SPPU Data Analysis Timepoints

Elise

2024-06-09

Set the right working directory.

```
rm(list = ls())

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

Data importation

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

```
list.files()
```

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

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

# plant_info
plant_info <- lapply(plant_info, factor)

# 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)

platform <- "SPPU"

# 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"


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

```
## [1] "SPPU : The variables for S_timeseries are S_Height_cm, S_Height_pixel, S_Area_c
msquared, S_Area_pixel, S_Perimeter_cm, S_Perimeter_pixel, S_Compactness, S_Width_cm, S
_Width_pixel"
```

```
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

```
# 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”.

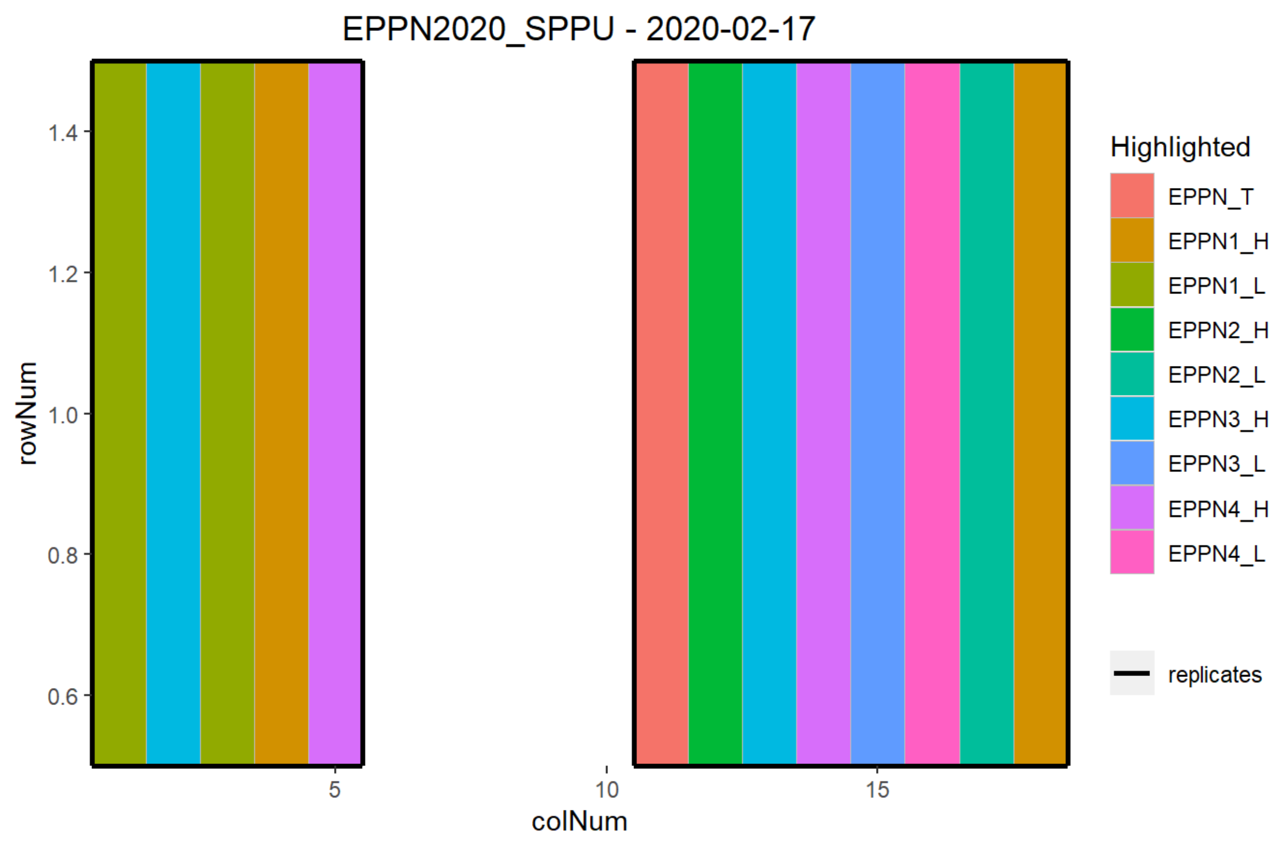
```
timePoint_S <- createTimePoints(dat = S_timeseries,
                              experimentName = "EPPN2020_SPPU",
                              genotype = "Genotype",
                              timePoint = "Date",
                              plotId = "Unit.ID",
                              rowNum = "Row",
                              colNum = "Column",
                              repId = "Replication",
                              addCheck = TRUE,
                              checkGenotypes = "EPPN_T")
```

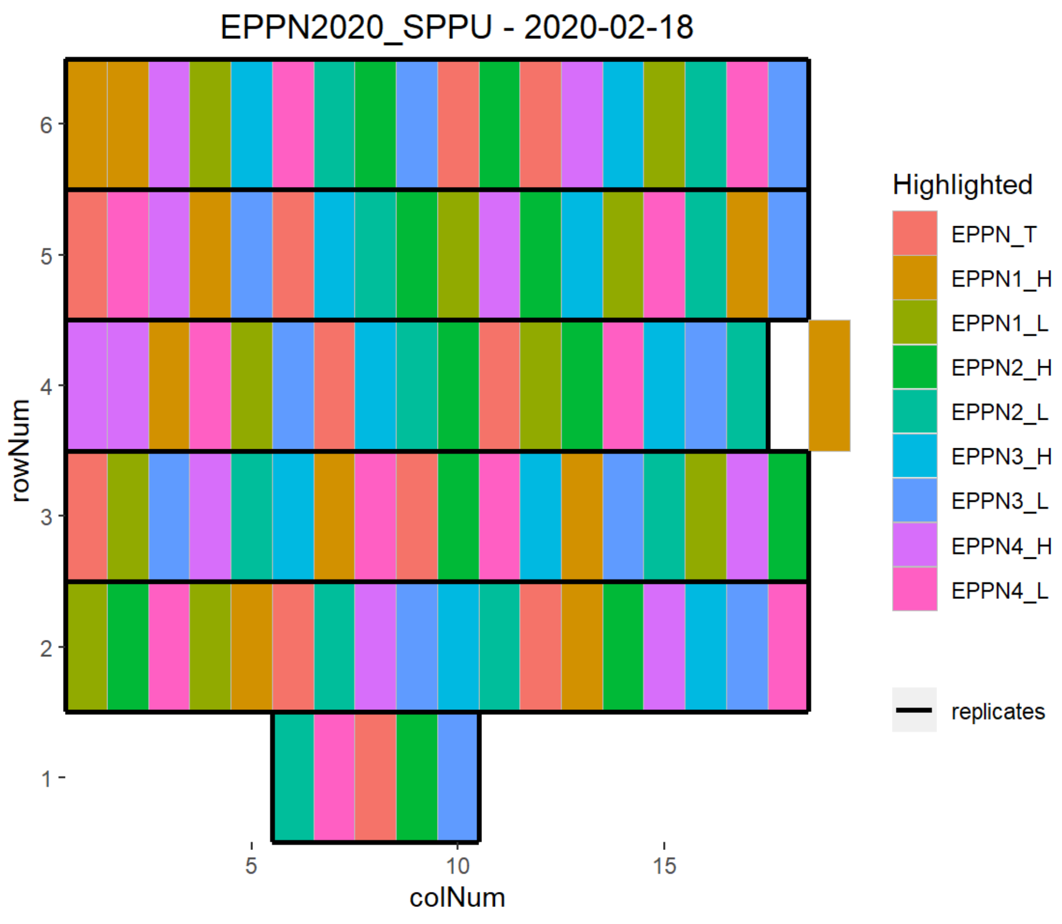
```
## Warning: More than 5 plotIds have observations for less than 50% of the time points.
The first 5 are printed, to see them all run
##               attr(..., 'plotLimObs') on the output
## 102, 103, 104, 105, 106
```

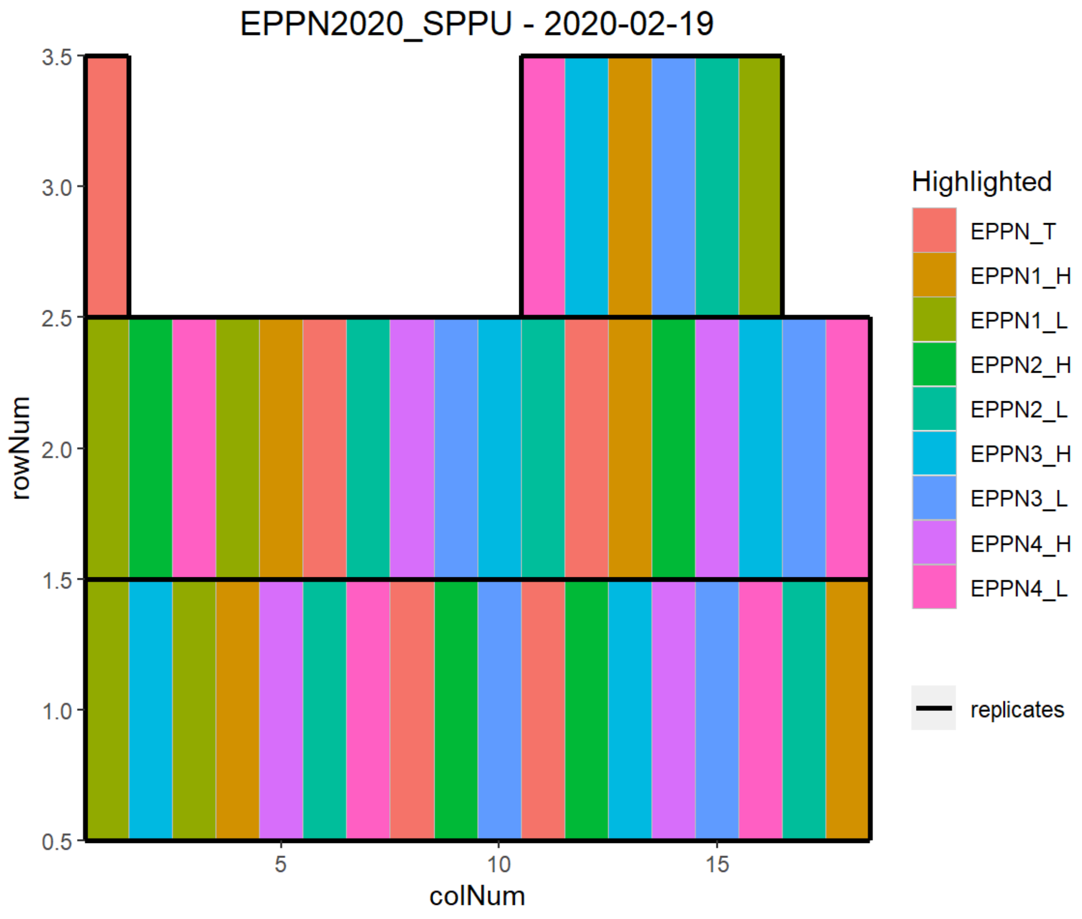
Gentoypic layout

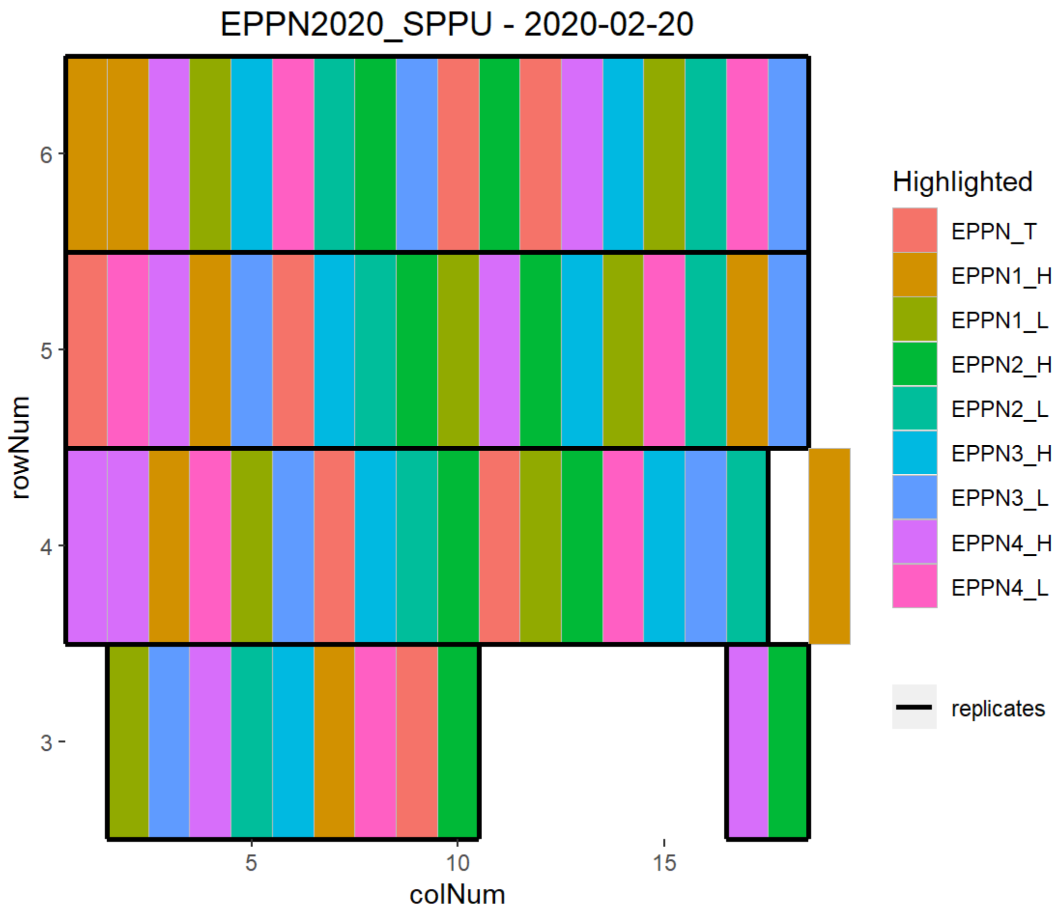
Check the layout of the platforms' genotypes.

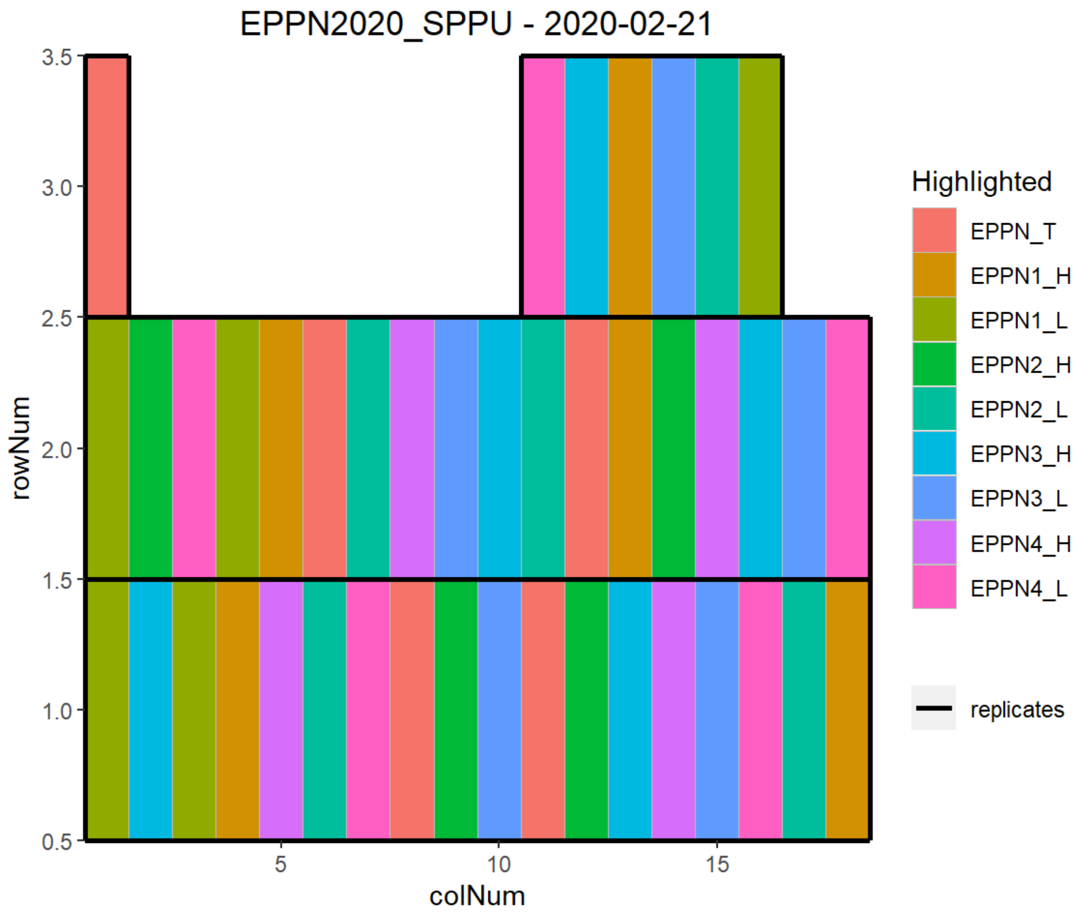
```
genotypes_list <- as.character(unique(S_timeseries$Genotype))

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

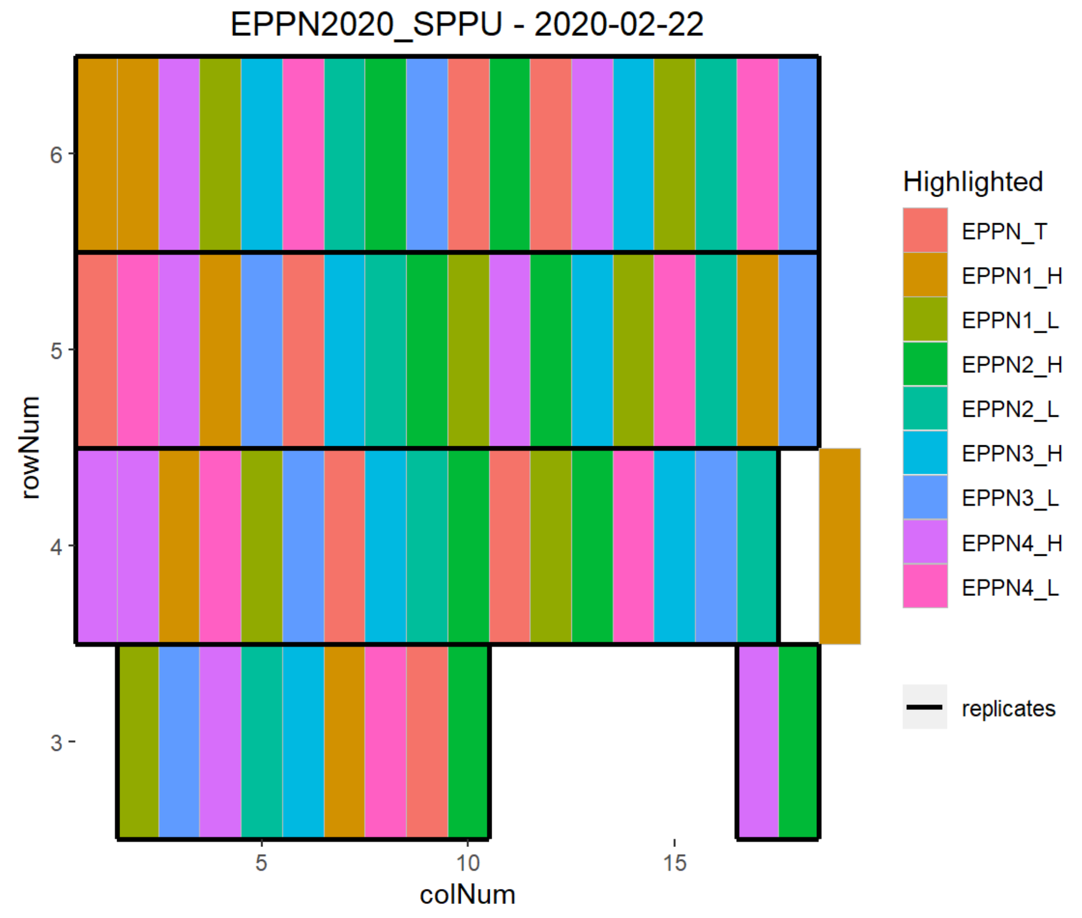

EPPN2020_SPPU - 2020-02-17
rowNum
1.4
1.2
1.0
0.8
0.6
5
10
15
colNum
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

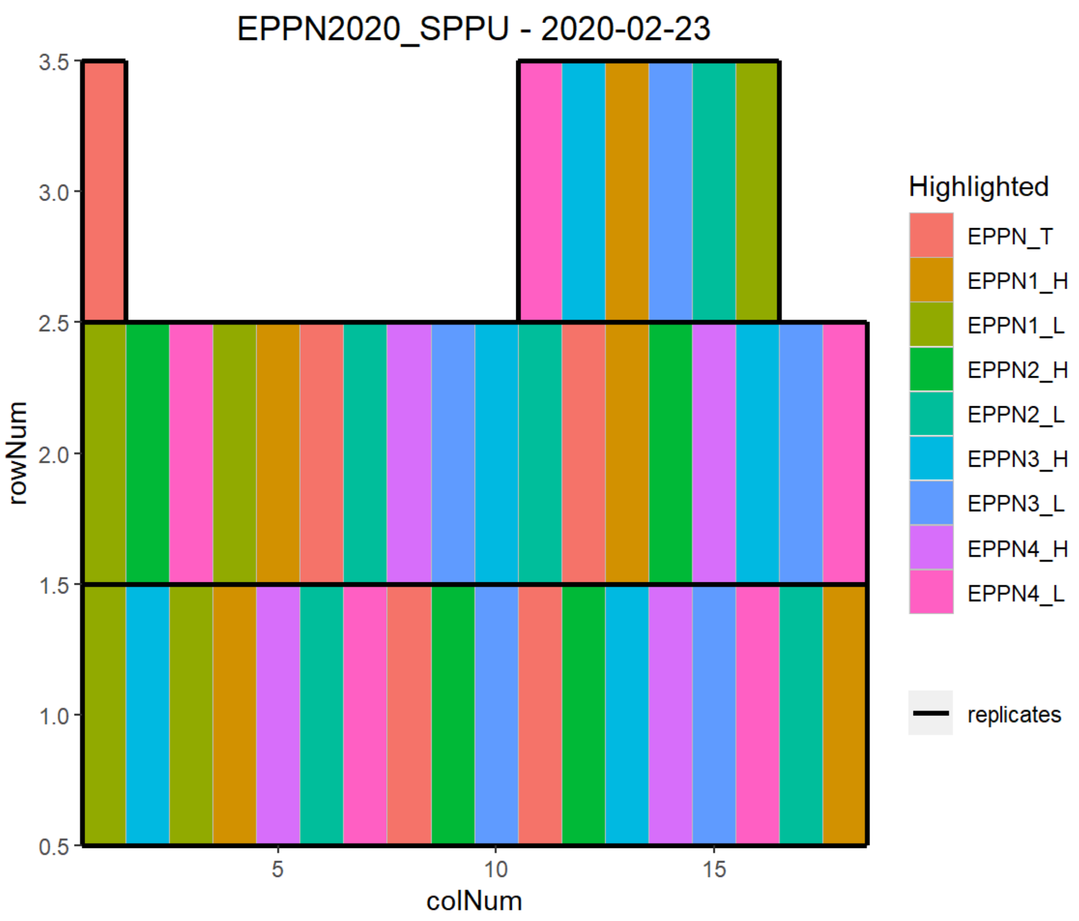

EPPN2020_SPPU - 2020-02-18
rowNum
6
5
4
3
2
1
5
10
15
colNum
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

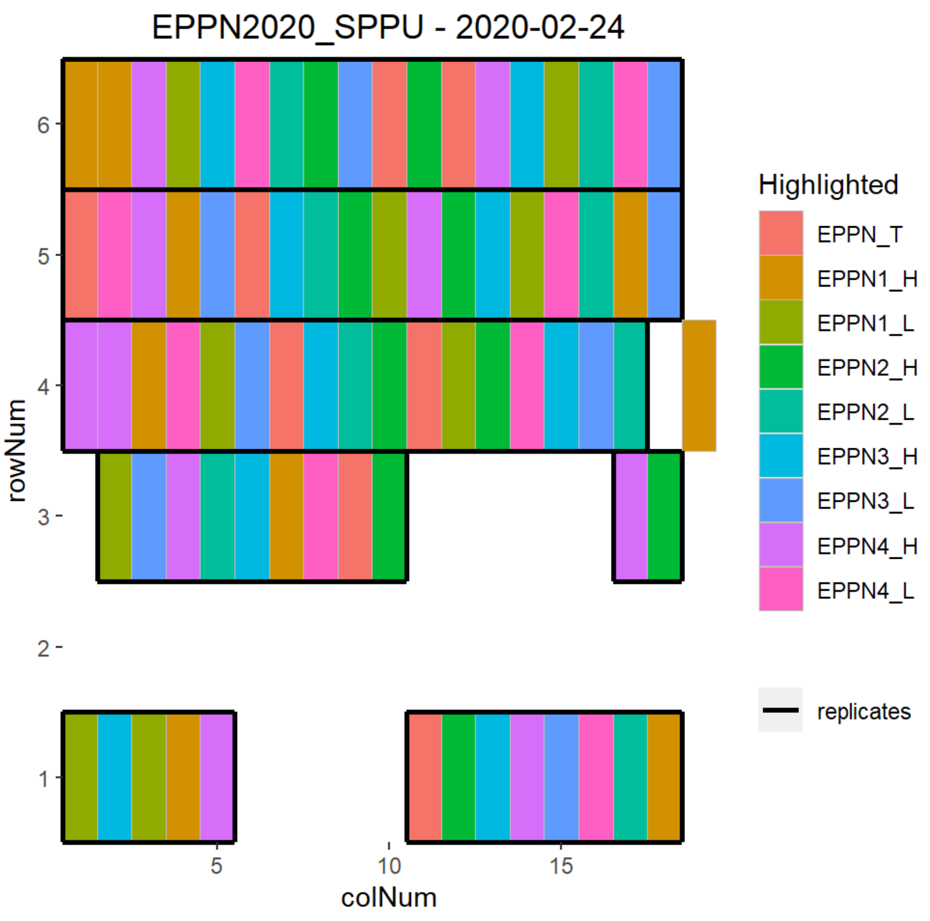

EPPN2020_SPPU - 2020-02-19
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
5
10
15
colNum
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

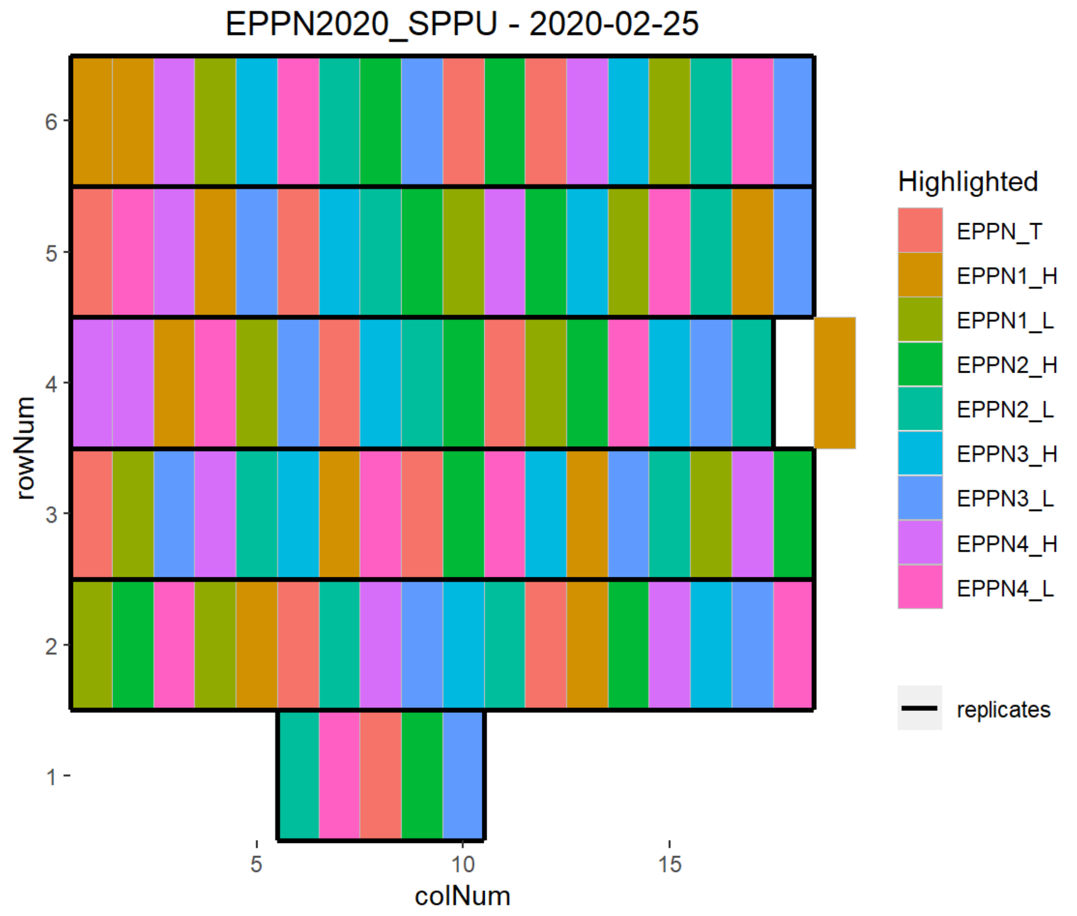

EPPN2020_SPPU - 2020-02-20
rowNum
6
5
4
3
5
10
15
colNum
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

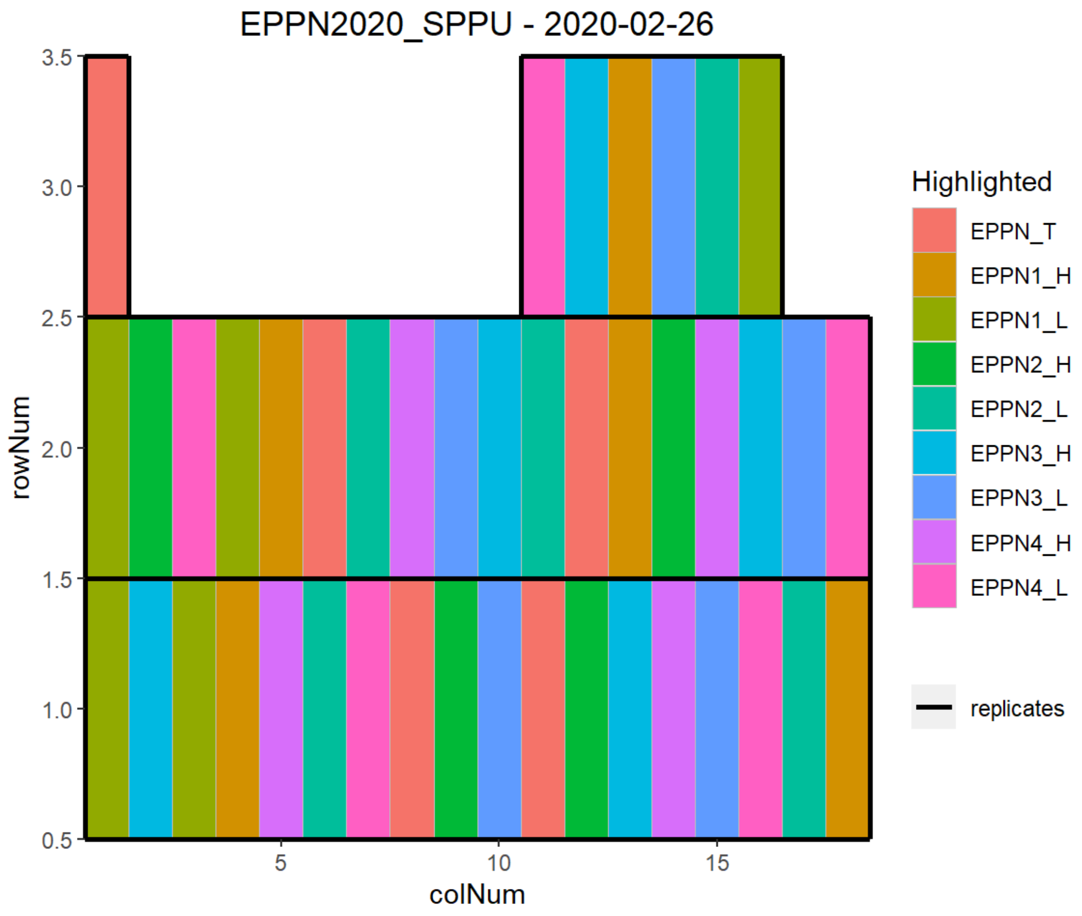

EPPN2020_SPPU - 2020-02-21
3.5
3.0
2.5
2.0
1.5
1.0
0.5
rowNum
5
10
15
colNum
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

