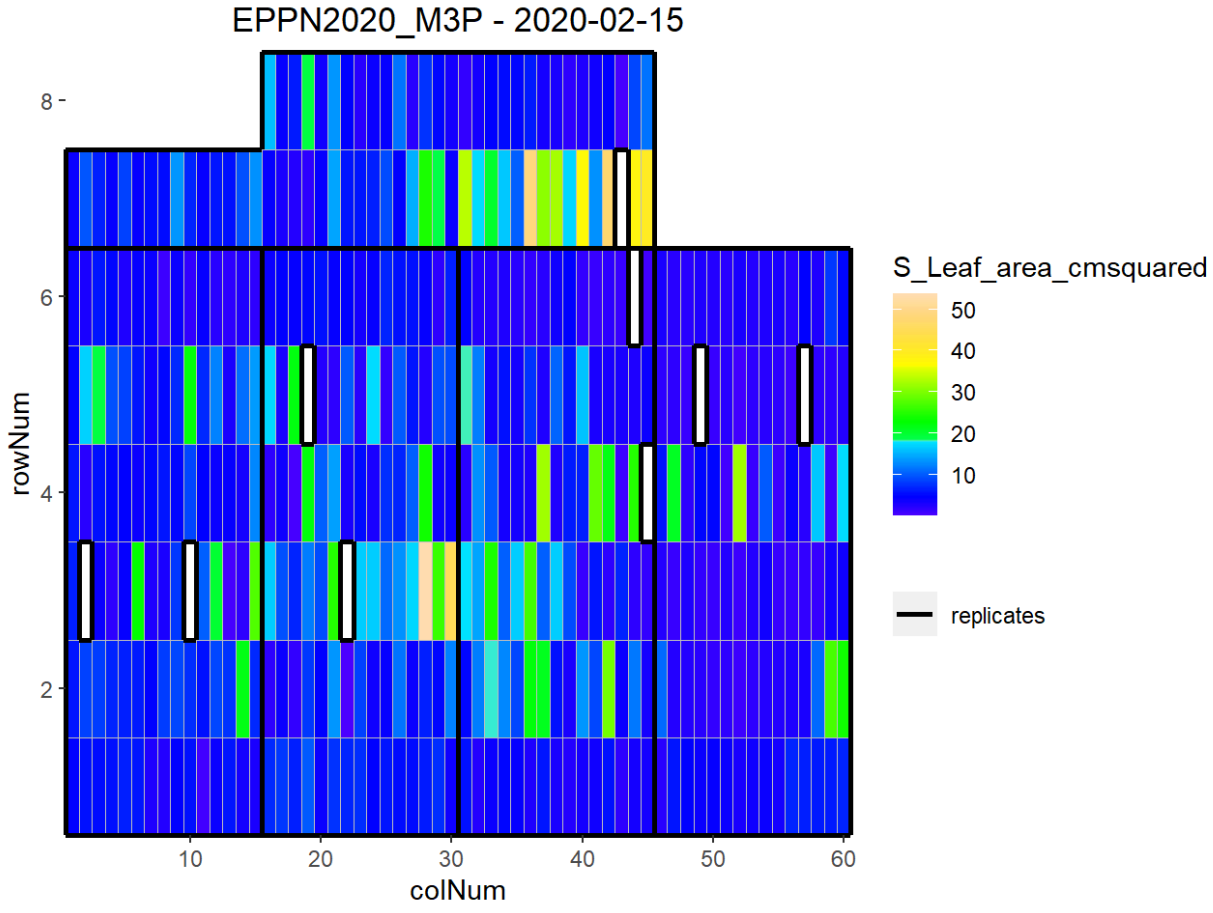

EPPN2020_M3P - 2020-03-29
rowNum
colNum
S_Height_cm
1600
1200
800
400
replicates

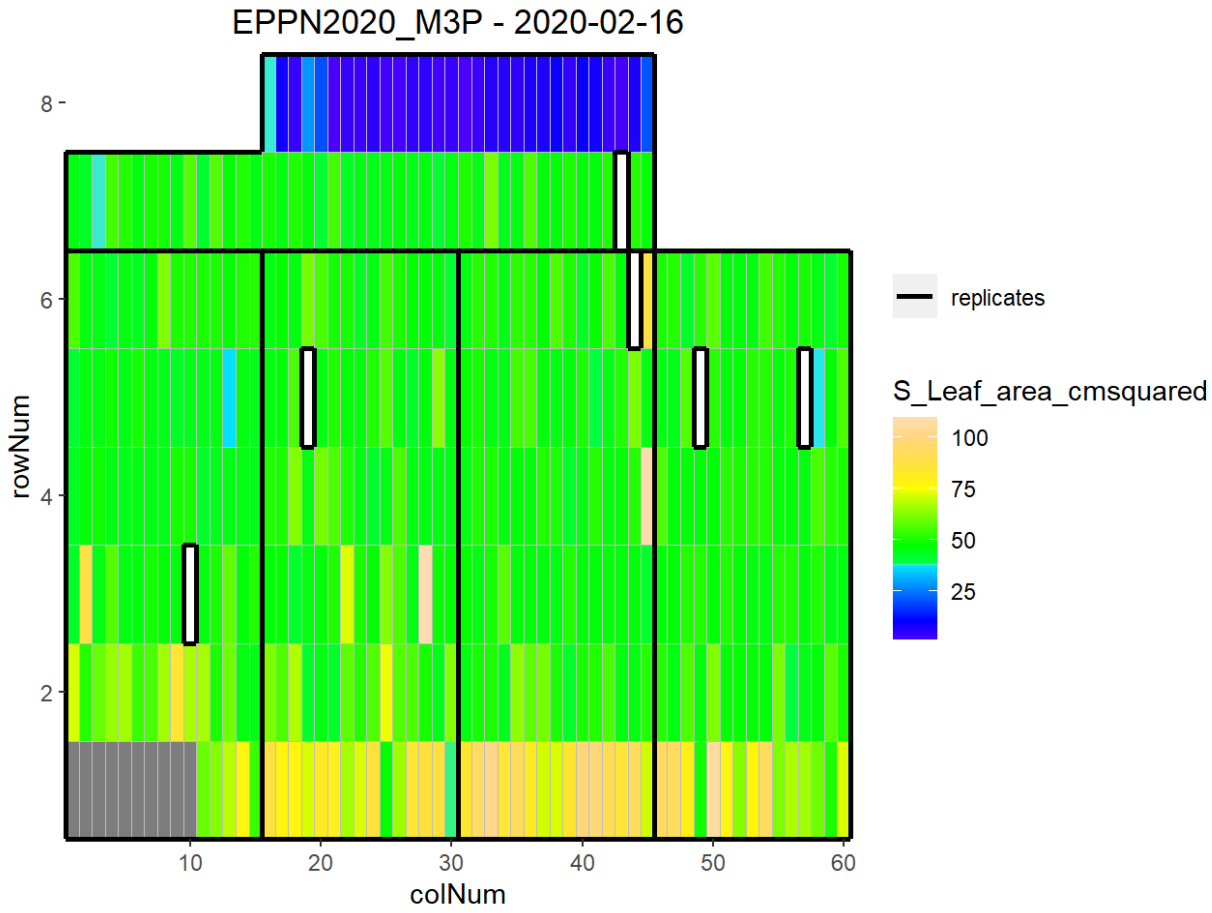

EPPN2020_M3P - 2020-03-30
rowNum
colNum
S_Height_cm
1500
1000
500
replicates

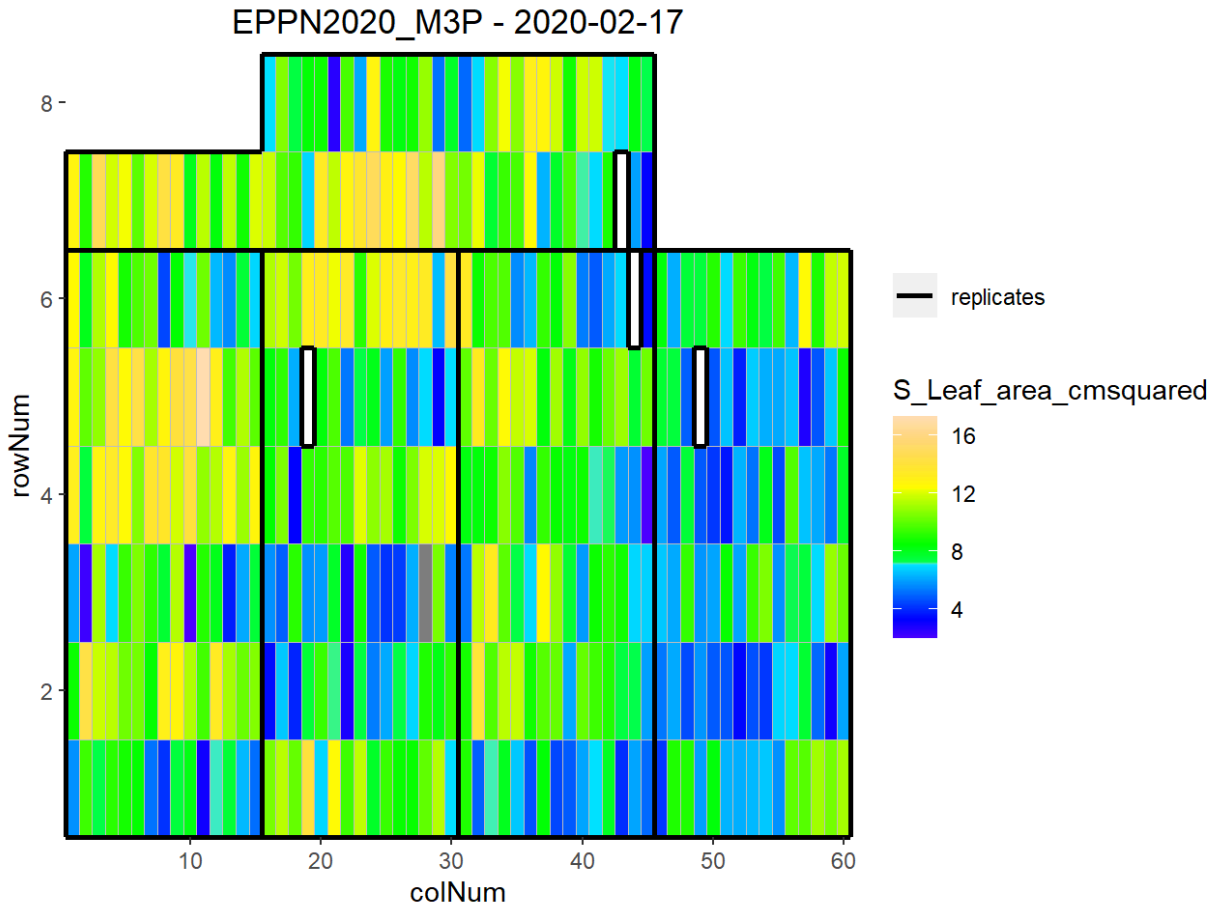

EPPN2020_M3P - 2020-03-31
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Height_cm
1500
1000
500

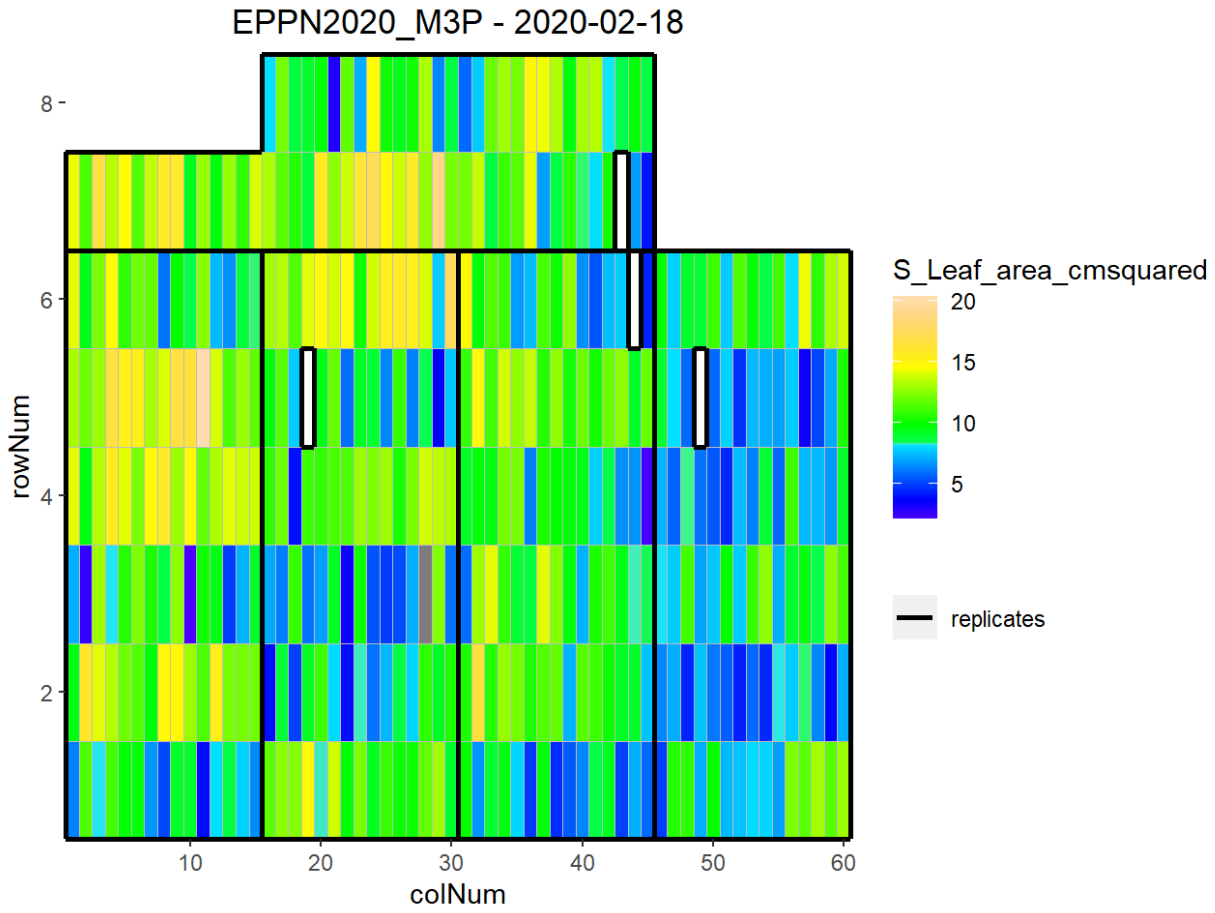

EPPN2020_M3P - 2020-04-01
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Height_cm
1500
1000
500
replicates

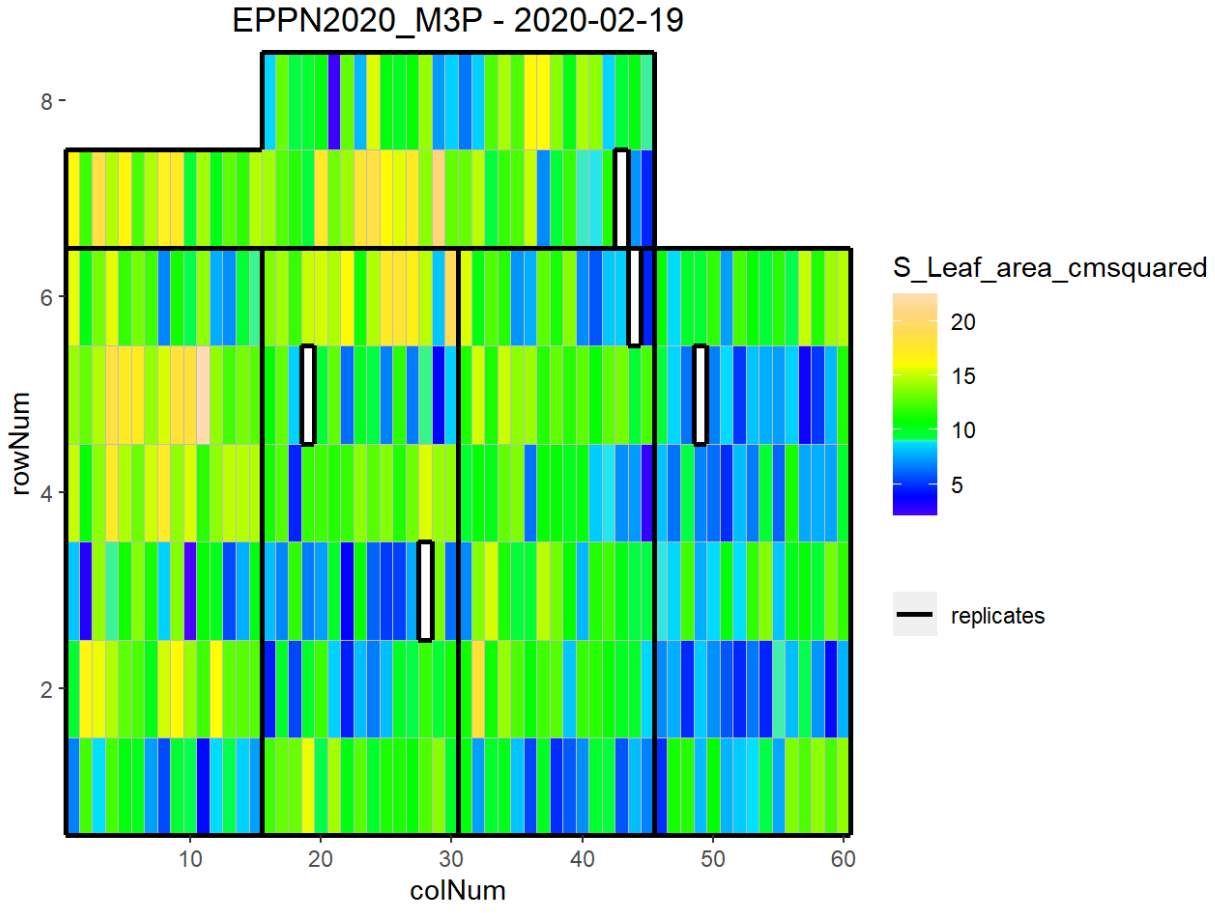

EPPN2020_M3P - 2020-02-14
rowNum
colNum
S_Leaf_area_cmsquared
60
40
20
replicates

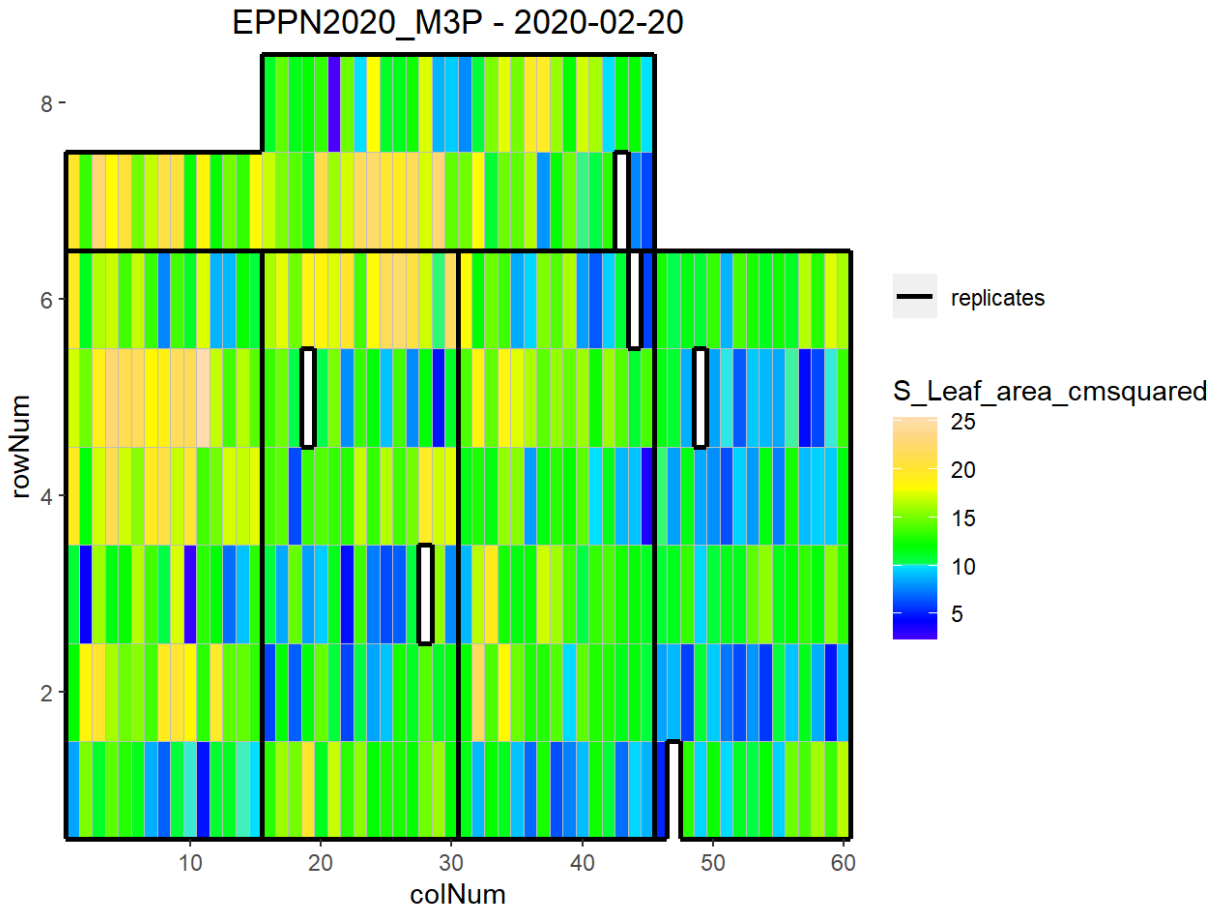

EPPN2020_M3P - 2020-02-15
rowNum
colNum
S_Leaf_area_cmsquared
50
40
30
20
10
replicates

