# NaPPI Data Analysis

Elise

2024-06-03

Set the right working directory.

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

# Data importation

Import the data sets extracted from the Data Preparation 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")
```

Convert the columns to factor and date formats.

```
# 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 NaPPI

# 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
```

Collect the variables of every data template and print the names of the variables. This serves as a double check.

```
platform <- "NaPPI"

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

# timeseries
# no 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 three
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] "NaPPI : The variables for endpoint are DW_shoot_g, FW_shoot_g"
```

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

```
## [1] "NaPPI : The variables for S_timeseries are S_Height_cm, S_Height_pixel, S_Area_cmsqu
ared, S_Area_pixel, S_Perimeter_cm, S_Perimeter_pixel, S_Compactness, S_Width_cm, S_Width_pi
xel"
```

Add a column Plant_type with three levels, H L and T. This variable is useful to test for heterosis effects.

```
endpoint$Plant_type <- substr(endpoint$Genotype, nchar(as.character(endpoint$Genotype)), nch
ar(as.character(endpoint$Genotype)))
S_timeseries$Plant_type <- substr(S_timeseries$Genotype, nchar(as.character(S_timeseries$Gen
otype)), nchar(as.character(S_timeseries$Genotype)))
```

# 1. Endpoint dataframe

## A. Exploration of data

### Exploration tables using the rstatix, janitor and skimr packages

```
endpoint %>%
  count(Genotype)
```

```
##    Genotype  n
## 1  EPPN01_H  9
## 2  EPPN02_H  9
## 3  EPPN03_H 10
## 4  EPPN04_H  7
## 5  EPPN05_H 13
## 6  EPPN06_H 18
## 7  EPPN07_L  2
## 8  EPPN08_H  7
## 9  EPPN09_H 11
## 10 EPPN10_H  7
## 11 EPPN10_L  2
## 12 EPPN11_H 10
## 13 EPPN11_L  3
## 14 EPPN12_H 10
## 15 EPPN13_H  5
## 16 EPPN20_T  3
```

```
endpoint %>%
  tabyl(Genotype, Column) %>%
  adorn_totals("row") %>%
  adorn_percentages("row") %>%
  adorn_pct_formatting() %>%
  adorn_ns() %>%
  adorn_title("combined")
```