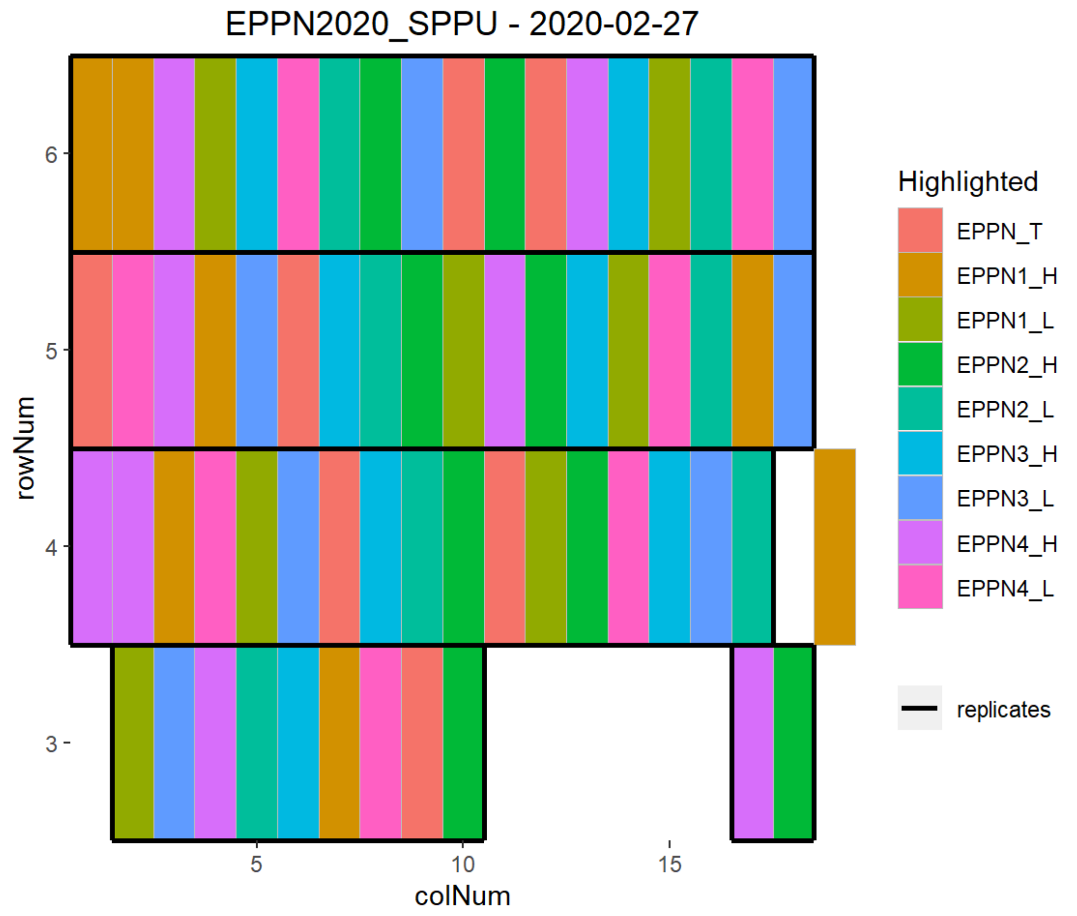

EPPN2020_SPPU - 2020-02-22
6
5
4
3
rowNum
5
10
15
colNum
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

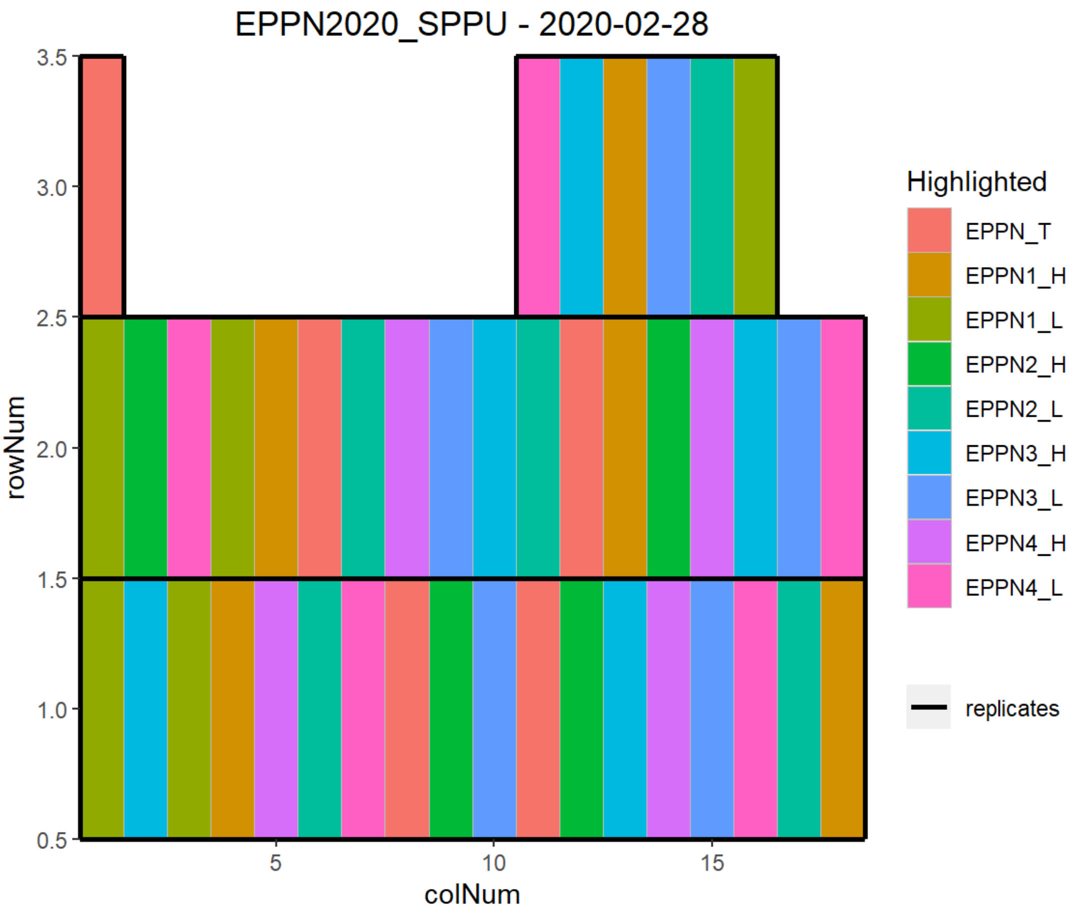

EPPN2020_SPPU - 2020-02-23
rowNum
colNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

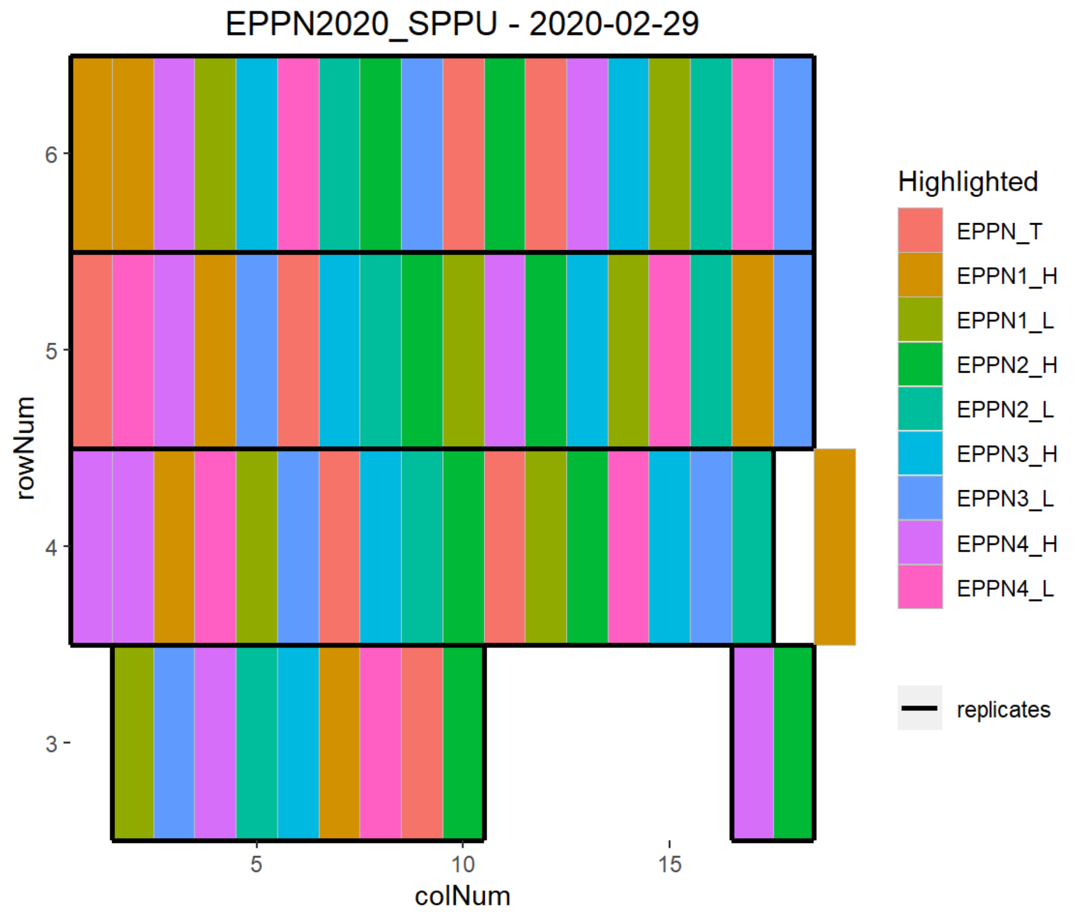

EPPN2020_SPPU - 2020-02-24
rowNum
colNum
6
5
4
3
2
1
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

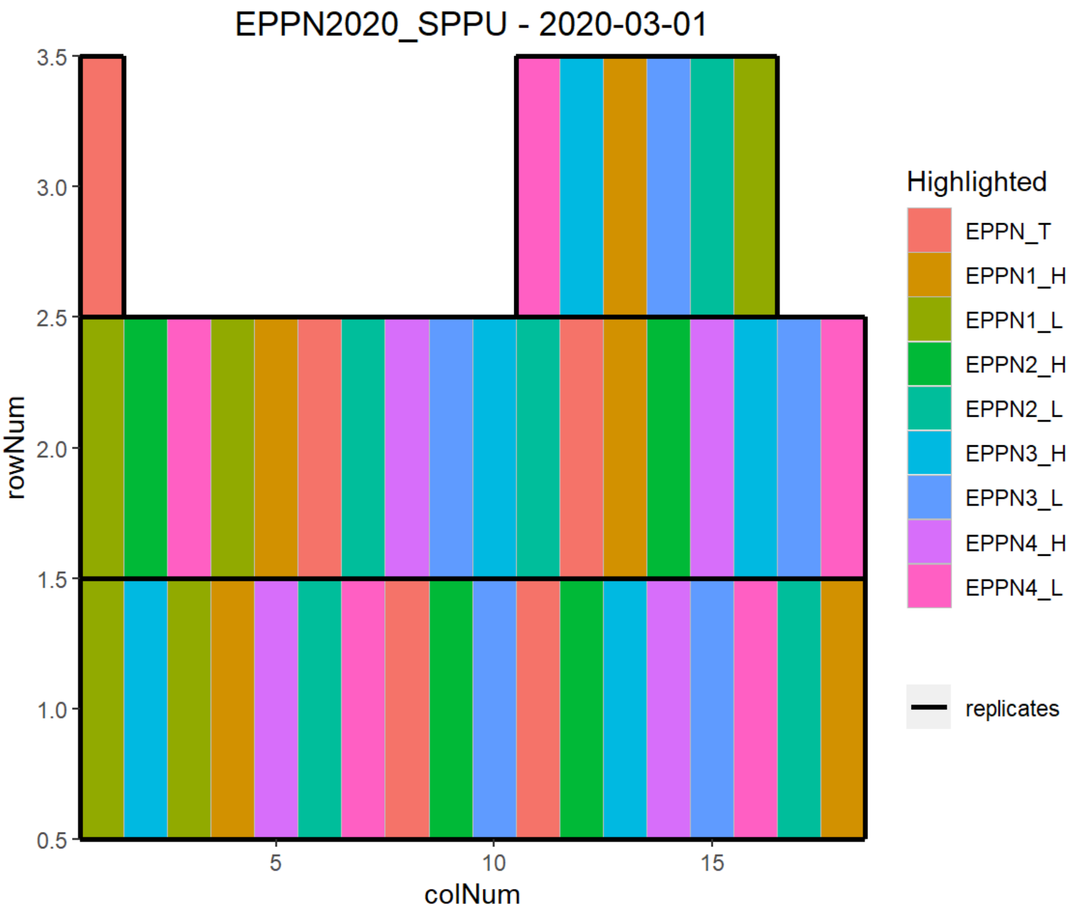

EPPN2020_SPPU - 2020-02-25
rowNum
6
5
4
3
2
1
colNum
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

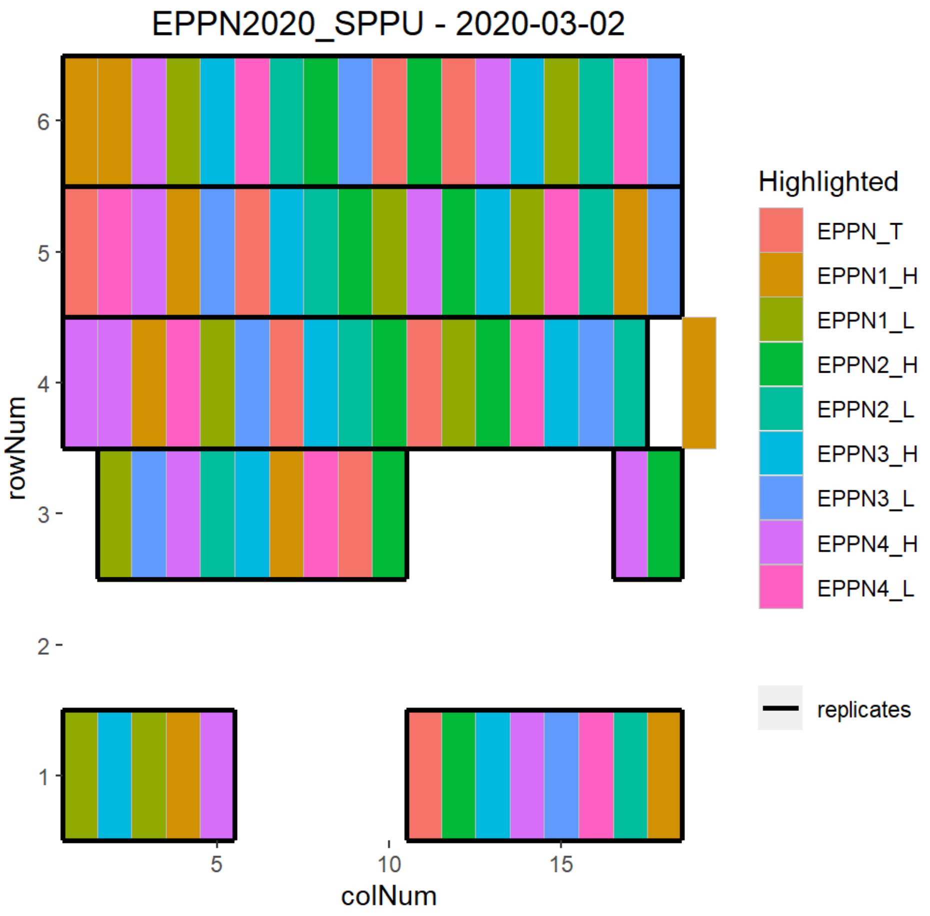

EPPN2020_SPPU - 2020-02-26
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

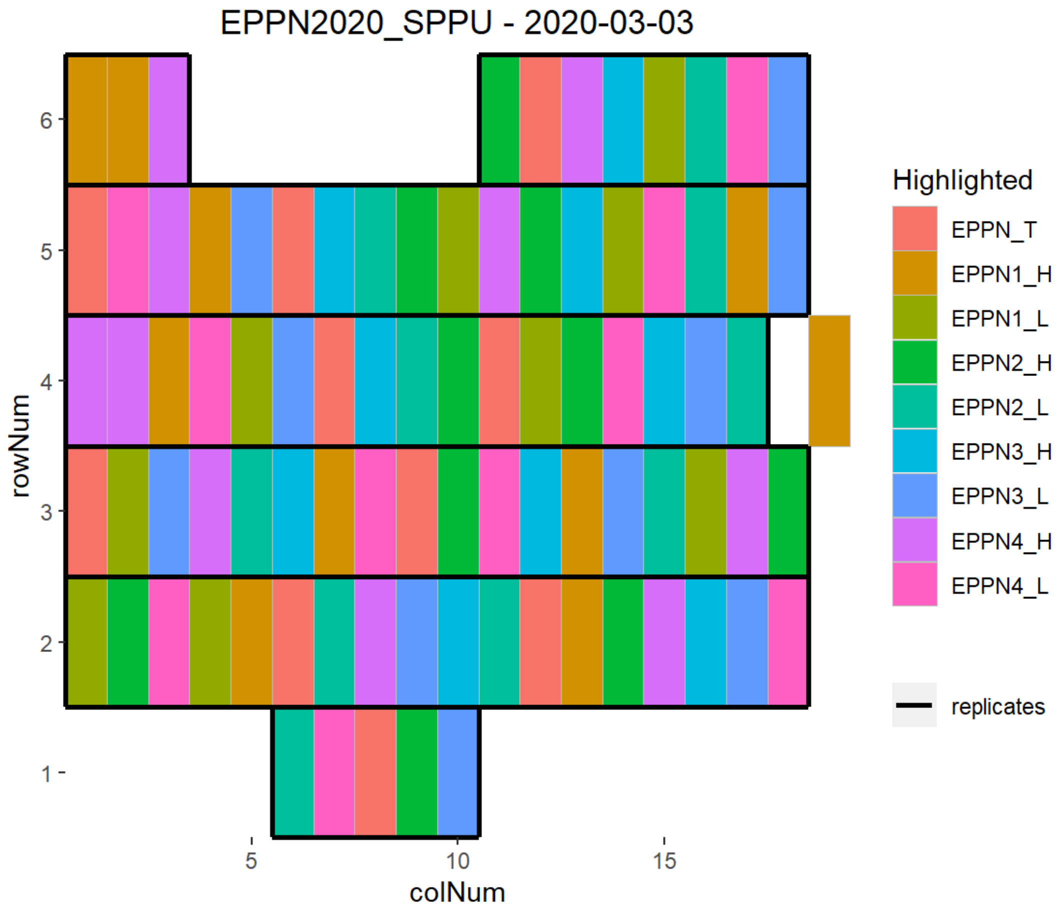

EPPN2020_SPPU - 2020-02-27
rowNum
colNum
3
4
5
6
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

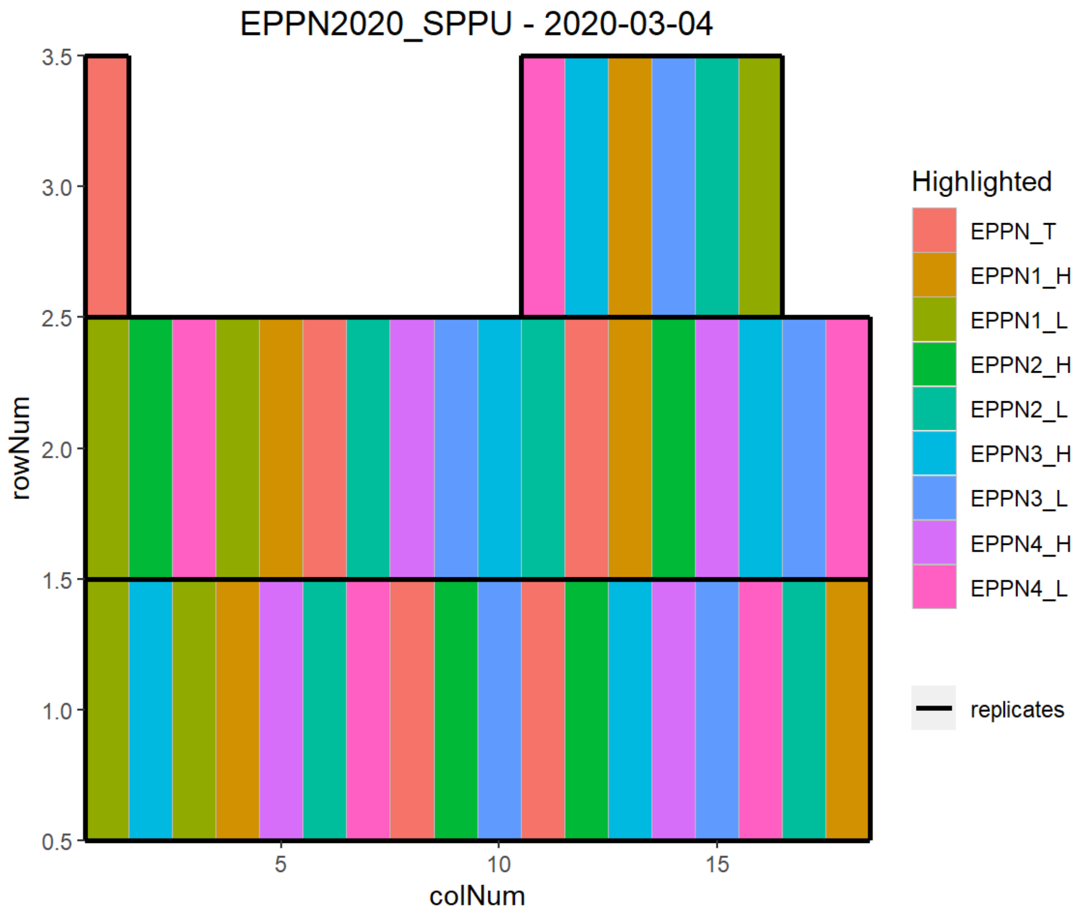

EPPN2020_SPPU - 2020-02-28
rowNum
colNum
0.5
1.0
1.5
2.0
2.5
3.0
3.5
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

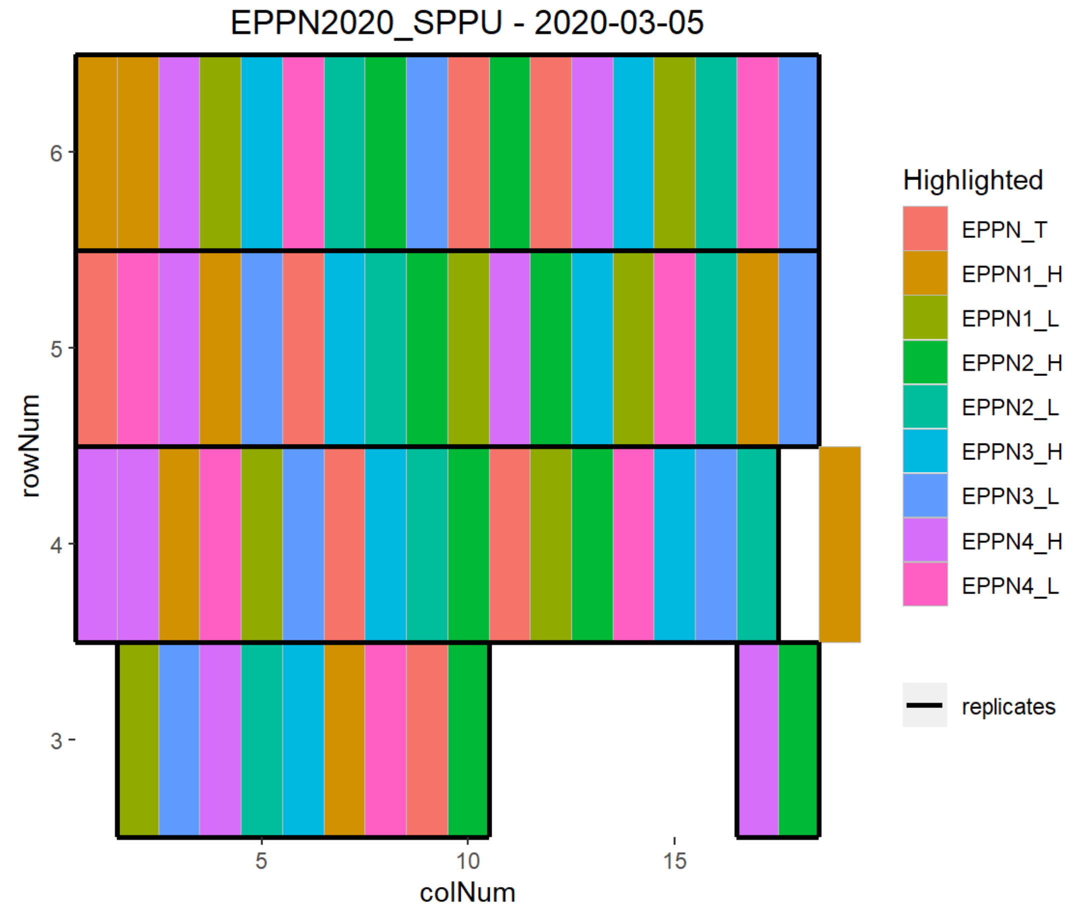

EPPN2020_SPPU - 2020-02-29
rowNum
6
5
4
3
colNum
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

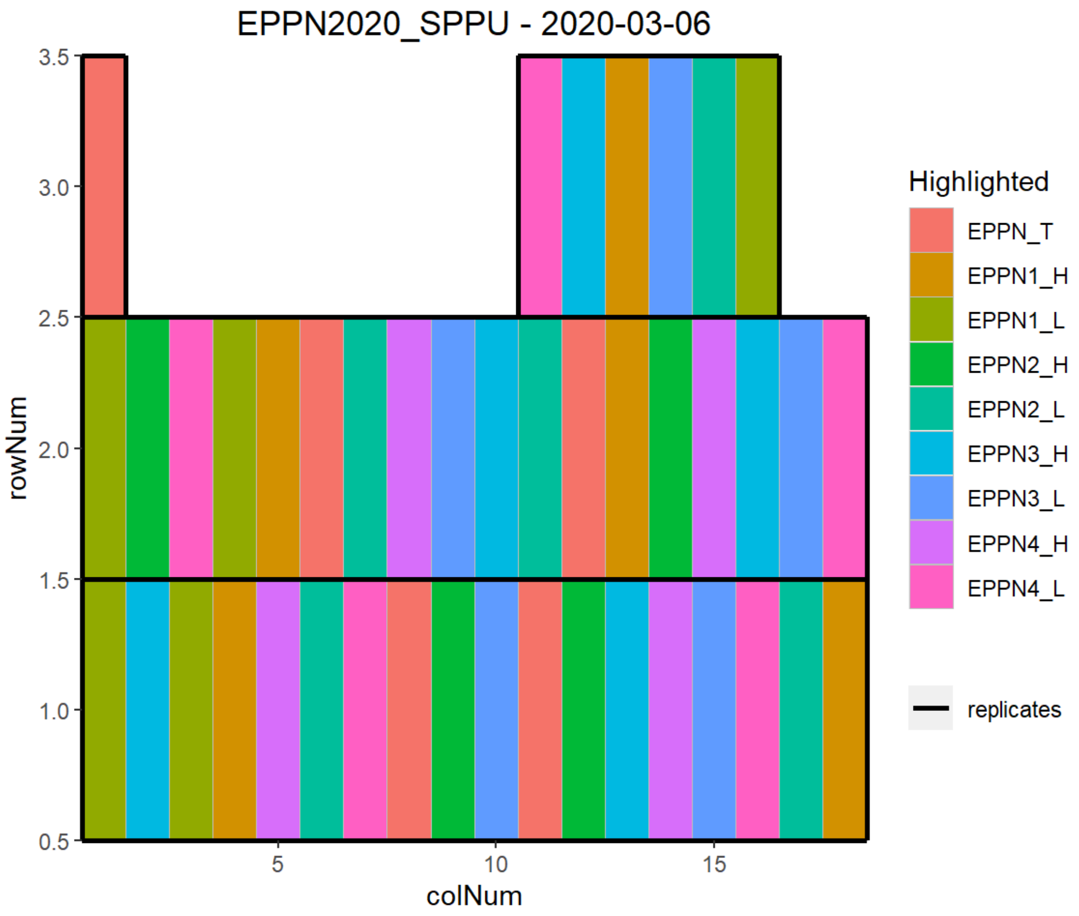

EPPN2020_SPPU - 2020-03-01
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

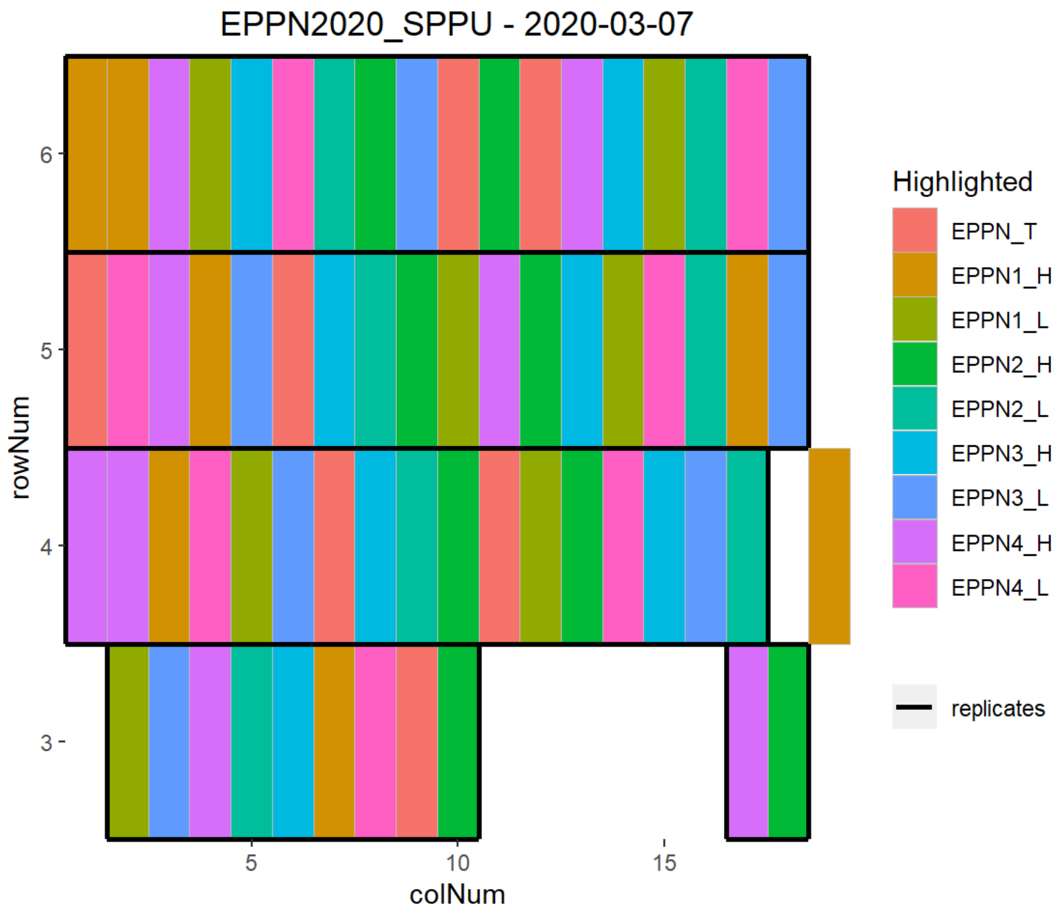

EPPN2020_SPPU - 2020-03-02
rowNum
6
5
4
3
2
1
colNum
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