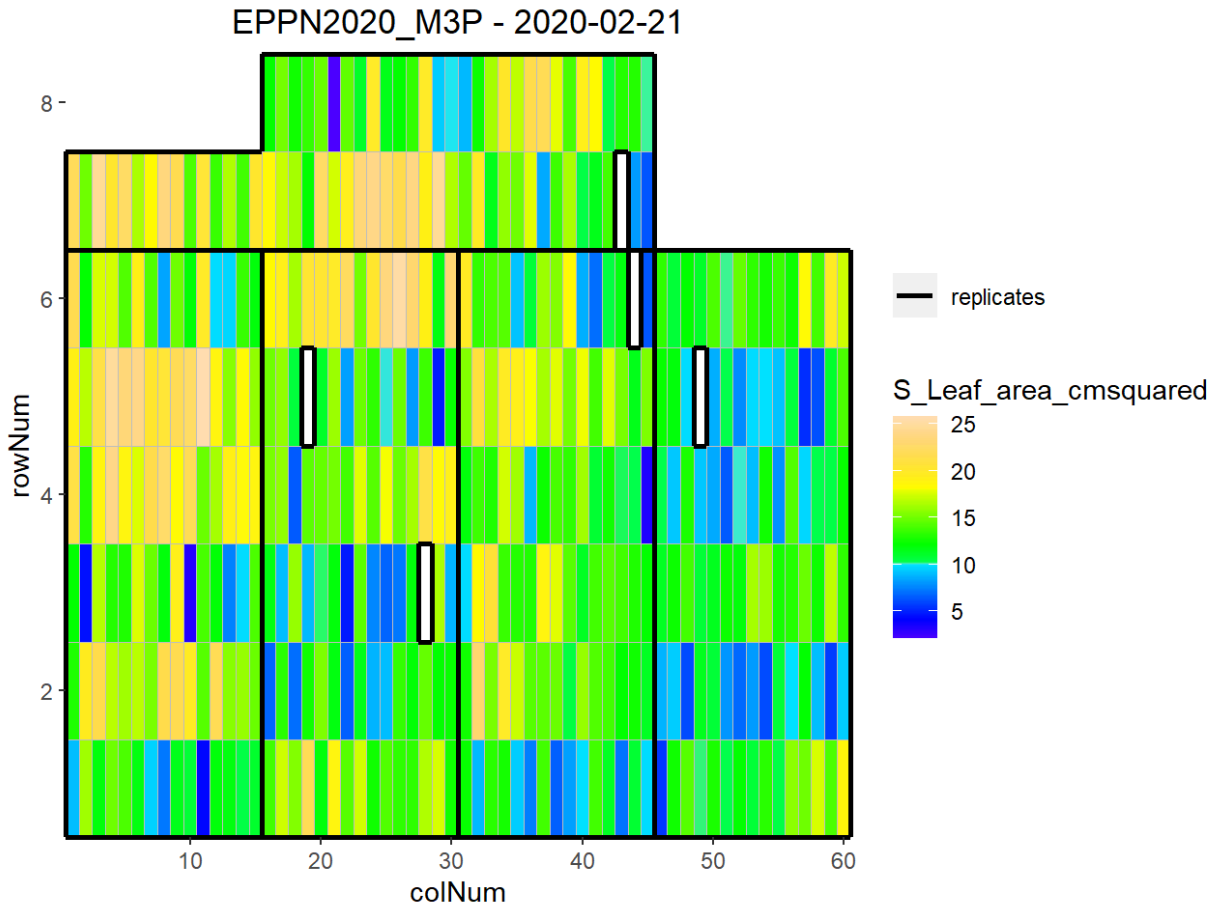

EPPN2020_M3P - 2020-02-16
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
100
75
50
25

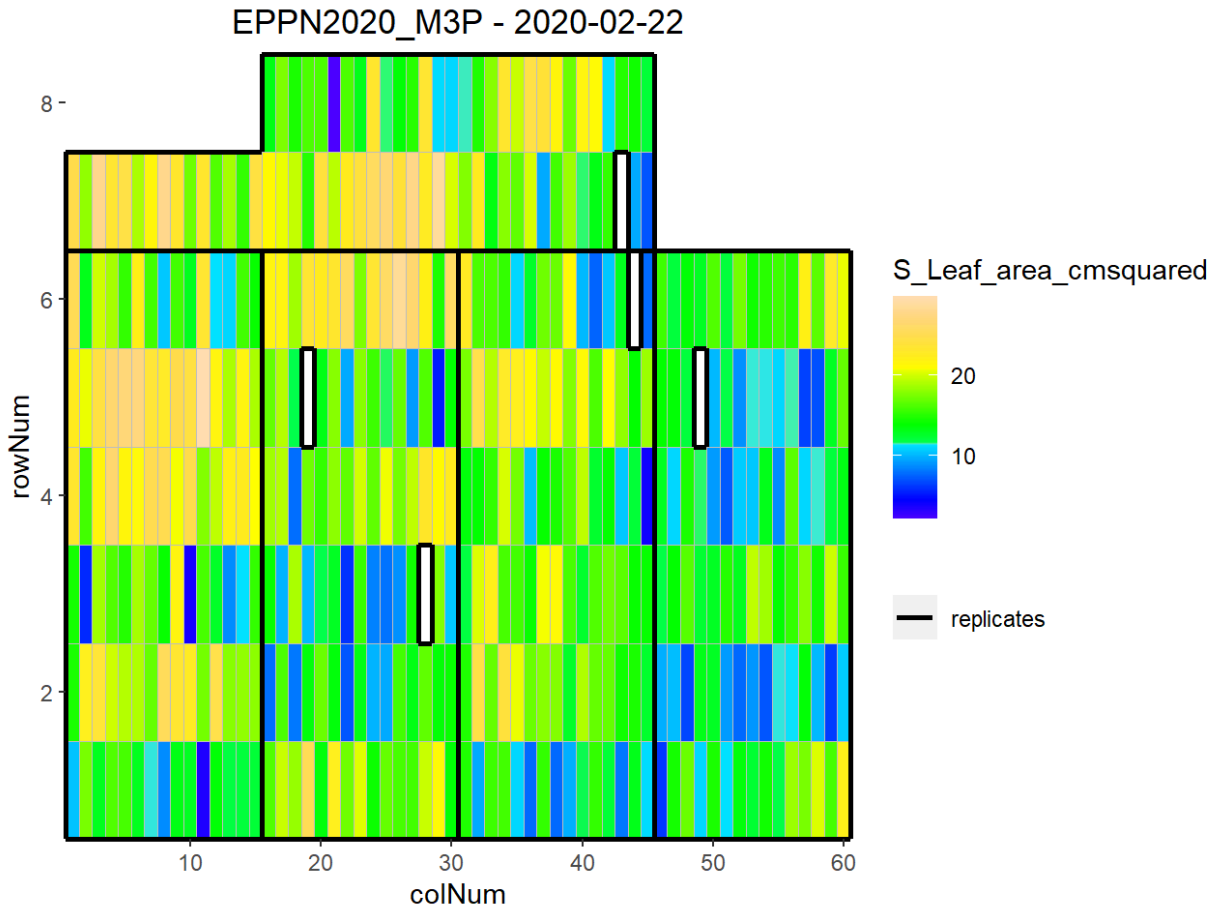

EPPN2020_M3P - 2020-02-17
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
16
12
8
4

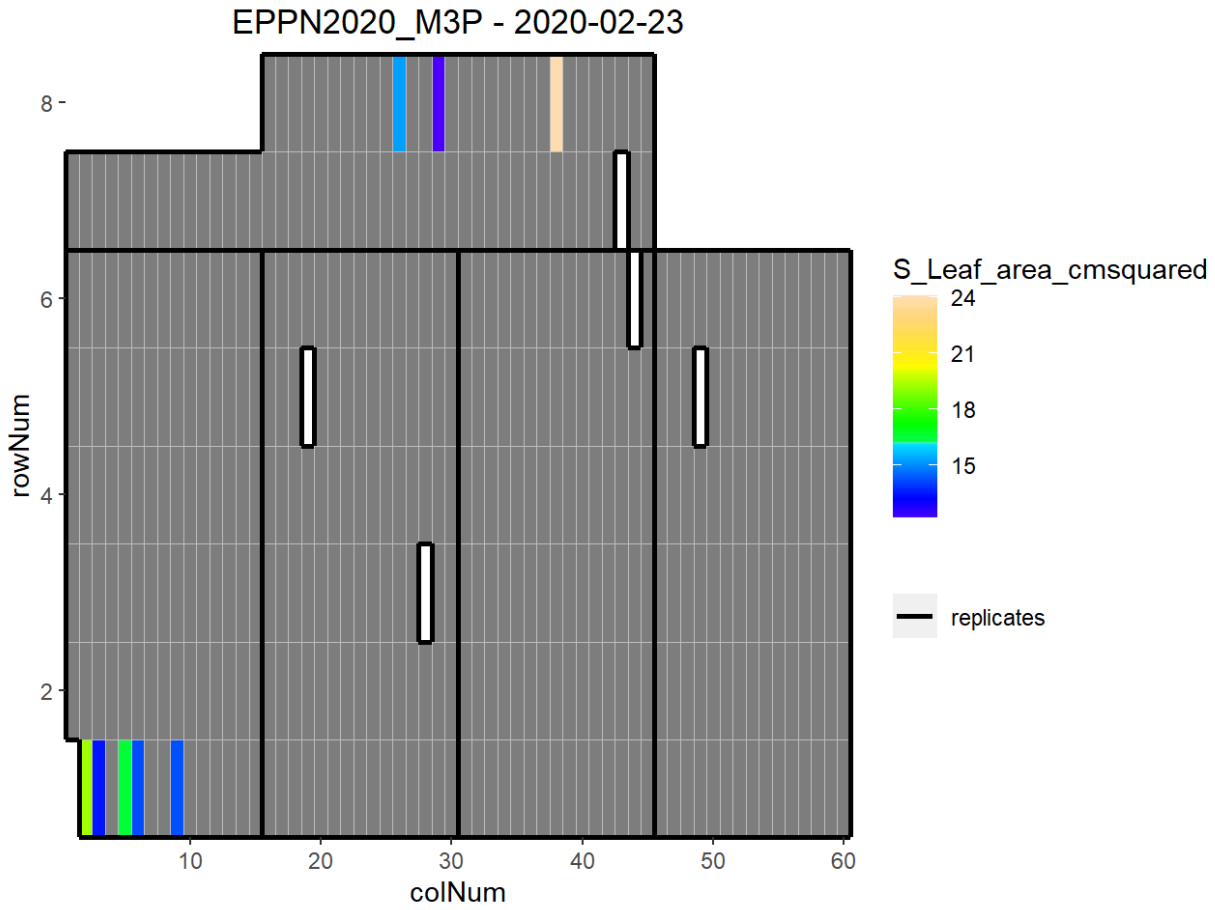

EPPN2020_M3P - 2020-02-18
rowNum
colNum
S_Leaf_area_cmsquared
20
15
10
5
replicates

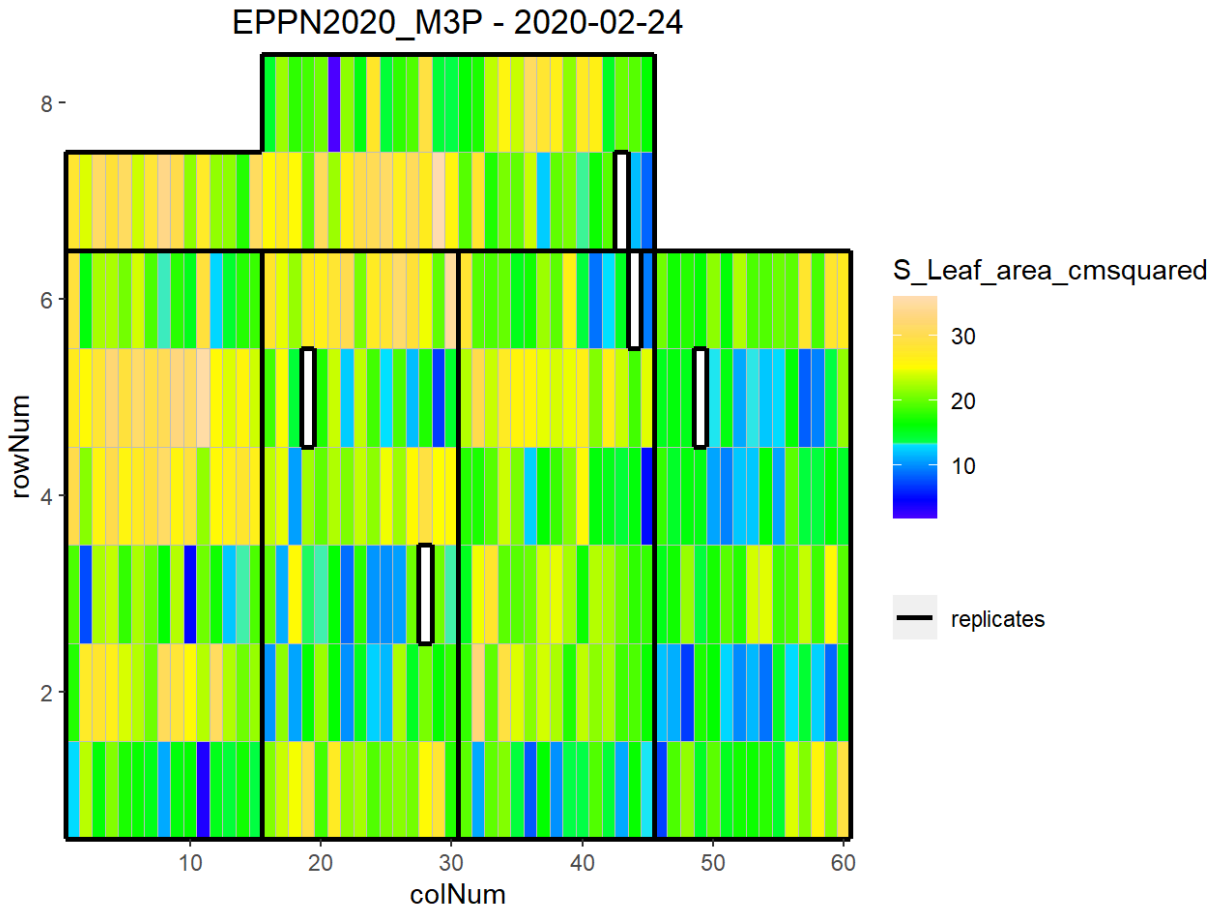

EPPN2020_M3P - 2020-02-19
rowNum
colNum
S_Leaf_area_cmsquared
20
15
10
5
replicates

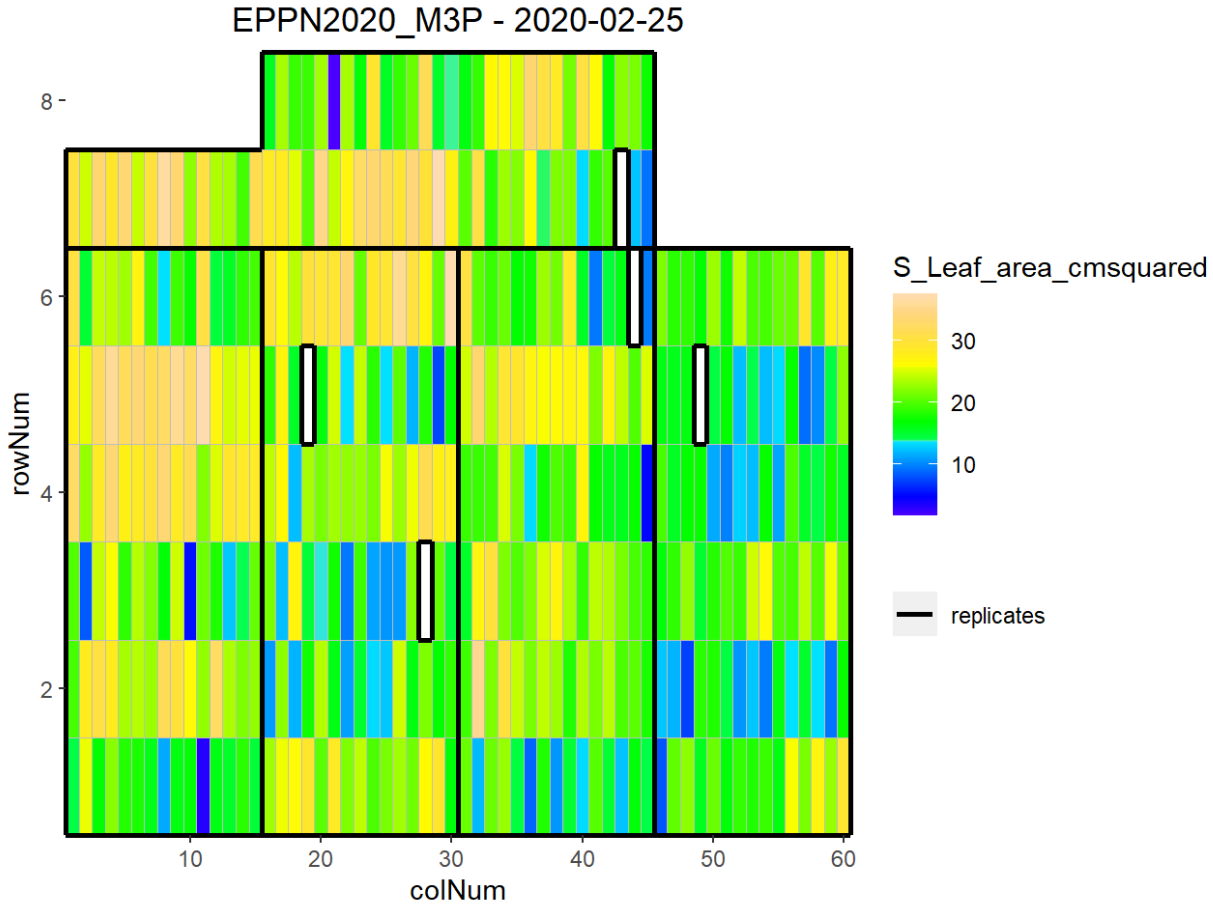

EPPN2020_M3P - 2020-02-20
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
25
20
15
10
5

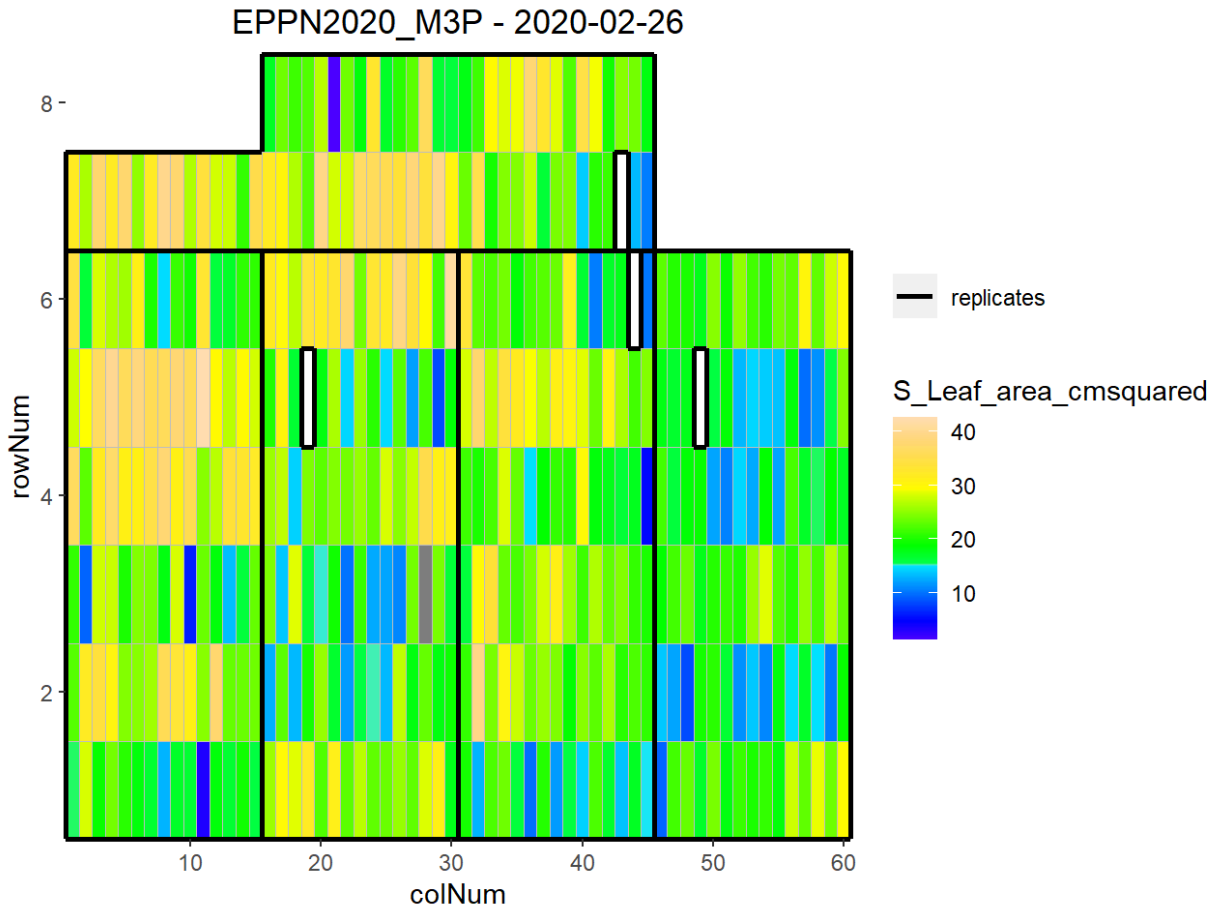

EPPN2020_M3P - 2020-02-21
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
25
20
15
10
5

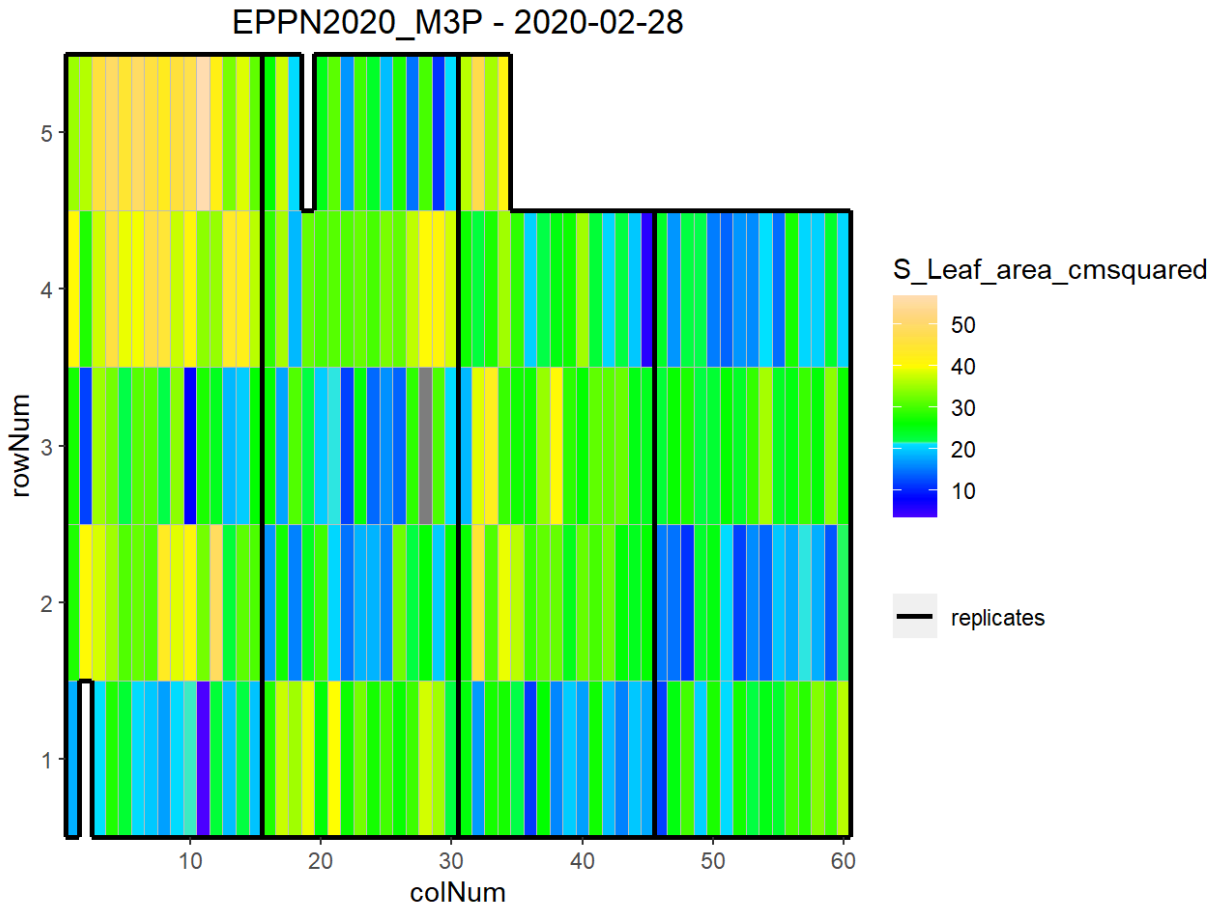

EPPN2020_M3P - 2020-02-22
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Leaf_area_cmsquared
20
10
replicates