```
##  Genotype/Column         1          2          3          4          5          6
##         EPPN01_H  0.0% (0) 11.1% (1) 11.1% (1)  0.0% (0) 11.1% (1)  0.0% (0)
##         EPPN02_H  0.0% (0)  0.0% (0)  0.0% (0) 11.1% (1)  0.0% (0) 11.1% (1)
##         EPPN03_H  0.0% (0) 10.0% (1) 10.0% (1)  0.0% (0)  0.0% (0) 10.0% (1)
##         EPPN04_H  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0) 14.3% (1)
##         EPPN05_H  7.7% (1)  0.0% (0)  7.7% (1)  0.0% (0)  7.7% (1)  0.0% (0)
##         EPPN06_H 11.1% (2)  5.6% (1) 11.1% (2)  5.6% (1)  0.0% (0)  5.6% (1)
##         EPPN07_L  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)
##         EPPN08_H  0.0% (0)  0.0% (0) 14.3% (1)  0.0% (0)  0.0% (0)  0.0% (0)
##         EPPN09_H  9.1% (1)  0.0% (0)  0.0% (0)  9.1% (1)  9.1% (1)  9.1% (1)
##         EPPN10_H  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0) 14.3% (1)  0.0% (0)
##         EPPN10_L 50.0% (1)  0.0% (0)  0.0% (0) 50.0% (1)  0.0% (0)  0.0% (0)
##         EPPN11_H  0.0% (0) 10.0% (1) 10.0% (1)  0.0% (0)  0.0% (0) 10.0% (1)
##         EPPN11_L  0.0% (0) 33.3% (1)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)
##         EPPN12_H  0.0% (0) 10.0% (1)  0.0% (0) 10.0% (1) 10.0% (1)  0.0% (0)
##         EPPN13_H  0.0% (0) 20.0% (1)  0.0% (0)  0.0% (0) 20.0% (1)  0.0% (0)
##         EPPN20_T 33.3% (1)  0.0% (0)  0.0% (0) 33.3% (1)  0.0% (0)  0.0% (0)
##            Total  4.8% (6)  5.6% (7)  5.6% (7)  4.8% (6)  4.8% (6)  4.8% (6)
##          7          8          9         10         11         12         13
##  11.1% (1)  0.0% (0)  0.0% (0) 11.1% (1)  0.0% (0) 11.1% (1) 11.1% (1)
##   0.0% (0) 11.1% (1) 11.1% (1)  0.0% (0)  0.0% (0) 11.1% (1)  0.0% (0)
##   0.0% (0) 10.0% (1)  0.0% (0) 10.0% (1)  0.0% (0) 10.0% (1)  0.0% (0)
##   0.0% (0) 14.3% (1)  0.0% (0) 14.3% (1) 14.3% (1)  0.0% (0) 14.3% (1)
##   7.7% (1)  7.7% (1)  7.7% (1)  0.0% (0)  7.7% (1)  0.0% (0)  7.7% (1)
##   5.6% (1)  0.0% (0)  5.6% (1)  0.0% (0)  5.6% (1)  0.0% (0)  5.6% (1)
##   0.0% (0)  0.0% (0) 50.0% (1)  0.0% (0)  0.0% (0) 50.0% (1)  0.0% (0)
##   0.0% (0)  0.0% (0)  0.0% (0) 14.3% (1) 14.3% (1)  0.0% (0)  0.0% (0)
##   0.0% (0) 18.2% (2)  9.1% (1)  0.0% (0)  0.0% (0)  9.1% (1)  9.1% (1)
##  14.3% (1)  0.0% (0) 14.3% (1)  0.0% (0)  0.0% (0) 14.3% (1)  0.0% (0)
##   0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)
##  10.0% (1)  0.0% (0)  0.0% (0) 10.0% (1) 10.0% (1)  0.0% (0)  0.0% (0)
##   0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)
##  10.0% (1)  0.0% (0) 10.0% (1)  0.0% (0) 10.0% (1)  0.0% (0) 10.0% (1)
##   0.0% (0) 20.0% (1)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)
##   0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)
##   4.8% (6)  5.6% (7)  5.6% (7)  4.0% (5)  4.8% (6)  4.8% (6)  4.8% (6)
##         14         15         16         17         18         19         20
##   0.0% (0) 11.1% (1)  0.0% (0) 11.1% (1)  0.0% (0)  0.0% (0)  0.0% (0)
##  11.1% (1) 11.1% (1)  0.0% (0)  0.0% (0) 11.1% (1) 11.1% (1)  0.0% (0)
##  10.0% (1)  0.0% (0) 10.0% (1) 10.0% (1)  0.0% (0) 10.0% (1)  0.0% (0)
##   0.0% (0) 14.3% (1)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0) 14.3% (1)
##   0.0% (0)  7.7% (1)  0.0% (0)  0.0% (0) 15.4% (2) 15.4% (2)  0.0% (0)
##   5.6% (1)  5.6% (1)  5.6% (1)  5.6% (1)  5.6% (1)  0.0% (0) 11.1% (2)
##   0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)
##  14.3% (1)  0.0% (0) 14.3% (1)  0.0% (0) 14.3% (1) 14.3% (1)  0.0% (0)
##   0.0% (0)  0.0% (0)  9.1% (1)  0.0% (0)  0.0% (0)  9.1% (1)  0.0% (0)
##  14.3% (1) 14.3% (1)  0.0% (0)  0.0% (0)  0.0% (0) 14.3% (1)  0.0% (0)
##   0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0)
##  10.0% (1) 10.0% (1)  0.0% (0) 10.0% (1)  0.0% (0)  0.0% (0) 10.0% (1)
##   0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0) 33.3% (1)  0.0% (0) 33.3% (1)
##   0.0% (0)  0.0% (0) 10.0% (1) 10.0% (1)  0.0% (0)  0.0% (0) 10.0% (1)
##   0.0% (0)  0.0% (0)  0.0% (0)  0.0% (0) 20.0% (1)  0.0% (0) 20.0% (1)
##   0.0% (0)  0.0% (0)  0.0% (0) 33.3% (1)  0.0% (0)  0.0% (0)  0.0% (0)
##   4.8% (6)  5.6% (7)  4.0% (5)  4.8% (6)  5.6% (7)  5.6% (7)  5.6% (7)
```

```
endpoint %>%
  tabyl(Genotype, Row) %>%
  adorn_totals("row") %>%
  adorn_percentages("row") %>%
  adorn_pct_formatting() %>%
  adorn_ns() %>%
  adorn_title("combined")
```