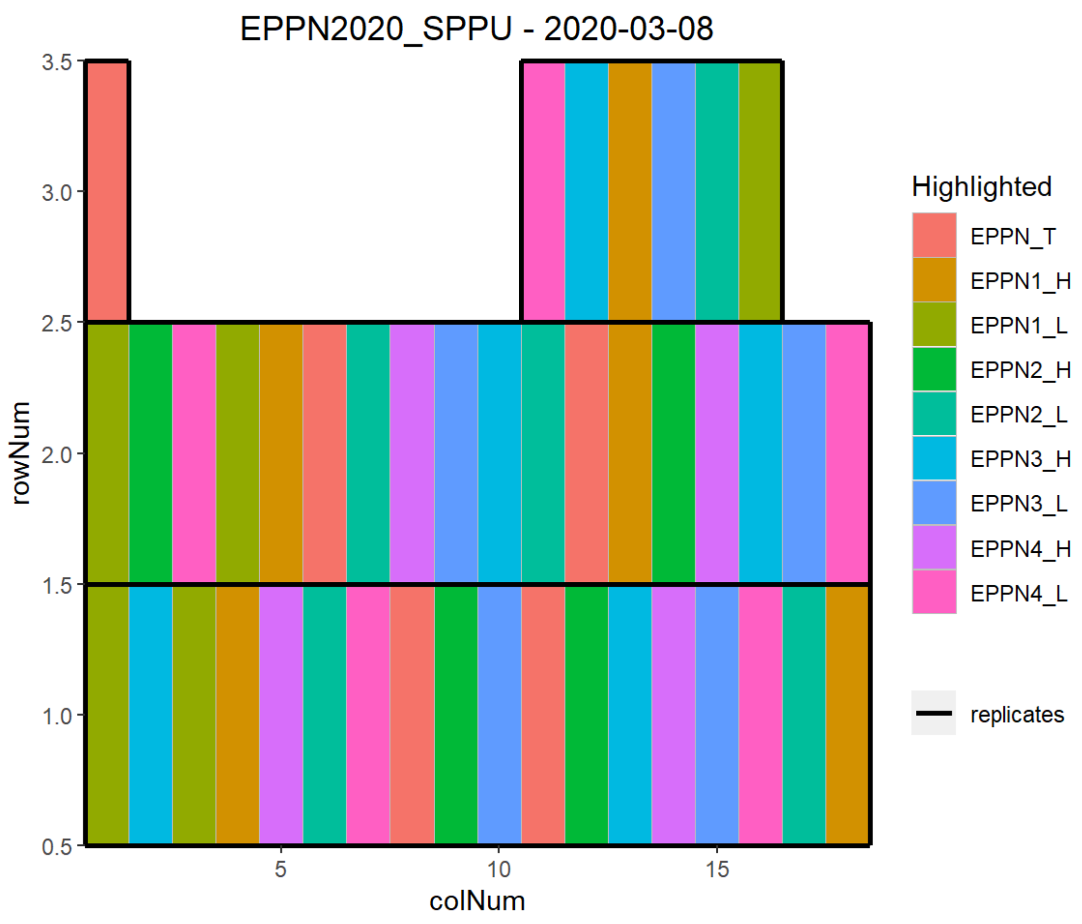

EPPN2020_SPPU - 2020-03-03
rowNum
6
5
4
3
2
1
colNum
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

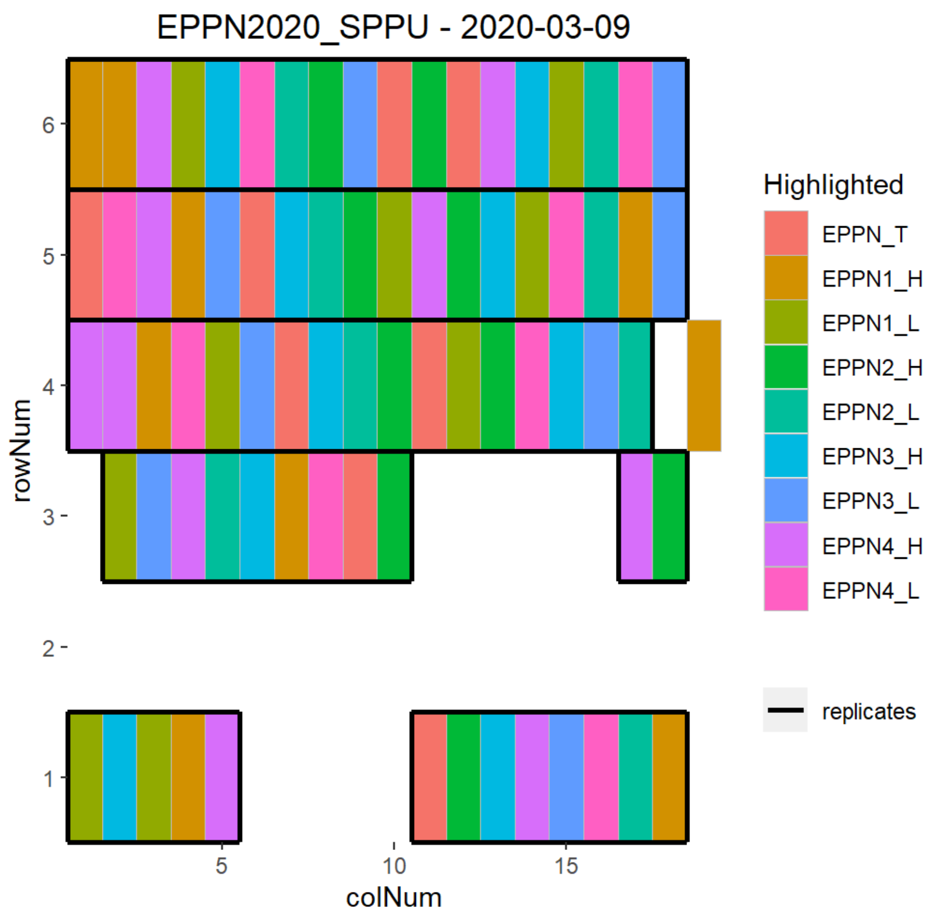

EPPN2020_SPPU - 2020-03-04
rowNum
colNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates

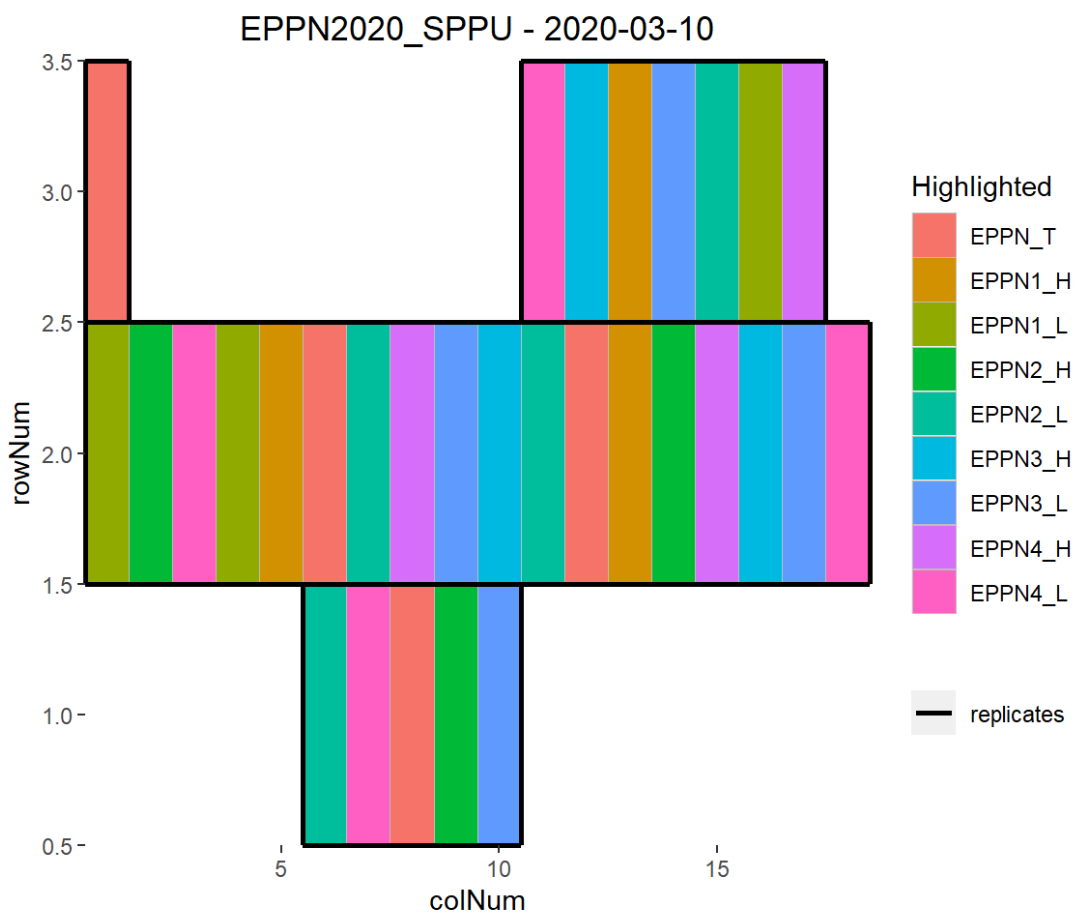

EPPN2020_SPPU - 2020-03-05
rowNum
colNum
6
5
4
3
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates


EPPN2020_SPPU - 2020-03-06
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates


EPPN2020_SPPU - 2020-03-07
rowNum
6
5
4
3
colNum
5
10
15
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates


EPPN2020_SPPU - 2020-03-08
3.5
3.0
2.5
2.0
1.5
1.0
0.5
rowNum
5
10
15
colNum
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates


EPPN2020_SPPU - 2020-03-09
6
5
4
3
2
1
rowNum
5
10
15
colNum
Highlighted
EPPN_T
EPPN1_H
EPPN1_L
EPPN2_H
EPPN2_L
EPPN3_H
EPPN3_L
EPPN4_H
EPPN4_L
replicates



1. endpoint

No data

2. S_timeseries

Raw data

View timePoint object

```
summary(timePoint_S)
```

```
## timePoint_S contains data for experiment EPPN2020_SPPU.
##
## It contains 23 time points.
## First time point: 2020-02-17
## Last time point: 2020-03-10
##
## The following genotypes are defined as check genotypes: EPPN_T.
```

```
getTimePoints(timePoint_S)
```

```
##    timeNumber  timePoint
## 1           1 2020-02-17
## 2           2 2020-02-18
## 3           3 2020-02-19
## 4           4 2020-02-20
## 5           5 2020-02-21
## 6           6 2020-02-22
## 7           7 2020-02-23
## 8           8 2020-02-24
## 9           9 2020-02-25
## 10         10 2020-02-26
## 11         11 2020-02-27
## 12         12 2020-02-28
## 13         13 2020-02-29
## 14         14 2020-03-01
## 15         15 2020-03-02
## 16         16 2020-03-03
## 17         17 2020-03-04
## 18         18 2020-03-05
## 19         19 2020-03-06
## 20         20 2020-03-07
## 21         21 2020-03-08
## 22         22 2020-03-09
## 23         23 2020-03-10
```

```
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.

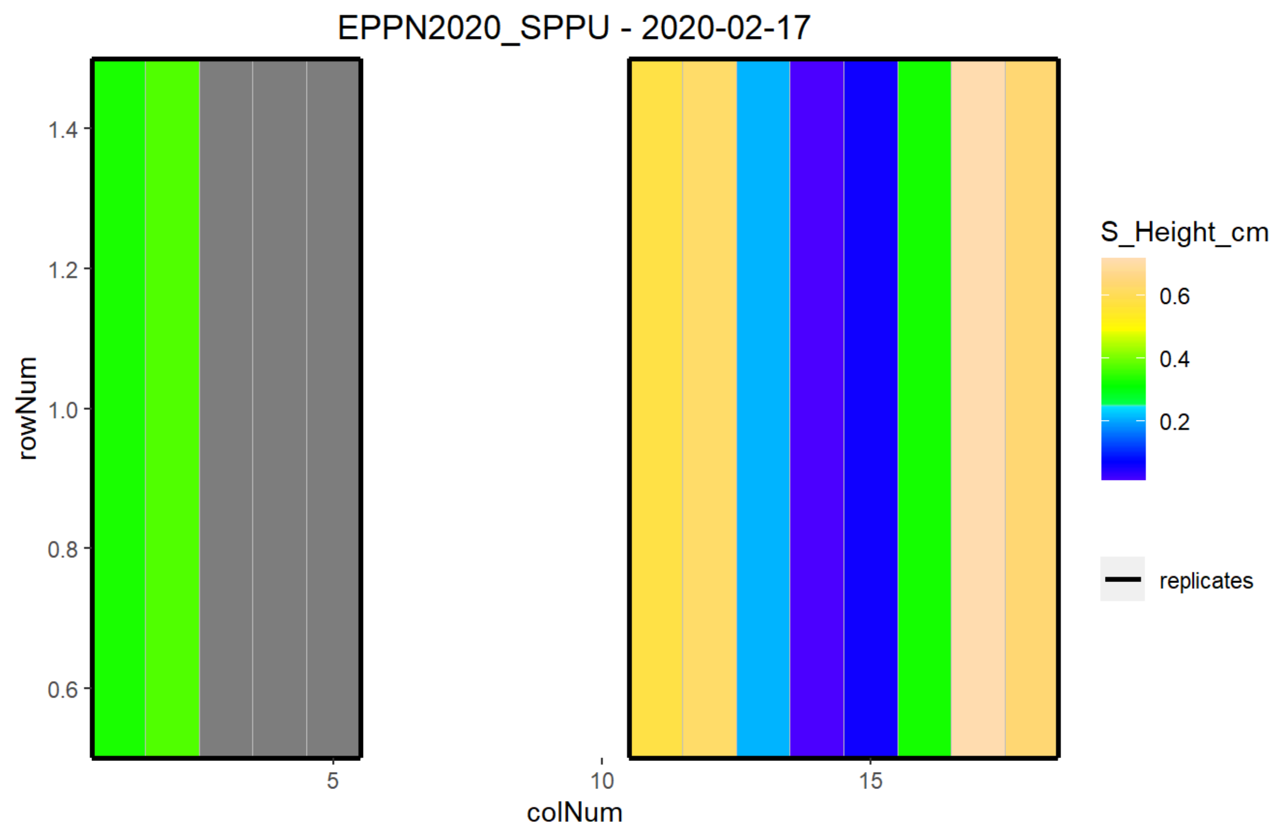
Height is computed for 6 platforms out of 9 and area for 4 out of 9.

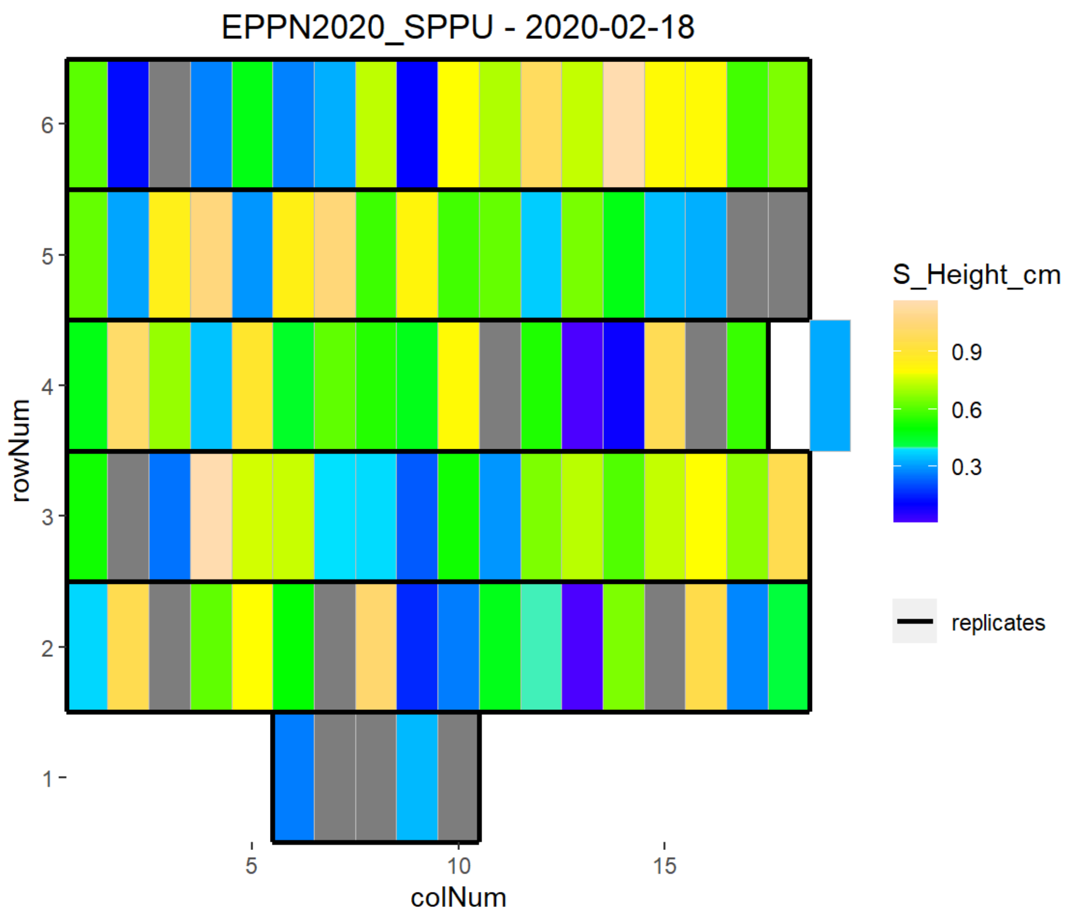
```
var_voulues <- c(variables_S[1], variables_S[3])
traits <- var_voulues

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

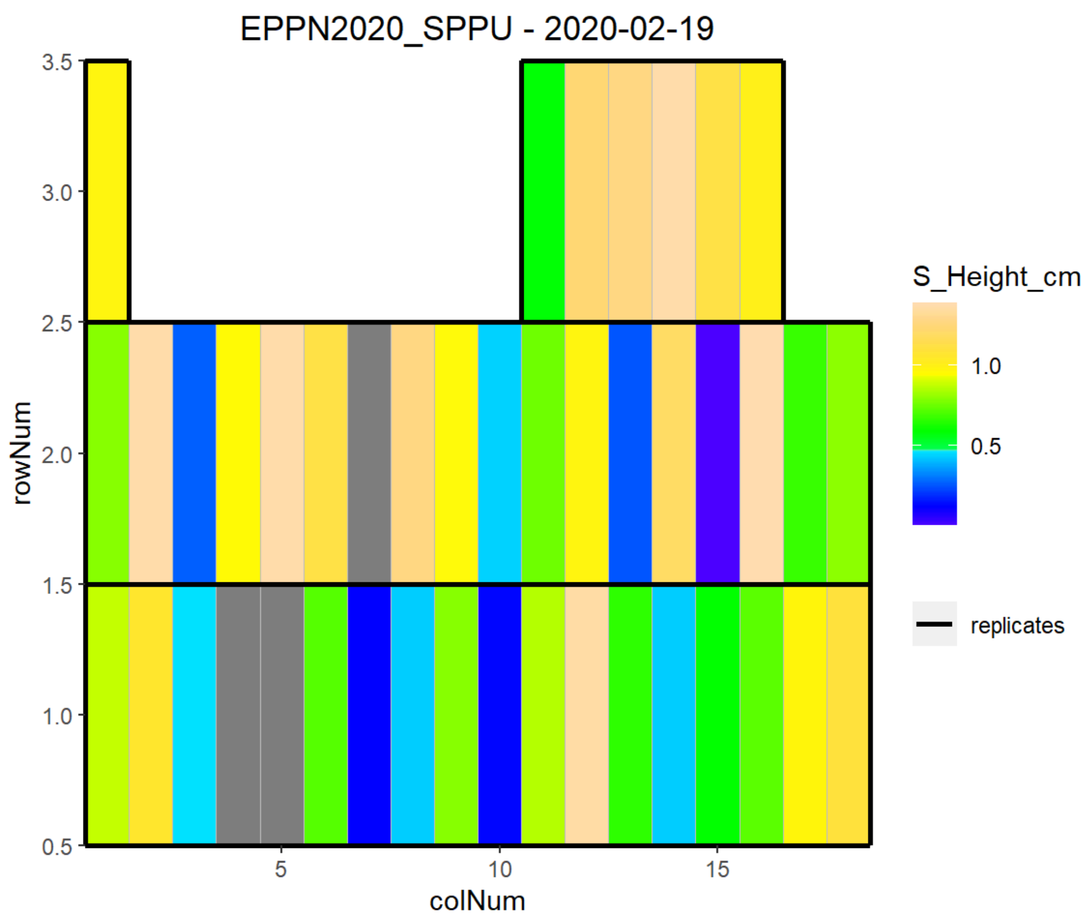
```
## [1] "How many observations for S_Height_cm"
## 2020-02-17 2020-02-18 2020-02-19 2020-02-20 2020-02-21 2020-02-22 2020-02-23
##         10         83         40         64         42         65         42
## 2020-02-24 2020-02-25 2020-02-26 2020-02-27 2020-02-28 2020-02-29 2020-03-01
##         78         94         42         65         42         65         42
## 2020-03-02 2020-03-03 2020-03-04 2020-03-05 2020-03-06 2020-03-07 2020-03-08
##         78         87         42         65         42         65         42
## 2020-03-09 2020-03-10
##         78         30
## [1] "How many observations for S_Area_cmsquared"
## 2020-02-17 2020-02-18 2020-02-19 2020-02-20 2020-02-21 2020-02-22 2020-02-23
##         10         83         40         64         42         65         42
## 2020-02-24 2020-02-25 2020-02-26 2020-02-27 2020-02-28 2020-02-29 2020-03-01
##         78         94         42         65         42         65         42
## 2020-03-02 2020-03-03 2020-03-04 2020-03-05 2020-03-06 2020-03-07 2020-03-08
##         78         87         42         65         42         65         42
## 2020-03-09 2020-03-10
##         78         30
```

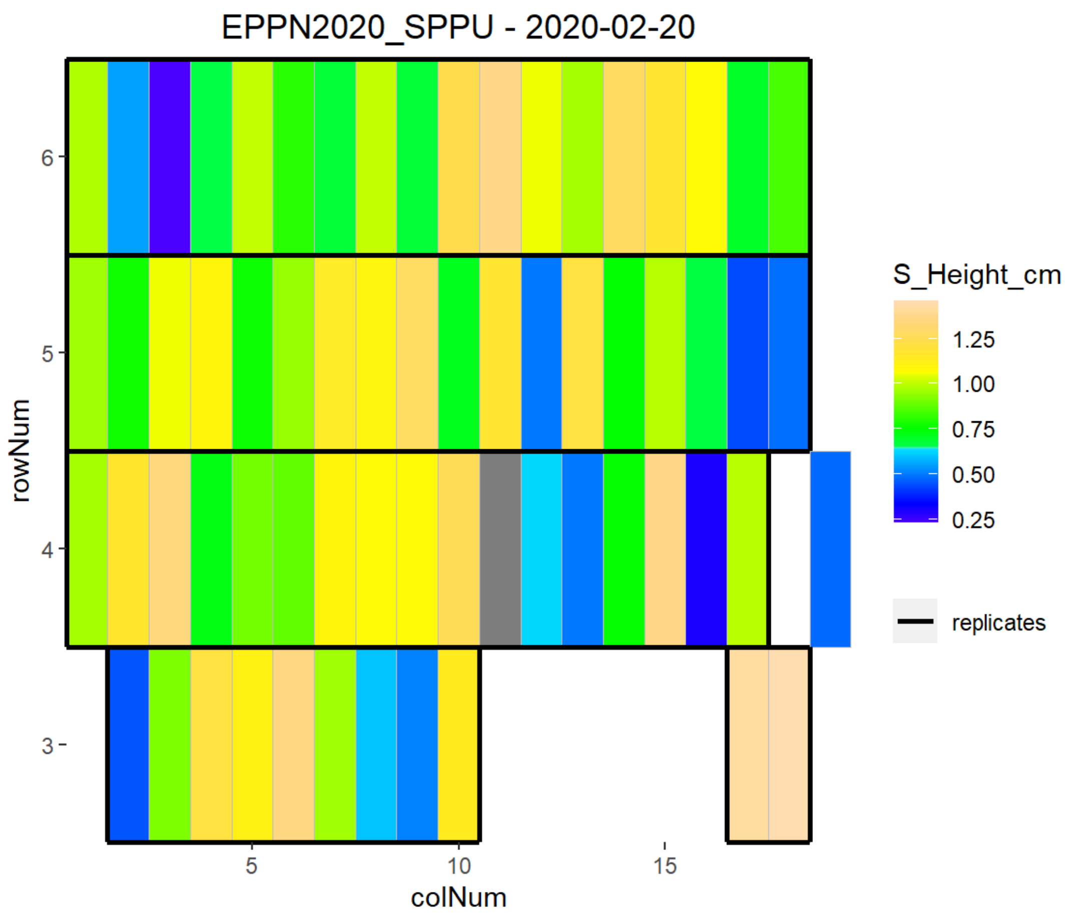
Check the heatmap of the raw data per time point

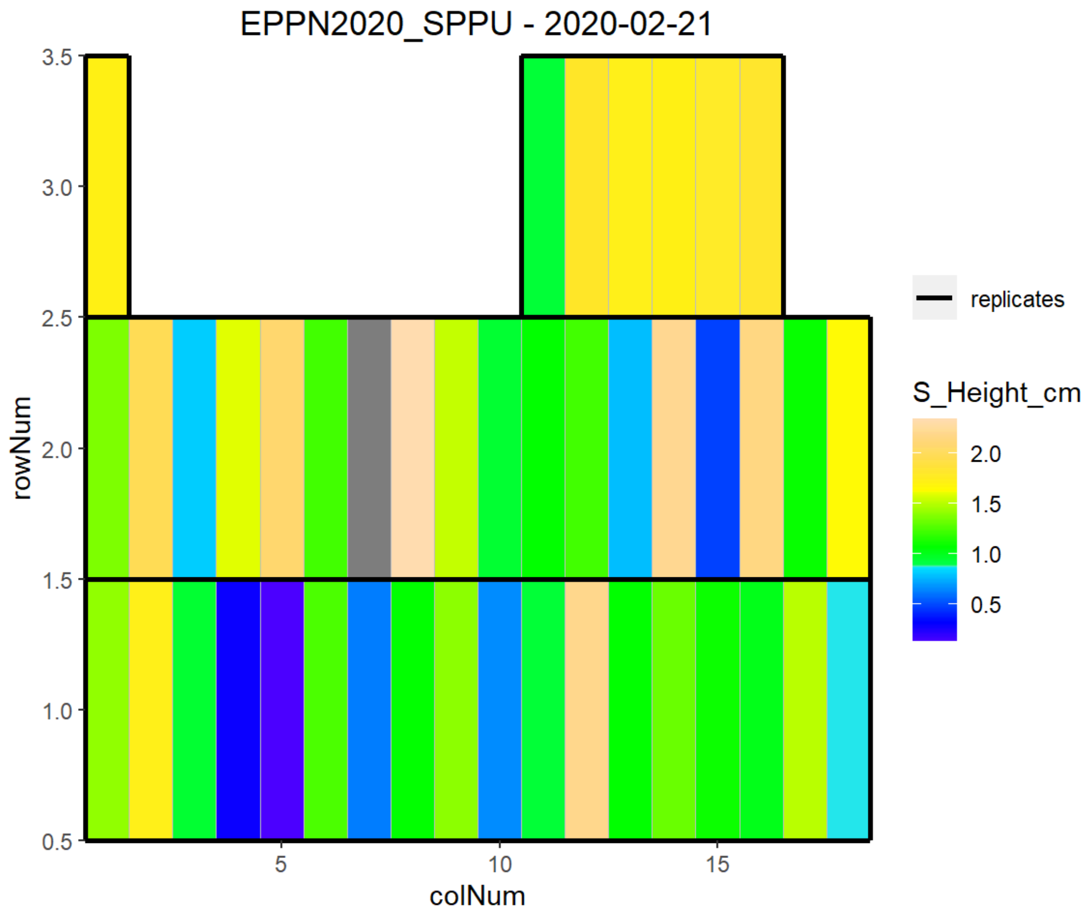
```
for (trait_name in traits) {
  for (tp in 1:length(num_timepoints$timeNumber)) {
    plot(timePoint_S,
         plotType = "layout",
         timePoints = tp,
         traits = trait_name)
  }
}
```

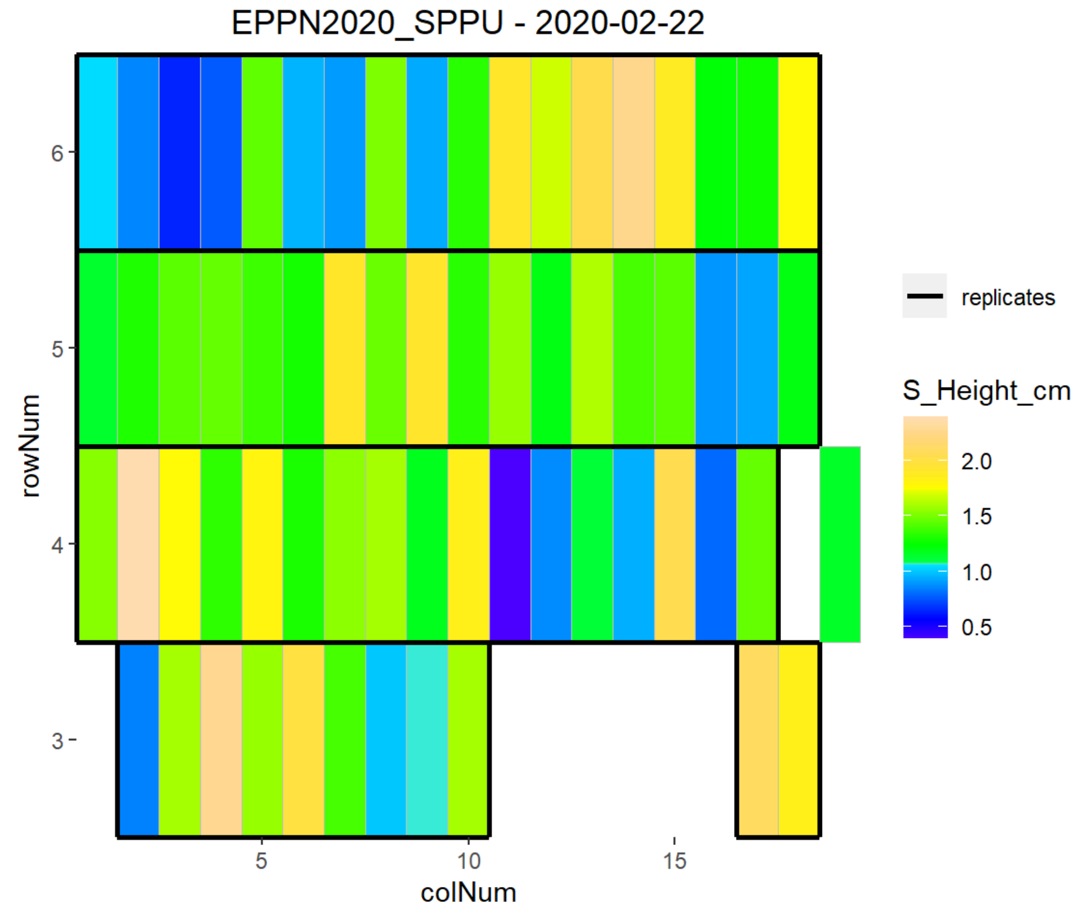

EPPN2020_SPPU - 2020-02-17
rowNum
1.4
1.2
1.0
0.8
0.6
colNum
5
10
15
S_Height_cm
0.6
0.4
0.2
replicates