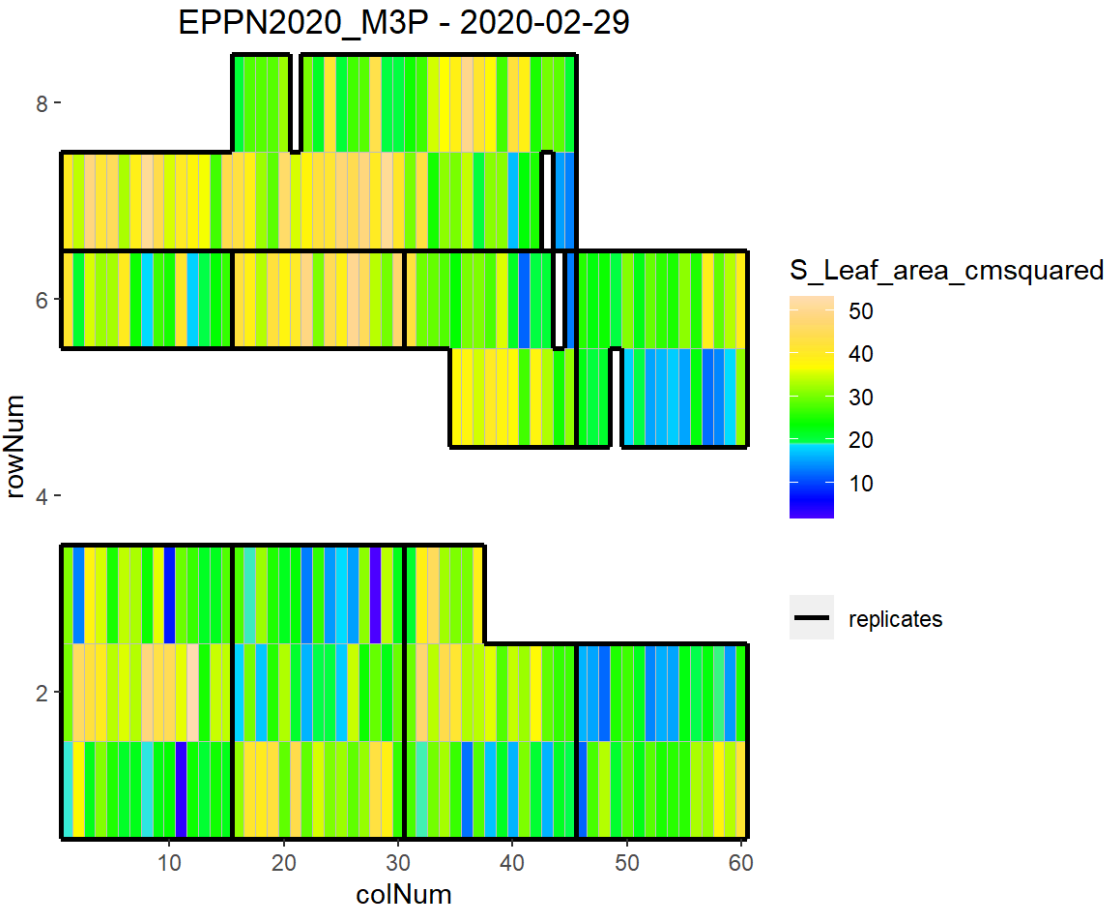

EPPN2020_M3P - 2020-02-23
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Leaf_area_cmsquared
24
21
18
15
replicates

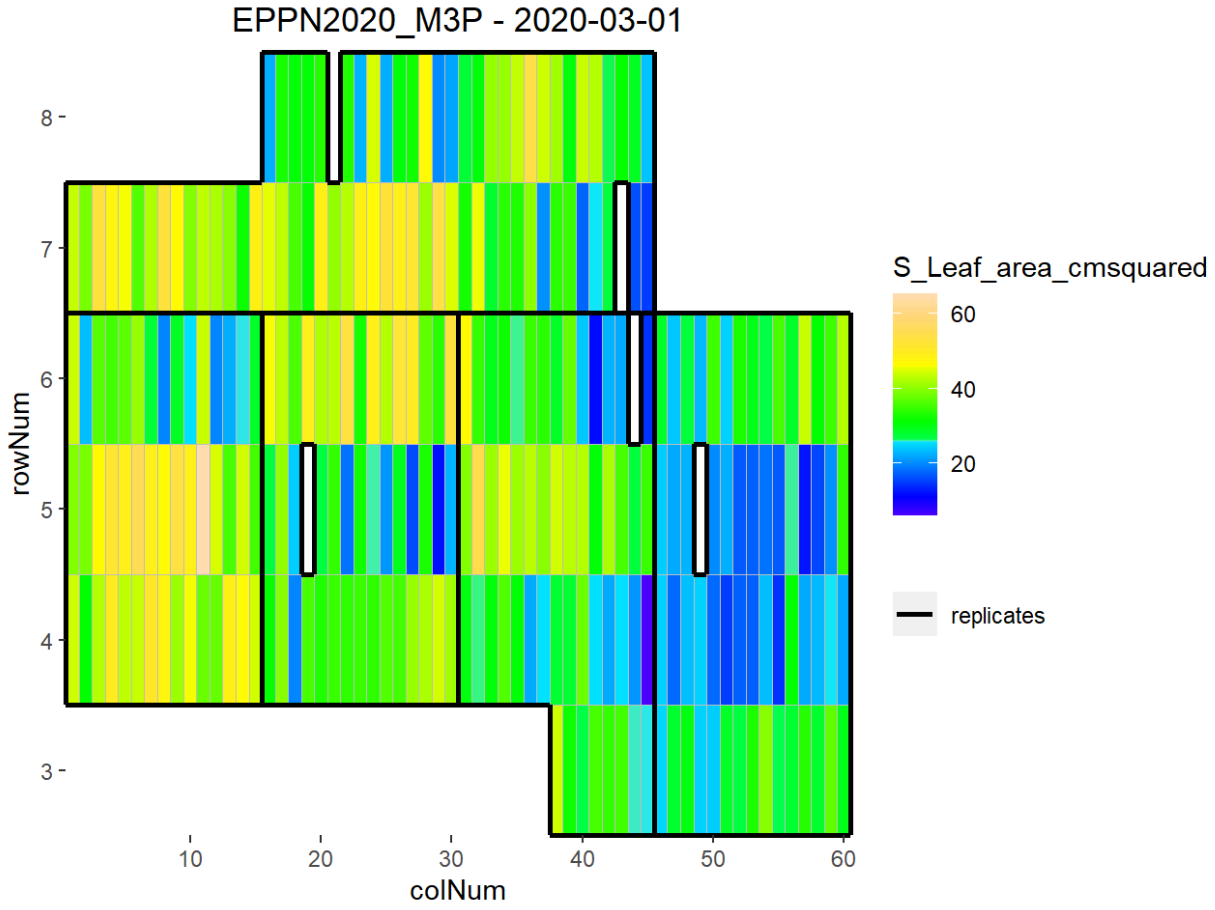

EPPN2020_M3P - 2020-02-24
rowNum
colNum
S_Leaf_area_cmsquared
30
20
10
replicates

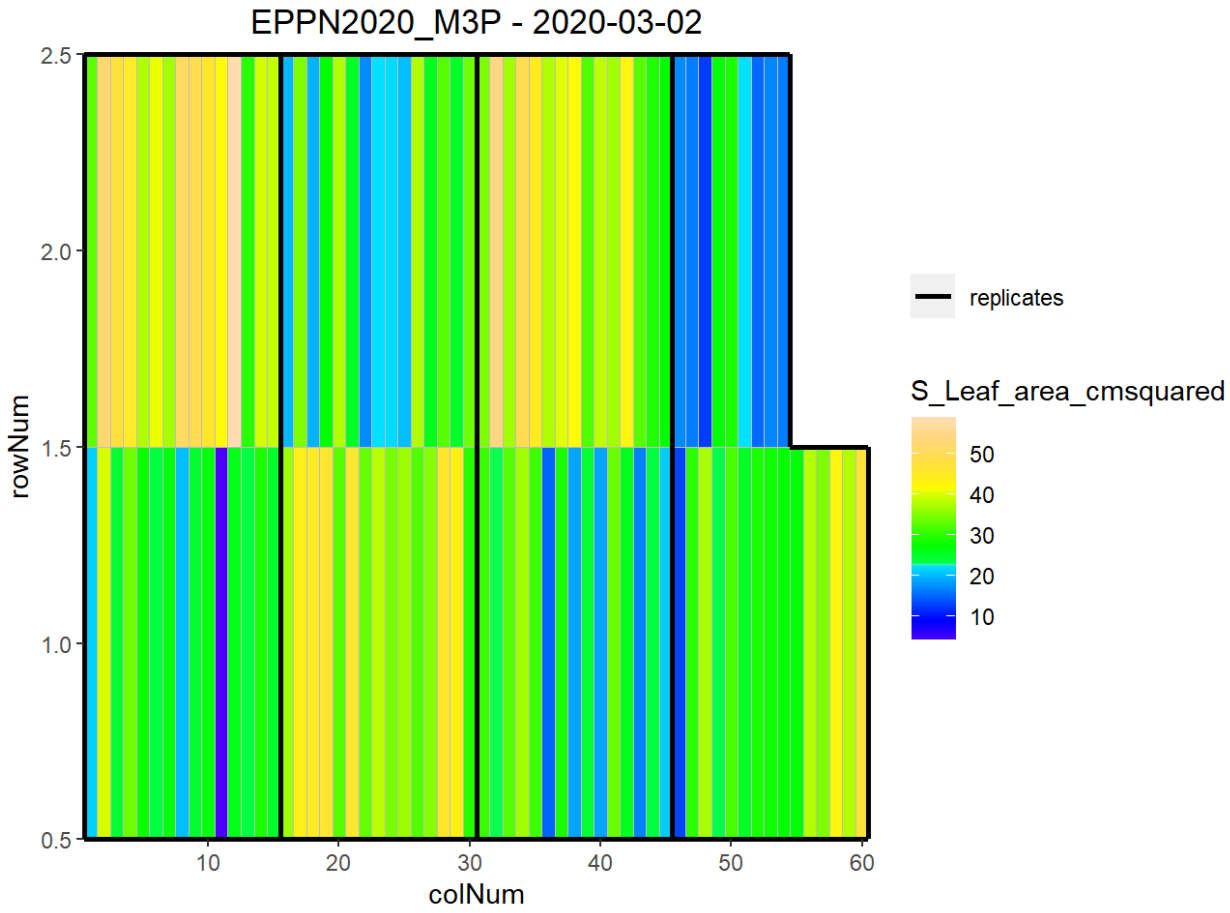

EPPN2020_M3P - 2020-02-25
rowNum
colNum
S_Leaf_area_cmsquared
30
20
10
replicates

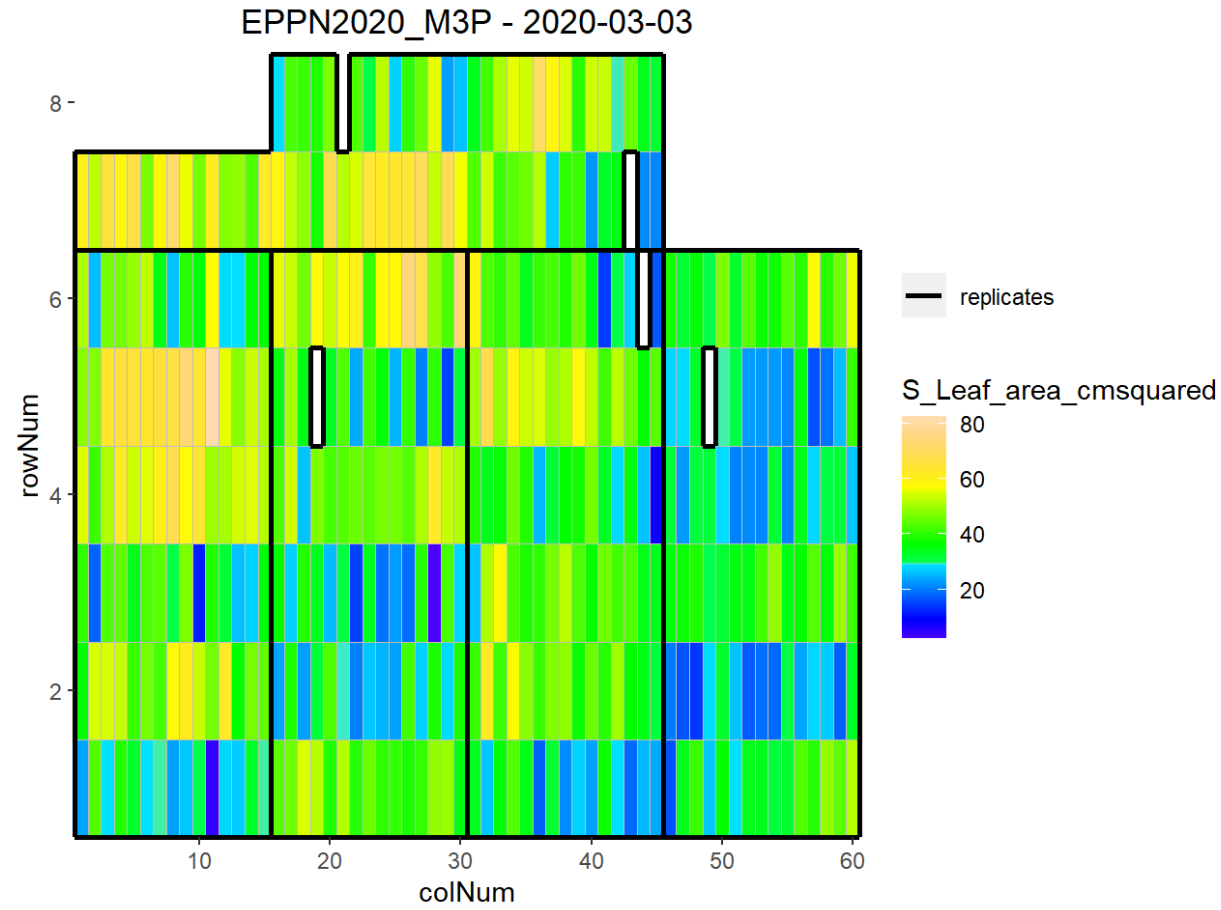

EPPN2020_M3P - 2020-02-26
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
40
30
20
10

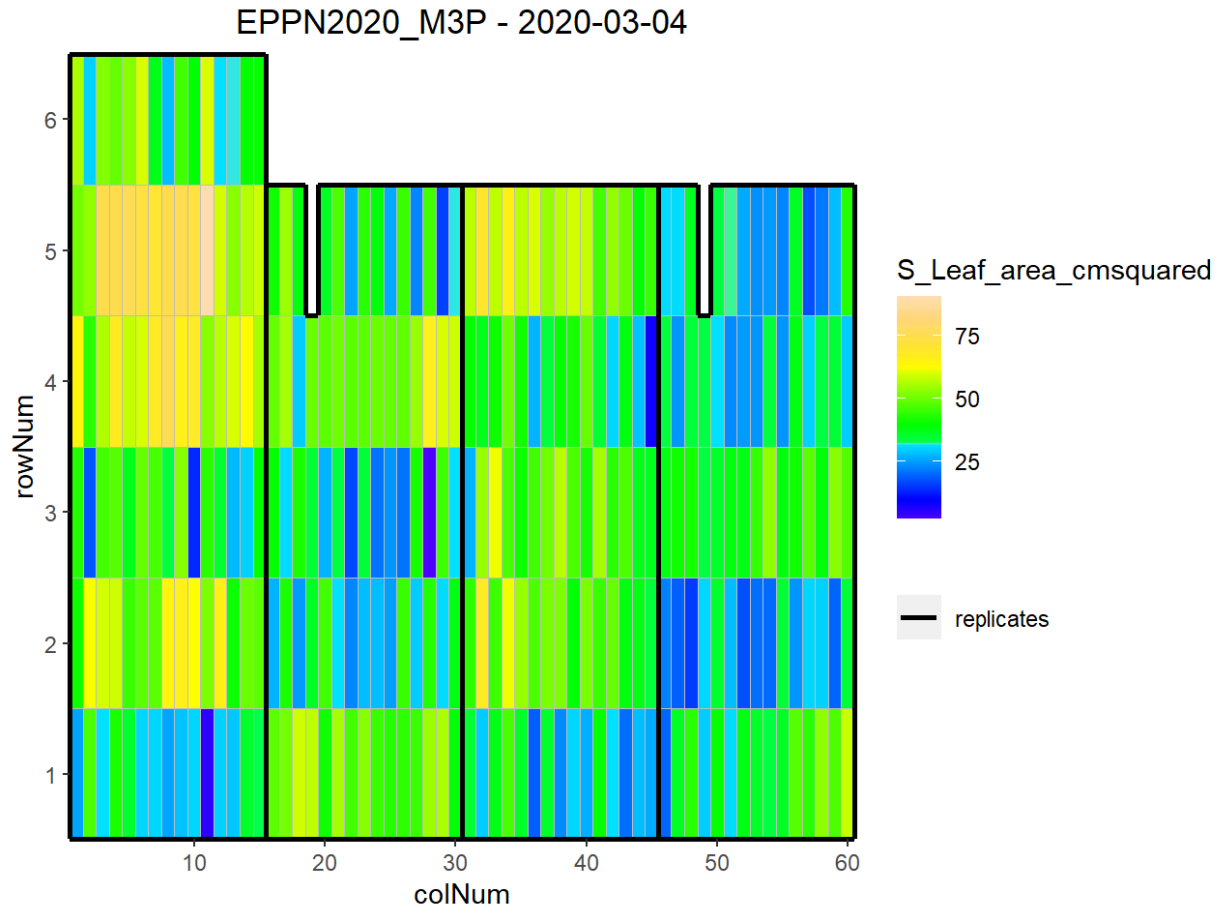

EPPN2020_M3P - 2020-02-28
rowNum
colNum
5
4
3
2
1
10
20
30
40
50
60
S_Leaf_area_cmsquared
50
40
30
20
10
replicates

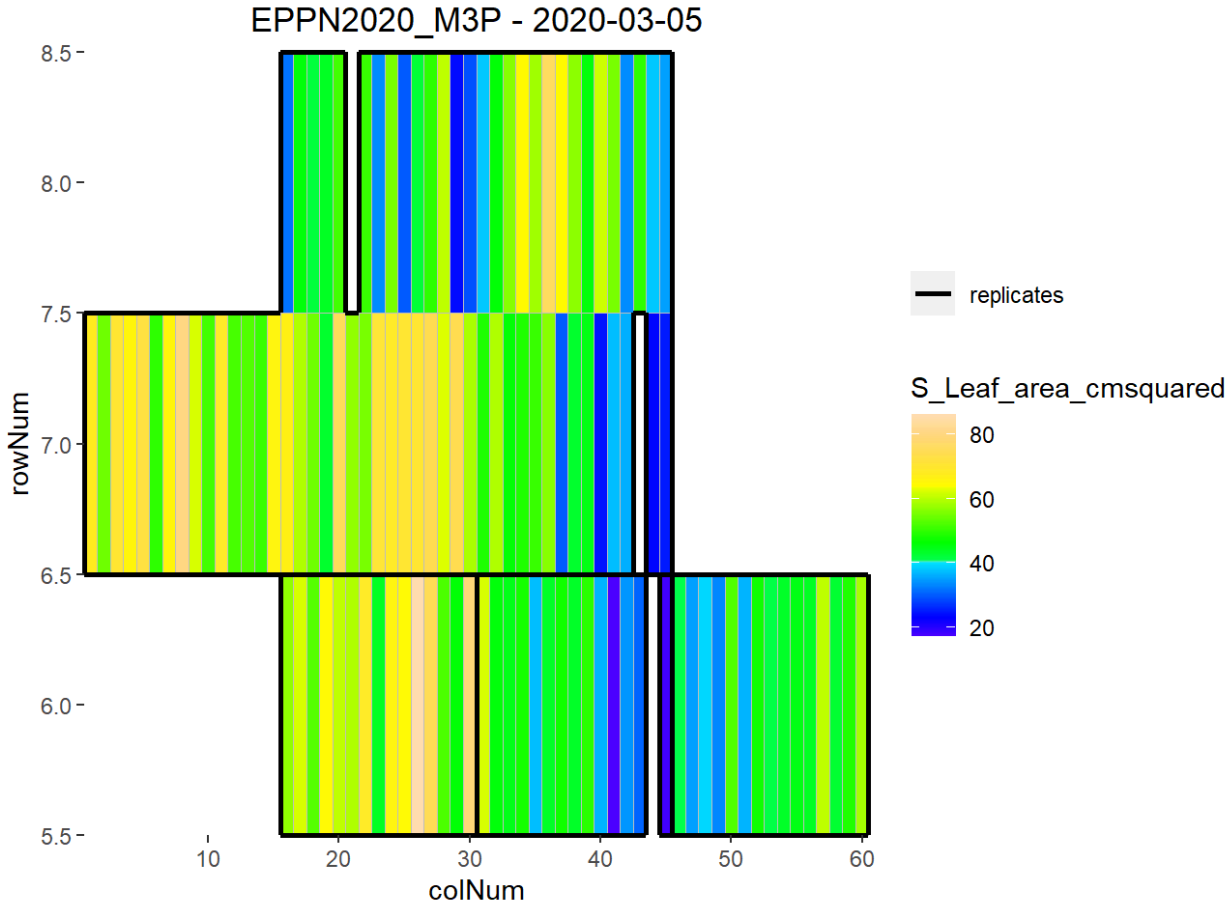

EPPN2020_M3P - 2020-02-29
rowNum
colNum
S_Leaf_area_cmsquared
50
40
30
20
10
replicates