```
##  Genotype/Row          1           2           3           4           5           6
##      EPPN01_H 11.1%  (1) 11.1%  (1) 11.1%  (1) 22.2%  (2) 22.2%  (2) 11.1%  (1)
##      EPPN02_H 11.1%  (1) 22.2%  (2) 11.1%  (1) 22.2%  (2) 22.2%  (2)  0.0%  (0)
##      EPPN03_H 10.0%  (1) 10.0%  (1) 20.0%  (2) 10.0%  (1) 10.0%  (1) 20.0%  (2)
##      EPPN04_H 14.3%  (1) 14.3%  (1) 14.3%  (1) 14.3%  (1) 14.3%  (1) 14.3%  (1)
##      EPPN05_H  7.7%  (1)  7.7%  (1)  7.7%  (1) 23.1%  (3) 15.4%  (2) 23.1%  (3)
##      EPPN06_H 11.1%  (2) 11.1%  (2) 27.8%  (5) 16.7%  (3) 11.1%  (2) 16.7%  (3)
##      EPPN07_L  0.0%  (0)  0.0%  (0) 50.0%  (1)  0.0%  (0) 50.0%  (1)  0.0%  (0)
##      EPPN08_H 28.6%  (2) 14.3%  (1) 14.3%  (1) 14.3%  (1) 14.3%  (1)  0.0%  (0)
##      EPPN09_H 18.2%  (2) 27.3%  (3)  0.0%  (0)  9.1%  (1) 18.2%  (2)  9.1%  (1)
##      EPPN10_H 14.3%  (1)  0.0%  (0) 28.6%  (2) 14.3%  (1)  0.0%  (0) 28.6%  (2)
##      EPPN10_L 50.0%  (1) 50.0%  (1)  0.0%  (0)  0.0%  (0)  0.0%  (0)  0.0%  (0)
##      EPPN11_H 20.0%  (2) 10.0%  (1) 10.0%  (1) 20.0%  (2) 10.0%  (1) 10.0%  (1)
##      EPPN11_L 33.3%  (1)  0.0%  (0) 33.3%  (1)  0.0%  (0)  0.0%  (0)  0.0%  (0)
##      EPPN12_H 10.0%  (1) 20.0%  (2) 10.0%  (1) 10.0%  (1) 20.0%  (2) 20.0%  (2)
##      EPPN13_H  0.0%  (0) 40.0%  (2)  0.0%  (0)  0.0%  (0)  0.0%  (0) 60.0%  (3)
##      EPPN20_T 33.3%  (1)  0.0%  (0)  0.0%  (0) 33.3%  (1) 33.3%  (1)  0.0%  (0)
##         Total 14.3% (18) 14.3% (18) 14.3% (18) 15.1% (19) 14.3% (18) 15.1% (19)
##           7
##  11.1%  (1)
##  11.1%  (1)
##  20.0%  (2)
##  14.3%  (1)
##  15.4%  (2)
##   5.6%  (1)
##   0.0%  (0)
##  14.3%  (1)
##  18.2%  (2)
##  14.3%  (1)
##   0.0%  (0)
##  20.0%  (2)
##  33.3%  (1)
##  10.0%  (1)
##   0.0%  (0)
##   0.0%  (0)
##  12.7% (16)
```

```
endpoint %>%
  count(Genotype)
```

```
##    Genotype  n
## 1  EPPN01_H  9
## 2  EPPN02_H  9
## 3  EPPN03_H 10
## 4  EPPN04_H  7
## 5  EPPN05_H 13
## 6  EPPN06_H 18
## 7  EPPN07_L  2
## 8  EPPN08_H  7
## 9  EPPN09_H 11
## 10 EPPN10_H  7
## 11 EPPN10_L  2
## 12 EPPN11_H 10
## 13 EPPN11_L  3
## 14 EPPN12_H 10
## 15 EPPN13_H  5
## 16 EPPN20_T  3
```

```
get_summary_stats(data = endpoint,
                  variables,
                  type = "common")
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## ℹ Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(variables)
##
##   # Now:
##   data %>% select(all_of(variables))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## # A tibble: 2 × 10
##   variable       n   min   max median   iqr  mean    sd    se    ci
##   <fct>      <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 DW_shoot_g   125  17.7  117    33.9  9.51  35.9  12.3  1.10  2.19
## 2 FW_shoot_g   125  13.2  347.  177.  89.0  178.   67.1  6.00 11.9
```