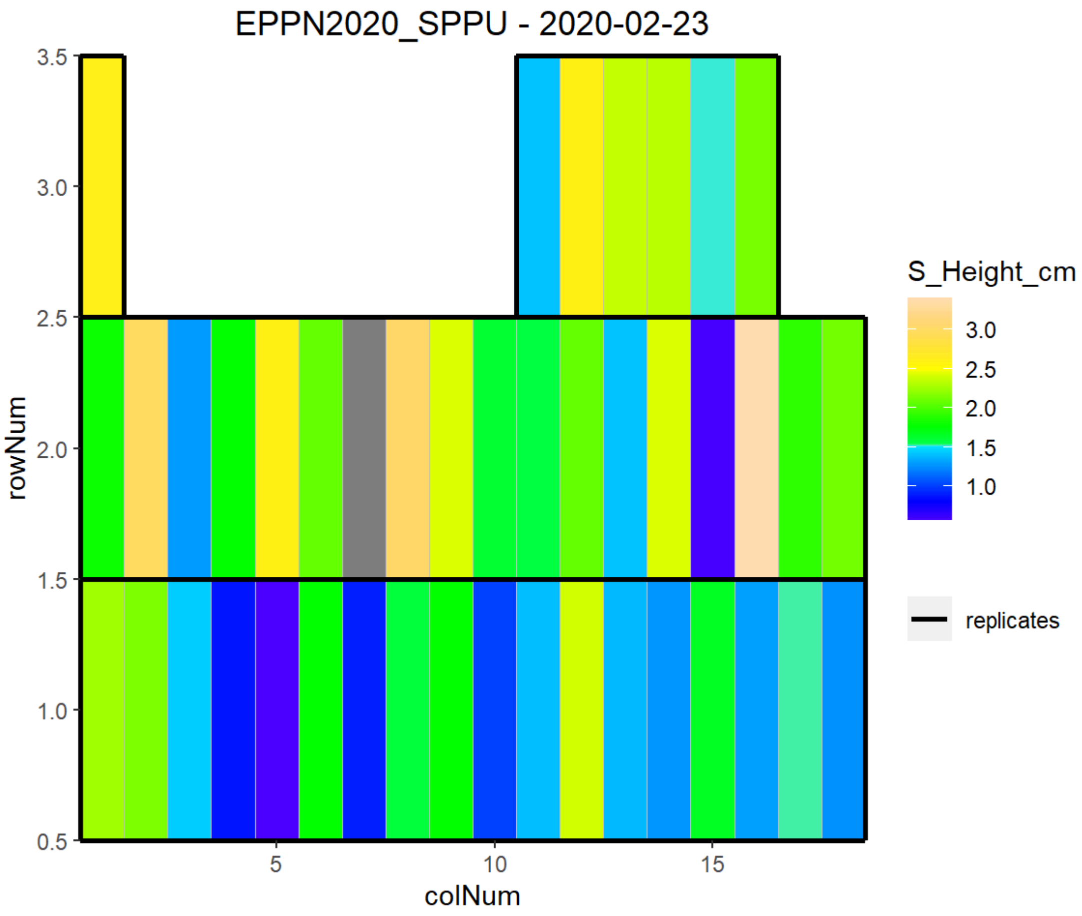

EPPN2020_SPPU - 2020-02-18
rowNum
6
5
4
3
2
1
colNum
5
10
15
S_Height_cm
0.9
0.6
0.3
replicates

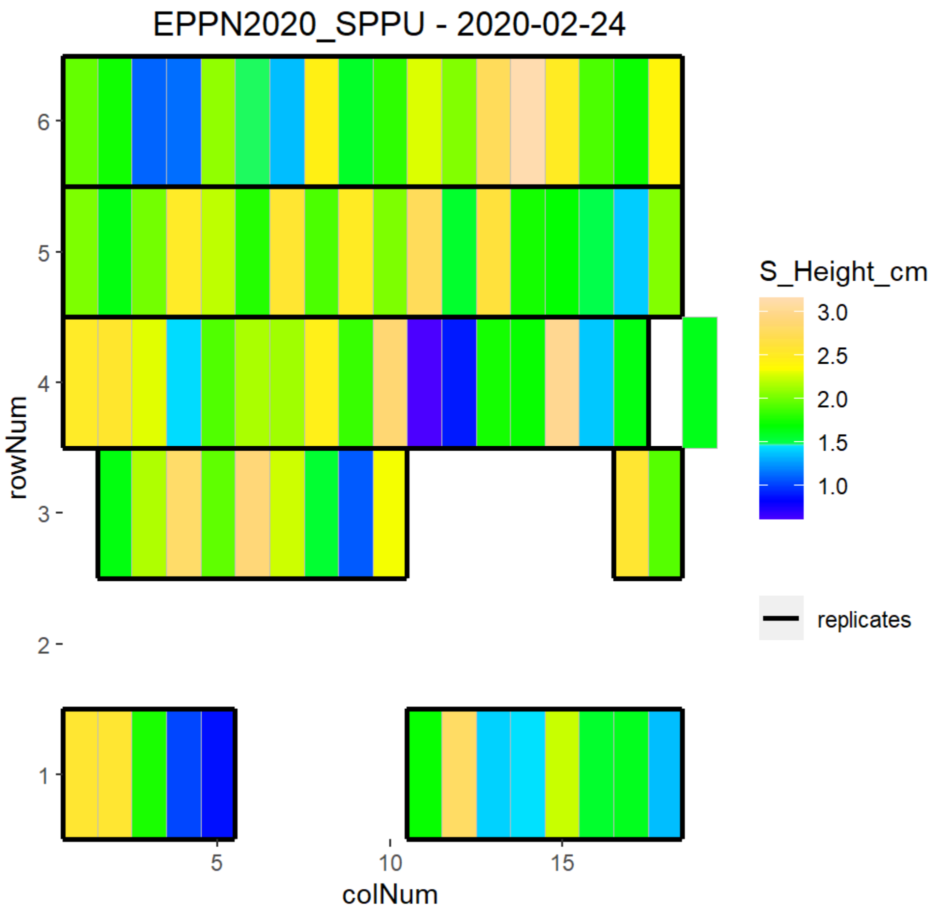

EPPN2020_SPPU - 2020-02-19
3.5
3.0
2.5
2.0
1.5
1.0
0.5
rowNum
5
10
15
colNum
S_Height_cm
1.0
0.5
replicates

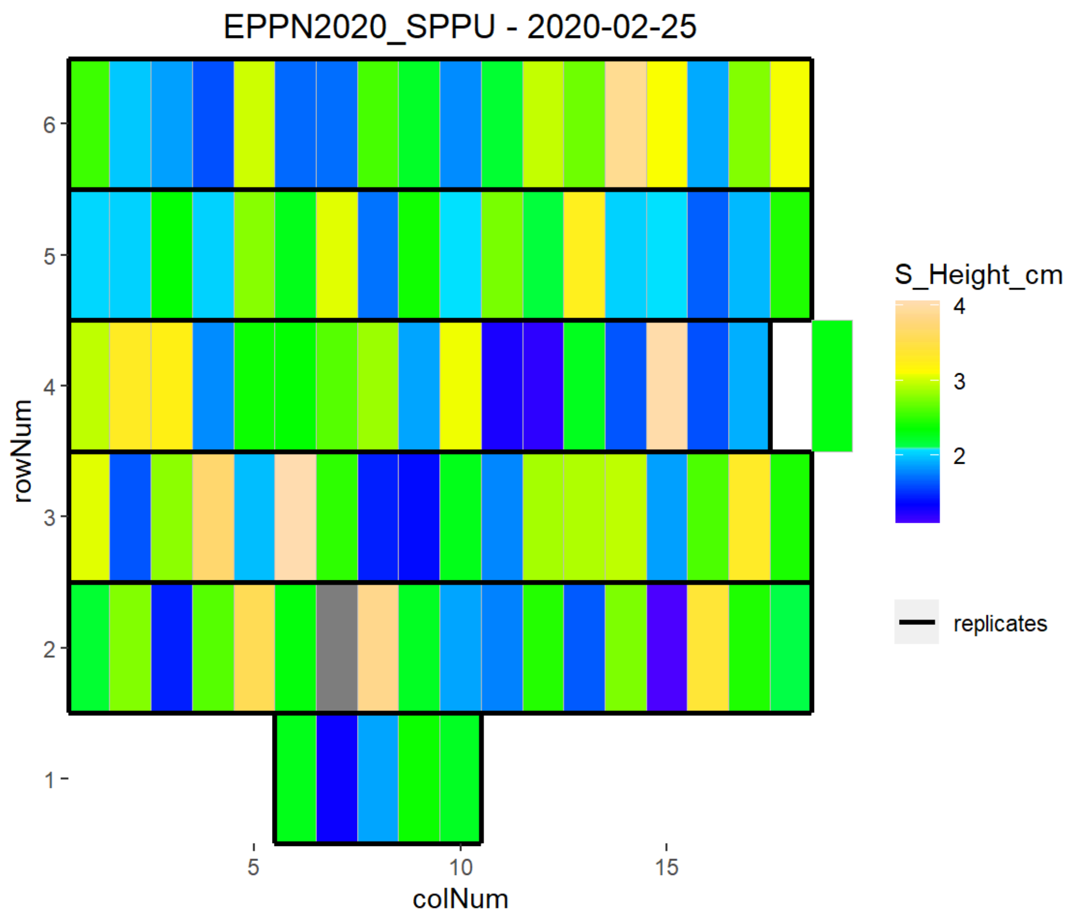

EPPN2020_SPPU - 2020-02-20
6
5
4
3
rowNum
5
10
15
colNum
S_Height_cm
1.25
1.00
0.75
0.50
0.25
replicates

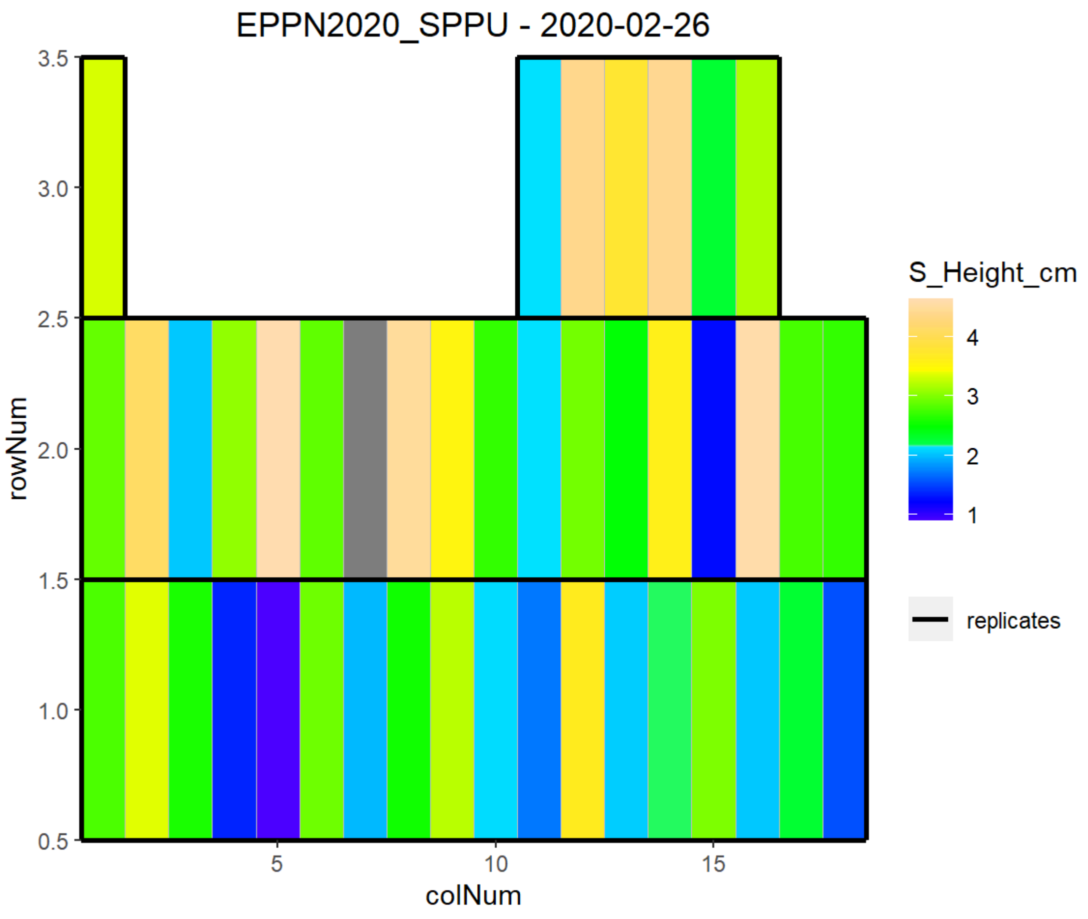

EPPN2020_SPPU - 2020-02-21
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
replicates
S_Height_cm
2.0
1.5
1.0
0.5

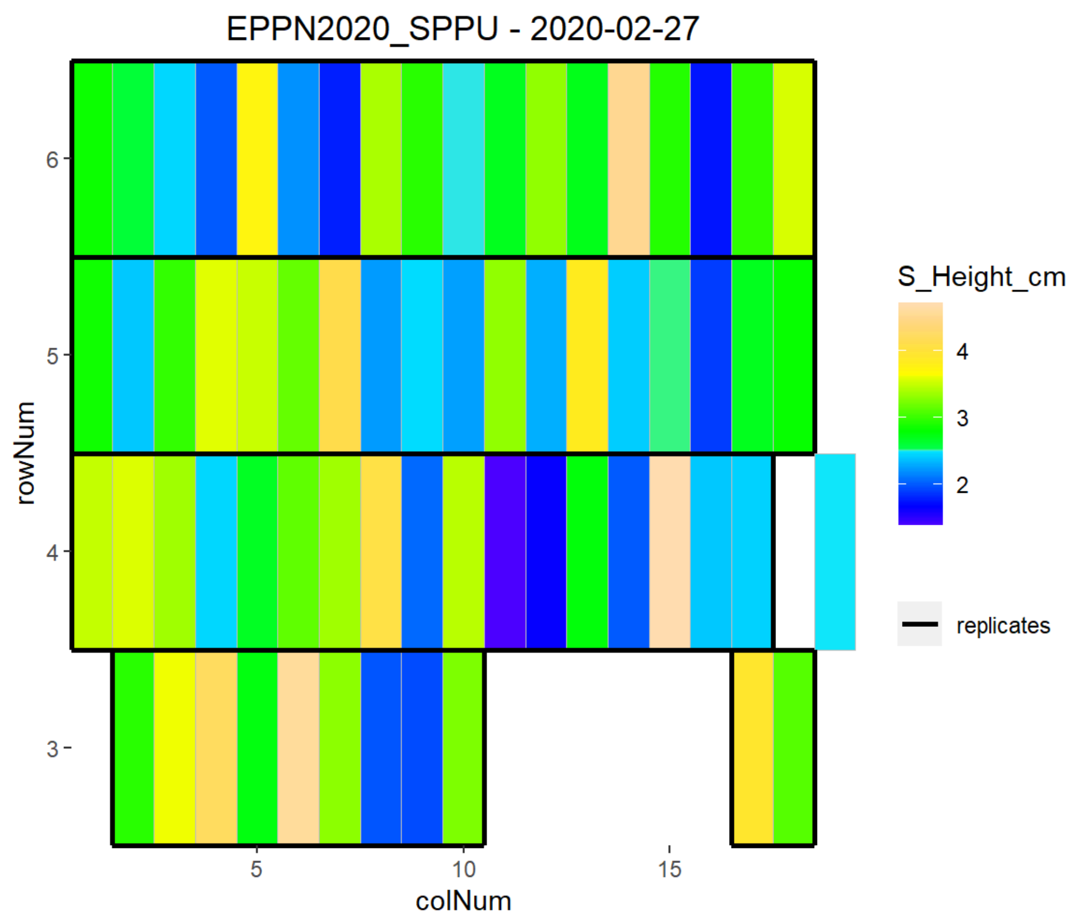

EPPN2020_SPPU - 2020-02-22
rowNum
6
5
4
3
colNum
5
10
15
replicates
S_Height_cm
2.0
1.5
1.0
0.5

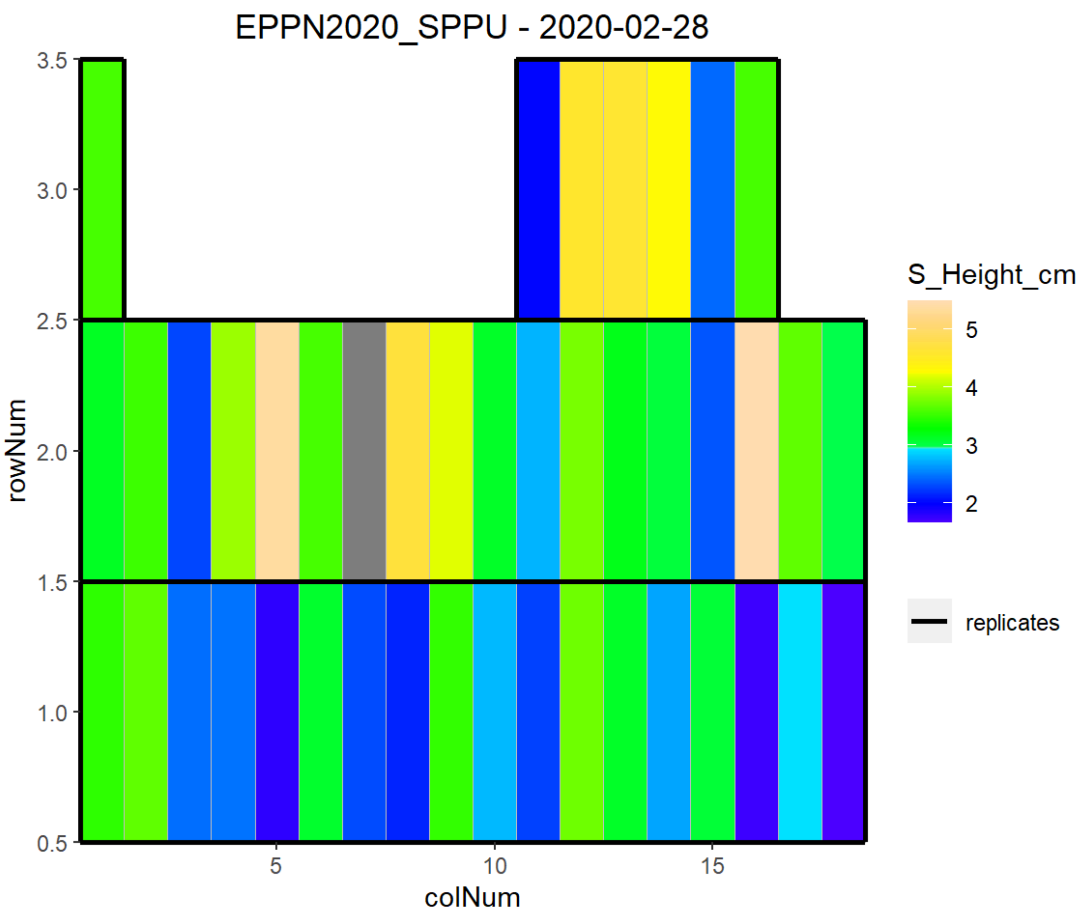

EPPN2020_SPPU - 2020-02-23
rowNum
colNum
S_Height_cm
3.0
2.5
2.0
1.5
1.0
replicates

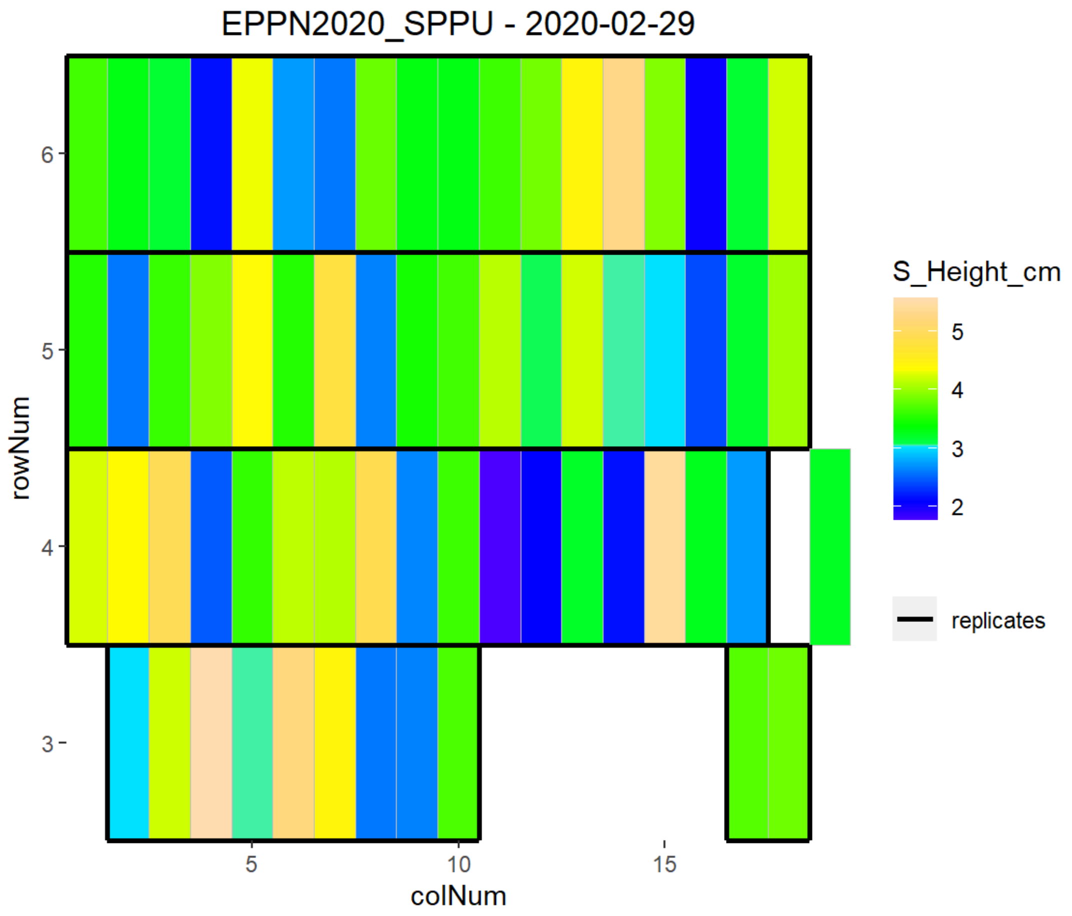

EPPN2020_SPPU - 2020-02-24
rowNum
colNum
S_Height_cm
3.0
2.5
2.0
1.5
1.0
replicates

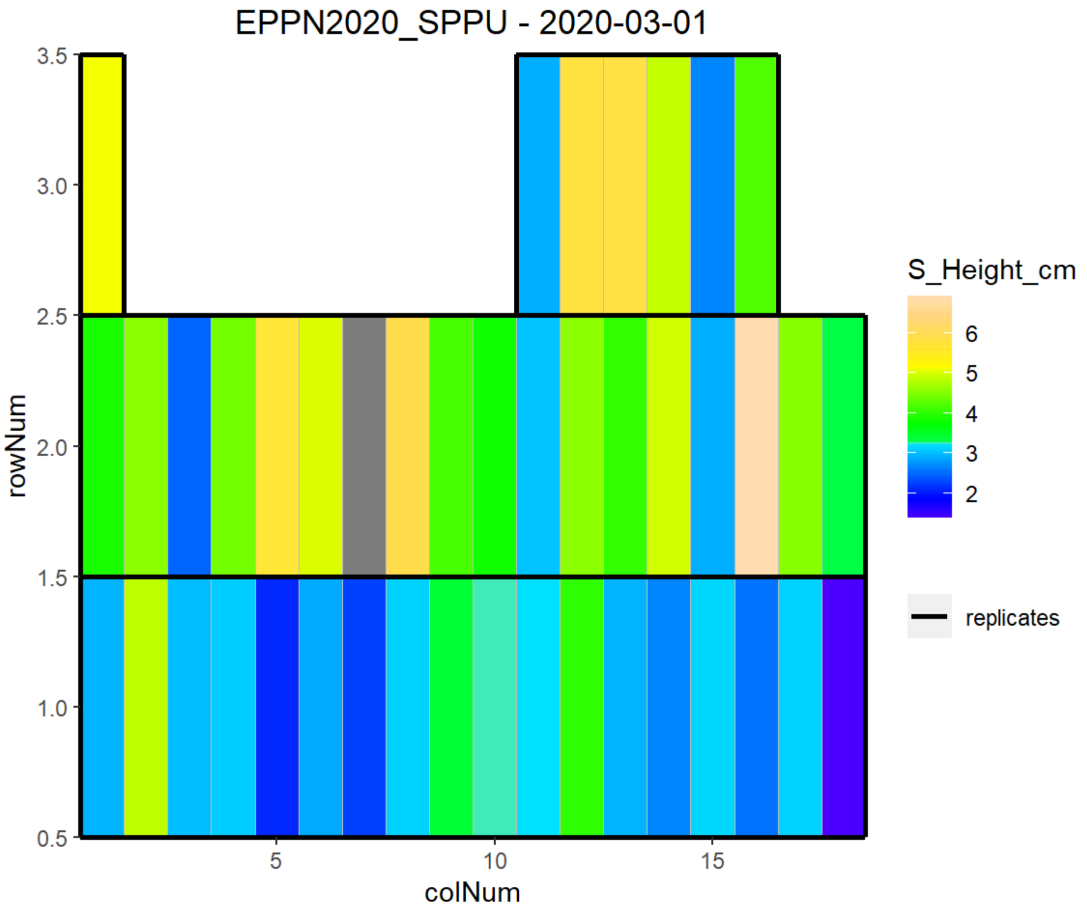

EPPN2020_SPPU - 2020-02-25
rowNum
1
2
3
4
5
6
colNum
5
10
15
S_Height_cm
4
3
2
replicates

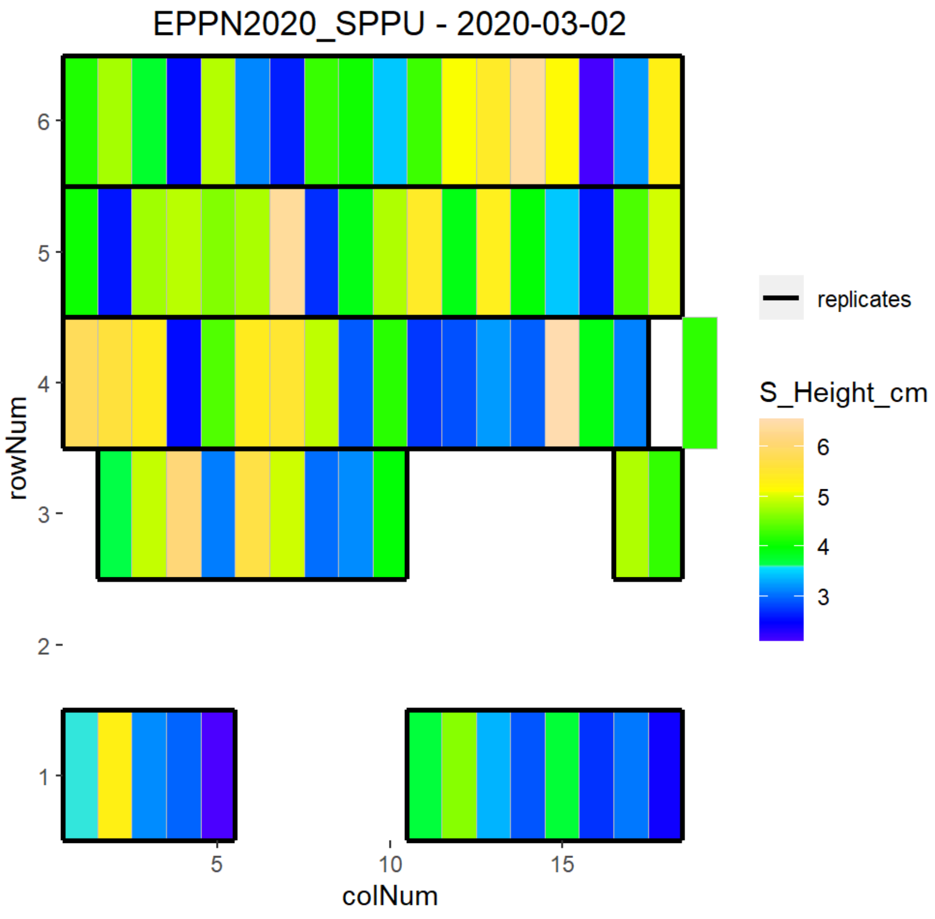

EPPN2020_SPPU - 2020-02-26
rowNum
0.5
1.0
1.5
2.0
2.5
3.0
3.5
colNum
5
10
15
S_Height_cm
4
3
2
1
replicates

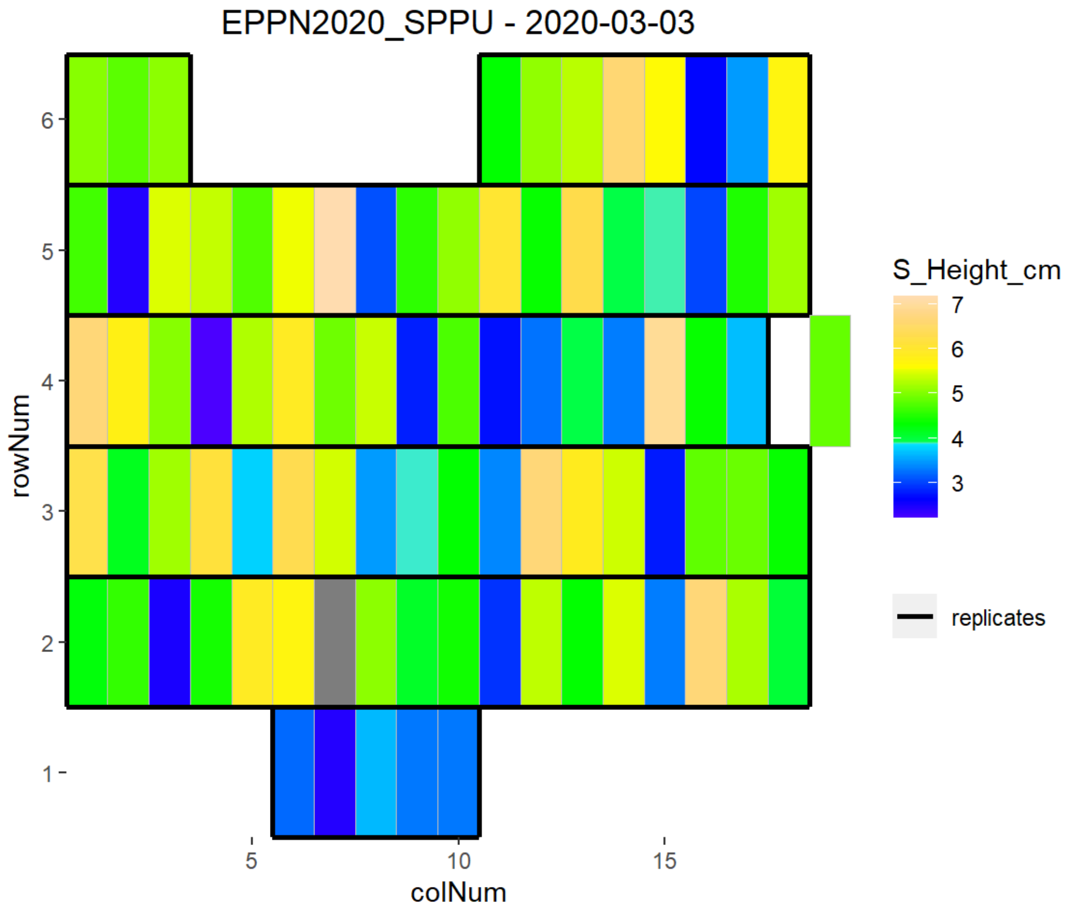

EPPN2020_SPPU - 2020-02-27
rowNum
6
5
4
3
colNum
5
10
15
S_Height_cm
4
3
2
replicates

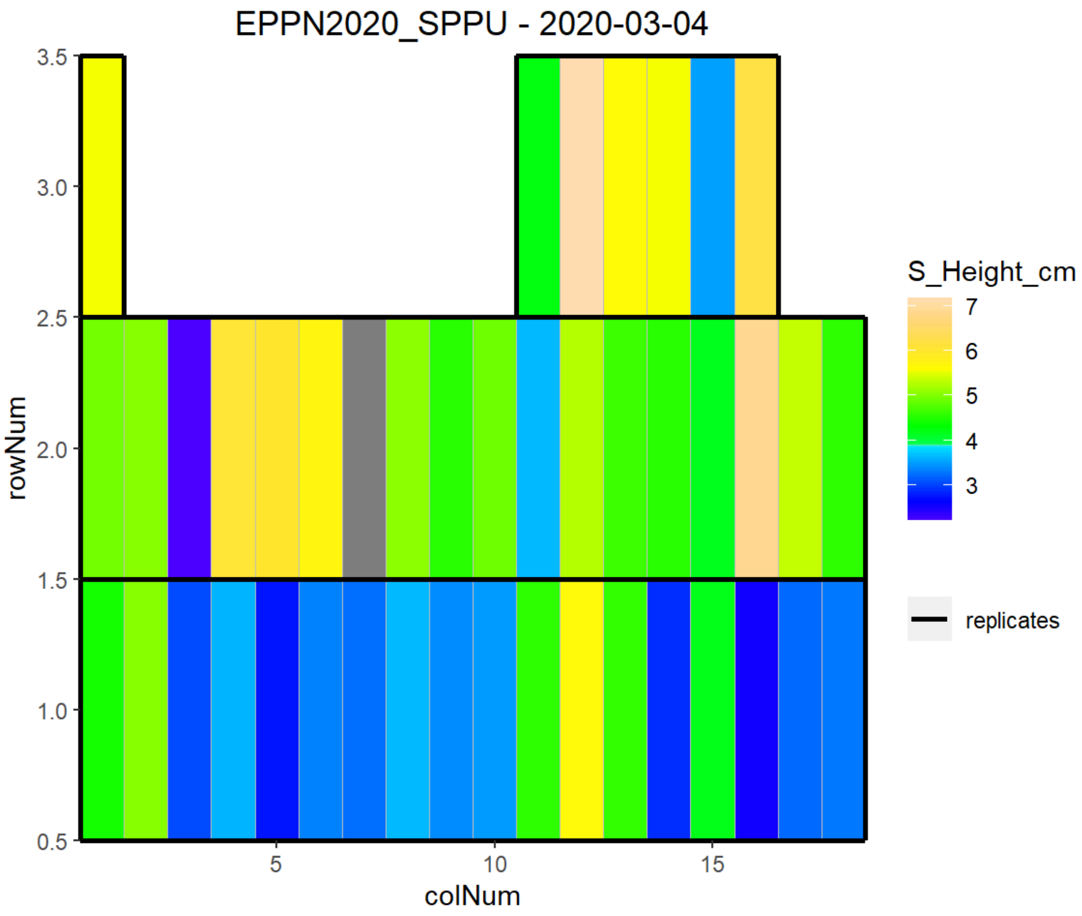

EPPN2020_SPPU - 2020-02-28
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
S_Height_cm
5
4
3
2
replicates