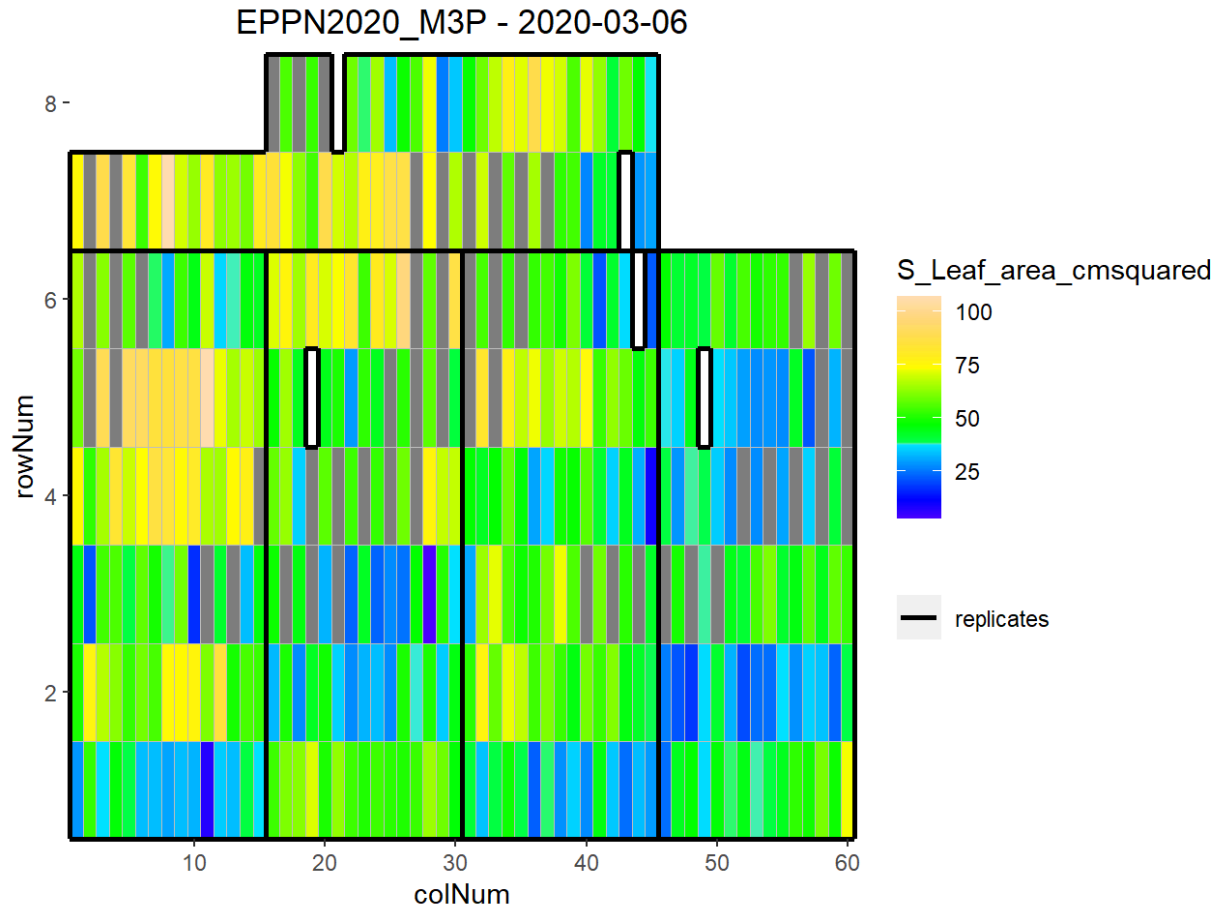

EPPN2020_M3P - 2020-03-01
rowNum
colNum
S_Leaf_area_cmsquared
60
40
20
replicates

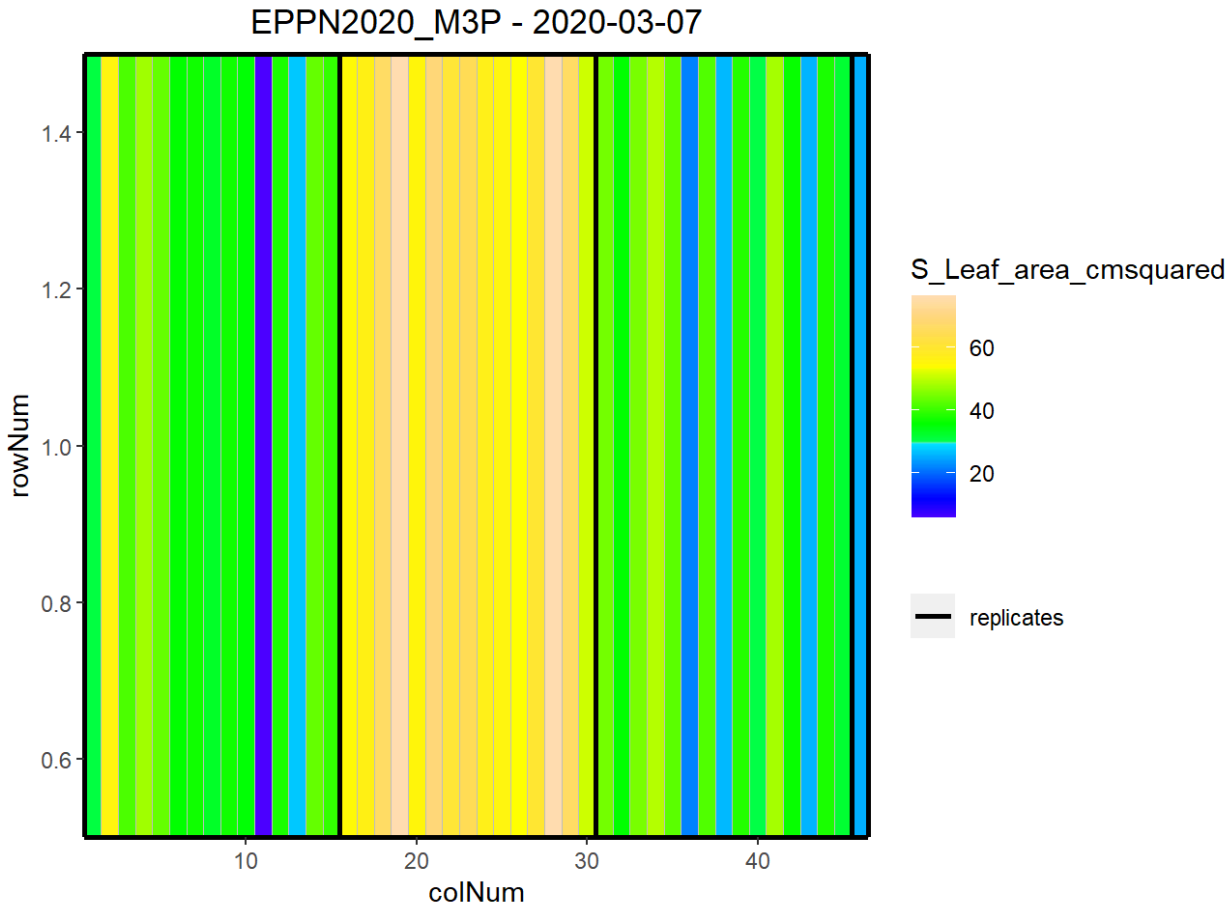

EPPN2020_M3P - 2020-03-02
rowNum
colNum
2.5
2.0
1.5
1.0
0.5
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
50
40
30
20
10

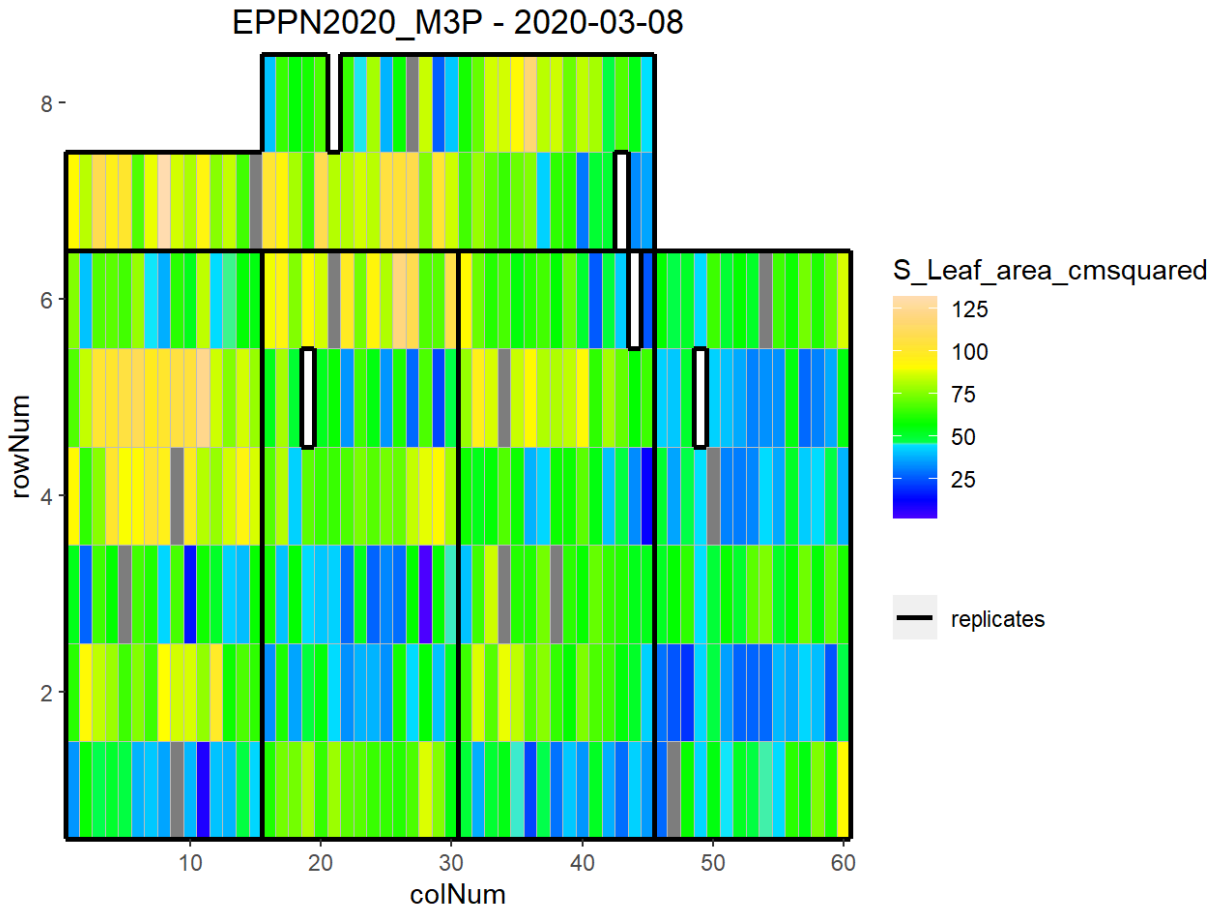

EPPN2020_M3P - 2020-03-03
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
80
60
40
20

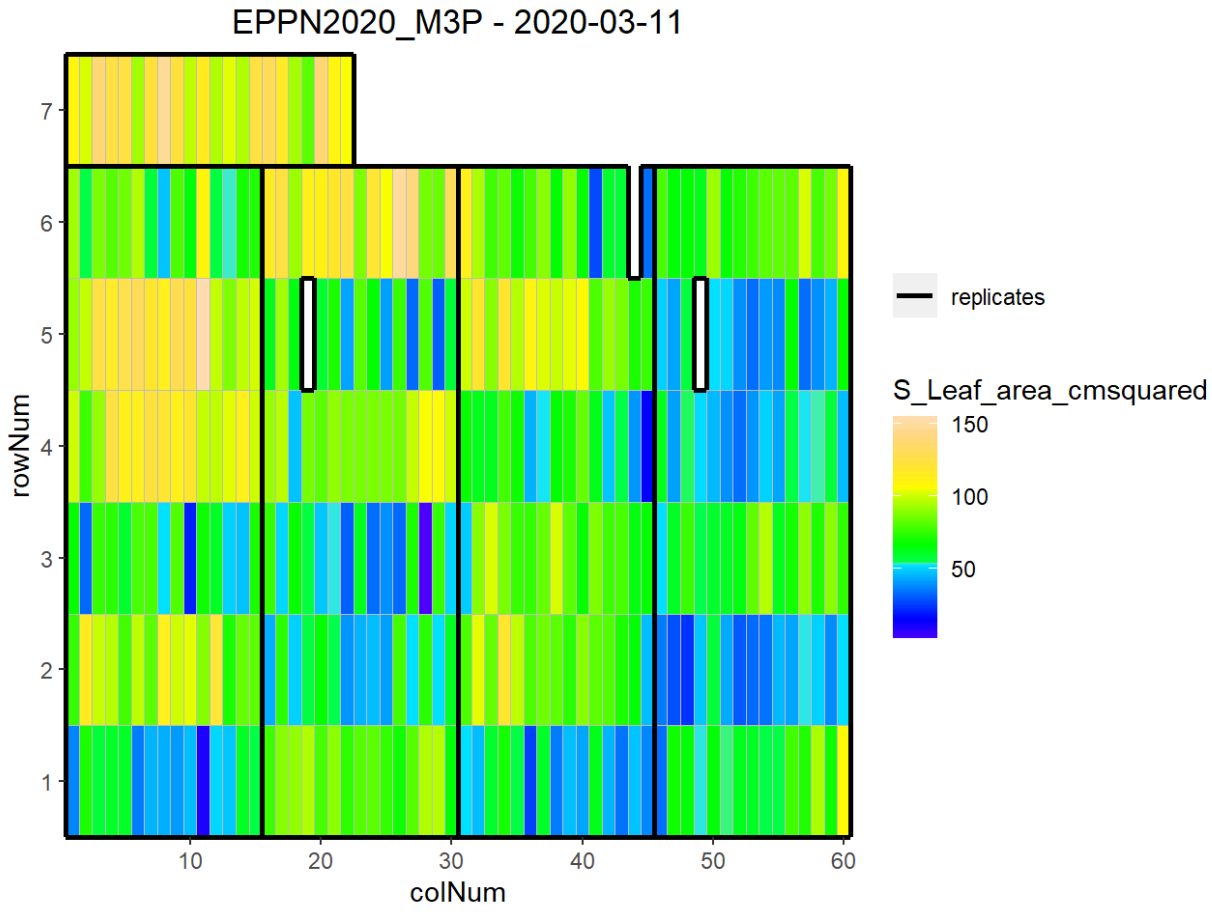

EPPN2020_M3P - 2020-03-04
rowNum
colNum
S_Leaf_area_cmsquared
75
50
25
replicates

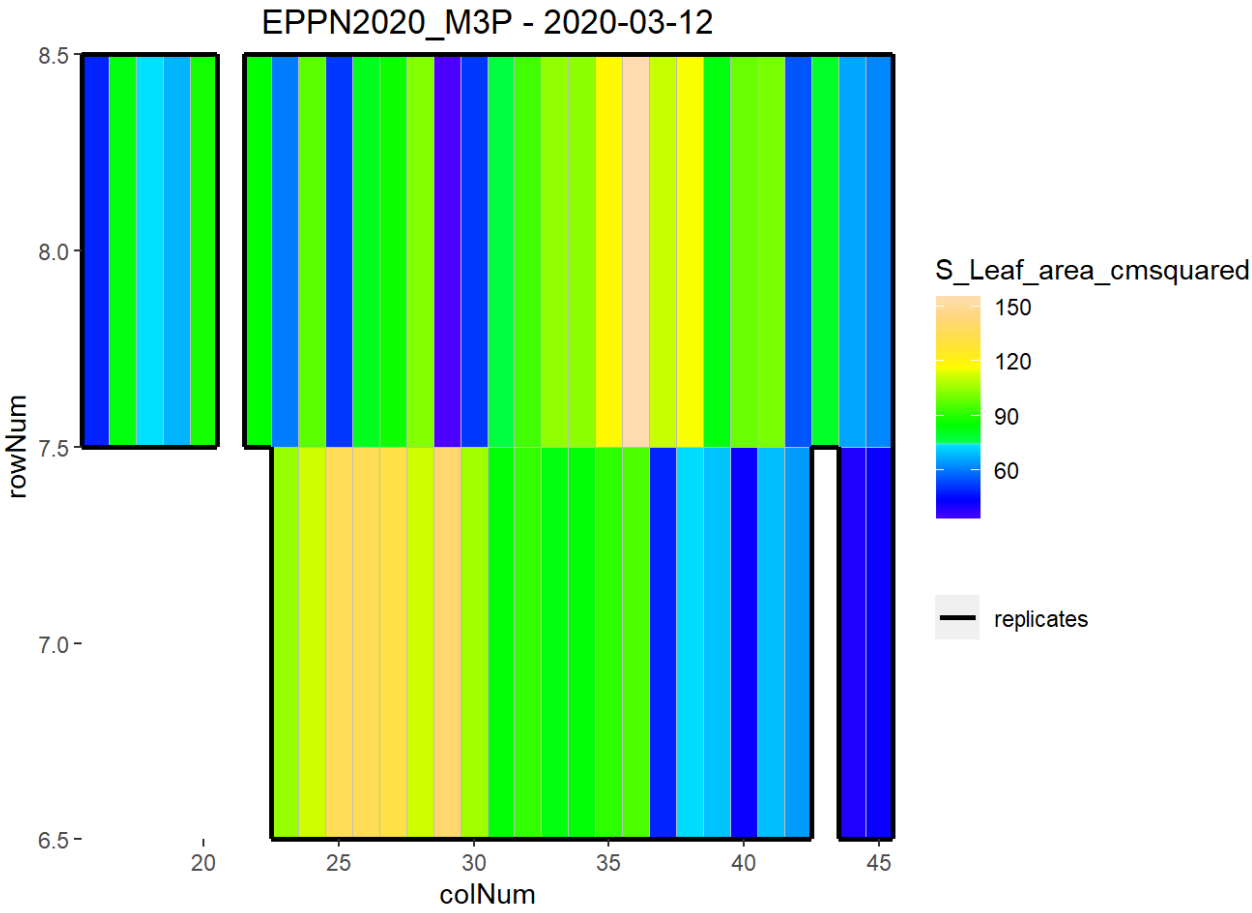

EPPN2020_M3P - 2020-03-05
rowNum
colNum
replicates
S_Leaf_area_cmsquared
80
60
40
20

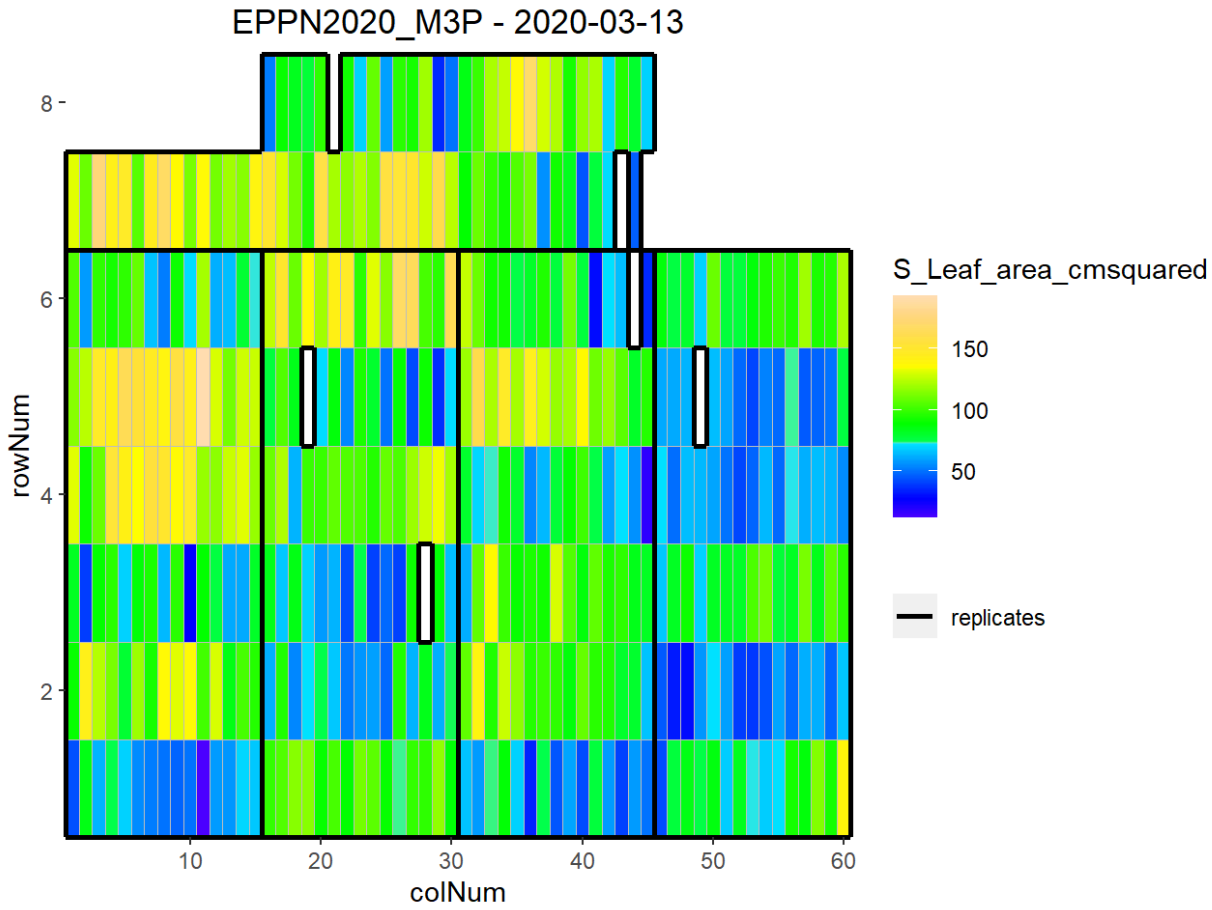

EPPN2020_M3P - 2020-03-06
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Leaf_area_cmsquared
100
75
50
25
replicates

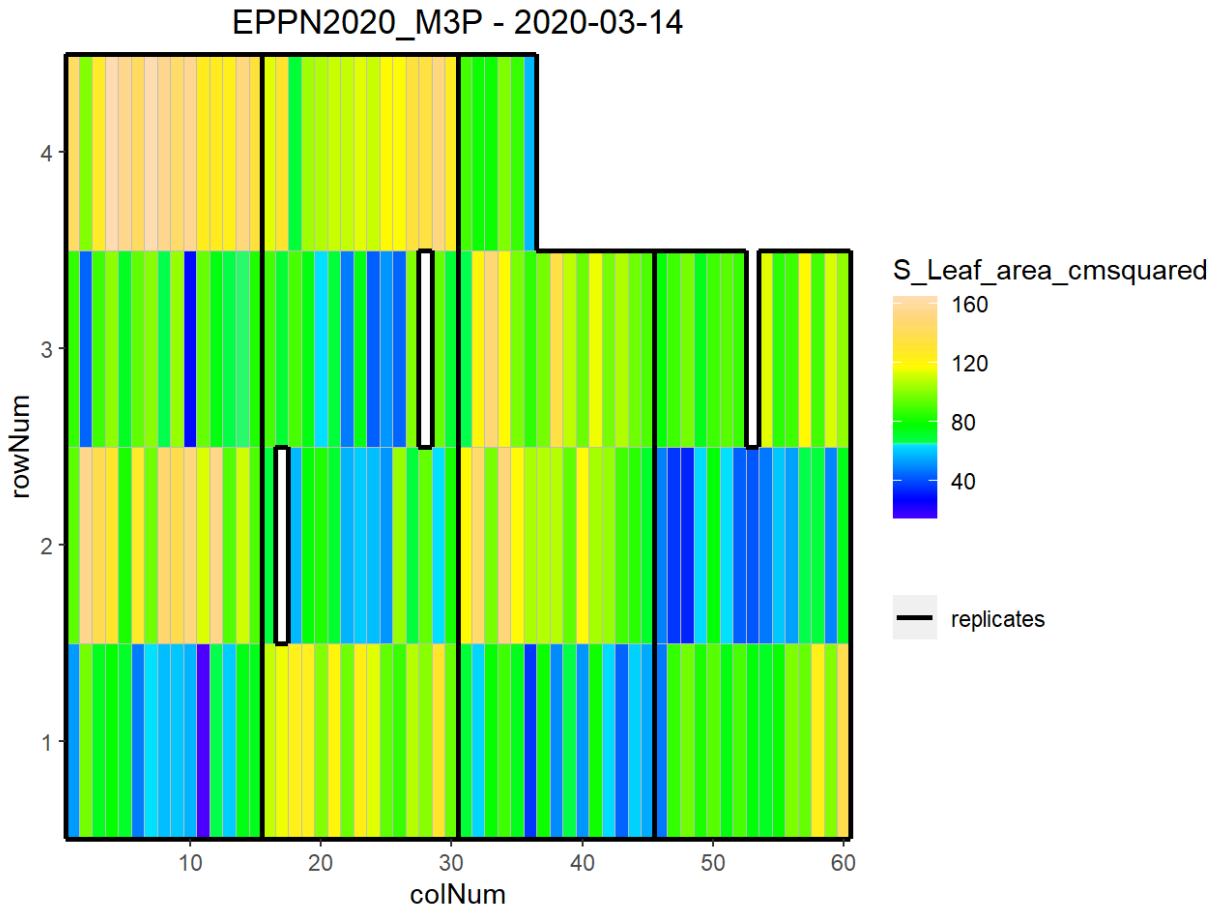

EPPN2020_M3P - 2020-03-07
rowNum
colNum
1.4
1.2
1.0
0.8
0.6
10
20
30
40
S_Leaf_area_cmsquared
60
40
20
replicates

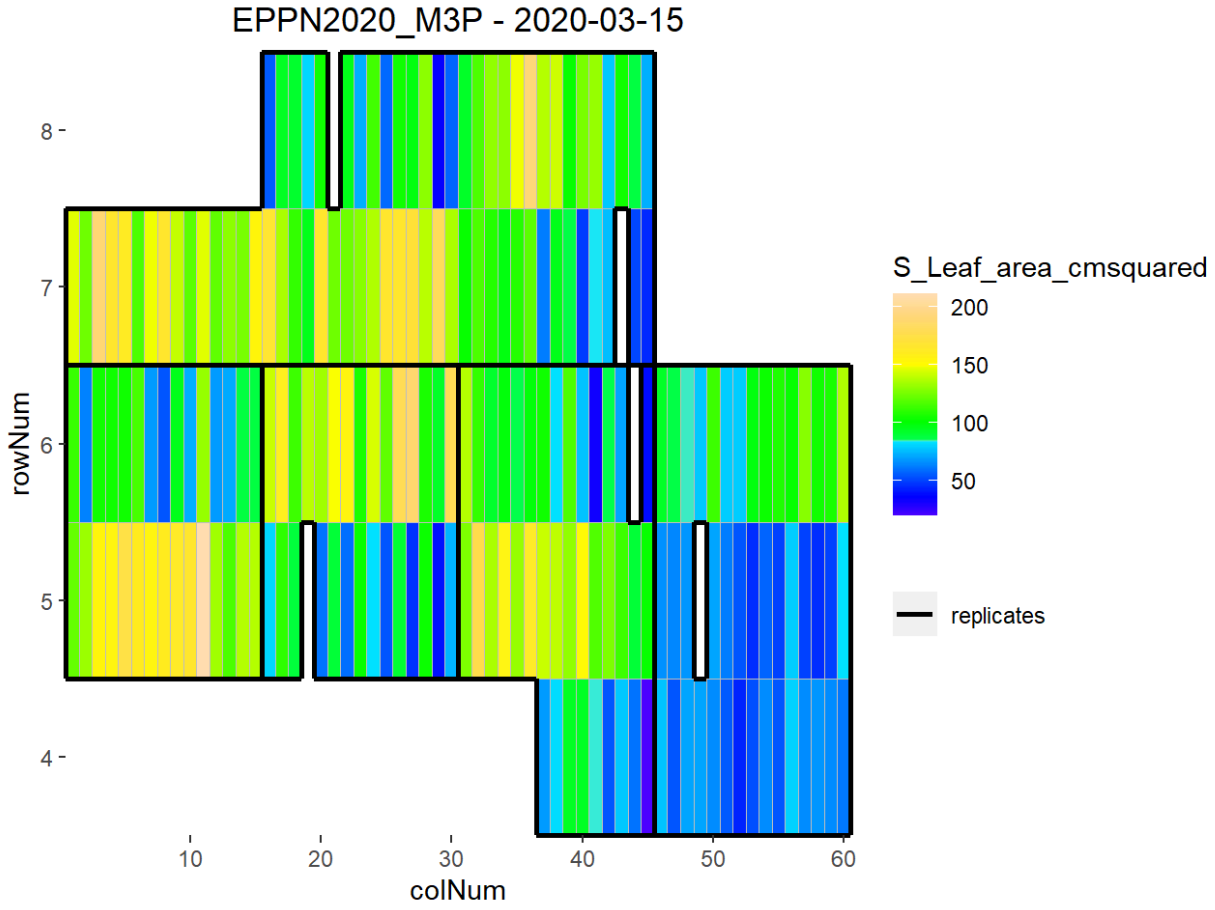

EPPN2020_M3P - 2020-03-08
rowNum
colNum
S_Leaf_area_cmsquared
125
100
75
50
25
replicates

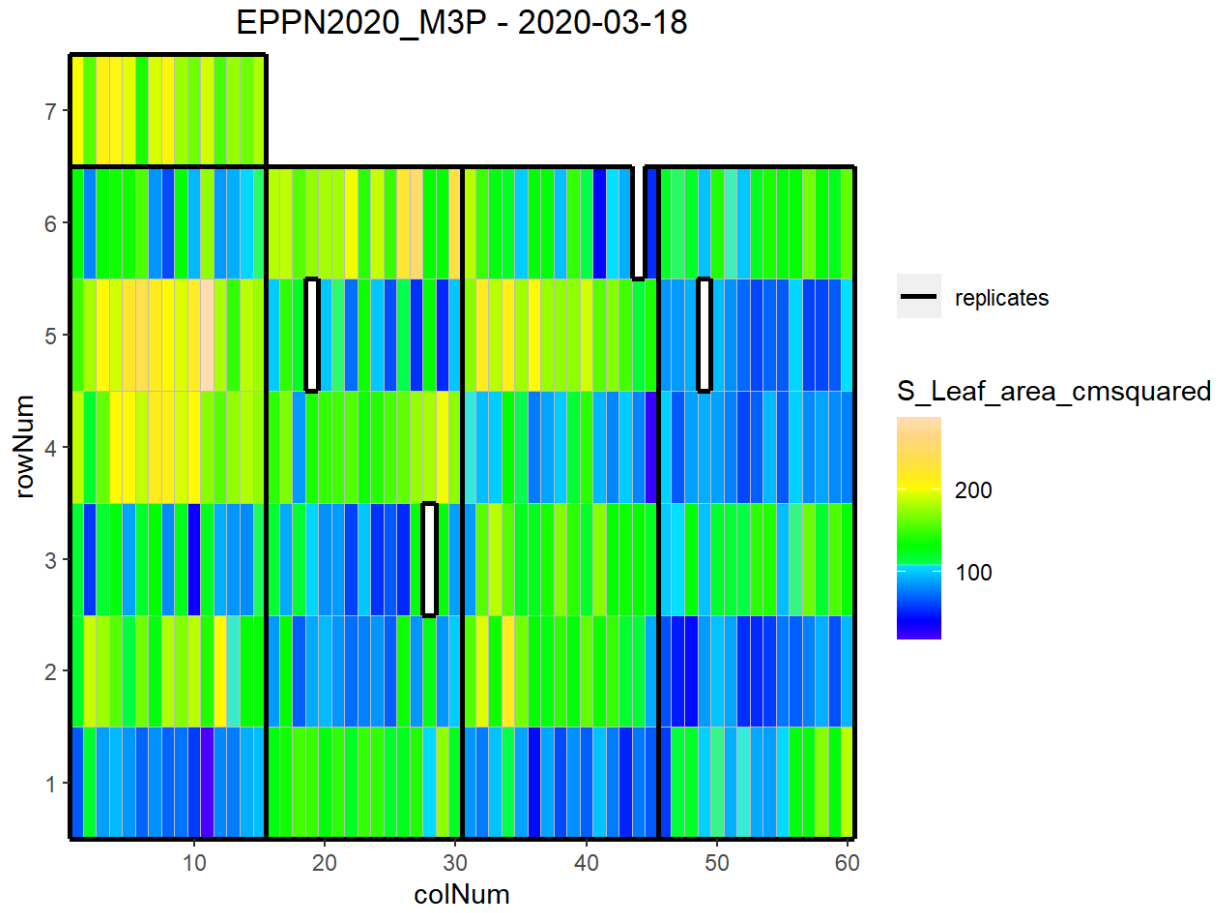

EPPN2020_M3P - 2020-03-11
rowNum
colNum
replicates
S_Leaf_area_cmsquared
150
100
50

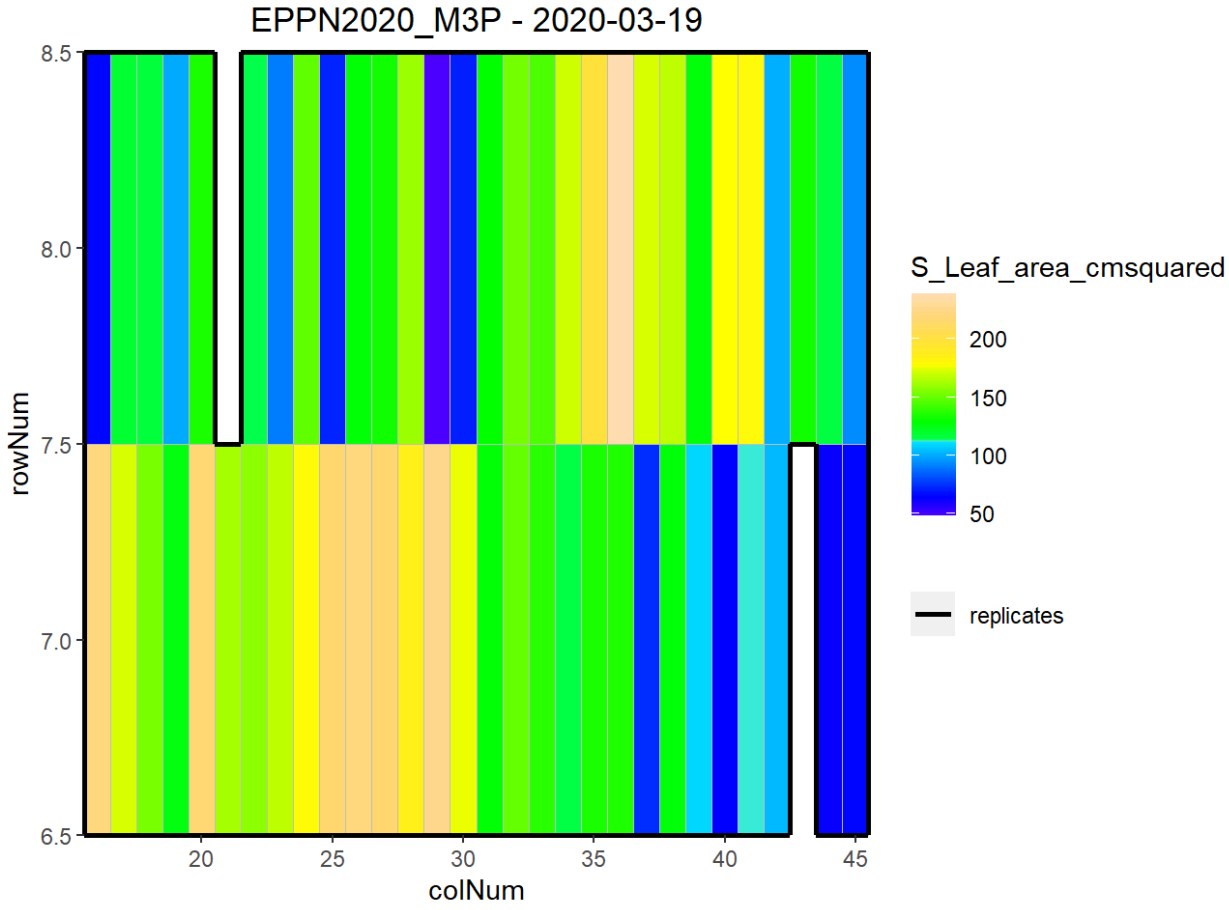

EPPN2020_M3P - 2020-03-12
rowNum
colNum
8.5
8.0
7.5
7.0
6.5
20
25
30
35
40
45
S_Leaf_area_cmsquared
150
120
90
60
replicates

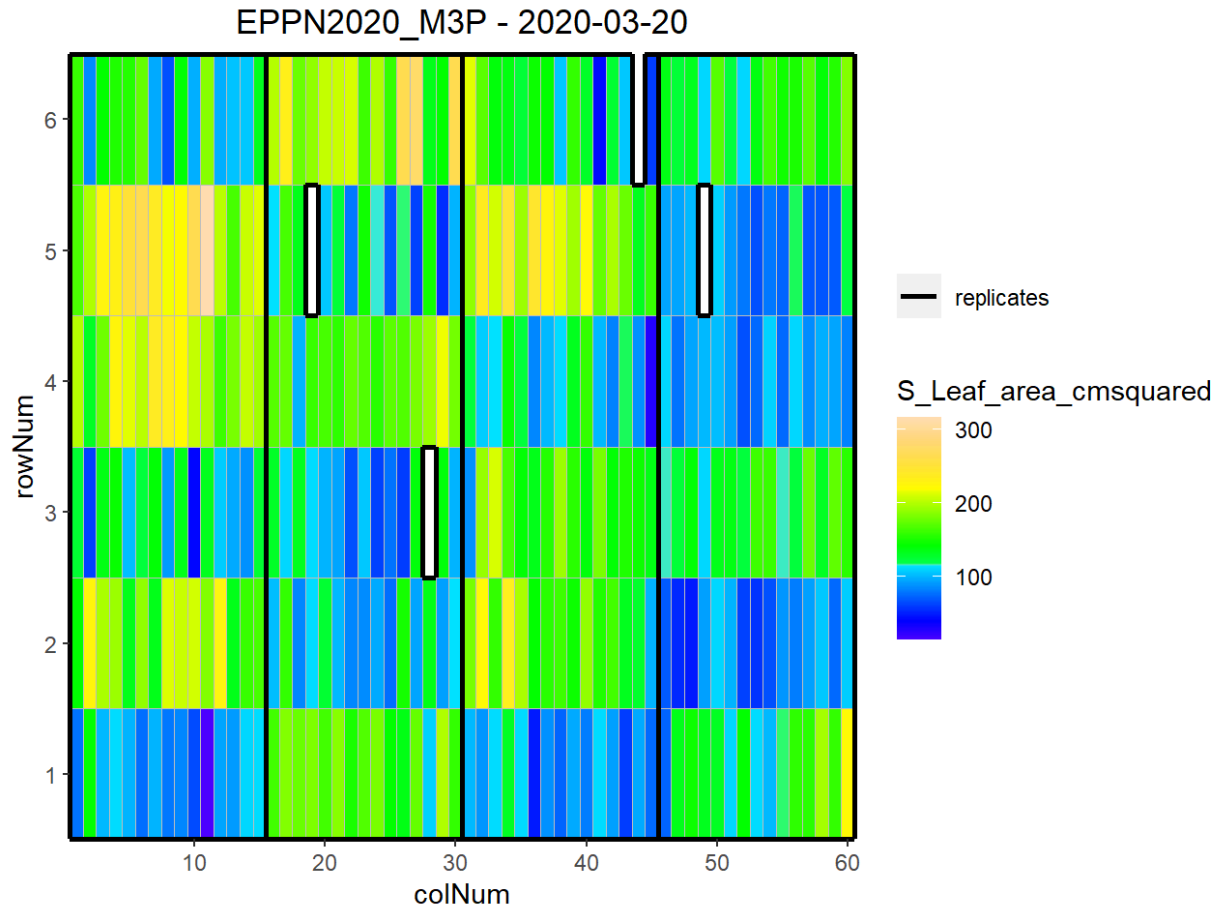

EPPN2020_M3P - 2020-03-13
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Leaf_area_cmsquared
150
100
50
replicates

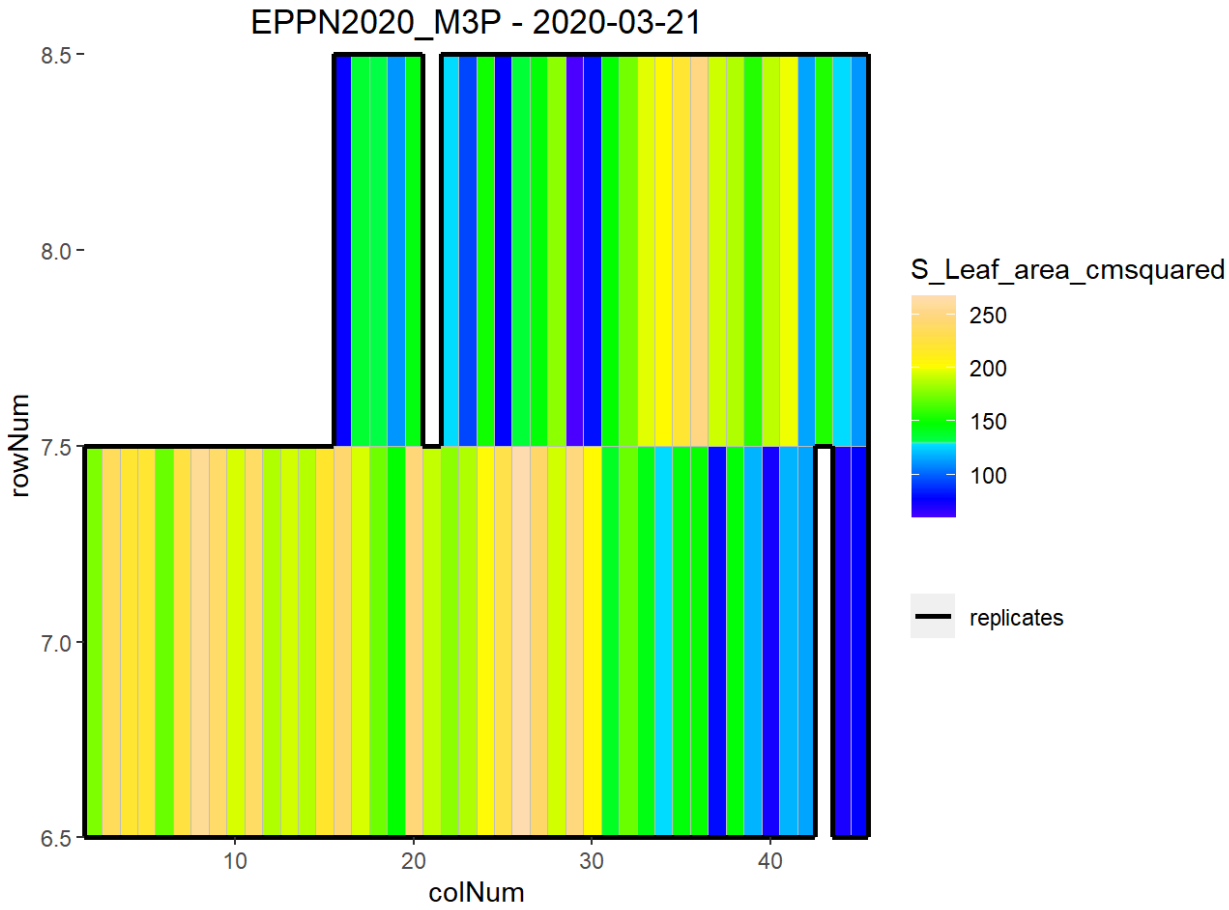

EPPN2020_M3P - 2020-03-14
rowNum
colNum
S_Leaf_area_cmsquared
160
120
80
40
replicates

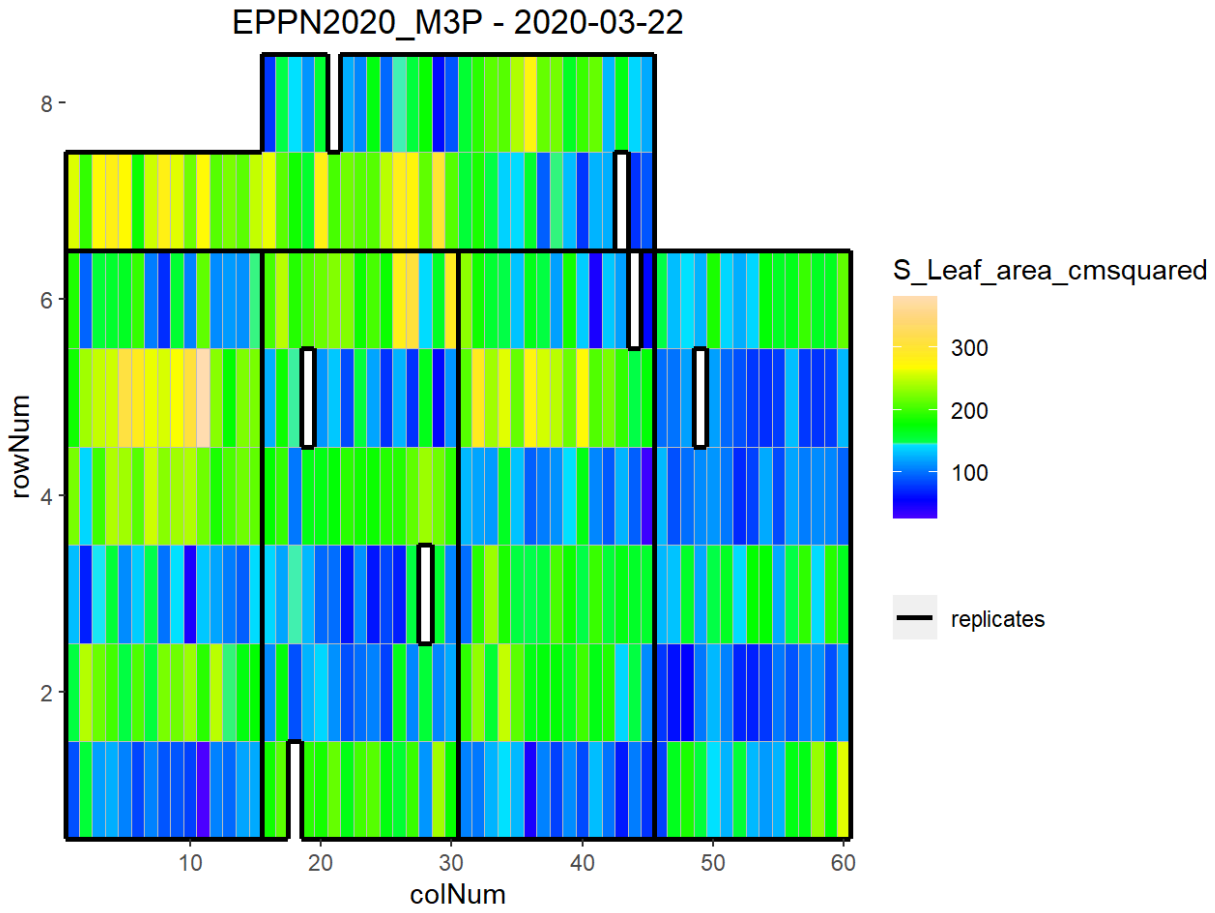

EPPN2020_M3P - 2020-03-15
rowNum
colNum
S_Leaf_area_cmsquared
200
150
100
50
replicates