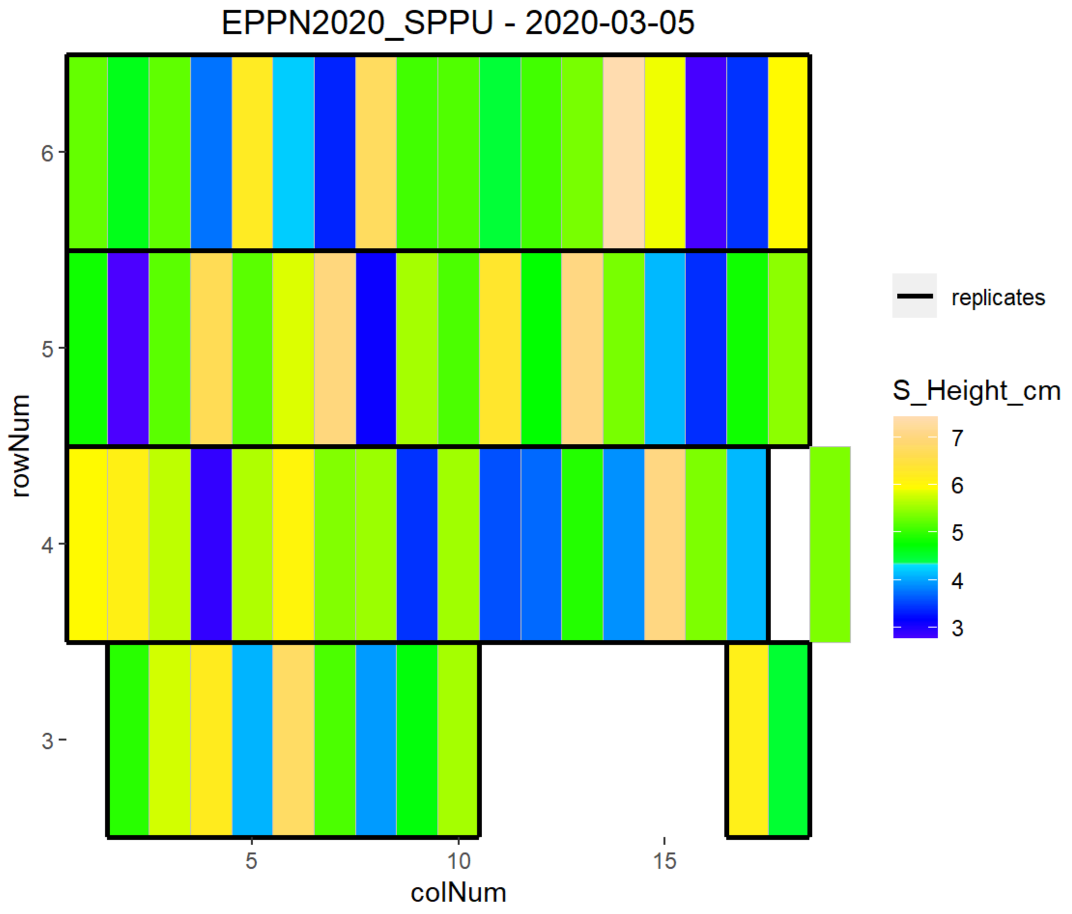

EPPN2020_SPPU - 2020-02-29
rowNum
6
5
4
3
colNum
5
10
15
S_Height_cm
5
4
3
2
replicates

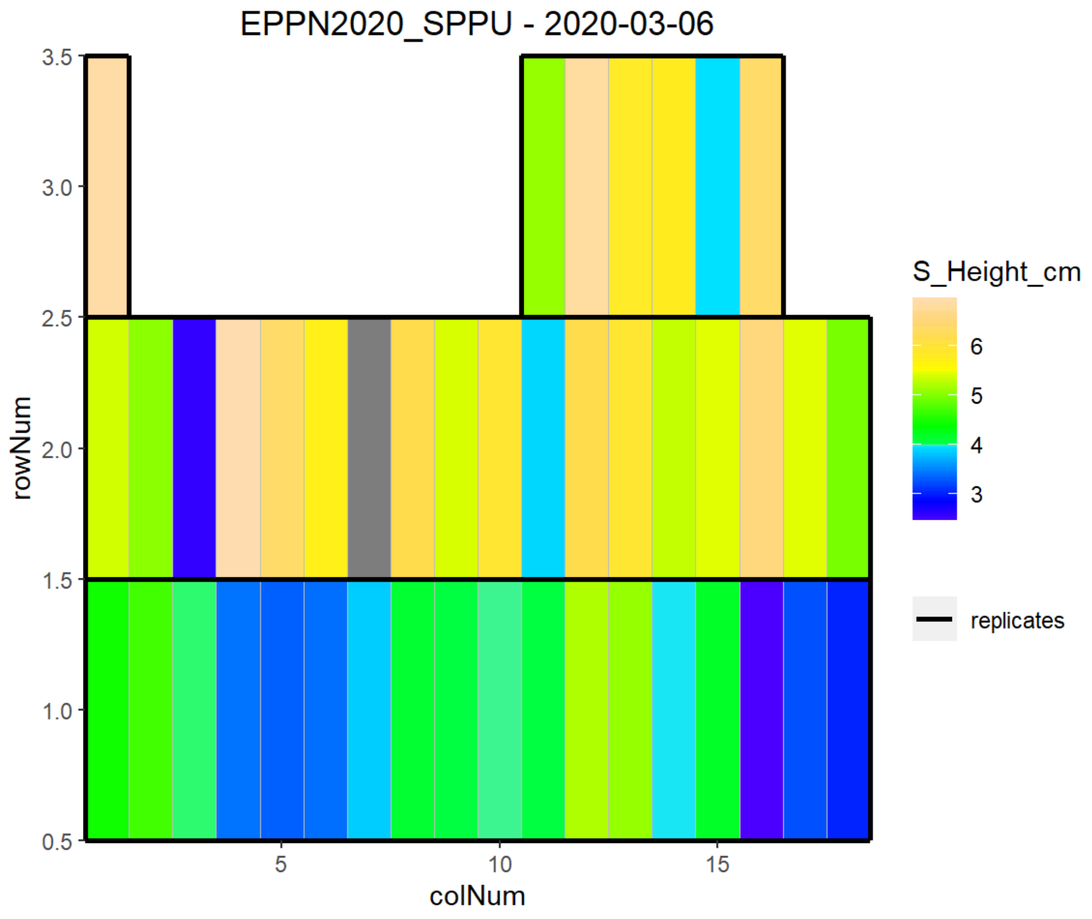

EPPN2020_SPPU - 2020-03-01
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
S_Height_cm
6
5
4
3
2
replicates

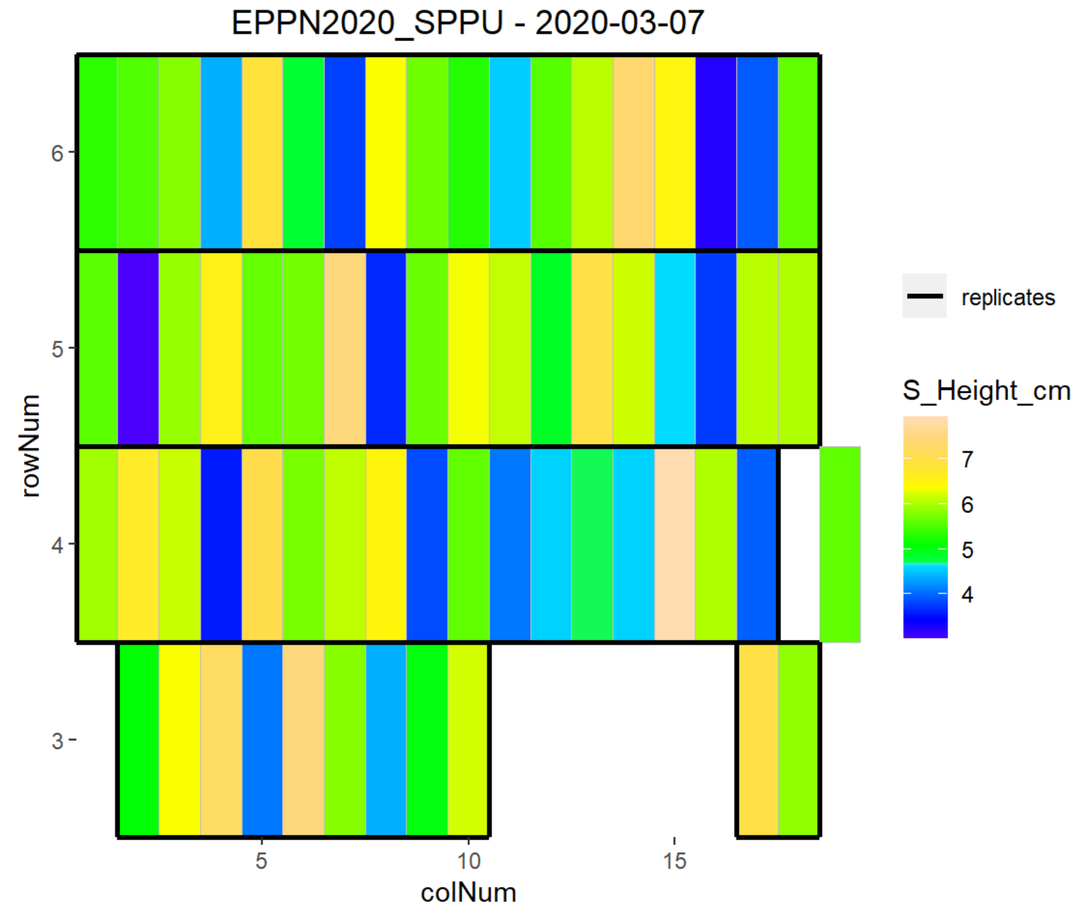

EPPN2020_SPPU - 2020-03-02
rowNum
6
5
4
3
2
1
colNum
5
10
15
replicates
S_Height_cm
6
5
4
3

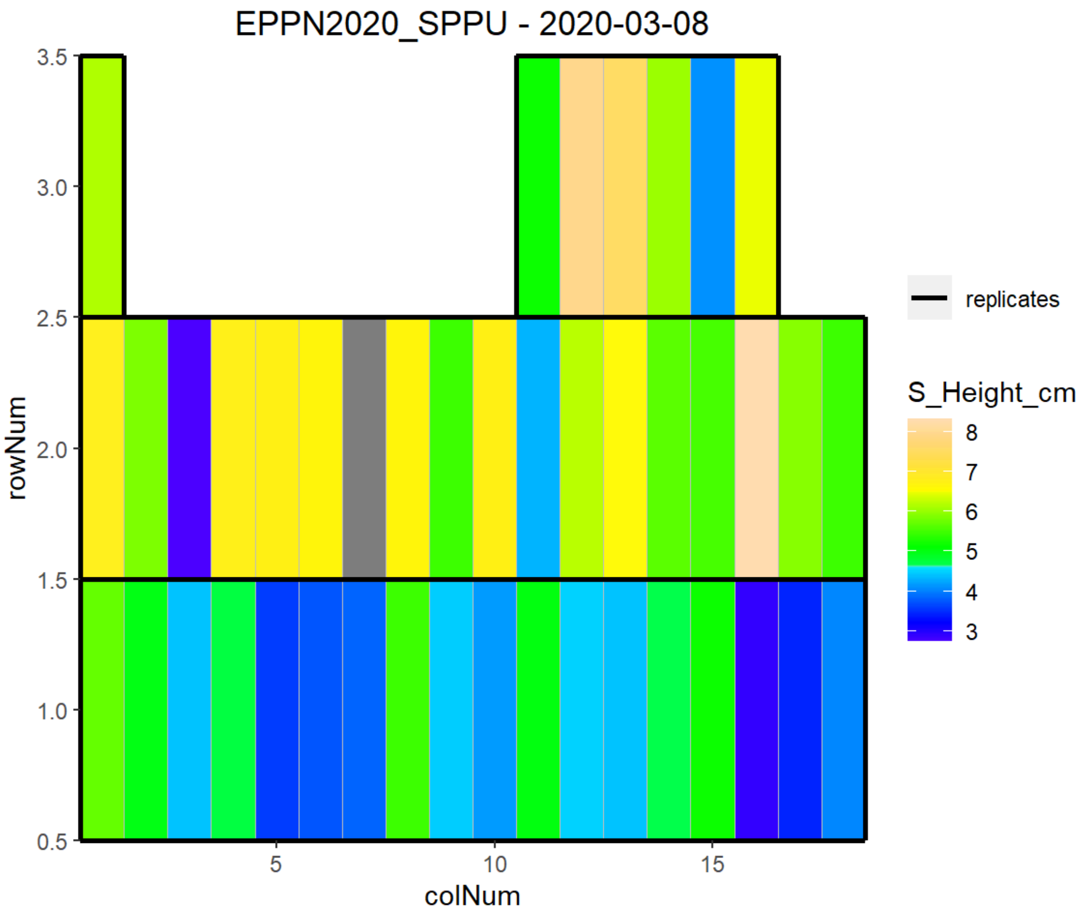

EPPN2020_SPPU - 2020-03-03
rowNum
6
5
4
3
2
1
colNum
5
10
15
S_Height_cm
7
6
5
4
3
replicates

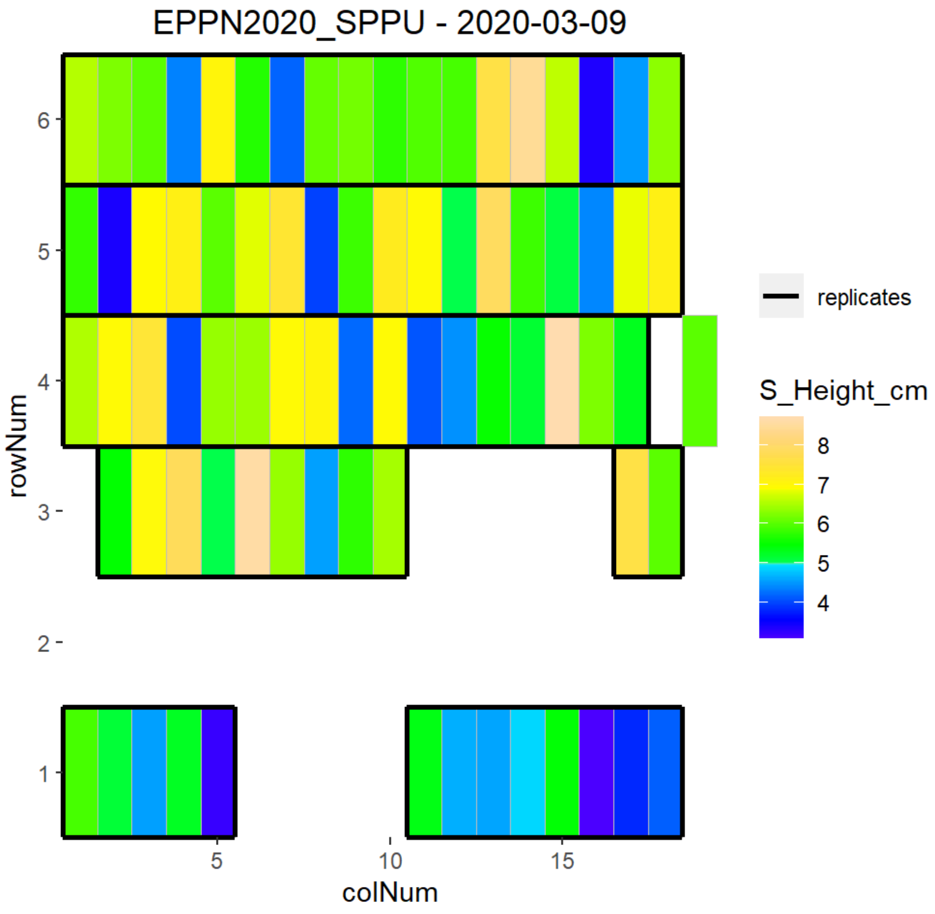

EPPN2020_SPPU - 2020-03-04
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
S_Height_cm
7
6
5
4
3
replicates

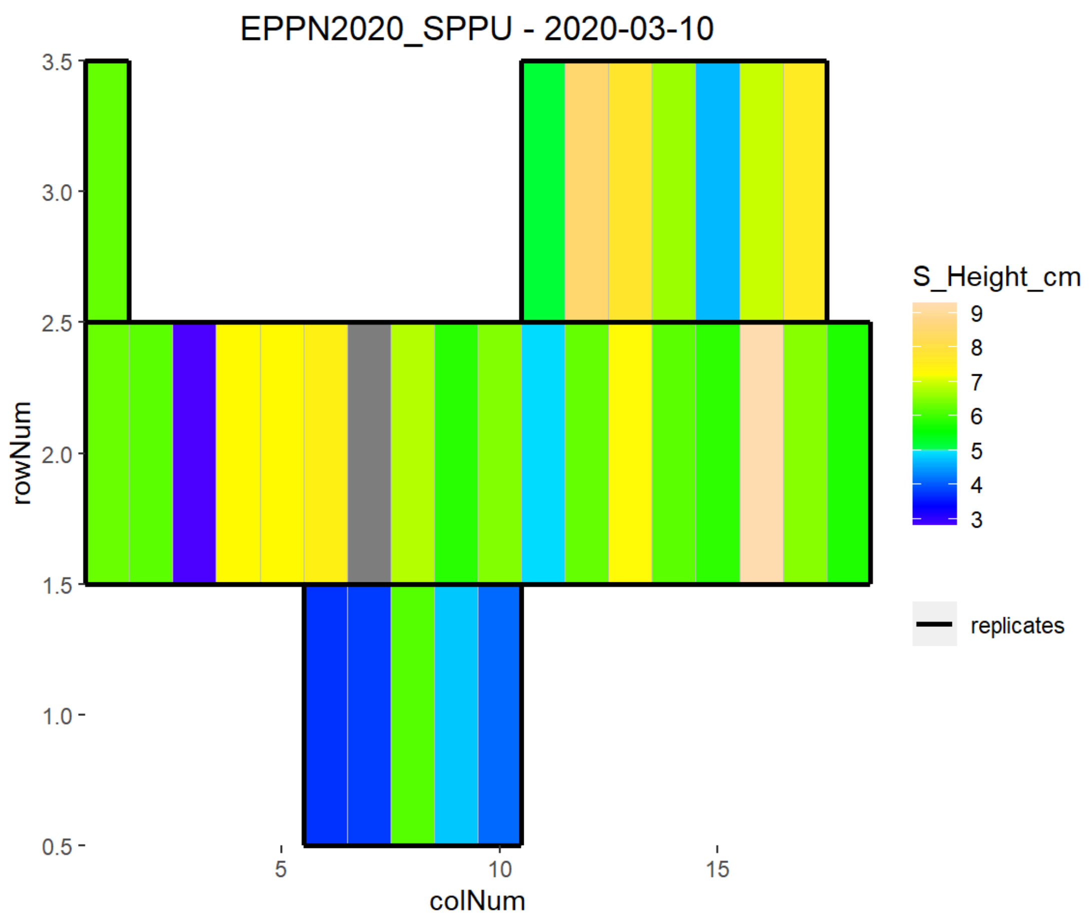

EPPN2020_SPPU - 2020-03-05
rowNum
6
5
4
3
colNum
5
10
15
replicates
S_Height_cm
7
6
5
4
3

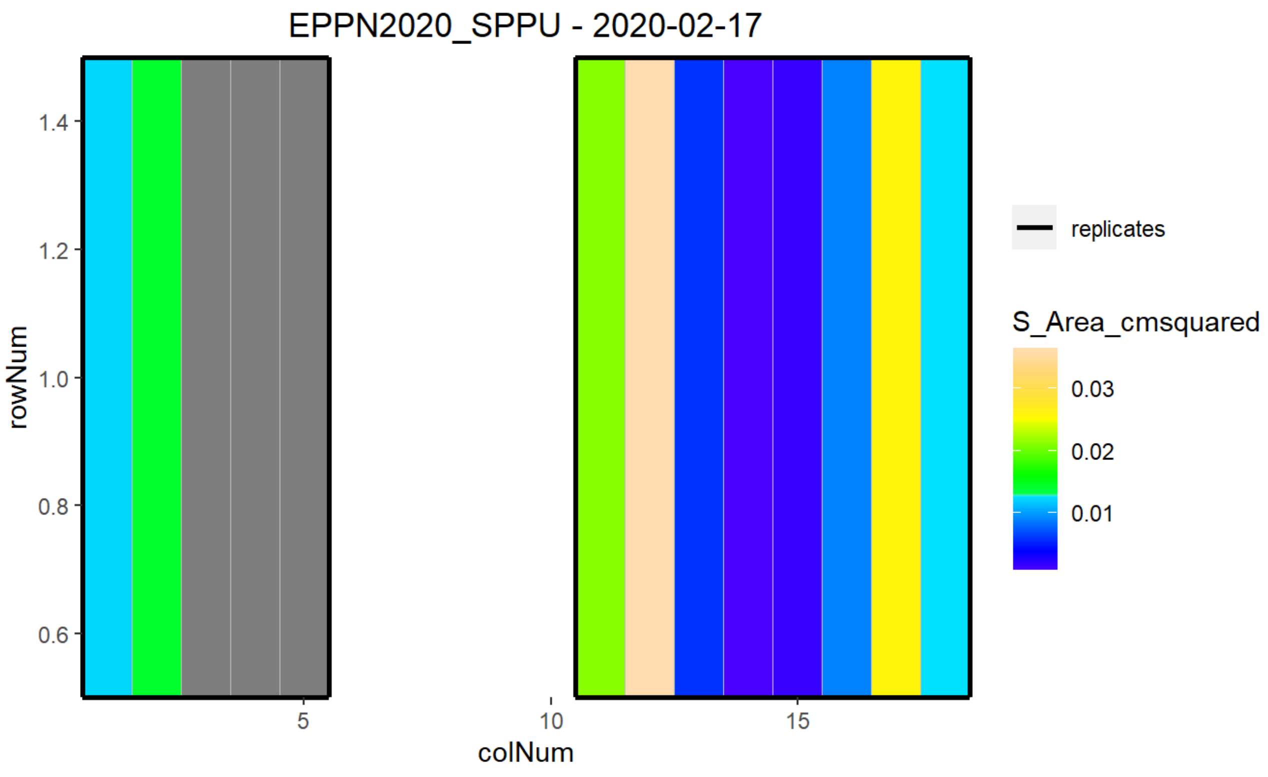

EPPN2020_SPPU - 2020-03-06
rowNum
colNum
S_Height_cm
6
5
4
3
replicates

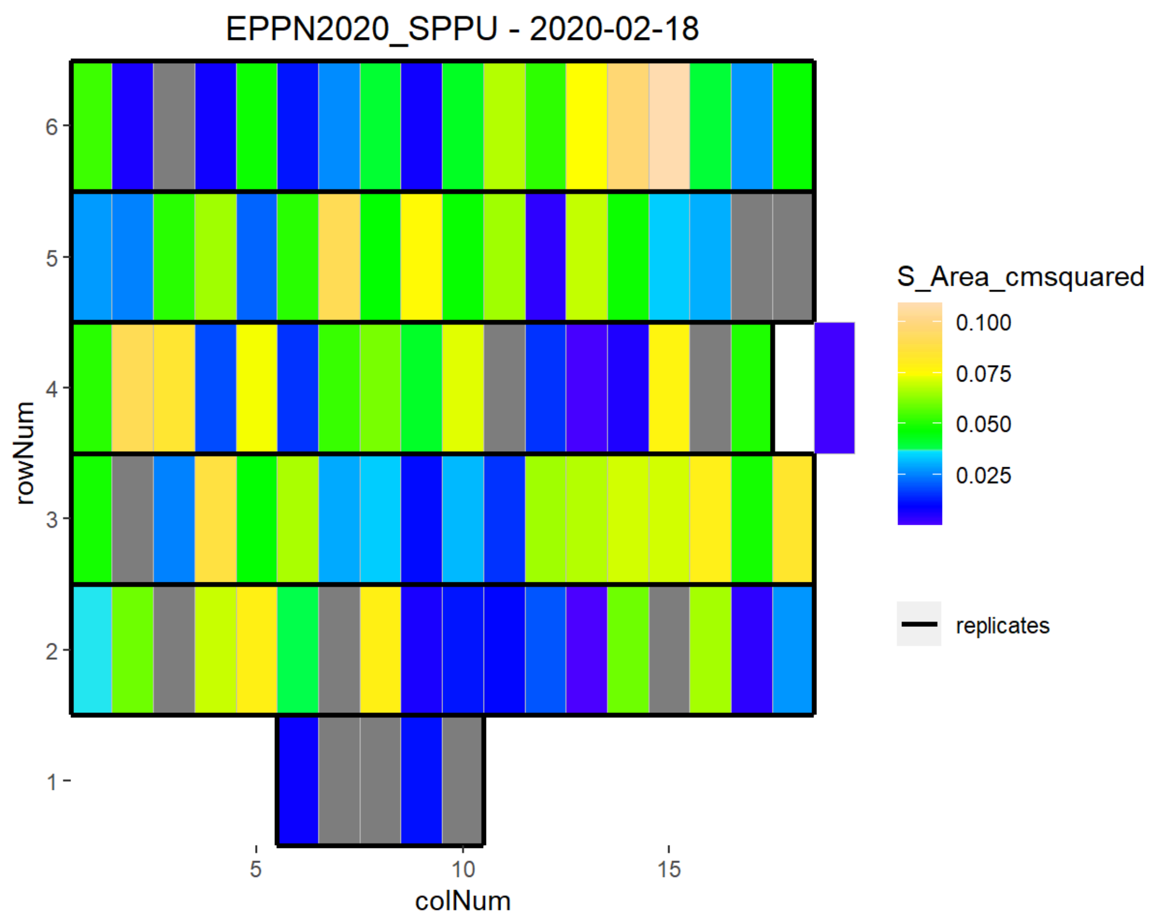

EPPN2020_SPPU - 2020-03-07
rowNum
colNum
replicates
S_Height_cm
7
6
5
4

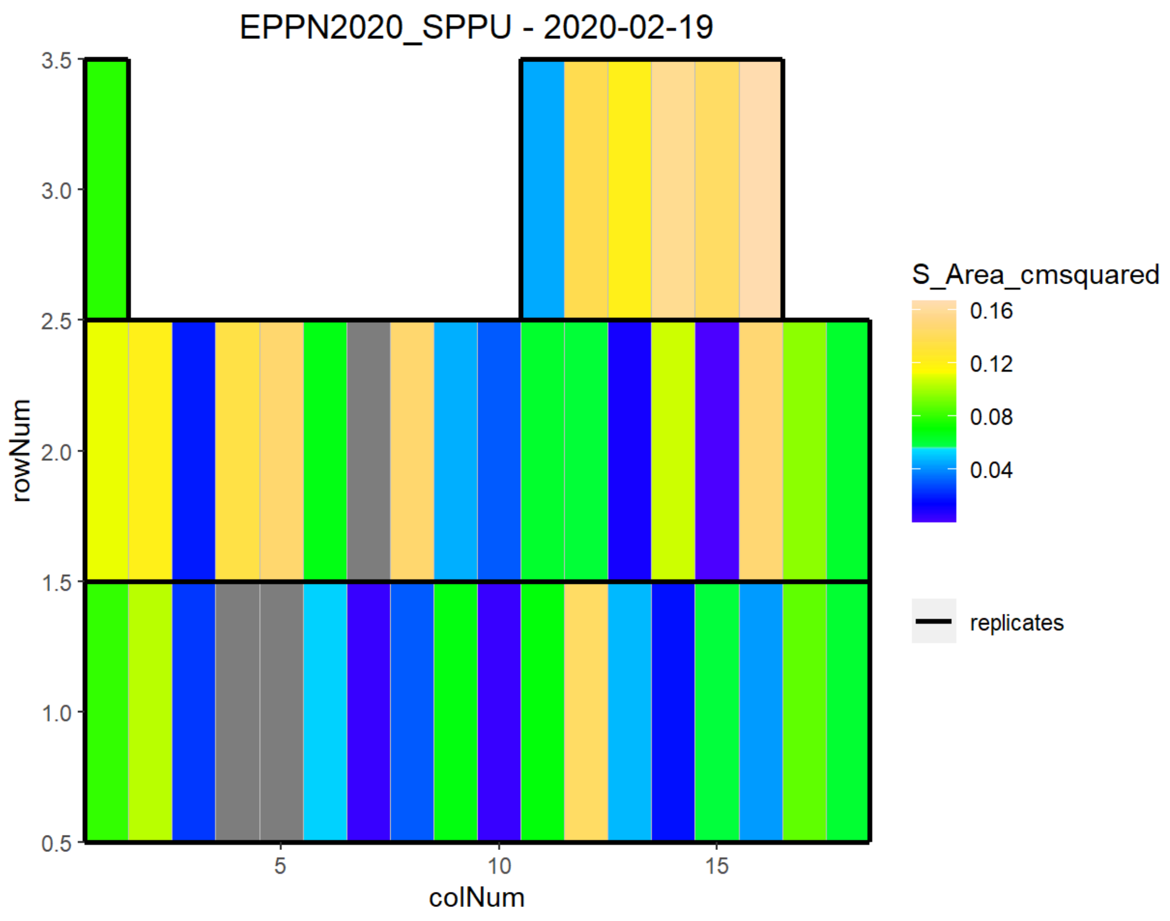

EPPN2020_SPPU - 2020-03-08
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
replicates
S_Height_cm
8
7
6
5
4
3