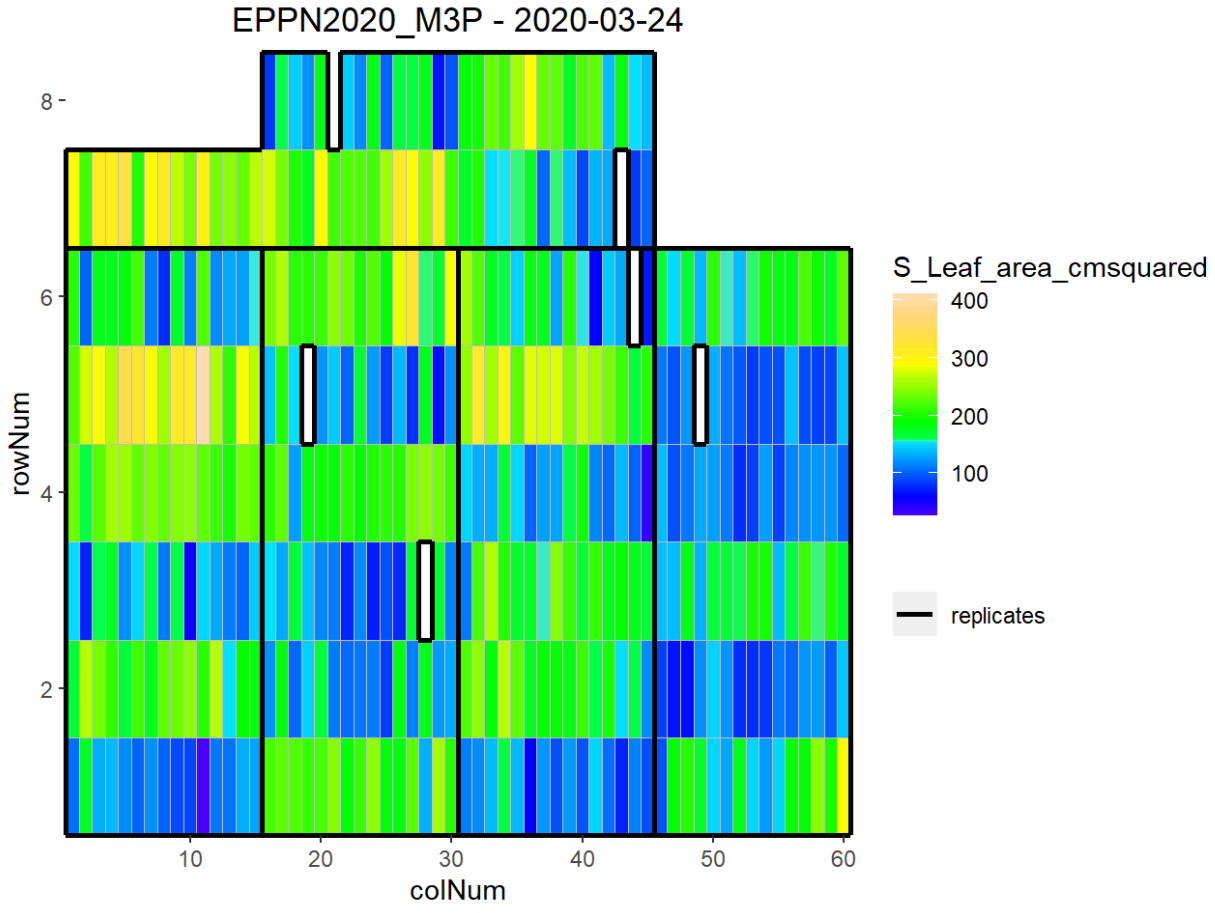

EPPN2020_M3P - 2020-03-18
rowNum
colNum
replicates
S_Leaf_area_cmsquared
200
100

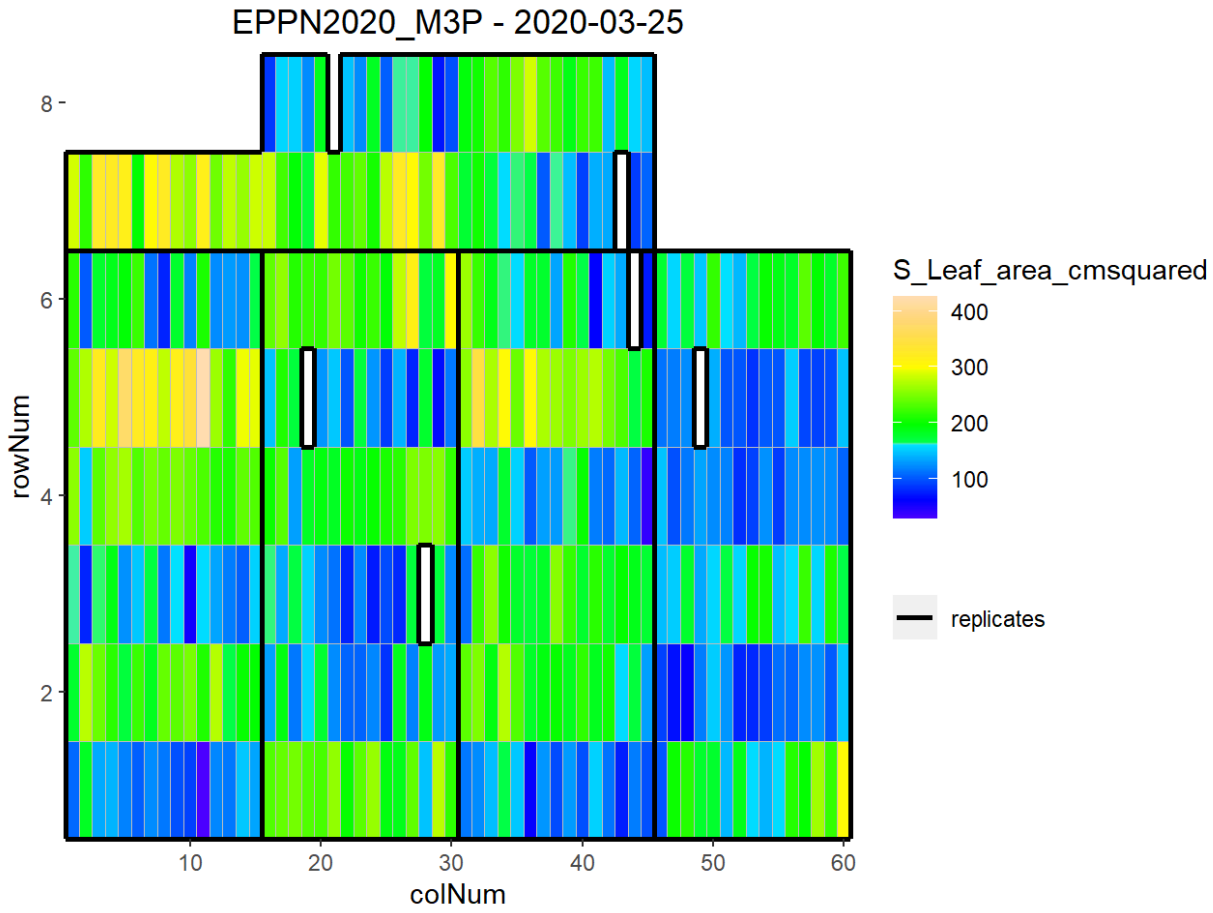

EPPN2020_M3P - 2020-03-19
rowNum
colNum
S_Leaf_area_cmsquared
200
150
100
50
replicates

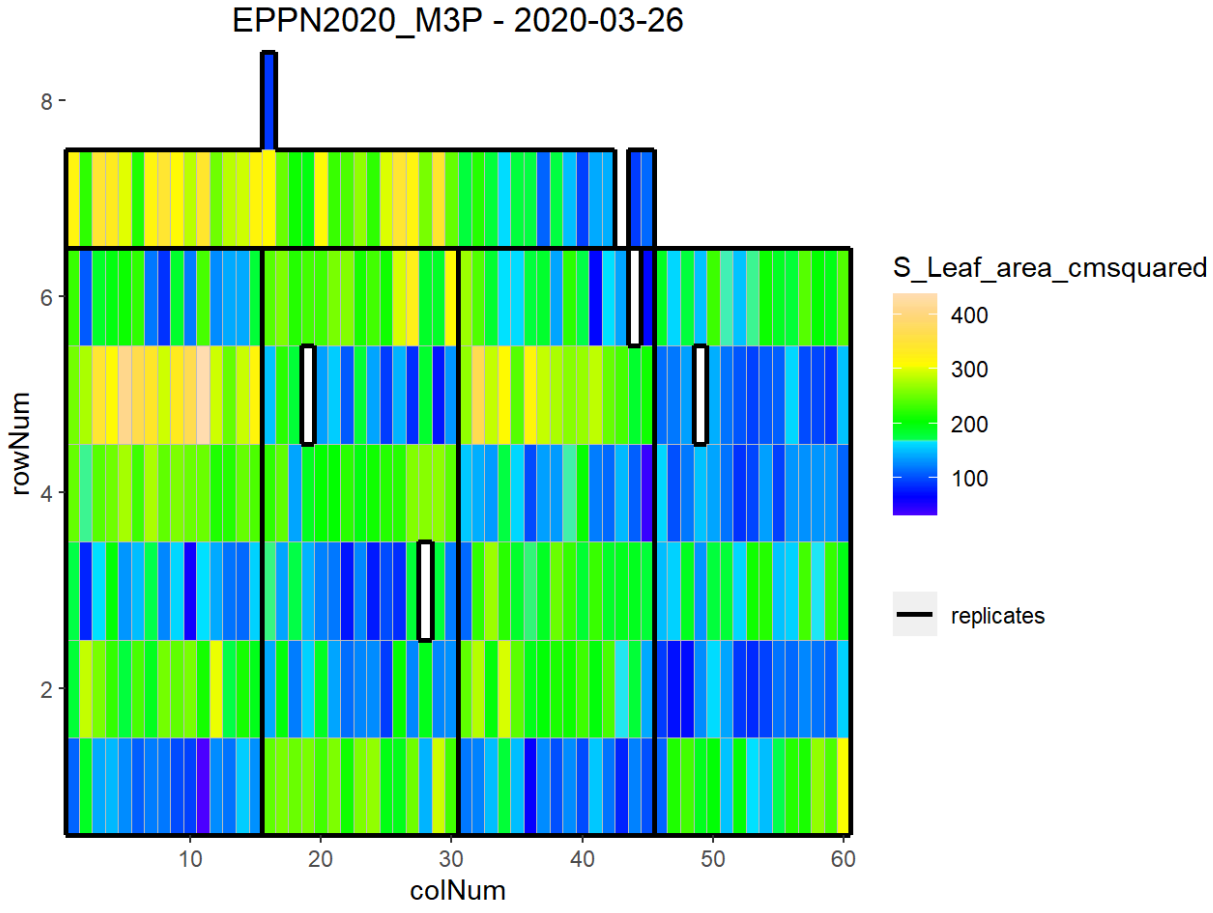

EPPN2020_M3P - 2020-03-20
rowNum
colNum
1
2
3
4
5
6
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
300
200
100

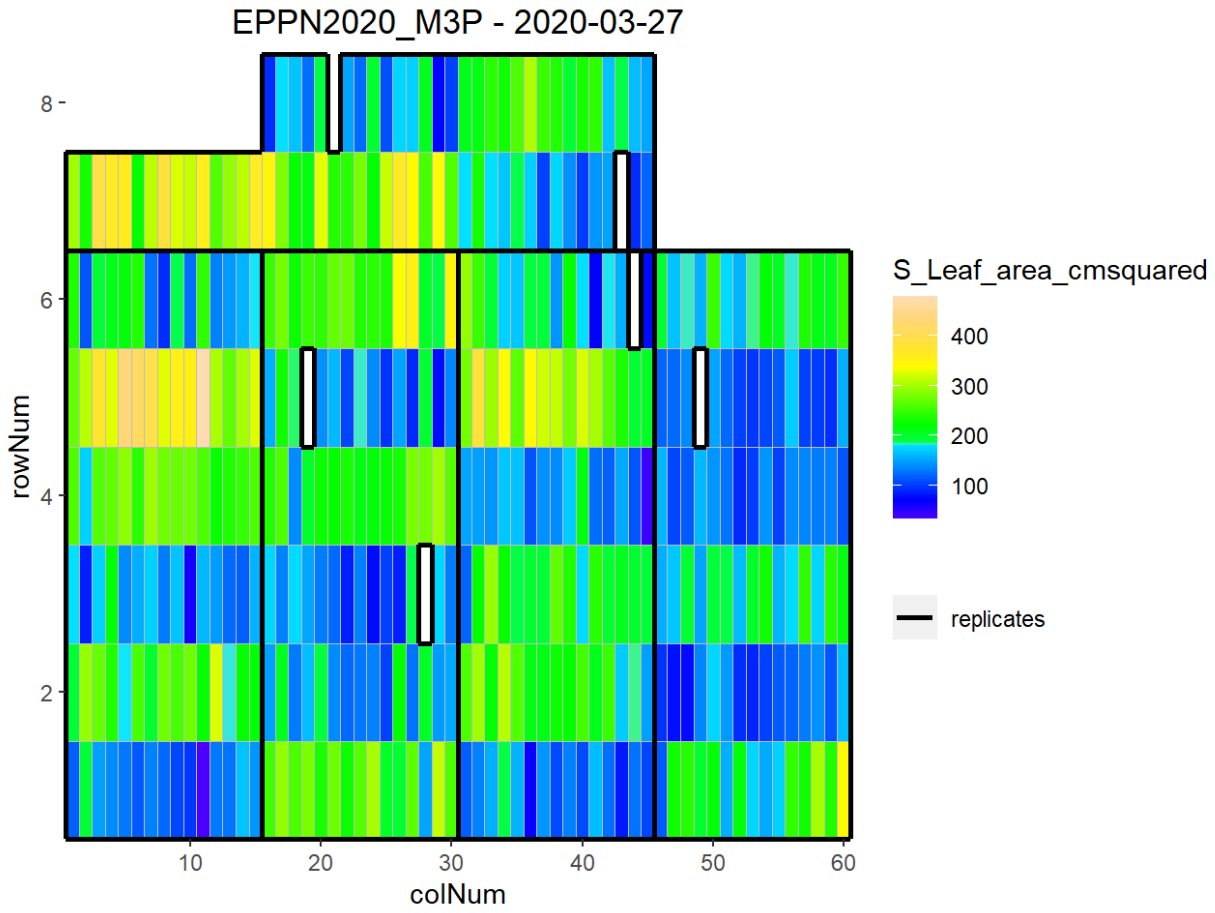

EPPN2020_M3P - 2020-03-21
rowNum
colNum
6.5
7.0
7.5
8.0
8.5
10
20
30
40
S_Leaf_area_cmsquared
250
200
150
100
replicates

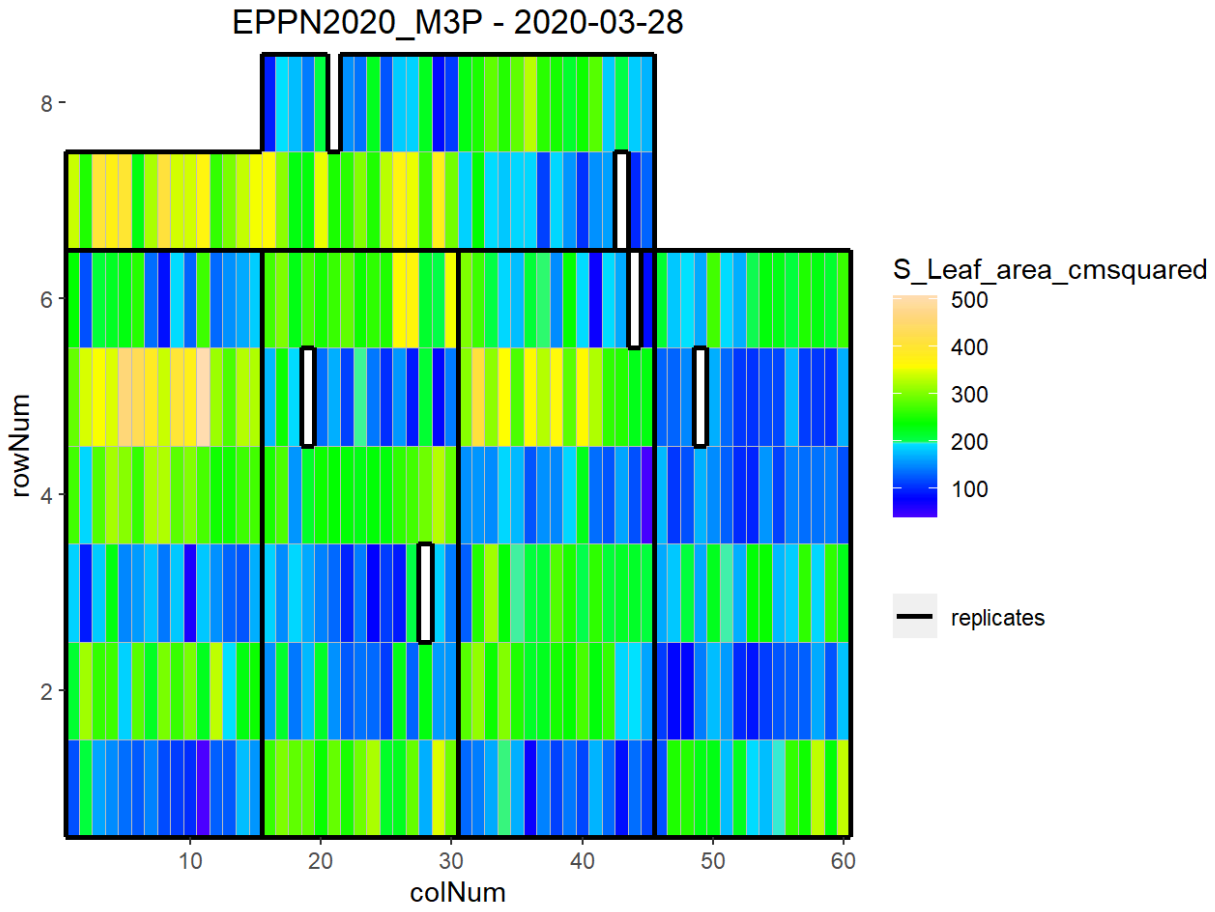

EPPN2020_M3P - 2020-03-22
rowNum
colNum
S_Leaf_area_cmsquared
300
200
100
replicates

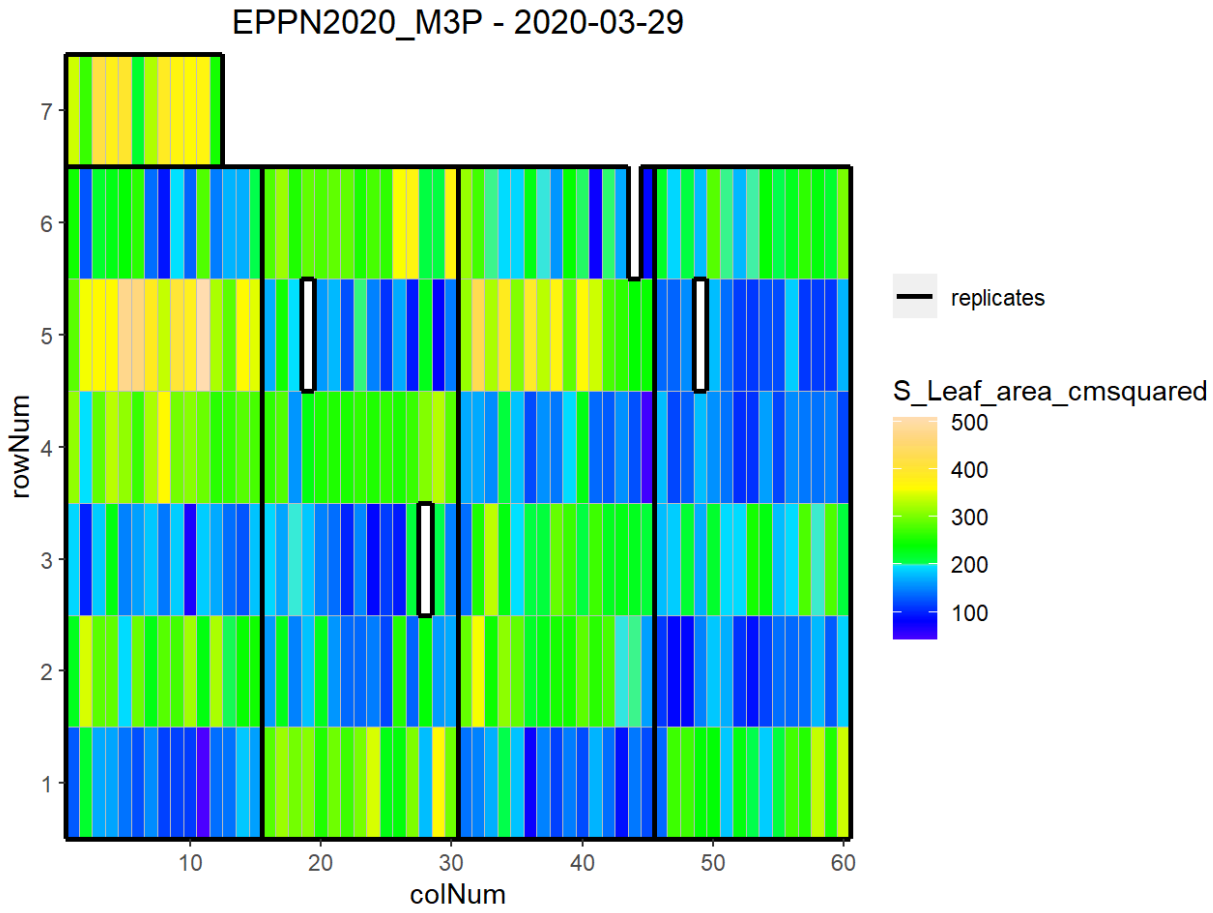

EPPN2020_M3P - 2020-03-24
rowNum
colNum
S_Leaf_area_cmsquared
400
300
200
100
replicates

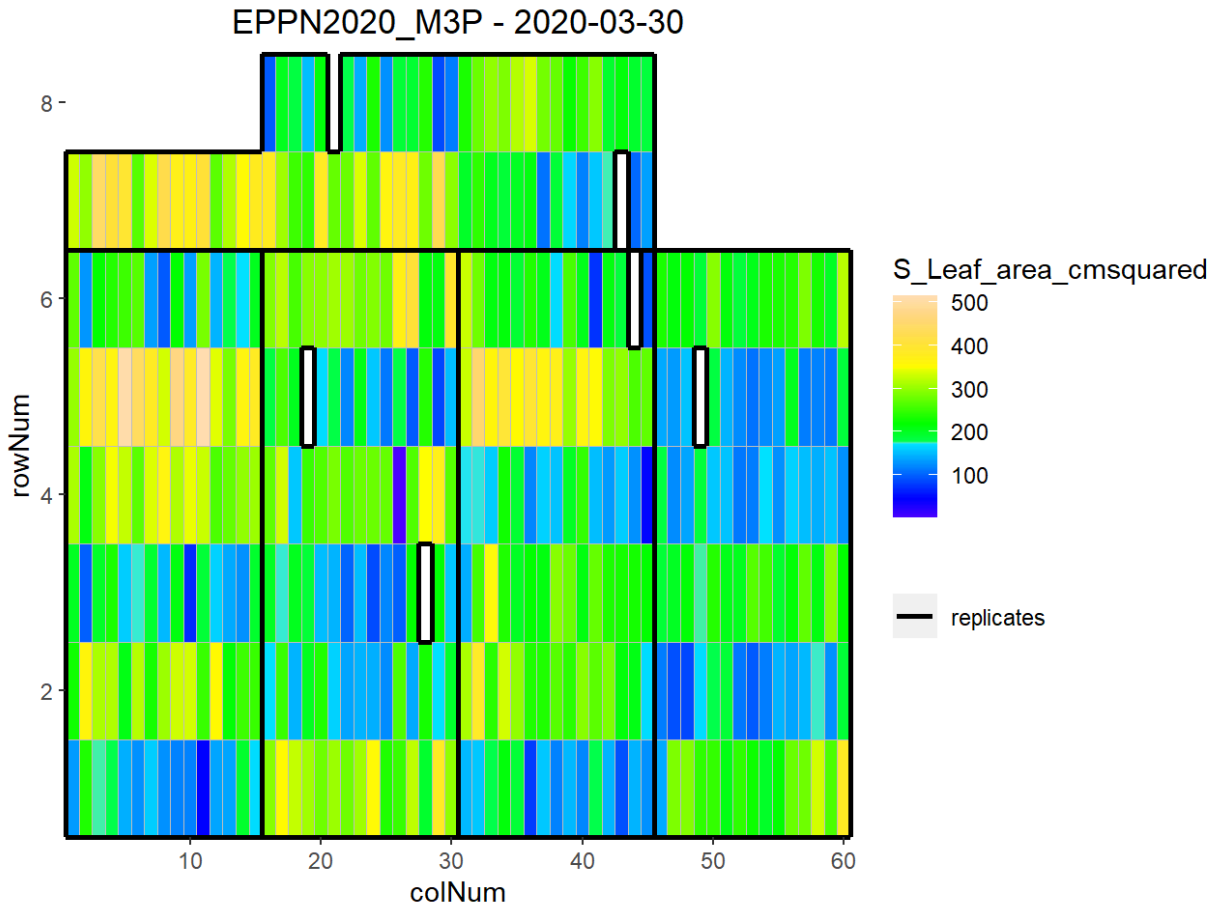

EPPN2020_M3P - 2020-03-25
rowNum
colNum
S_Leaf_area_cmsquared
400
300
200
100
replicates

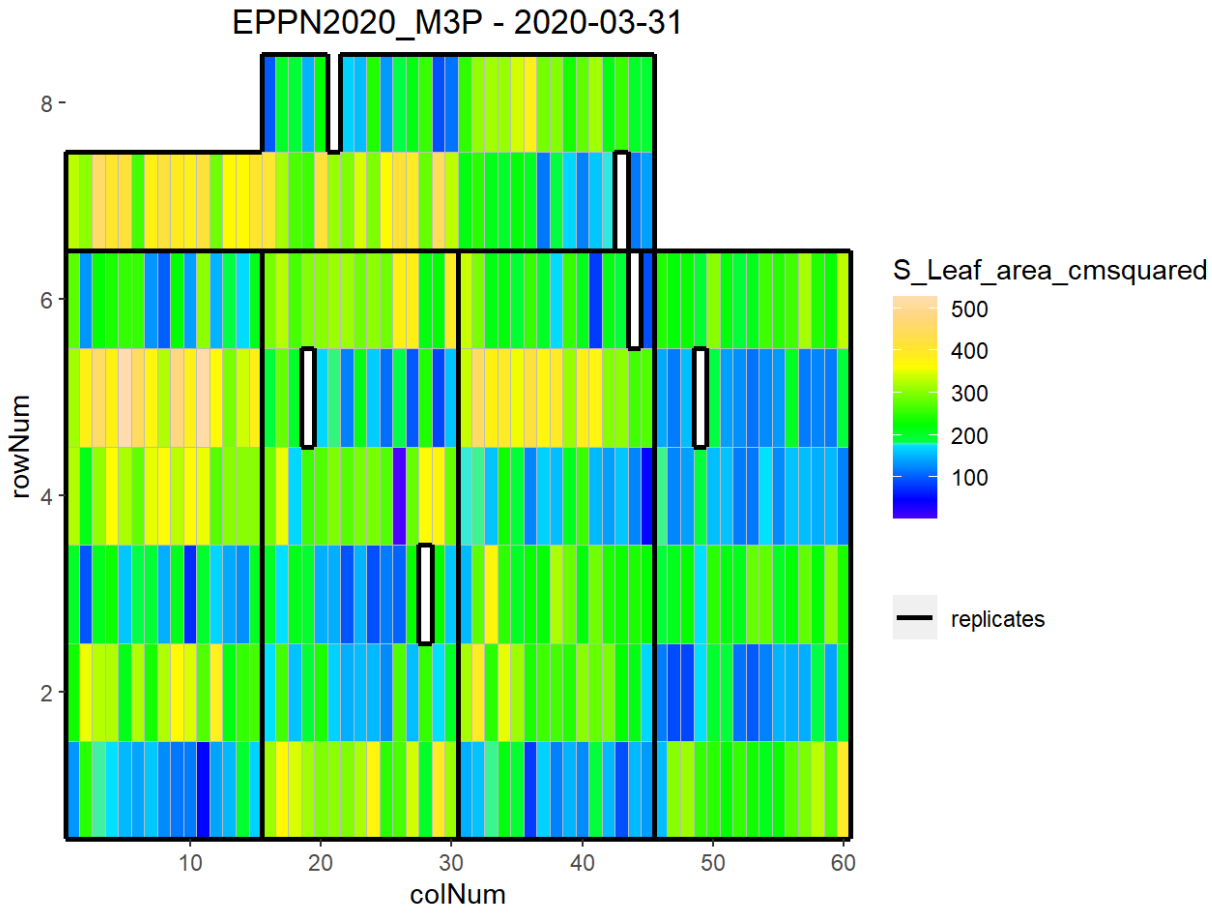

EPPN2020_M3P - 2020-03-26
rowNum
colNum
S_Leaf_area_cmsquared
400
300
200
100
replicates

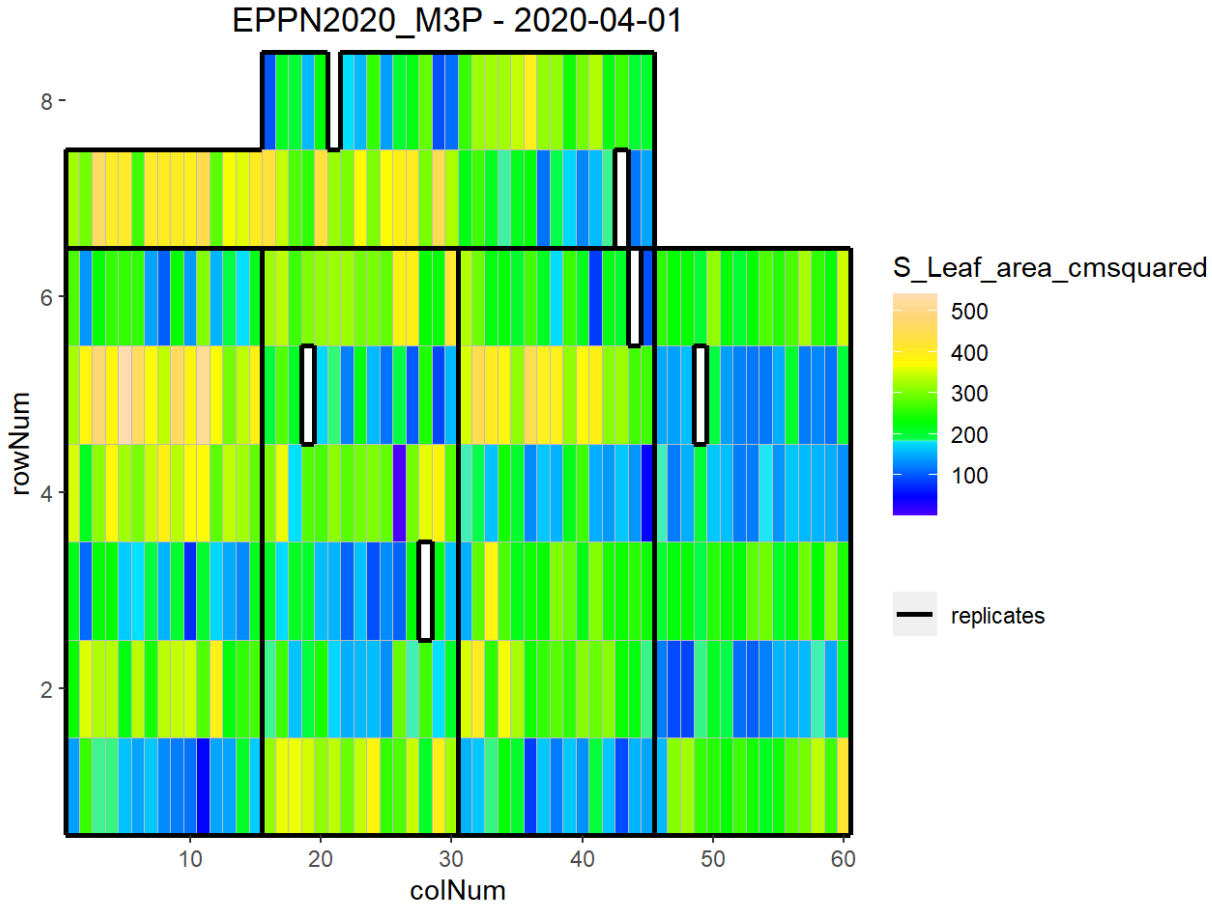

EPPN2020_M3P - 2020-03-27
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Leaf_area_cmsquared
400
300
200
100
replicates


EPPN2020_M3P - 2020-03-28
rowNum
colNum
8
6
4
2
10
20
30
40
50
60
S_Leaf_area_cmsquared
500
400
300
200
100
replicates


EPPN2020_M3P - 2020-03-29
rowNum
colNum
1
2
3
4
5
6
7
10
20
30
40
50
60
replicates
S_Leaf_area_cmsquared
500
400
300
200
100

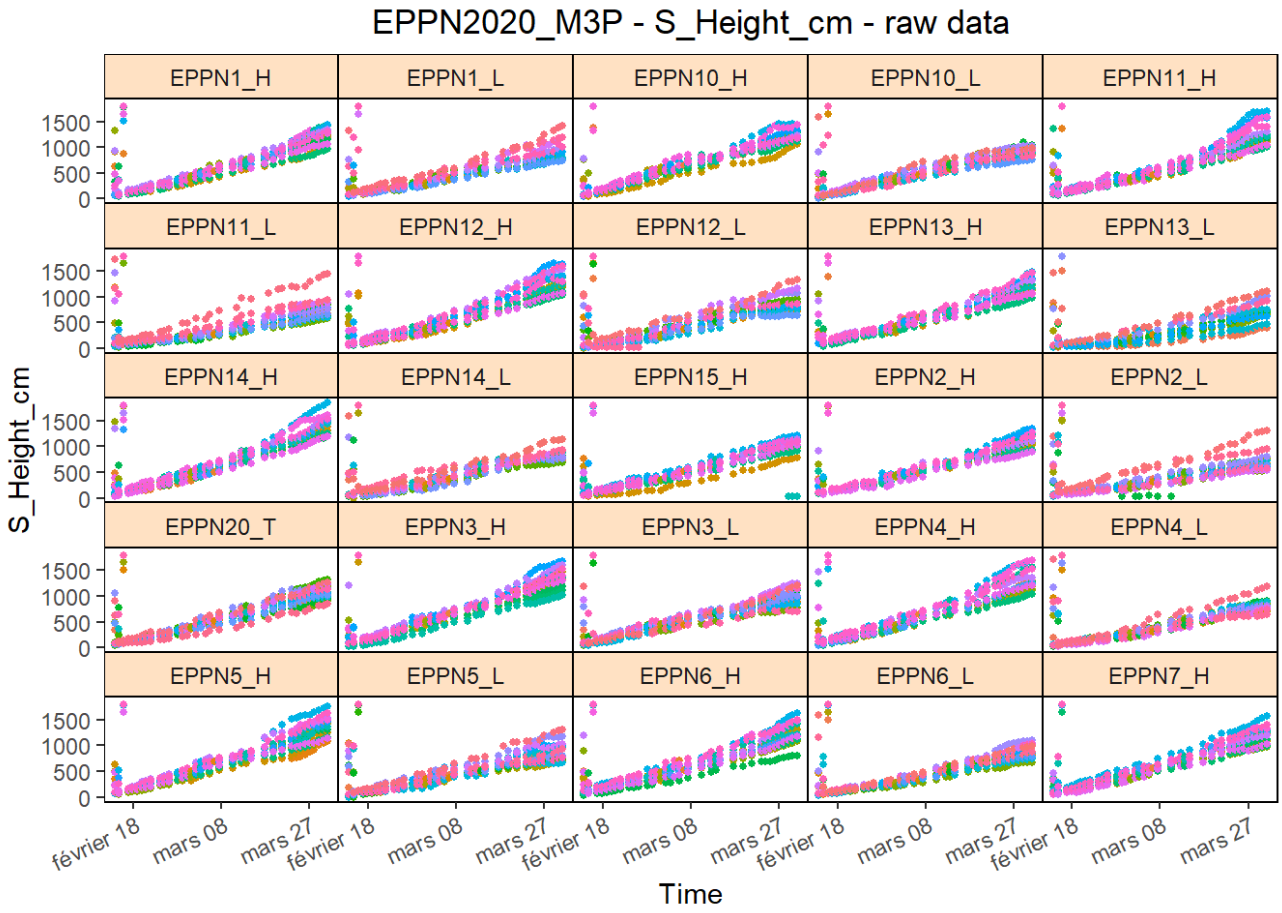

EPPN2020_M3P - 2020-03-30
rowNum
colNum
2
4
6
8
10
20
30
40
50
60
S_Leaf_area_cmsquared
500
400
300
200
100
replicates

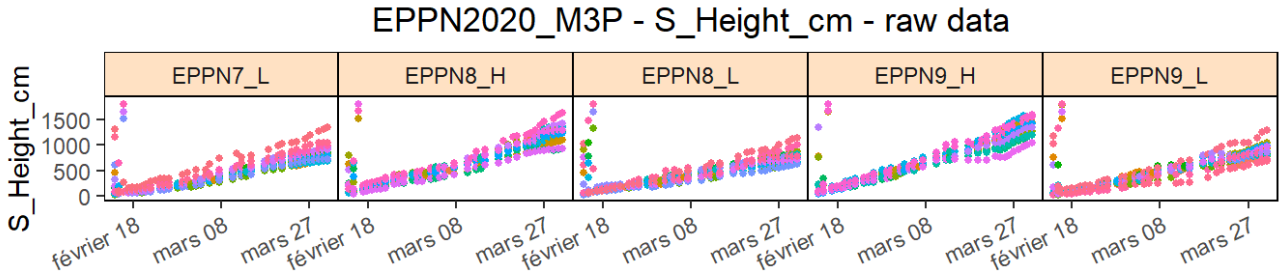




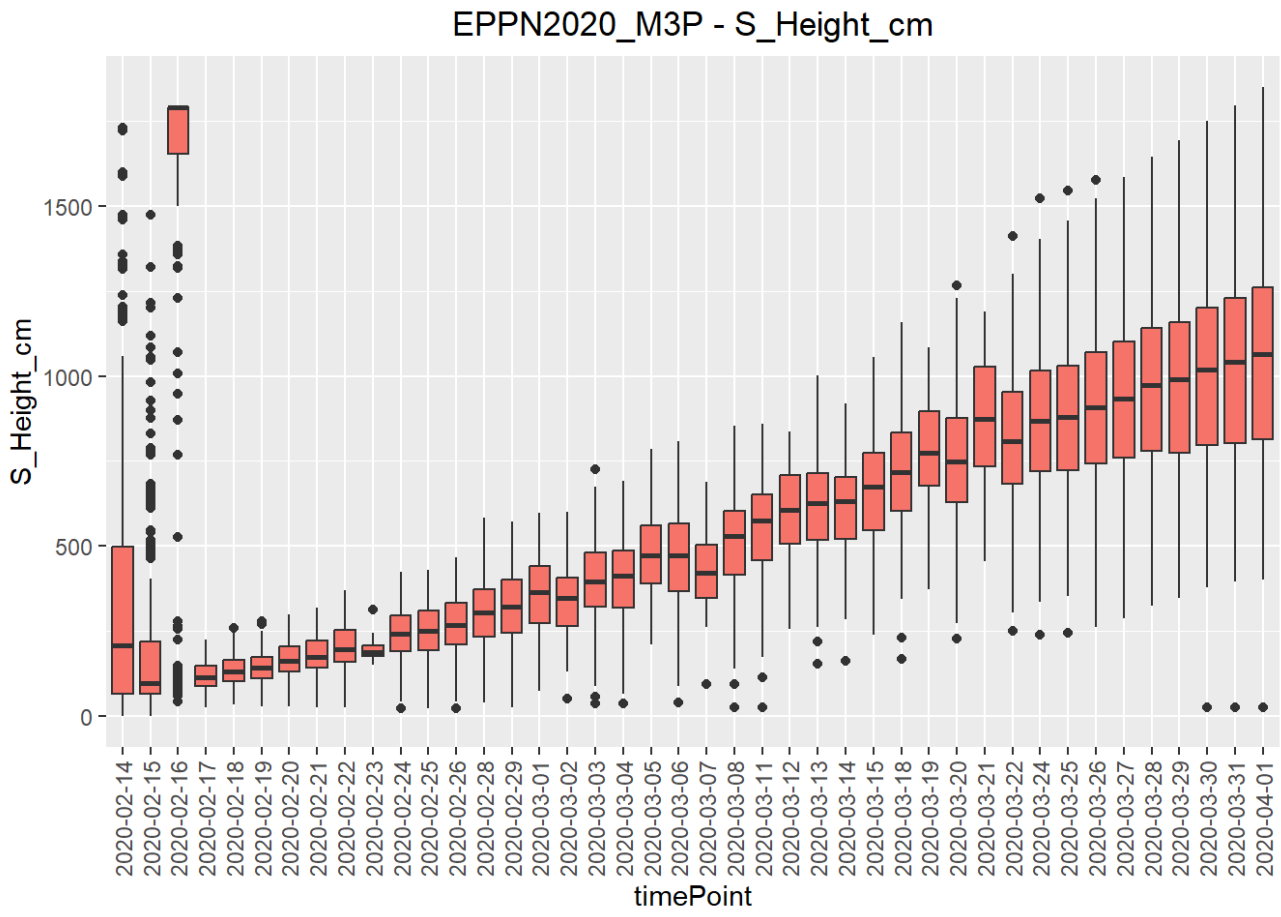
Time course, boxplots and correlation plots of data

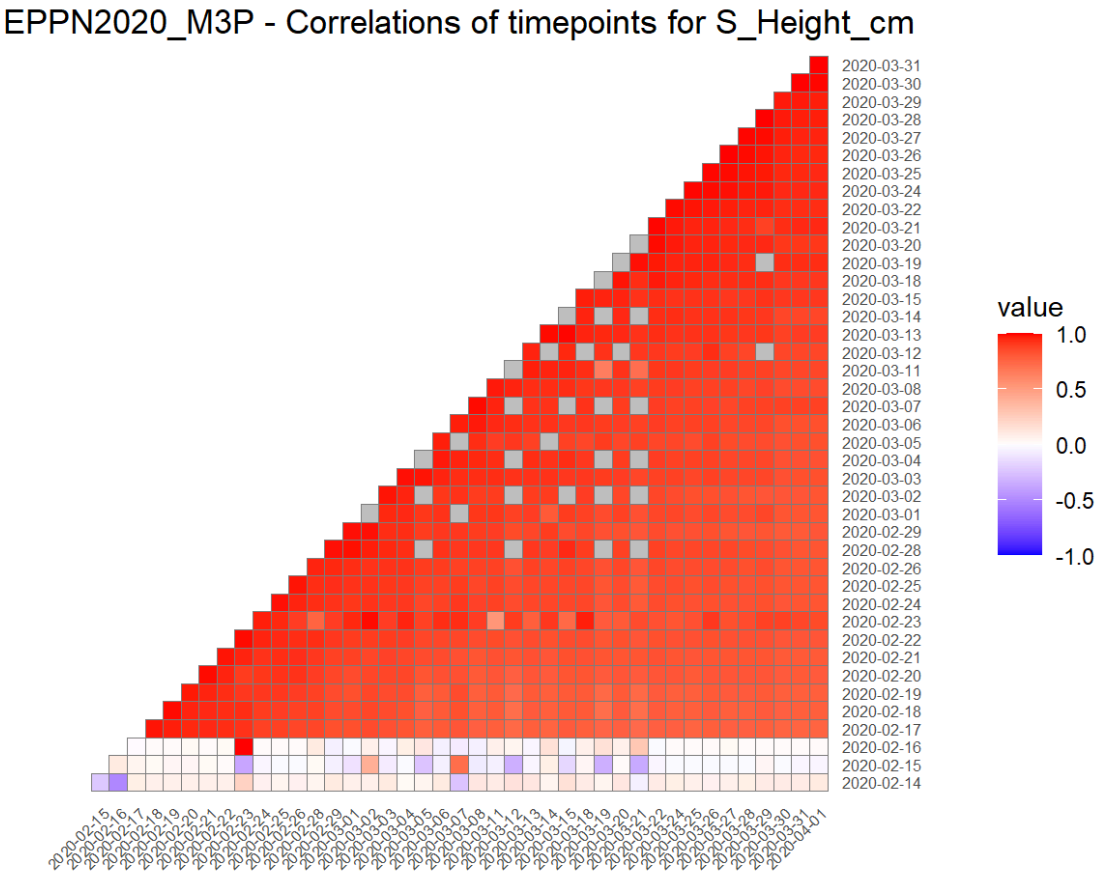
```
for (trait_name in traits) {
  single_outliers_name <- paste0("Single_outliers_", trait_name)

  if (exists(single_outliers_name)) {
    Single_outliers <- get(single_outliers_name)

    plot(Single_outliers,
         traits = trait_name,
         plotType = "raw")

    plot(Single_outliers,
         plotType = "box",
         traits = trait_name)

    plot(Single_outliers,
         plotType = "cor",
         traits = trait_name)

  } else {
    cat("No Single_outliers object found for trait", trait_name, "\n")
  }
}
```