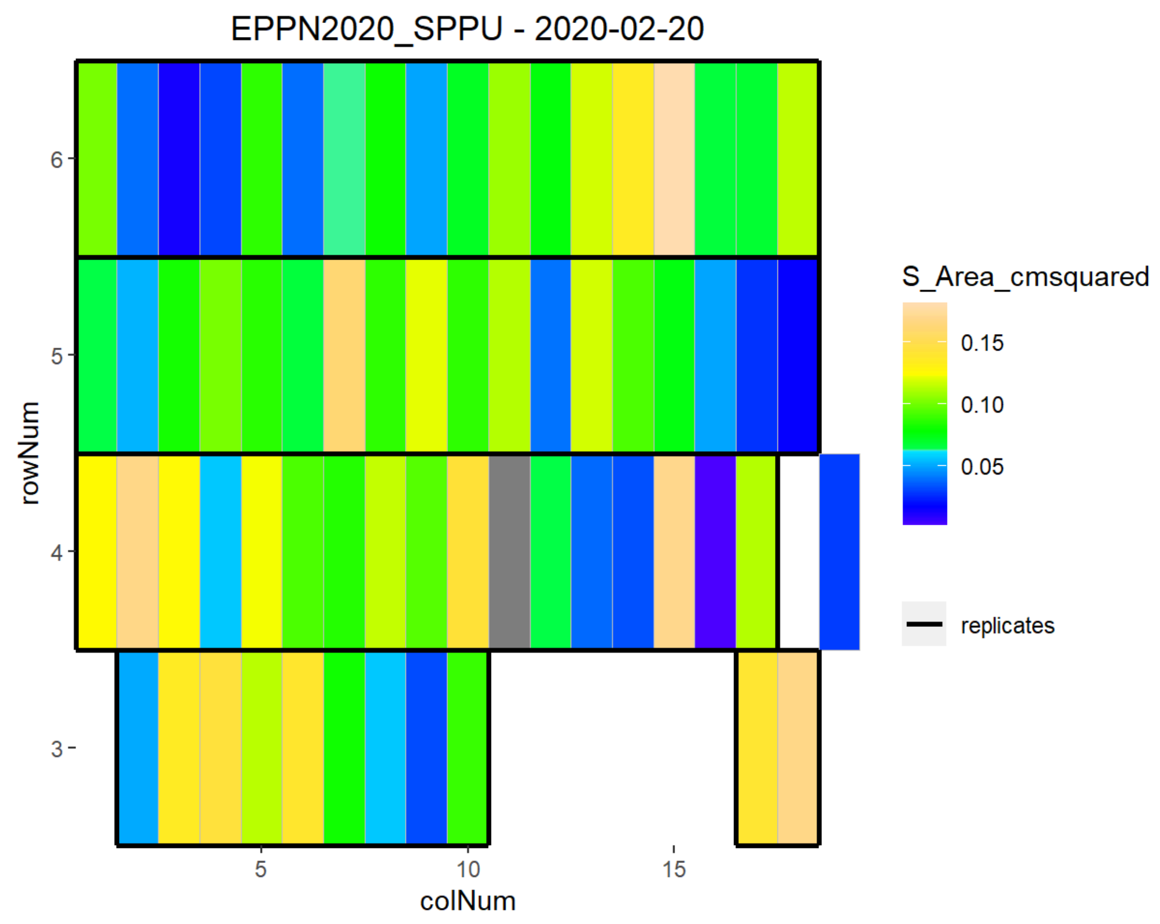

EPPN2020_SPPU - 2020-03-09
rowNum
6
5
4
3
2
1
colNum
5
10
15
replicates
S_Height_cm
8
7
6
5
4

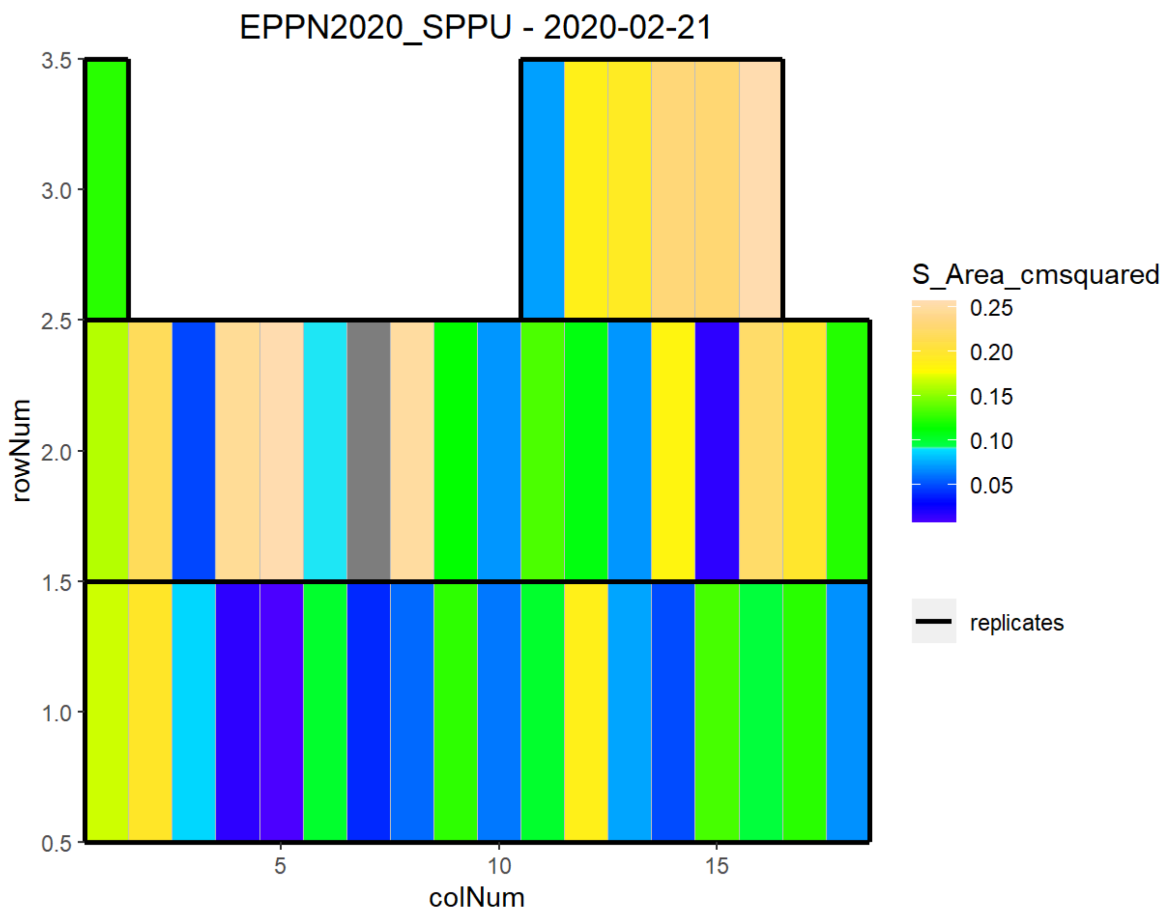

EPPN2020_SPPU - 2020-03-10
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
S_Height_cm
9
8
7
6
5
4
3
replicates

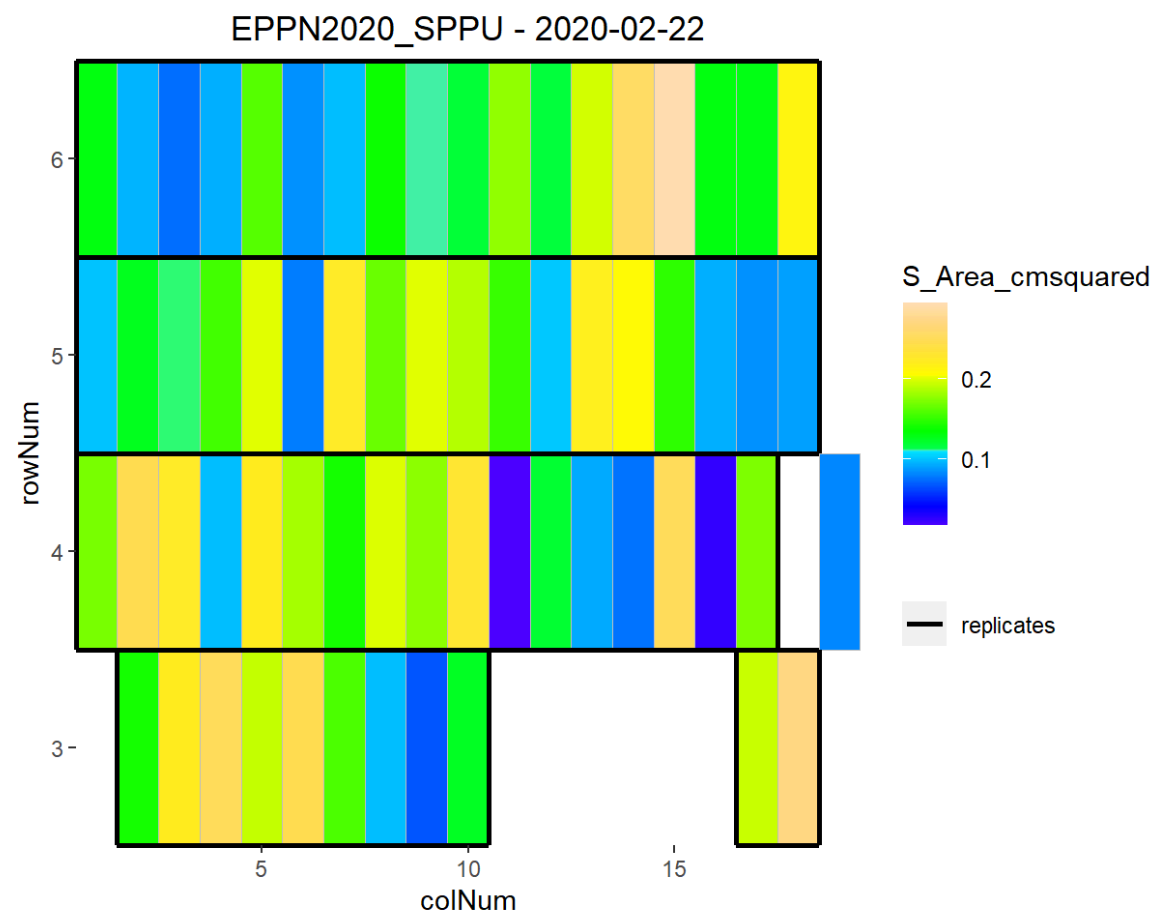

EPPN2020_SPPU - 2020-02-17
rowNum
1.4
1.2
1.0
0.8
0.6
colNum
5
10
15
replicates
S_Area_cmsquared
0.03
0.02
0.01

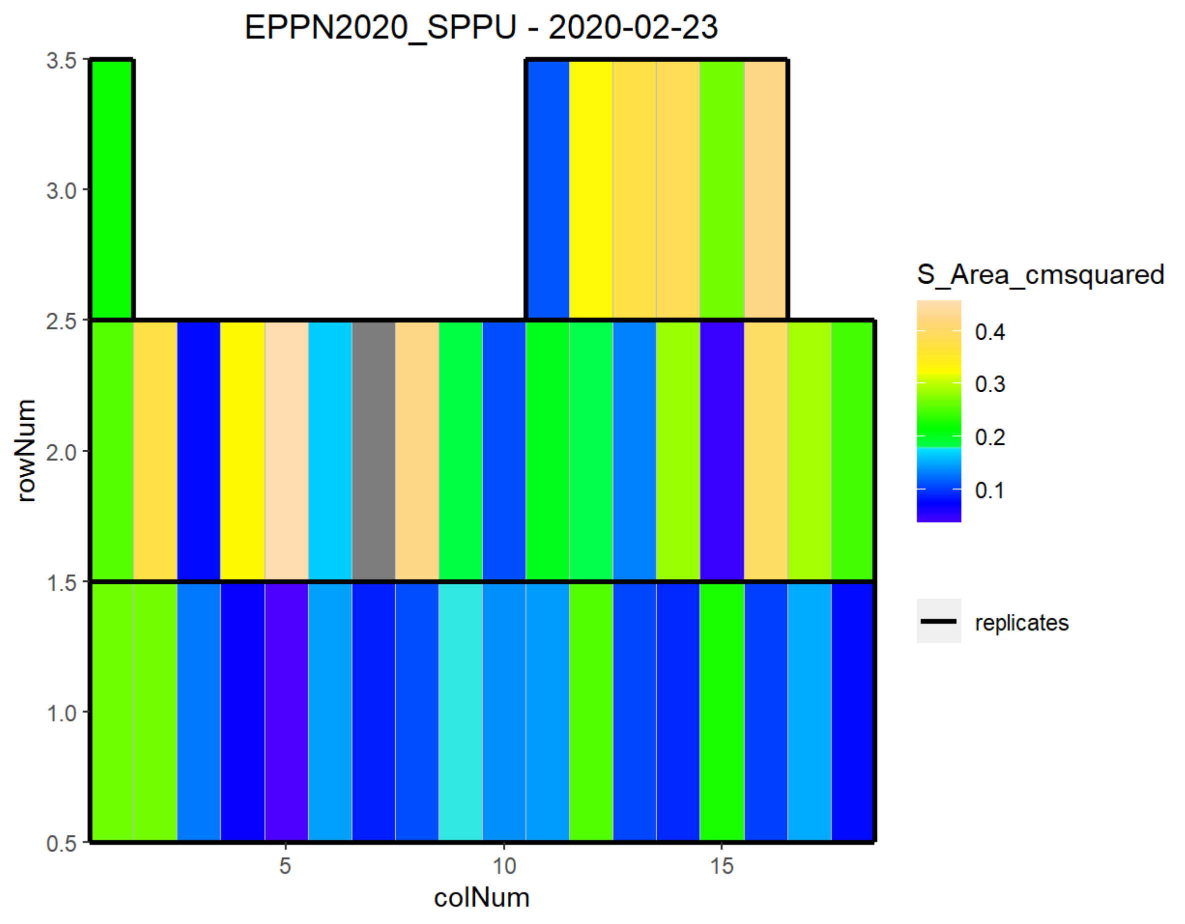

EPPN2020_SPPU - 2020-02-18
rowNum
6
5
4
3
2
1
colNum
5
10
15
S_Area_cmsquared
0.100
0.075
0.050
0.025
replicates

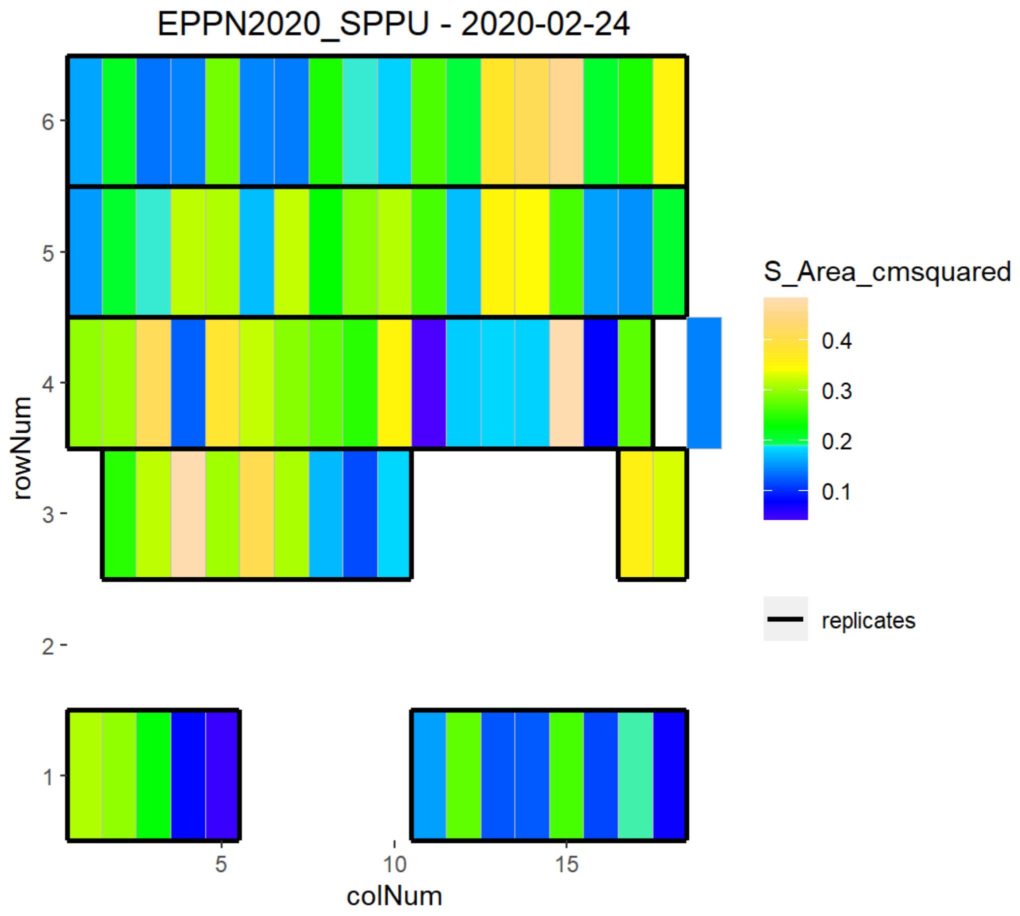

EPPN2020_SPPU - 2020-02-19
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
S_Area_cmsquared
0.16
0.12
0.08
0.04
replicates

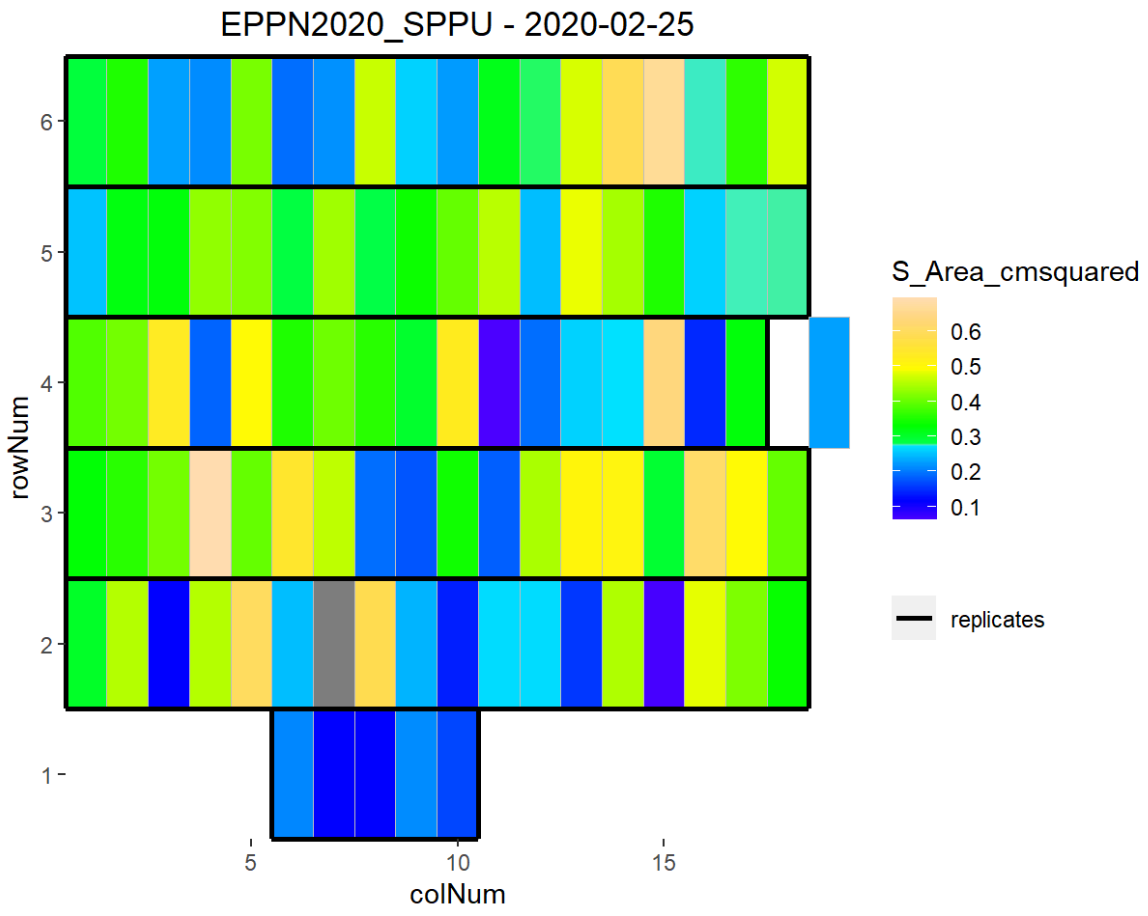

EPPN2020_SPPU - 2020-02-20
rowNum
6
5
4
3
colNum
5
10
15
S_Area_cmsquared
0.15
0.10
0.05
replicates

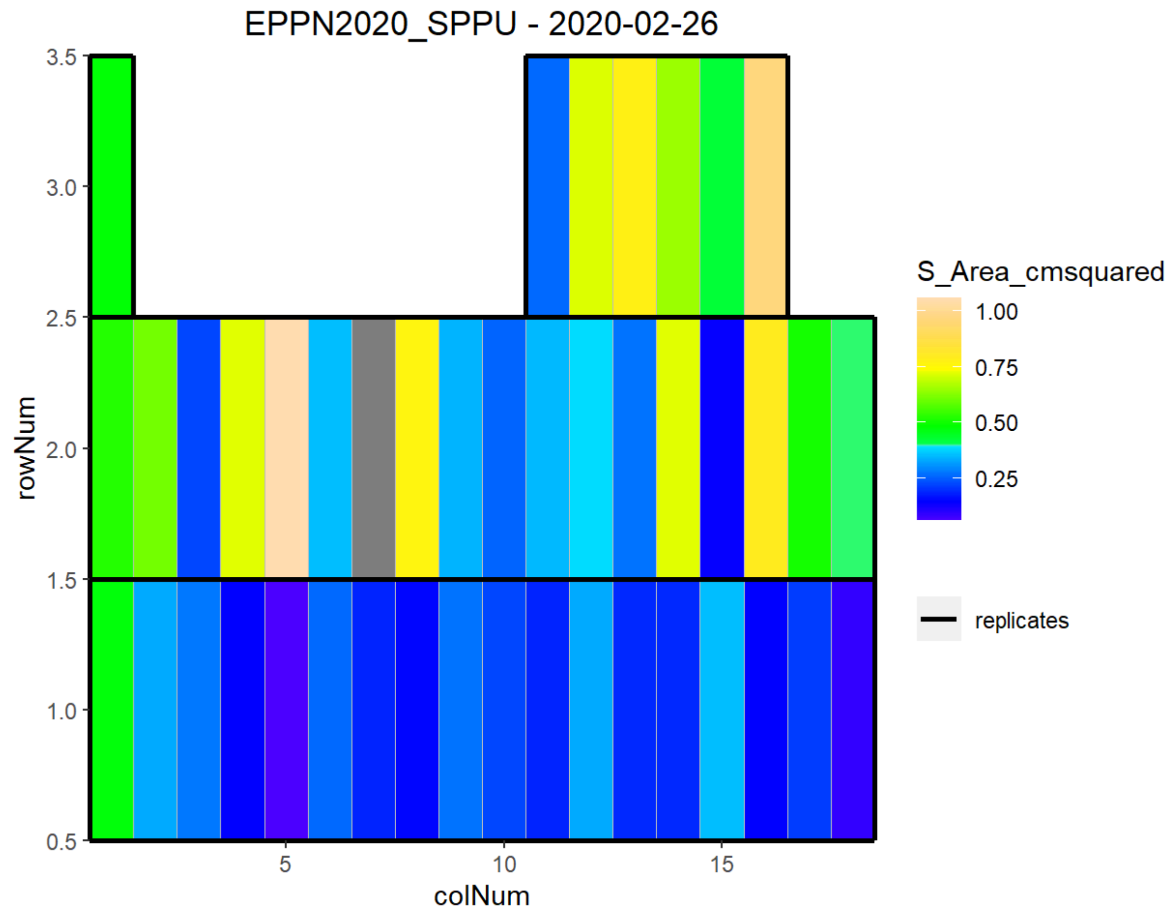

EPPN2020_SPPU - 2020-02-21
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
S_Area_cmsquared
0.25
0.20
0.15
0.10
0.05
replicates

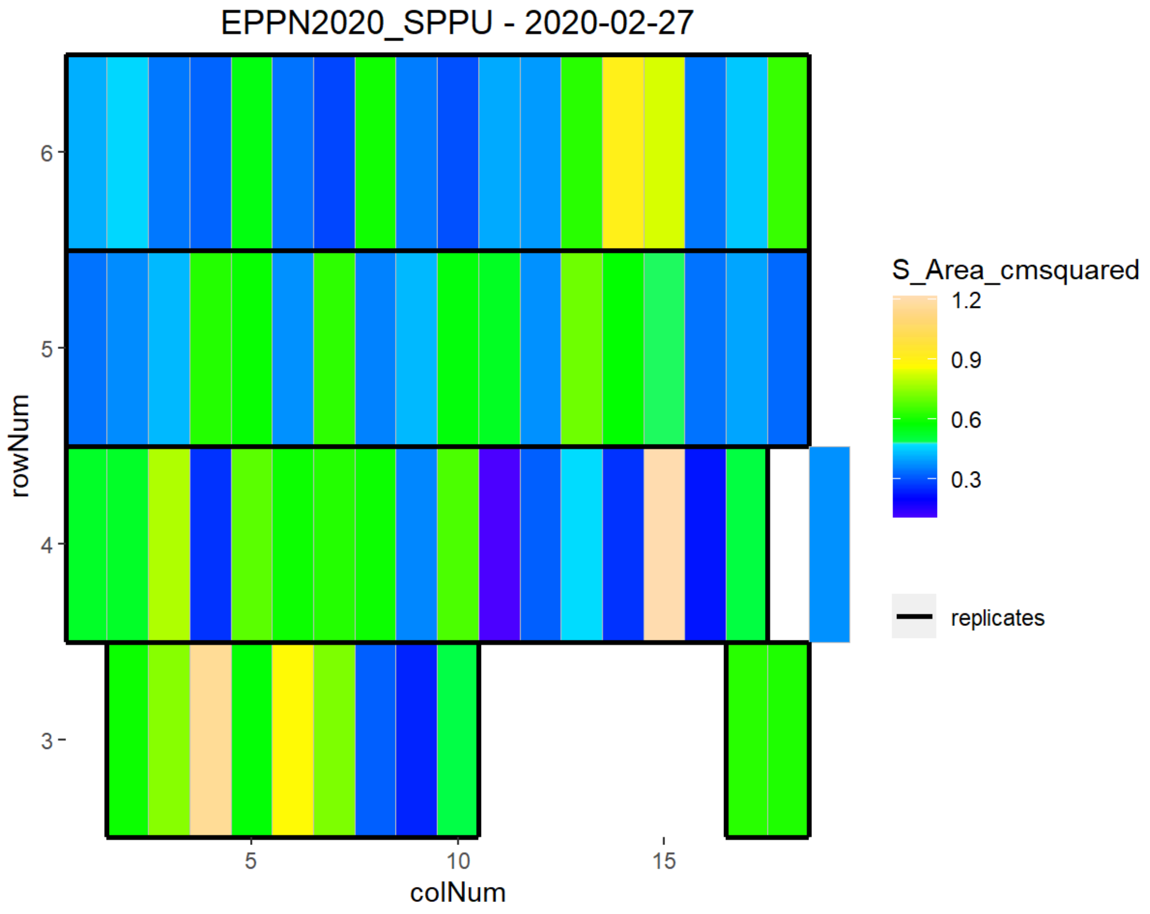

EPPN2020_SPPU - 2020-02-22
rowNum
6
5
4
3
colNum
5
10
15
S_Area_cmsquared
0.2
0.1
replicates

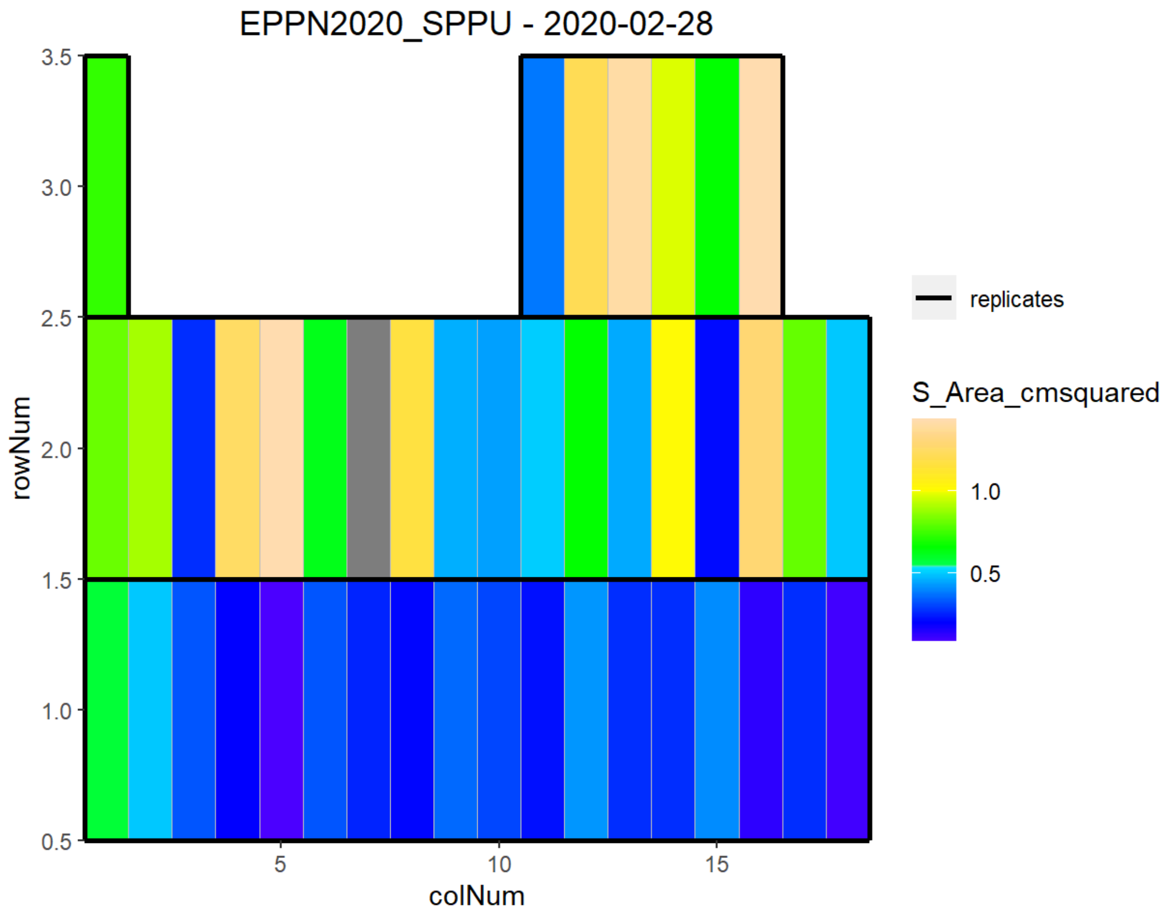

EPPN2020_SPPU - 2020-02-23
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
S_Area_cmsquared
0.4
0.3
0.2
0.1
replicates

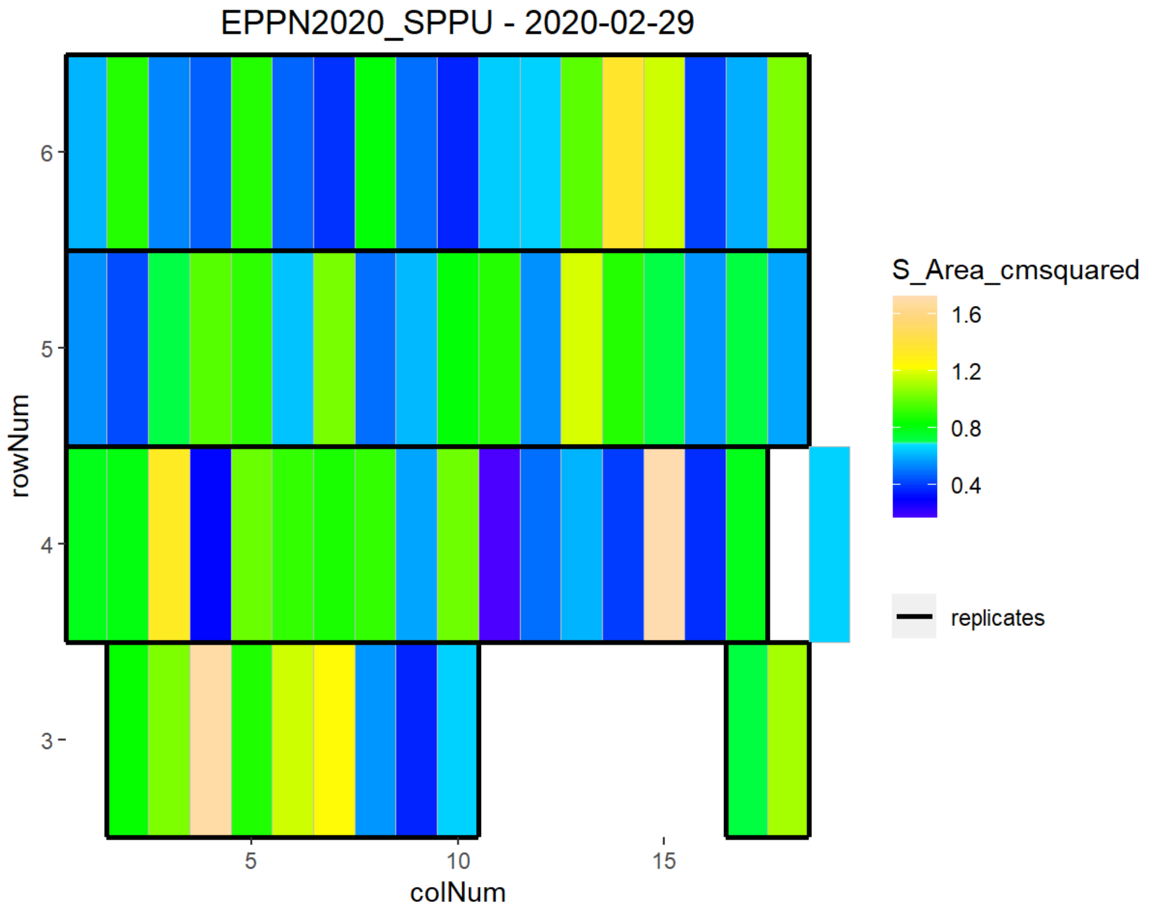

EPPN2020_SPPU - 2020-02-24
rowNum
6
5
4
3
2
1
colNum
5
10
15
S_Area_cmsquared
0.4
0.3
0.2
0.1
replicates

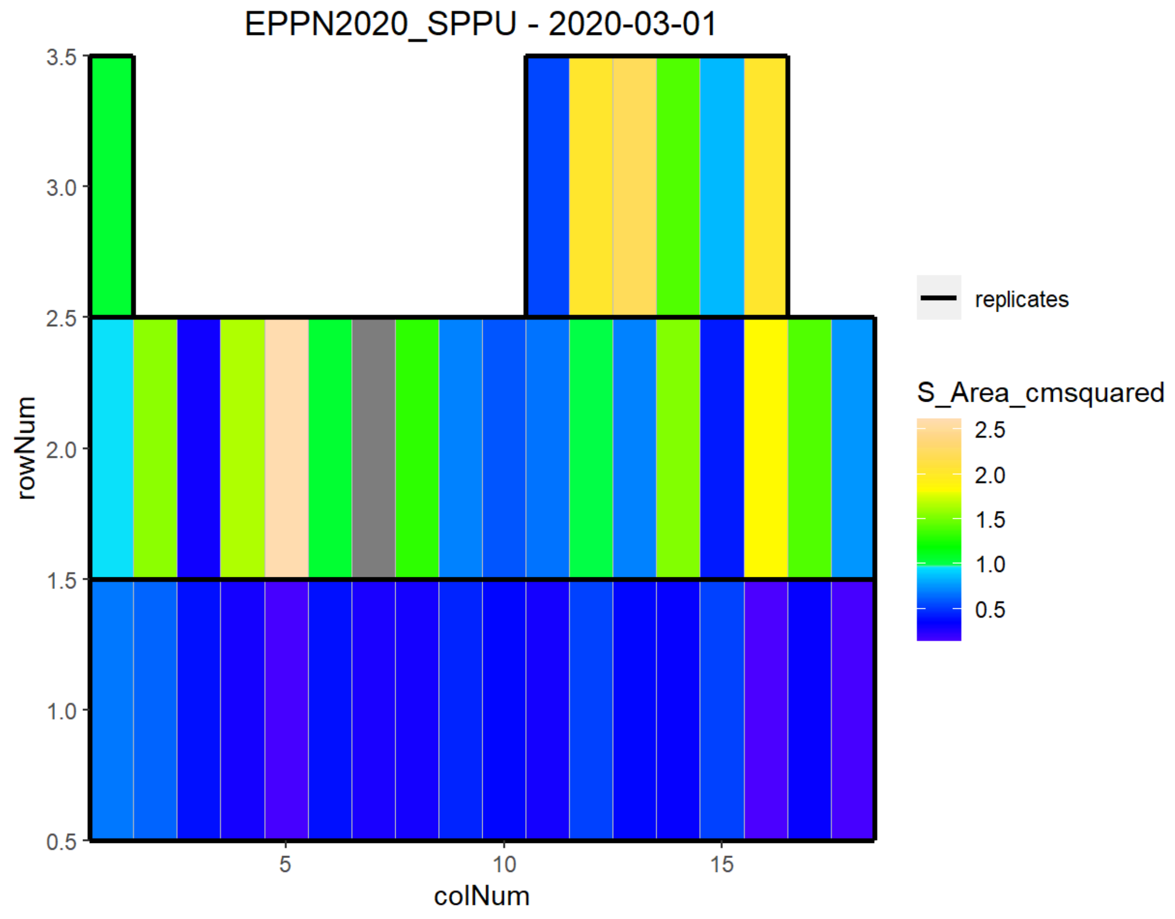

EPPN2020_SPPU - 2020-02-25
rowNum
6
5
4
3
2
1
colNum
5
10
15
S_Area_cmsquared
0.6
0.5
0.4
0.3
0.2
0.1
replicates

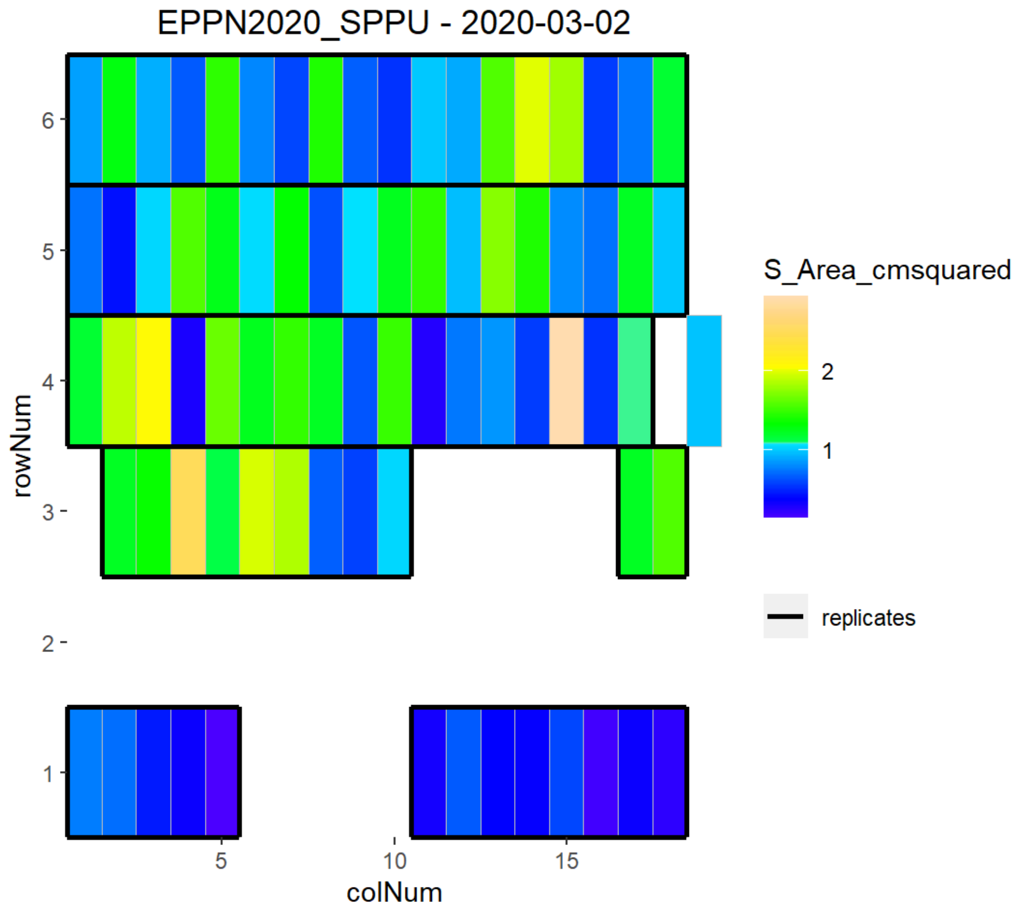

EPPN2020_SPPU - 2020-02-26
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
S_Area_cmsquared
1.00
0.75
0.50
0.25
replicates

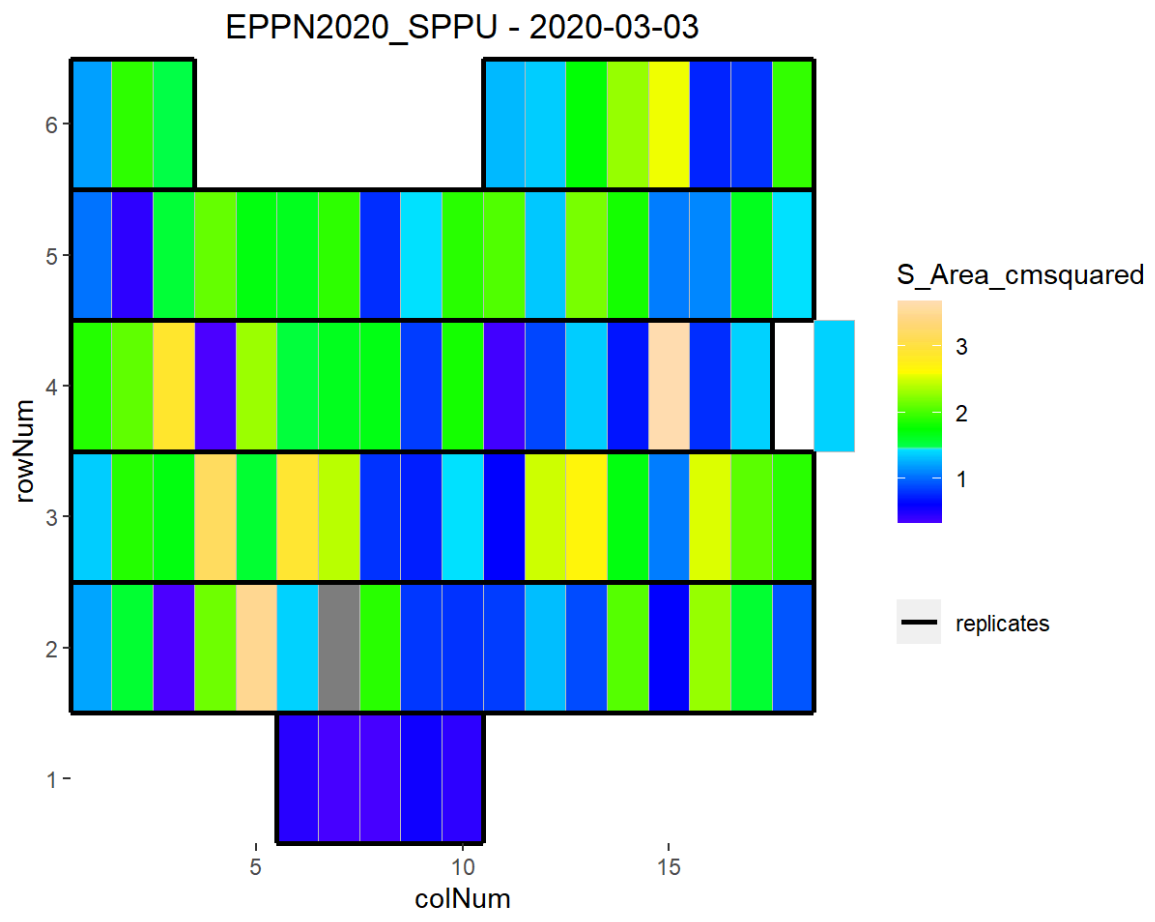

EPPN2020_SPPU - 2020-02-27
rowNum
6
5
4
3
colNum
5
10
15
S_Area_cmsquared
1.2
0.9
0.6
0.3
replicates

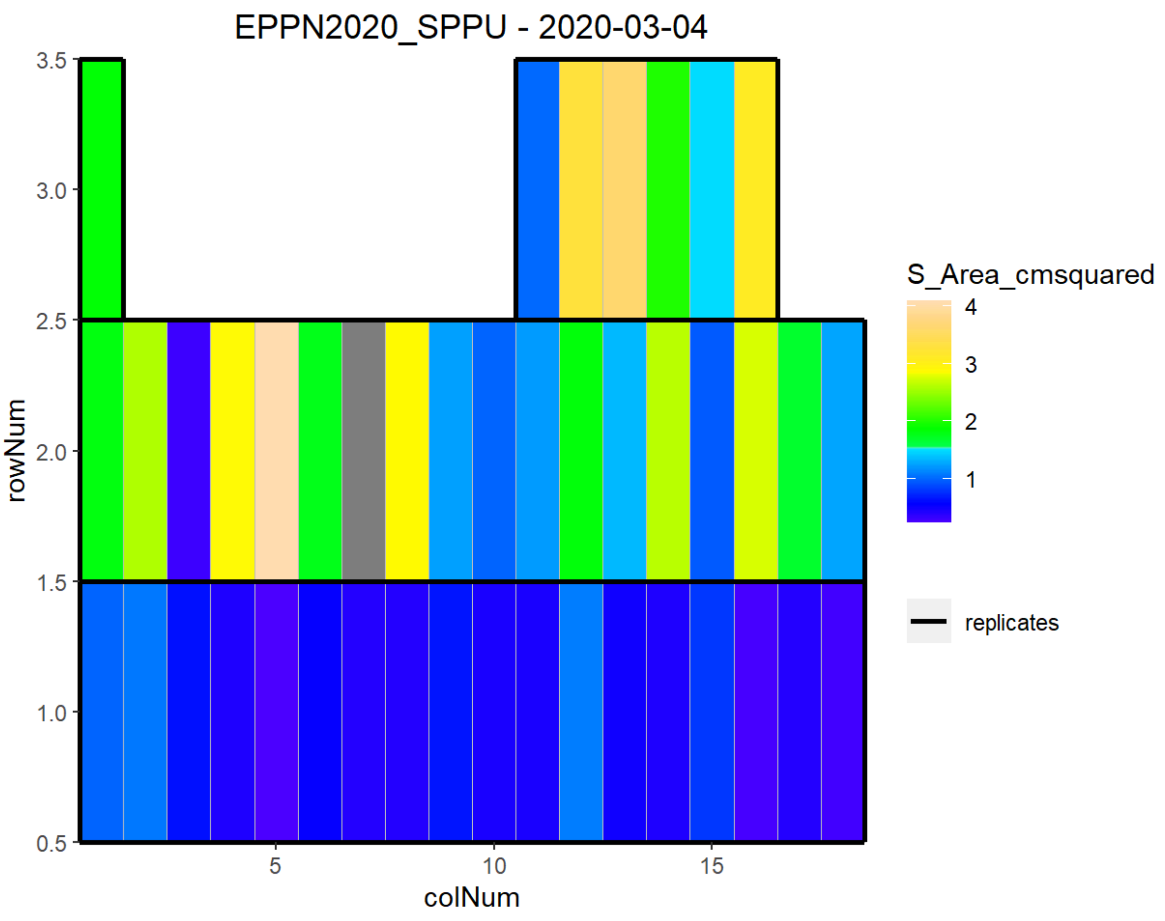

EPPN2020_SPPU - 2020-02-28
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
replicates
S_Area_cmsquared
1.0
0.5

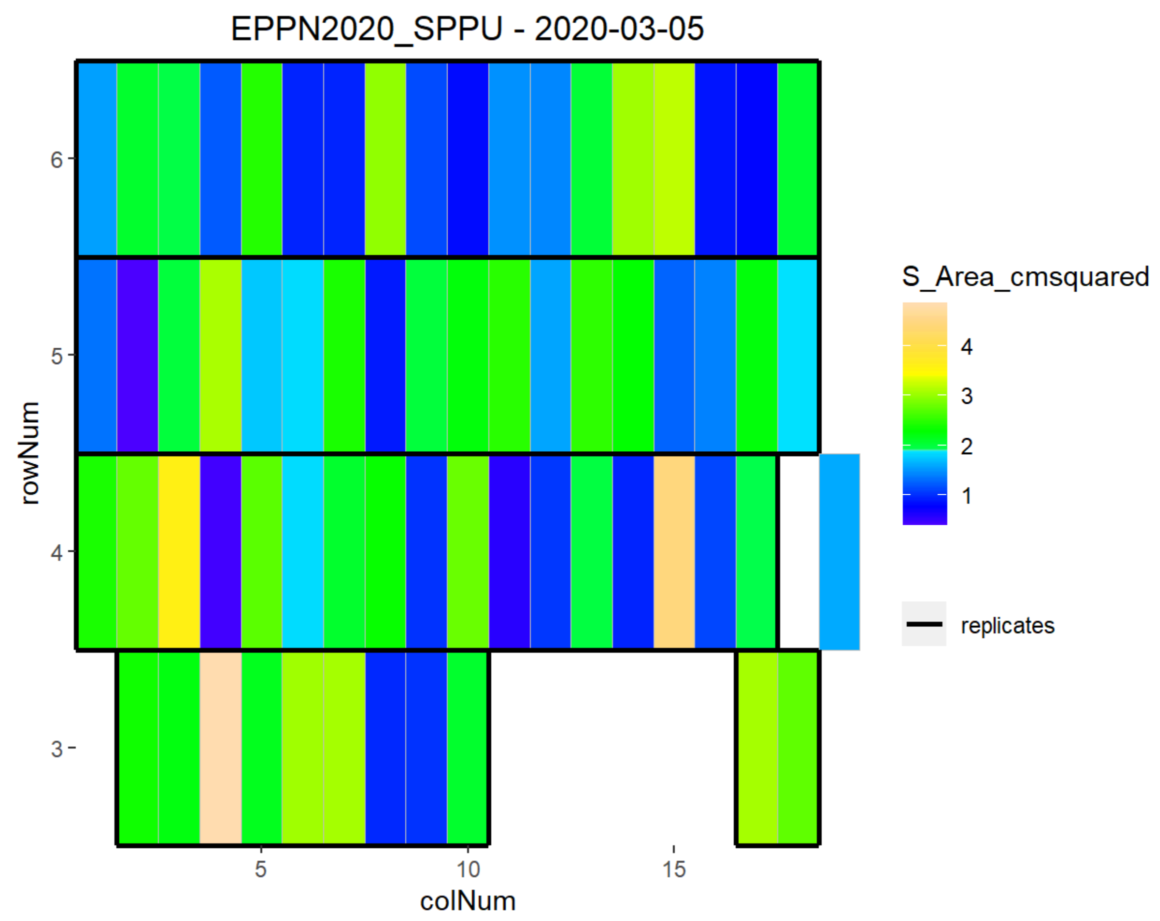

EPPN2020_SPPU - 2020-02-29
rowNum
6
5
4
3
colNum
5
10
15
S_Area_cmsquared
1.6
1.2
0.8
0.4
replicates

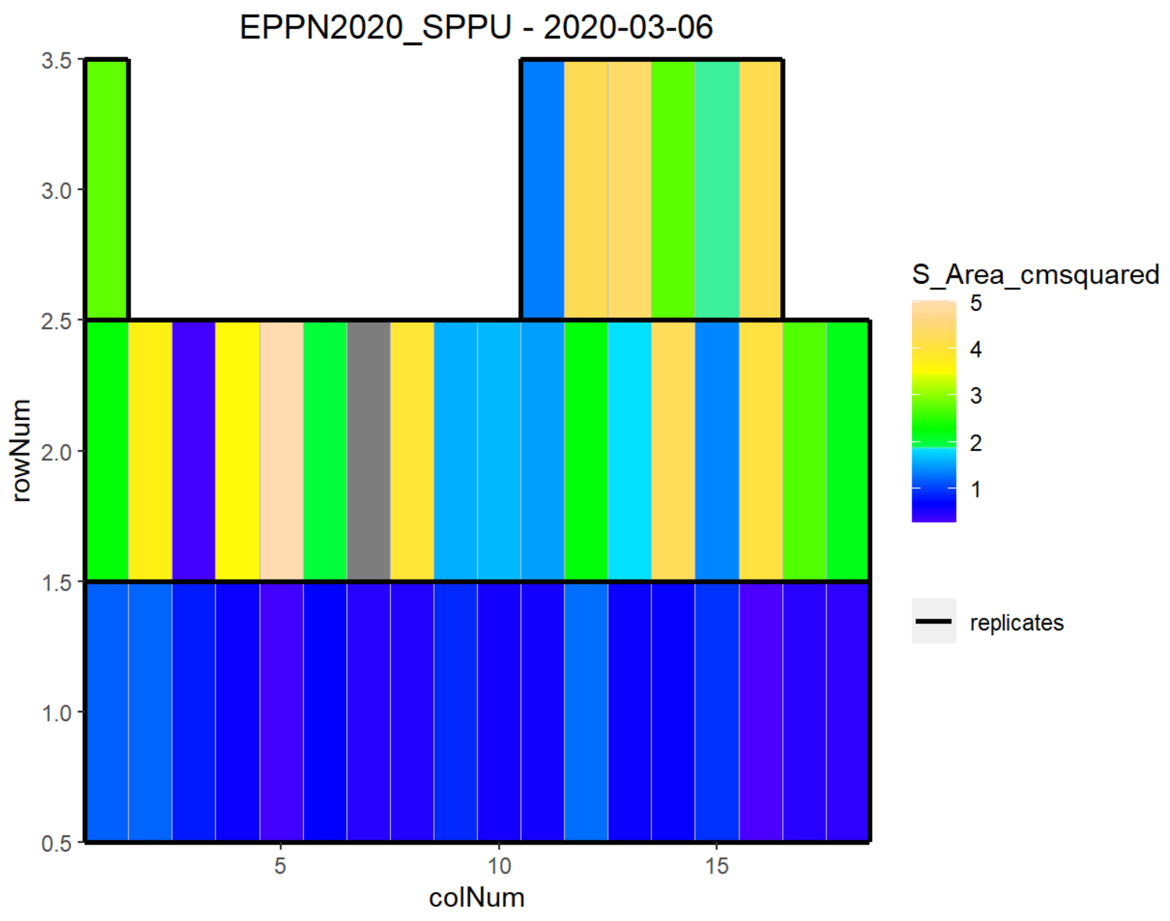

EPPN2020_SPPU - 2020-03-01
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
replicates
S_Area_cmsquared
2.5
2.0
1.5
1.0
0.5

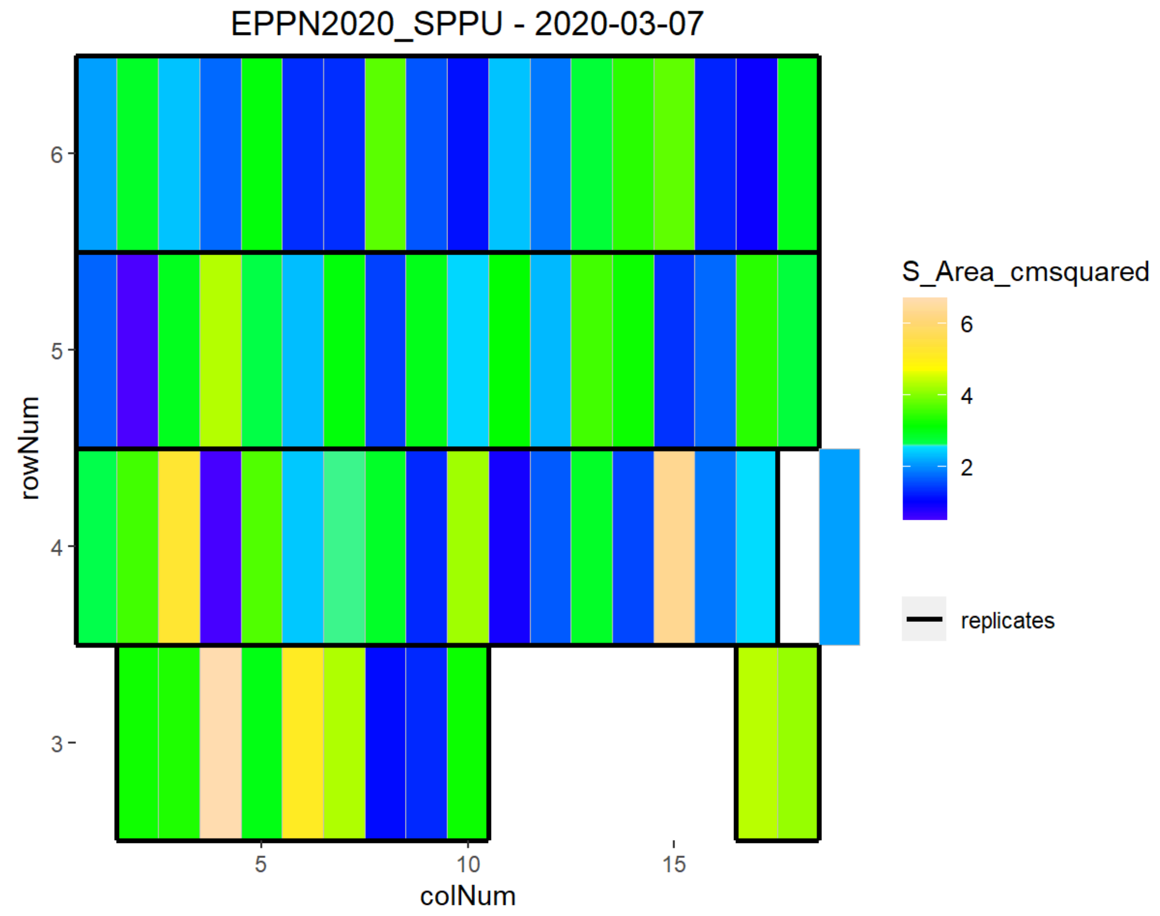

EPPN2020_SPPU - 2020-03-02
rowNum
6
5
4
3
2
1
colNum
5
10
15
S_Area_cmsquared
2
1
replicates

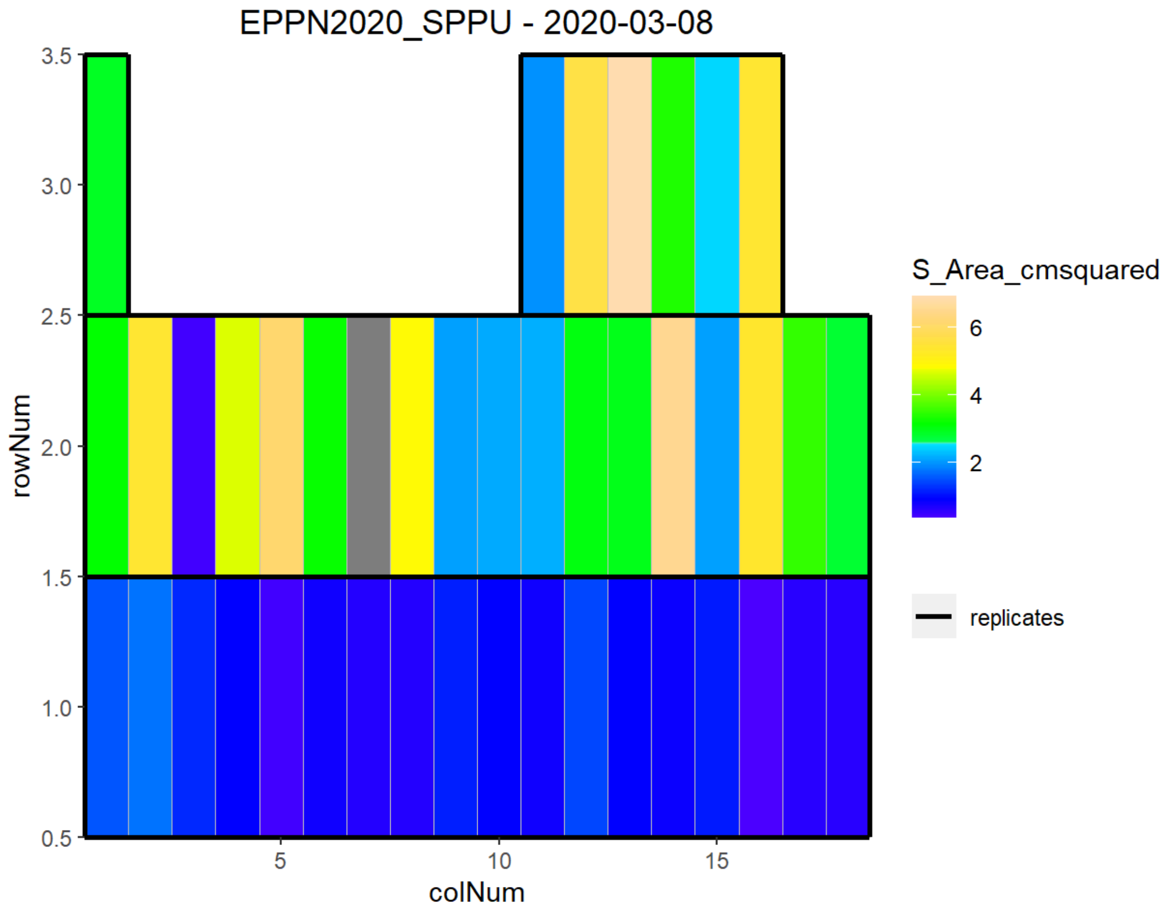

EPPN2020_SPPU - 2020-03-03
rowNum
colNum
S_Area_cmsquared
replicates

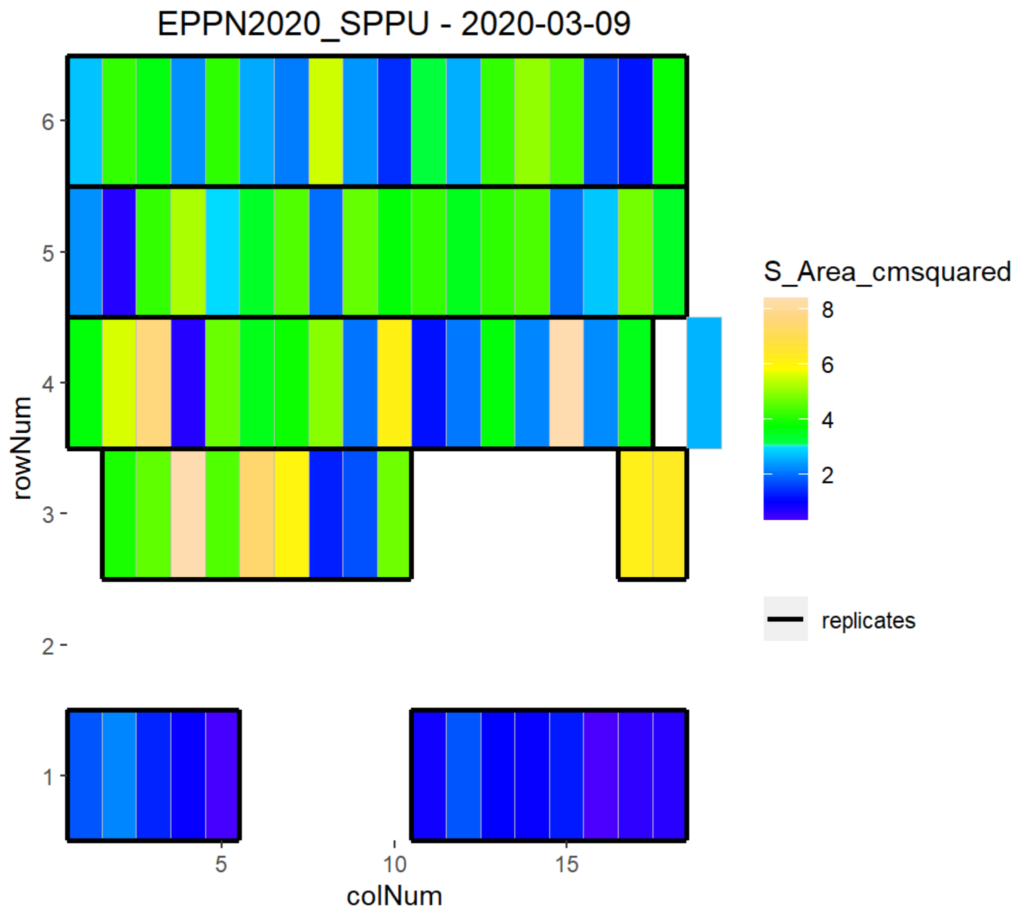

EPPN2020_SPPU - 2020-03-04
rowNum
colNum
S_Area_cmsquared
replicates

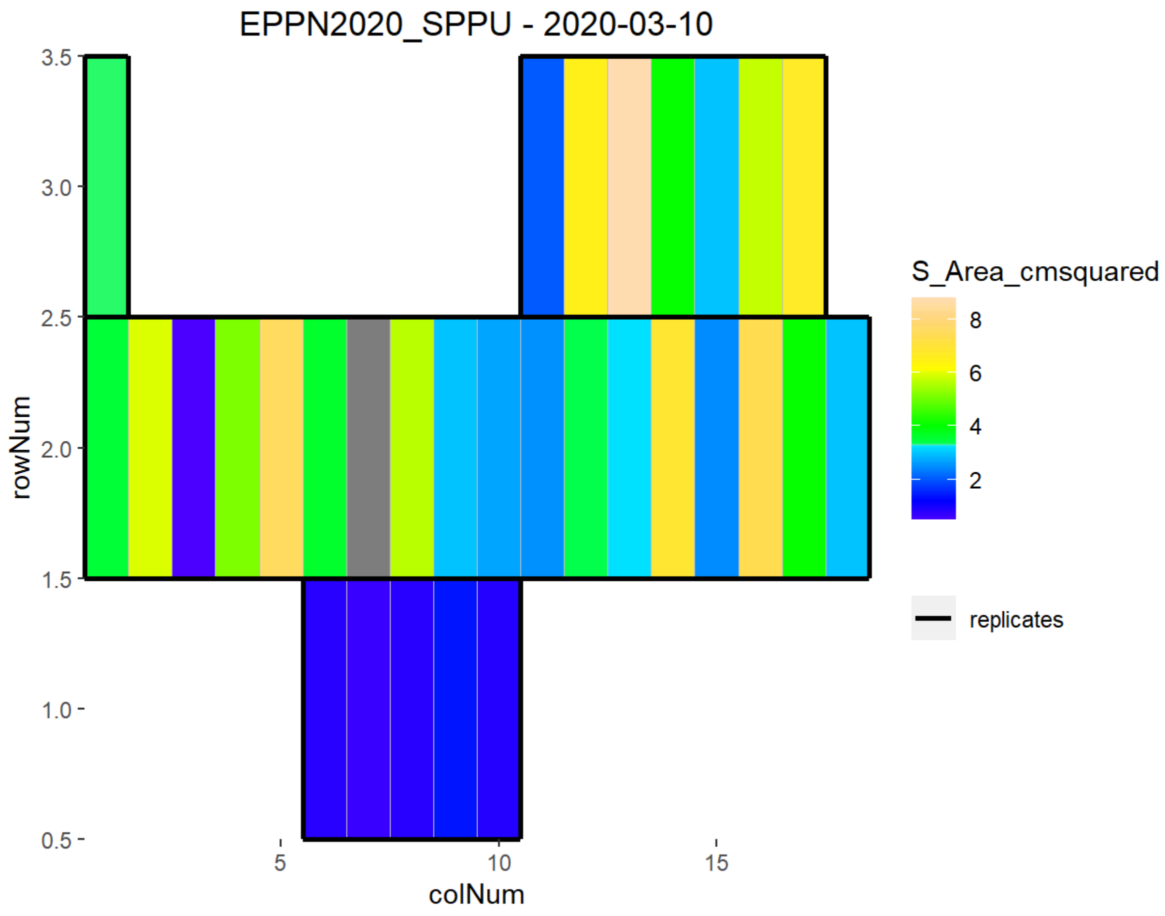

EPPN2020_SPPU - 2020-03-05
rowNum
6
5
4
3
colNum
5
10
15
S_Area_cmsquared
4
3
2
1
replicates