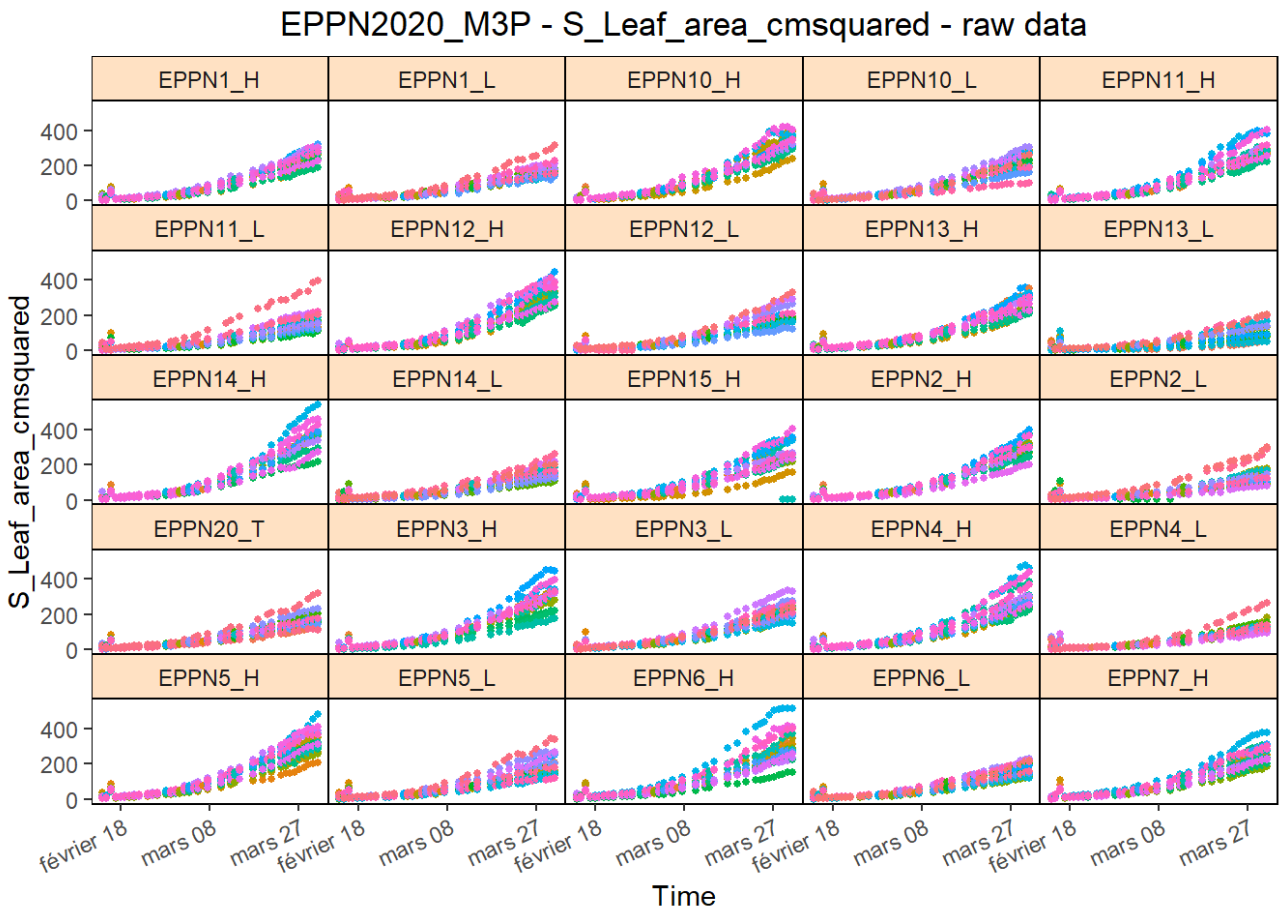

EPPN2020_M3P - S_Height_cm - raw data
EPPN1_H
EPPN1_L
EPPN10_H
EPPN10_L
EPPN11_H
EPPN11_L
EPPN12_H
EPPN12_L
EPPN13_H
EPPN13_L
EPPN14_H
EPPN14_L
EPPN15_H
EPPN2_H
EPPN2_L
EPPN20_T
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
EPPN5_H
EPPN5_L
EPPN6_H
EPPN6_L
EPPN7_H
S_Height_cm
0
500
1000
1500
février 18
mars 08
mars 27
Time

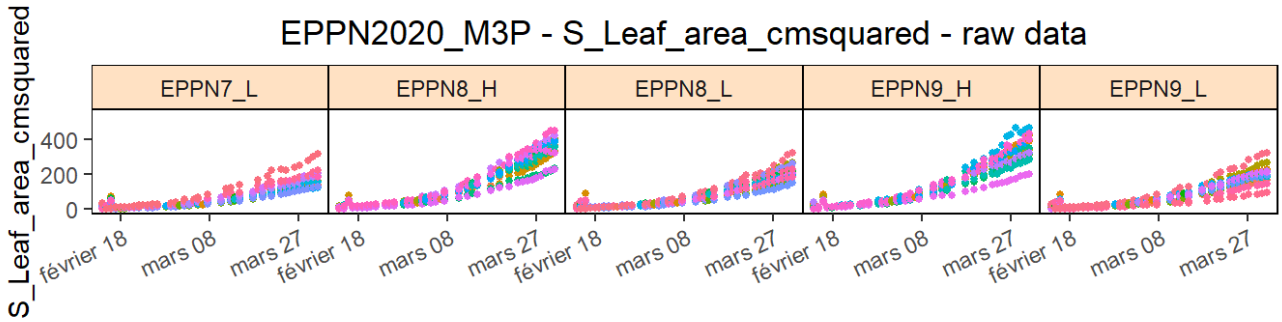

EPPN2020_M3P - S_Height_cm - raw data
EPPN7_L
EPPN8_H
EPPN8_L
EPPN9_H
EPPN9_L
S_Height_cm
0
500
1000
1500
février 18
mars 08
mars 27

Time

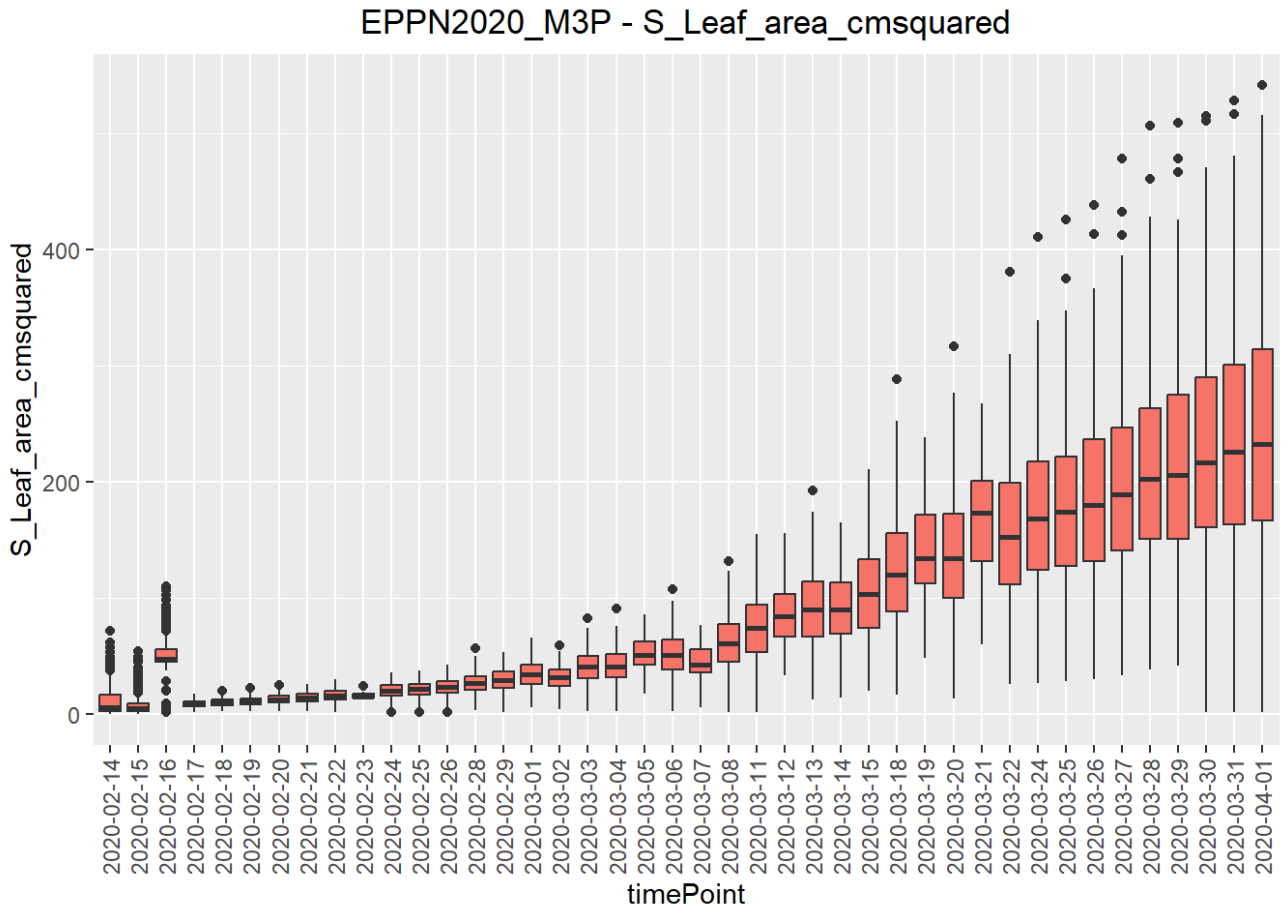

EPPN2020_M3P - S_Height_cm
S_Height_cm
0
500
1000
1500
2020-02-14
2020-02-15
2020-02-16
2020-02-17
2020-02-18
2020-02-19
2020-02-20
2020-02-21
2020-02-22
2020-02-23
2020-02-24
2020-02-25
2020-02-26
2020-02-28
2020-02-29
2020-03-01
2020-03-02
2020-03-03
2020-03-04
2020-03-05
2020-03-06
2020-03-07
2020-03-08
2020-03-11
2020-03-12
2020-03-13
2020-03-14
2020-03-15
2020-03-18
2020-03-19
2020-03-20
2020-03-21
2020-03-22
2020-03-24
2020-03-25
2020-03-26
2020-03-27
2020-03-28
2020-03-29
2020-03-30
2020-03-31
2020-04-01
timePoint

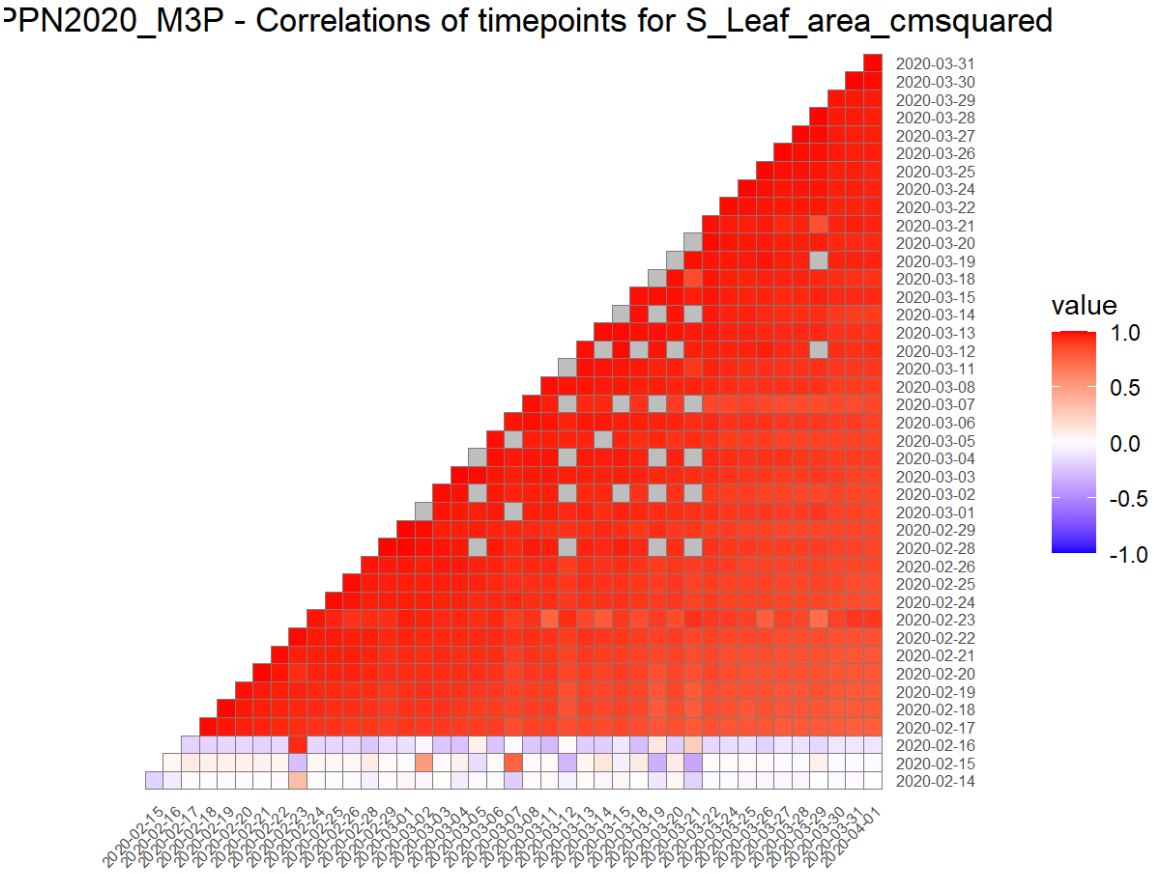

EPPN2020_M3P - Correlations of timepoints for S_Height_cm
value
1.0
0.5
0.0
-0.5
-1.0


EPPN2020_M3P - S_Leaf_area_cmsquared - raw data
EPPN1_H
EPPN1_L
EPPN10_H
EPPN10_L
EPPN11_H
EPPN11_L
EPPN12_H
EPPN12_L
EPPN13_H
EPPN13_L
EPPN14_H
EPPN14_L
EPPN15_H
EPPN2_H
EPPN2_L
EPPN20_T
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
EPPN5_H
EPPN5_L
EPPN6_H
EPPN6_L
EPPN7_H
S_Leaf_area_cmsquared
0
200
400
février 18
mars 08
mars 27
Time


EPPN2020_M3P - S_Leaf_area_cmsquared - raw data
EPPN7_L
EPPN8_H
EPPN8_L
EPPN9_H
EPPN9_L
S_Leaf_area_cmsquared
0
200
400
février 18
mars 08
mars 27

Time





2. Correction for spatial trends

Fit a model for all time points with no extra fixed effects.

```
#for (trait_name in traits) {
#  single_outliers_name <- paste0("Single_outliers_", trait_name)

#  if (exists(single_outliers_name)) {
#    Single_outliers <- get(single_outliers_name)

#    assign(paste0("modTP_", trait_name),
#           fitModels(TP = Single_outliers,
#                     trait = trait_name,
#                     geno.decomp = "Plant_type"))
#  } else {
#    assign(paste0("modTP_", trait_name),
#           fitModels(TP = timePoint_S,
#                     trait = trait_name,
#                     geno.decomp = "Plant_type"))
#  }
#}

print("Erreur dans SpATS::SpATS(response = trait, fixed = fixedForm, random = randForm,
:
  The design matrix associated to the fixed part of the model is not of full rank. Plea
se check that there are no redundant components in the fixed part of the model.")
```

```
## [1] "Erreur dans SpATS::SpATS(response = trait, fixed = fixedForm, random = randFor
m,  : \n  The design matrix associated to the fixed part of the model is not of full ra
nk. Please check that there are no redundant components in the fixed part of the mode
l."
```

Model visualisation

```
for (trait_name in traits) {
  mod_name <- paste0("modTP_", trait_name)

  if (exists(mod_name)) {
    mod <- get(mod_name)

    for (tp in 1:length(num_timepoints$timeNumber)) {
      plot(mod,
           timePoints = tp,
           plotType = "spatial",
           spaTrend = "percentage")
    }

    gif_file <- sprintf("%s/%s_mod.gif", datadir, trait_name)

    plot(mod,
         plotType = "timeLapse",
         spaTrend = "percentage",
         outFile = gif_file)
  } else {
    cat("No model found for", trait_name, "\n")
  }
}
```

```
## No model found for S_Height_cm
## No model found for S_Leaf_area_cmsquared
```

```
for (trait_name in traits) {
  mod_name <- paste0("modTP_", trait_name)
    if (exists(mod_name)) {
    mod <- get(mod_name)

    plot(mod,
         plotType = "rawPred",
         plotLine = TRUE)

    plot(mod,
         plotType = "corrPred",
         plotLine = TRUE)

    plot(mod,
         plotType = "herit",
         yLim = c(0, 0.5))

    plot(mod,
         plotType = "effDim",
         EDType = "ratio",
         yLim = c(0, 1))

    plot(mod,
         plotType = "corrPred")
  } else {
    cat("No model found for the trait", trait_name, "\n")
  }
}
```

```
## No model found for the trait S_Height_cm
## No model found for the trait S_Leaf_area_cmsquared
```

3. Outlier detection for series of observations

By using the splines.

fitModels

```
for (trait_name in traits) {
  Spatial_Corrected_name <- paste0("Spatial_Corrected_", trait_name)
  modTP_name <- paste0("modTP_", trait_name)
}
```

```
knots <- c(30)
mintimepoints <- c(9) # Minimal number of observations

for (trait_name in traits) {
  # Nom de la variable pour les données corrigées
  Spatial_Corrected_name <- paste0("Spatial_Corrected_", trait_name)
  modTP_name <- paste0("modTP_", trait_name)

  # Vérifier si le modèle existe
  if (exists(modTP_name)) {
    modTP <- get(modTP_name)
    Spatial_Corrected <- getCorrected(modTP)
    assign(Spatial_Corrected_name, Spatial_Corrected)

    # Ajuster les splines pour les données corrigées
    fit.spline <- fitSpline(inDat = Spatial_Corrected,
                            trait = paste0(trait_name, "_corr"),
                            knots = knots,
                            minNoTP = mintimepoints)

    # Extraire les tables de valeurs prédites et coefficients de splines
    predDat_name <- paste0("predDat_", trait_name)
    coefDat_name <- paste0("coefDat_", trait_name)

    assign(predDat_name, fit.spline$predDat)
    assign(coefDat_name, fit.spline$coefDat)
  } else {
    cat("No model found for", trait_name, "\n")
  }
}
```

```
## No model found for S_Height_cm
## No model found for S_Leaf_area_cmsquared
```

Plot the splines for a plant selection

detectSerieOut

```
thrCor <- c(0.9) # correlation threshold
thrPca <- c(30) # pca angle threshold
thrSlope <- c(0.7) # slope threshold

for (trait_name in traits) {
  Spatial_Corrected_name <- paste0("Spatial_Corrected_", trait_name)
  predDat_name <- paste0("predDat_", trait_name)
  coefDat_name <- paste0("coefDat_", trait_name)

  if (exists(Spatial_Corrected_name) && exists(predDat_name) && exists(coefDat_name)) {
    Spatial_Corrected <- get(Spatial_Corrected_name)
    predDat <- get(predDat_name)
    coefDat <- get(coefDat_name)

    Series_test <- detectSerieOut(corrDat = Spatial_Corrected,
                                  predDat = predDat,
                                  coefDat = coefDat,
                                  trait = paste0(trait_name, "_corr"),
                                  thrCor = thrCor,
                                  thrPca = thrPca,
                                  thrSlope = thrSlope,
                                  geno.decomp = "geno.decomp")

    plot(Series_test, genotypes = levels(factor(Series_test$genotype)))

    assign(paste0("Series_test_", trait_name), Series_test)

    assign(paste0("Spatial_Corrected_Out_", trait_name), Spatial_Corrected)
  } else {
    cat("No corrected data or prediction data found for", trait_name, "\n")
  }
}
```

```
## No corrected data or prediction data found for S_Height_cm
## No corrected data or prediction data found for S_Leaf_area_cmsquared
```

removeSerieOut

```
for (trait_name in traits) {
  # Nom de la variable pour les données corrigées
  Spatial_Corrected_name <- paste0("Spatial_Corrected_", trait_name)
  Series_test_name <- paste0("Series_test_", trait_name)

  # Extraire les données corrigées et les résultats des séries
  if (exists(Spatial_Corrected_name) && exists(Series_test_name)) {
    Spatial_Corrected <- get(Spatial_Corrected_name)
    Series_test <- get(Series_test_name)

    # Supprimer les outliers de la série
    Spatial_Corrected_Out <- removeSerieOut(dat = Spatial_Corrected, serieOut = Series_
test)

    # Assigner le résultat à une nouvelle variable
    assign(paste0("Spatial_Corrected_Out_", trait_name), Spatial_Corrected_Out)
  } else {
    cat("No corrected data or series test data found for", trait_name, "\n")
  }
}
```

```
## No corrected data or series test data found for S_Height_cm
## No corrected data or series test data found for S_Leaf_area_cmsquared
```

```
for (trait_name in traits) {
  Spatial_Corrected_Out_name <- paste0("Spatial_Corrected_Out_", trait_name)

  if (exists(Spatial_Corrected_Out_name)) {
    Spatial_Corrected_Out <- get(Spatial_Corrected_Out_name)
    output_file <- sprintf("%s/timeSeriesOutliers_%s.tsv", datadir, trait_name)
    readr::write_tsv(Spatial_Corrected_Out, output_file)

    cat("Data written to:", output_file, "\n")
  } else {
    cat("No corrected data found for", trait_name, "\n")
  }
}
```

```
## No corrected data found for S_Height_cm
## No corrected data found for S_Leaf_area_cmsquared
```

4. With the cleaned data, re-do the spatial correction

This is used to compare the values before and after.

Need to write a for loop for all the variables.

For S_Height_cm

Estimation of parameter from time series

For S_Area_cmsquared

Estimation of parameter from time series