EPPN2020_SPPU - 2020-03-06
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
colNum
5
10
15
S_Area_cmsquared
5
4
3
2
1
replicates


EPPN2020_SPPU - 2020-03-07
rowNum
6
5
4
3
5
10
15
colNum
S_Area_cmsquared
6
4
2
replicates

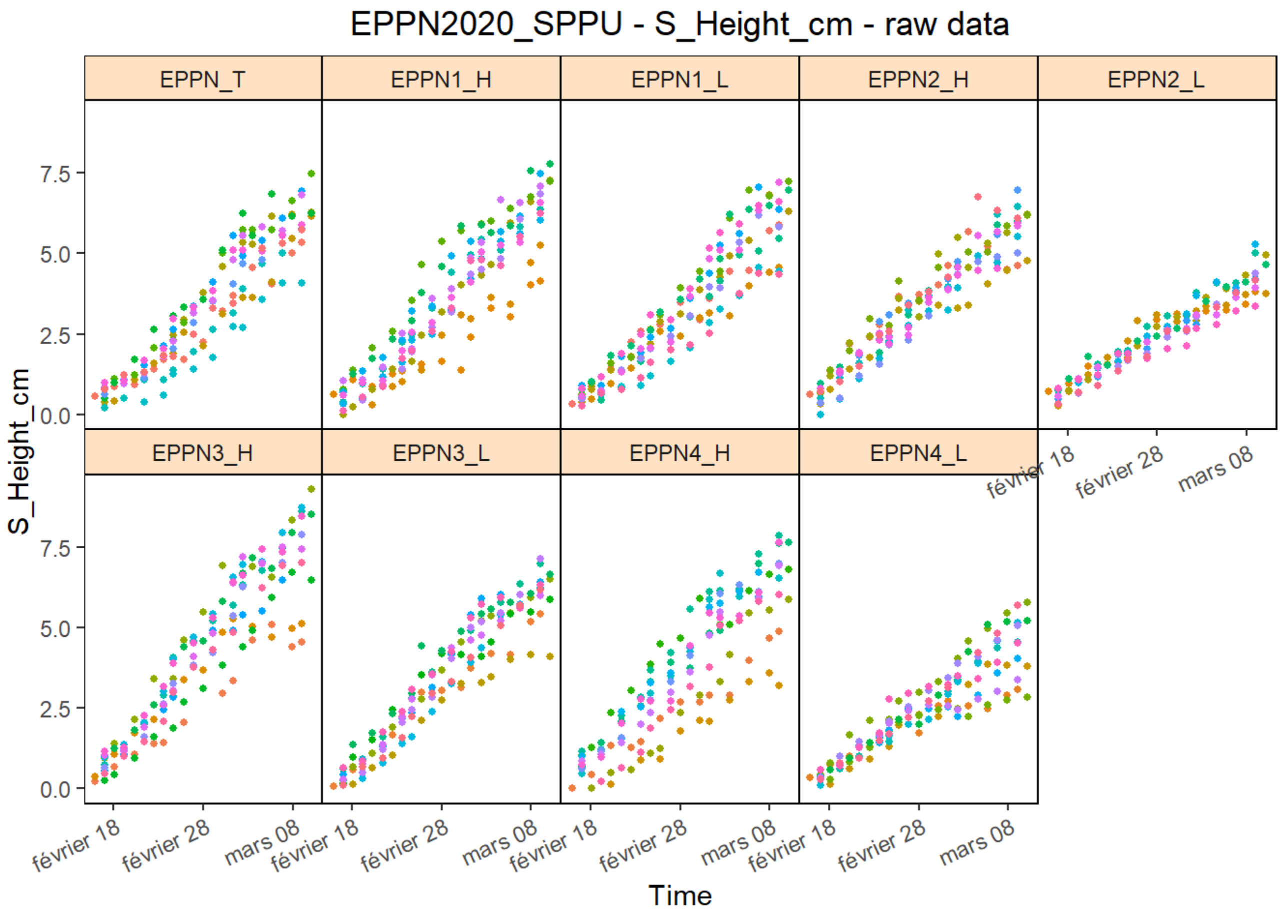

EPPN2020_SPPU - 2020-03-08
rowNum
3.5
3.0
2.5
2.0
1.5
1.0
0.5
5
10
15
colNum
S_Area_cmsquared
6
4
2
replicates

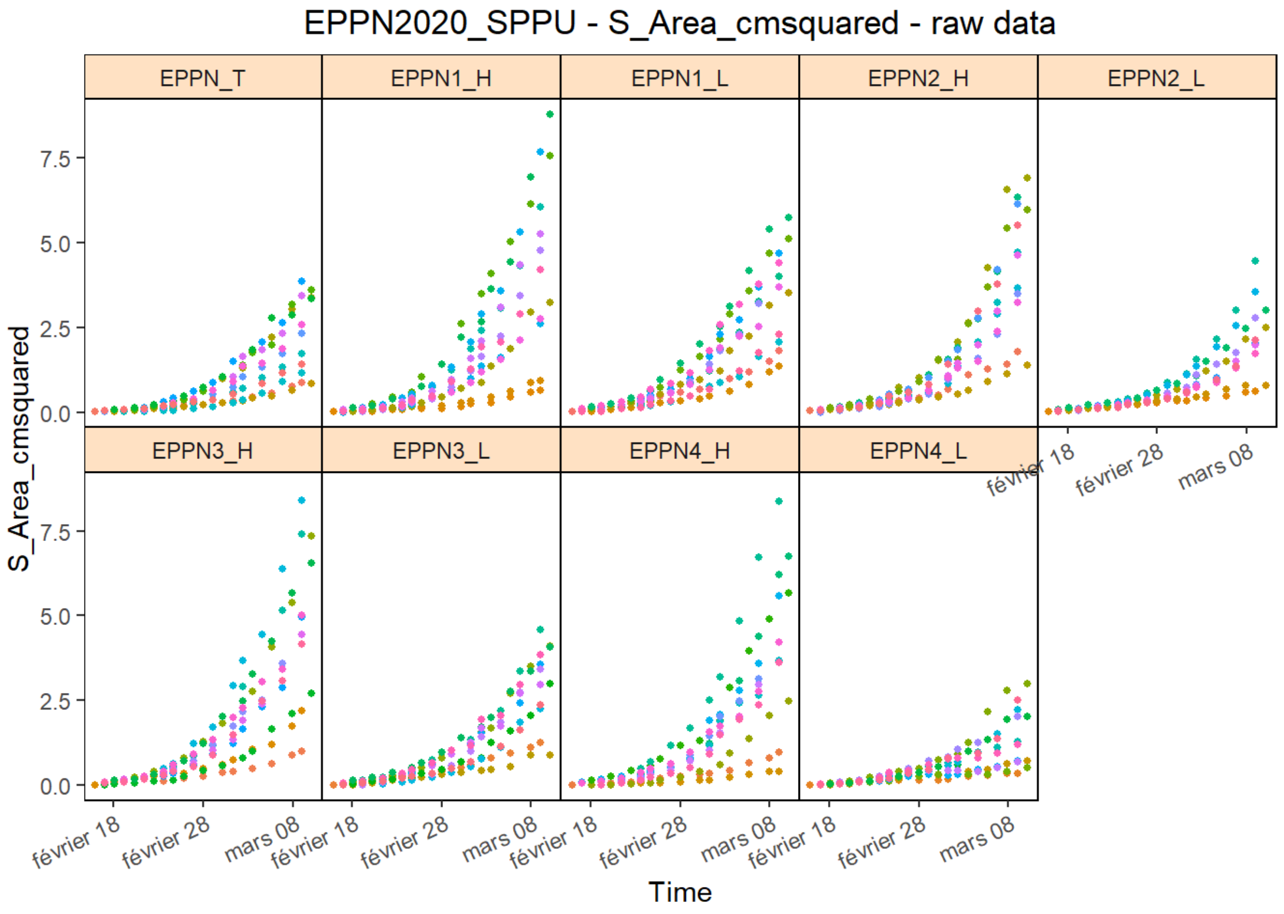




Check time course of raw data per time point

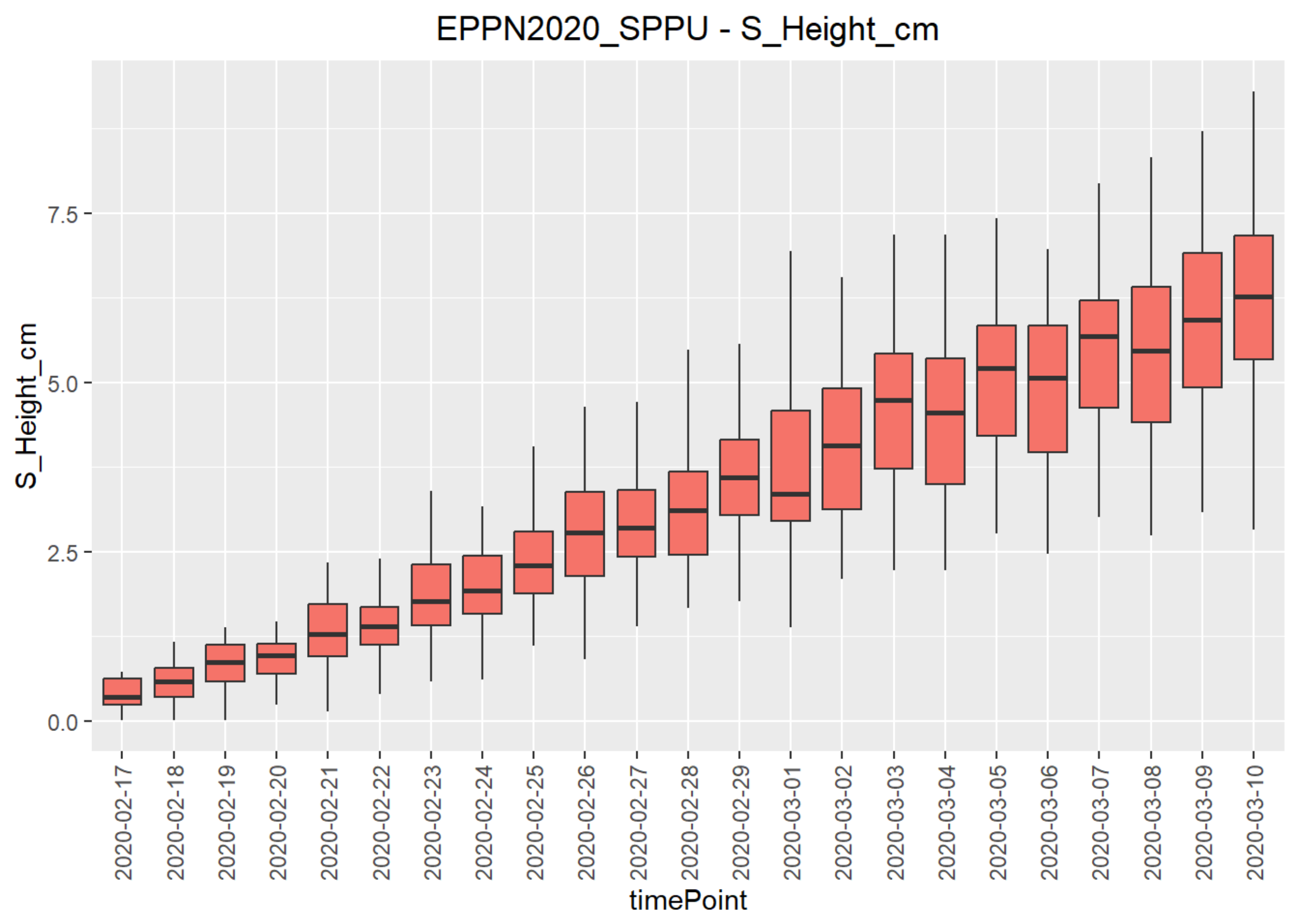
```
for (trait_name in traits) {
  plot(timePoint_S,
     traits = trait_name,
     plotType = "raw")
}
```

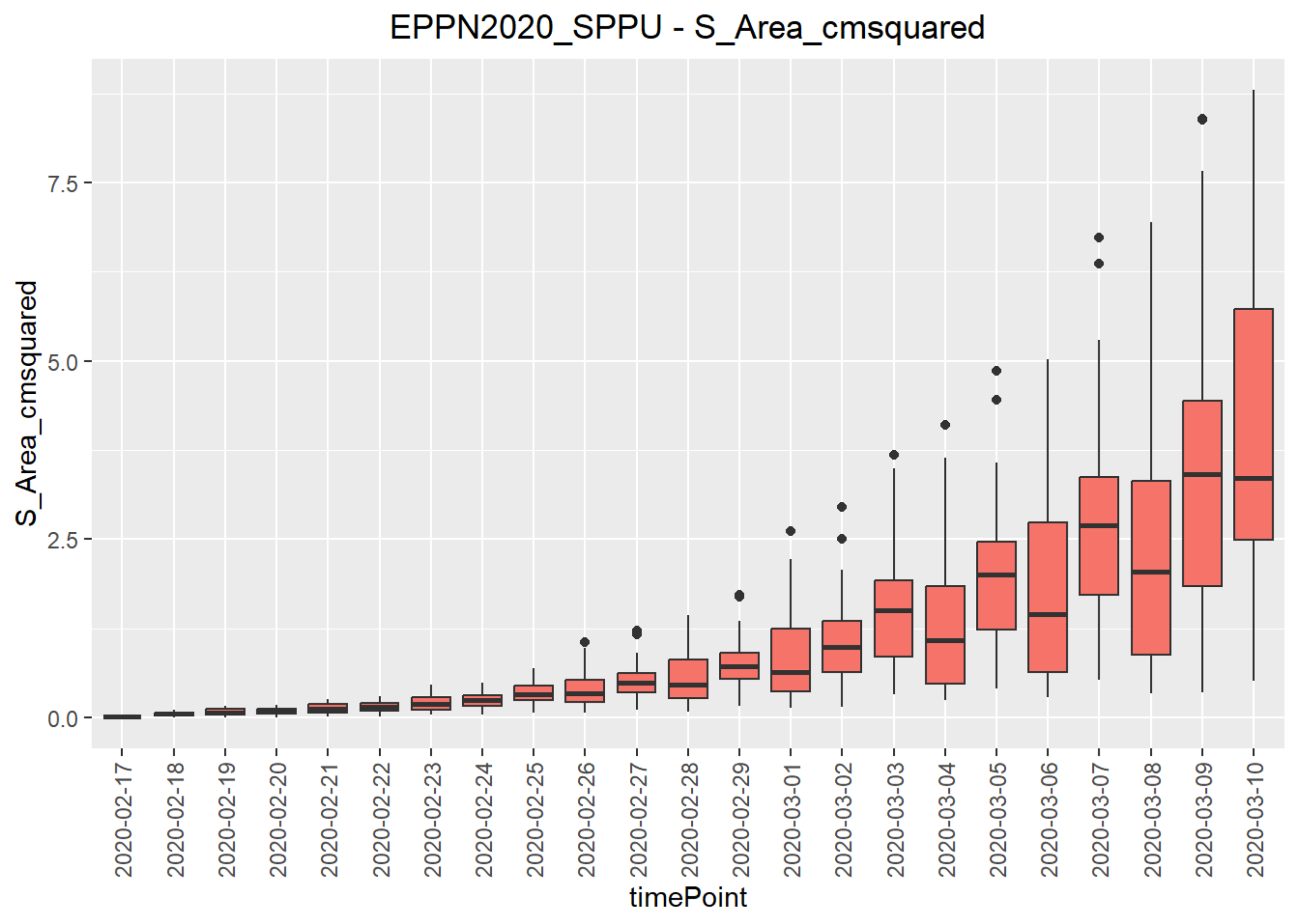




Check the boxplots of raw data per time point

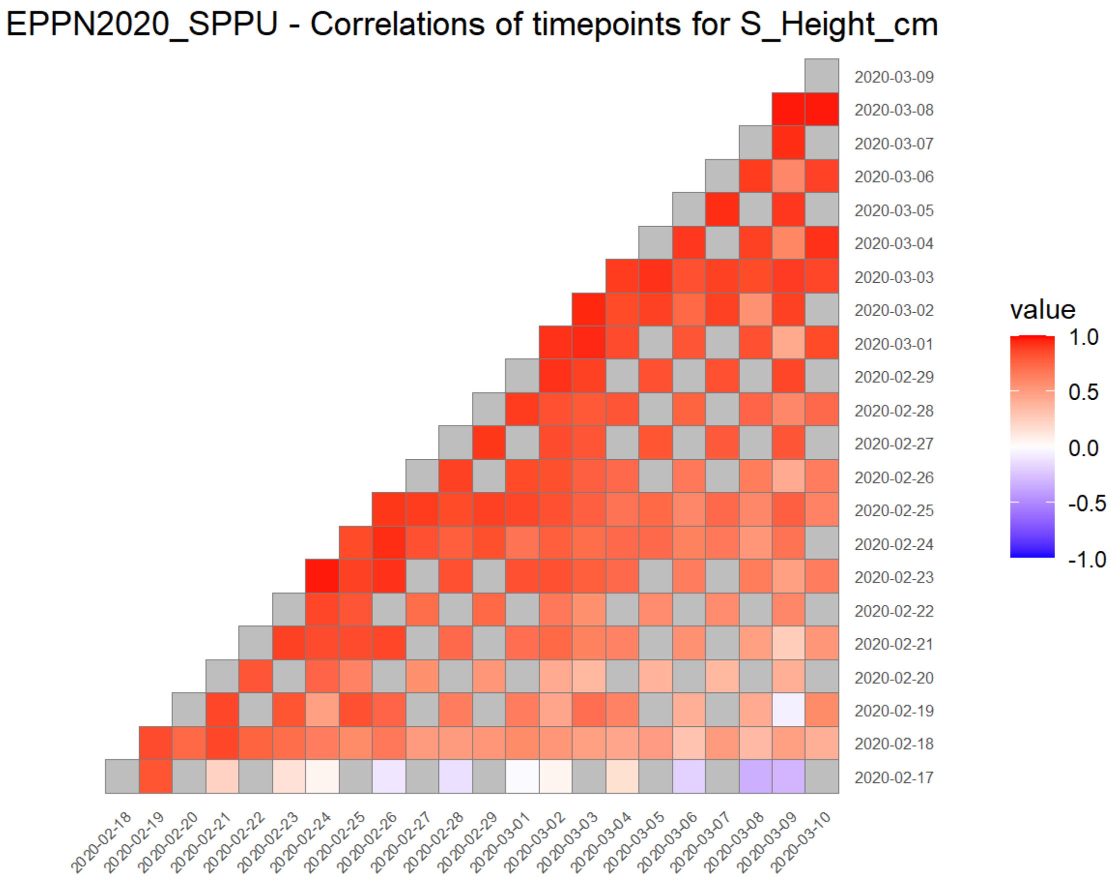
```r
for (trait_name in traits) {
  plot(timePoint_S,
     plotType = "box",
     traits = trait_name)
}
```

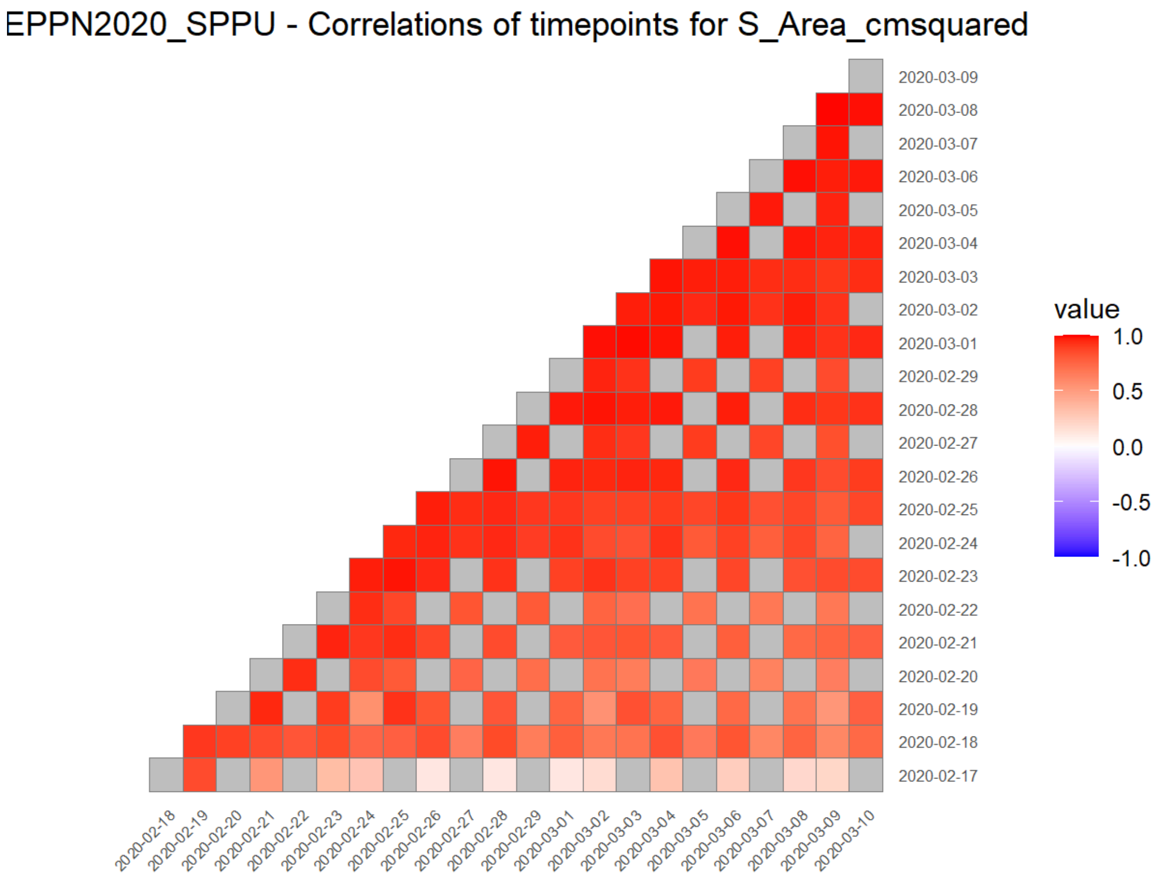




Check the correlation plots of raw data per time point

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



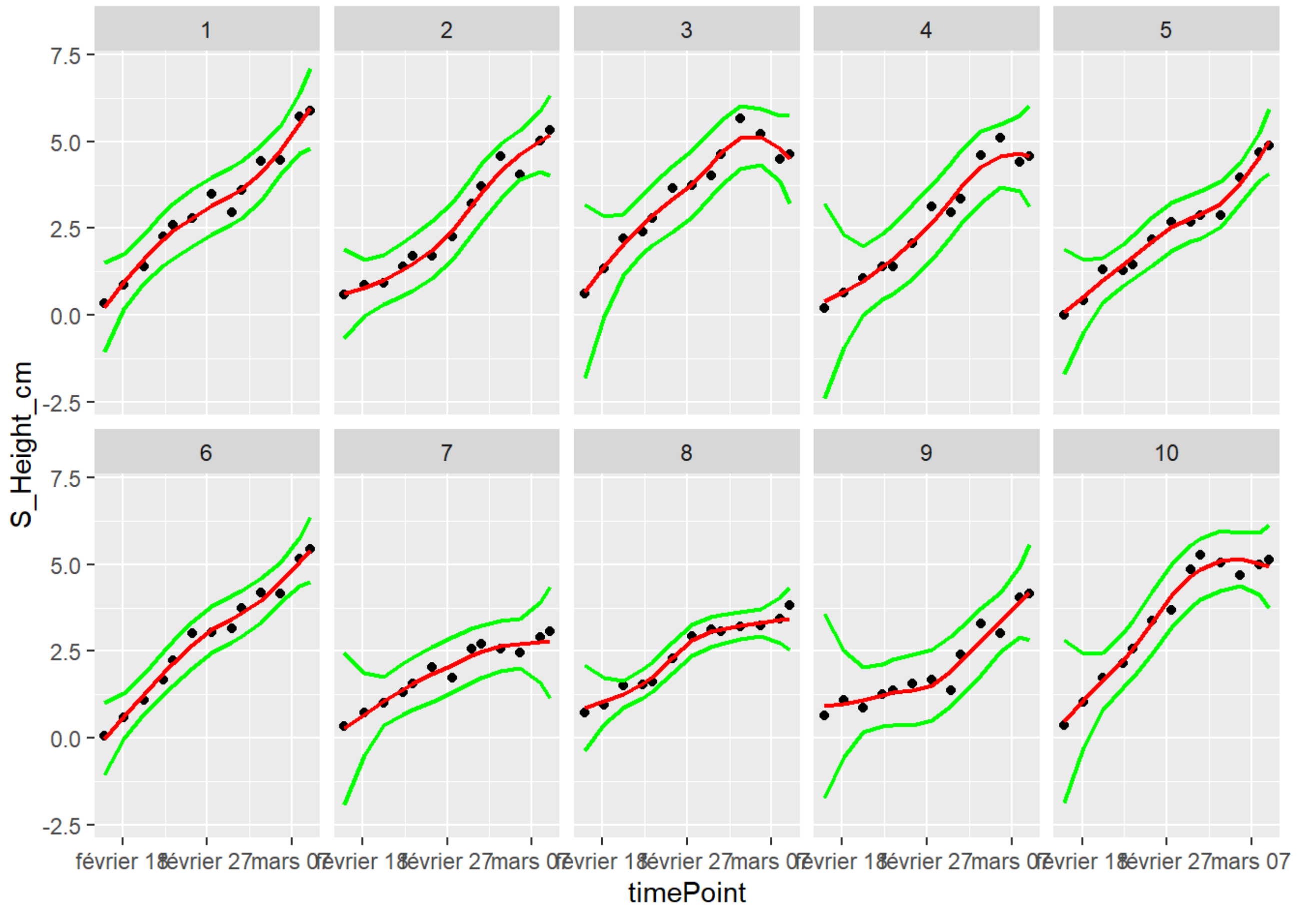


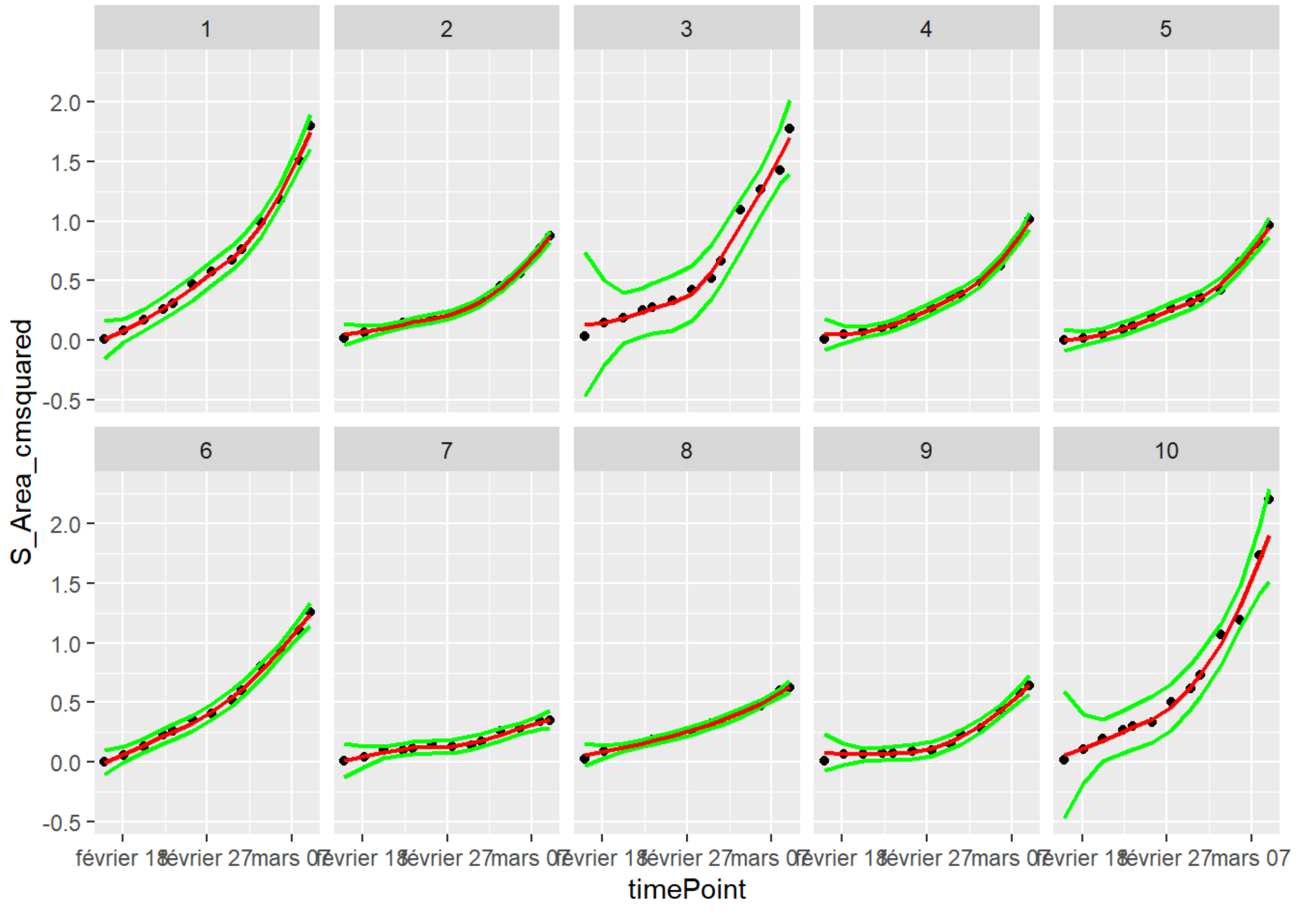
1. Detection of outliers for single observations

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")
  }
}
```

```
## No outlier for S_Height_cm
## No outlier for S_Area_cmsquared
```

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 `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]) :
  les contrastes ne peuvent être appliqués qu'aux facteurs ayant au moins deux niveau
x")
```

```
## [1] "Erreur dans `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]) : \n  les
contrastes ne peuvent être appliqués qu'aux facteurs ayant au moins deux niveaux"
```

Model visualisation

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)
}
```

Plot the splines for a plant selection

detectSerieOut

removeSerieOut

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