```
skim(endpoint[variables])
```

Data summary

| | |
|---|---|
| Name | endpoint[variables] |
| Number of rows | 126 |
| Number of columns | 2 |
| _______________________ | |
| Column type frequency: | |
| numeric | 2 |

| | |
|---|---|
| Group variables | None |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| DW_shoot_g | 1 | 0.99 | 35.93 | 12.35 | 17.73 | 28.79 | 33.94 | 38.30 | 117.0 | ▇▂▁▁▁ |
| FW_shoot_g | 1 | 0.99 | 177.54 | 67.14 | 13.23 | 130.80 | 177.25 | 219.75 | 346.9 | ▁▃▇▅▂ |

## Data visualization

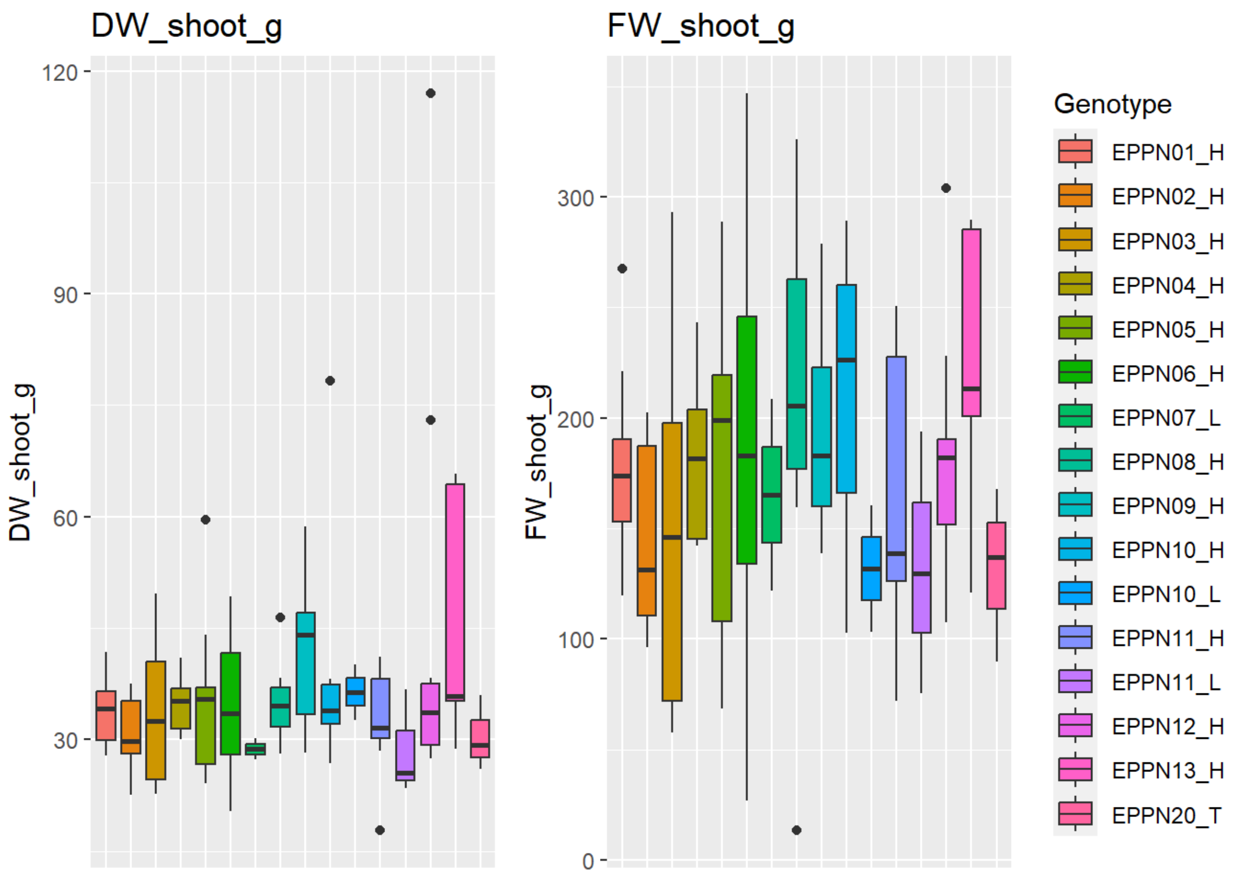
Using several functions that are located in the functions.R script

### Boxplots

```
create_boxplots(endpoint, variables, "Genotype")
```

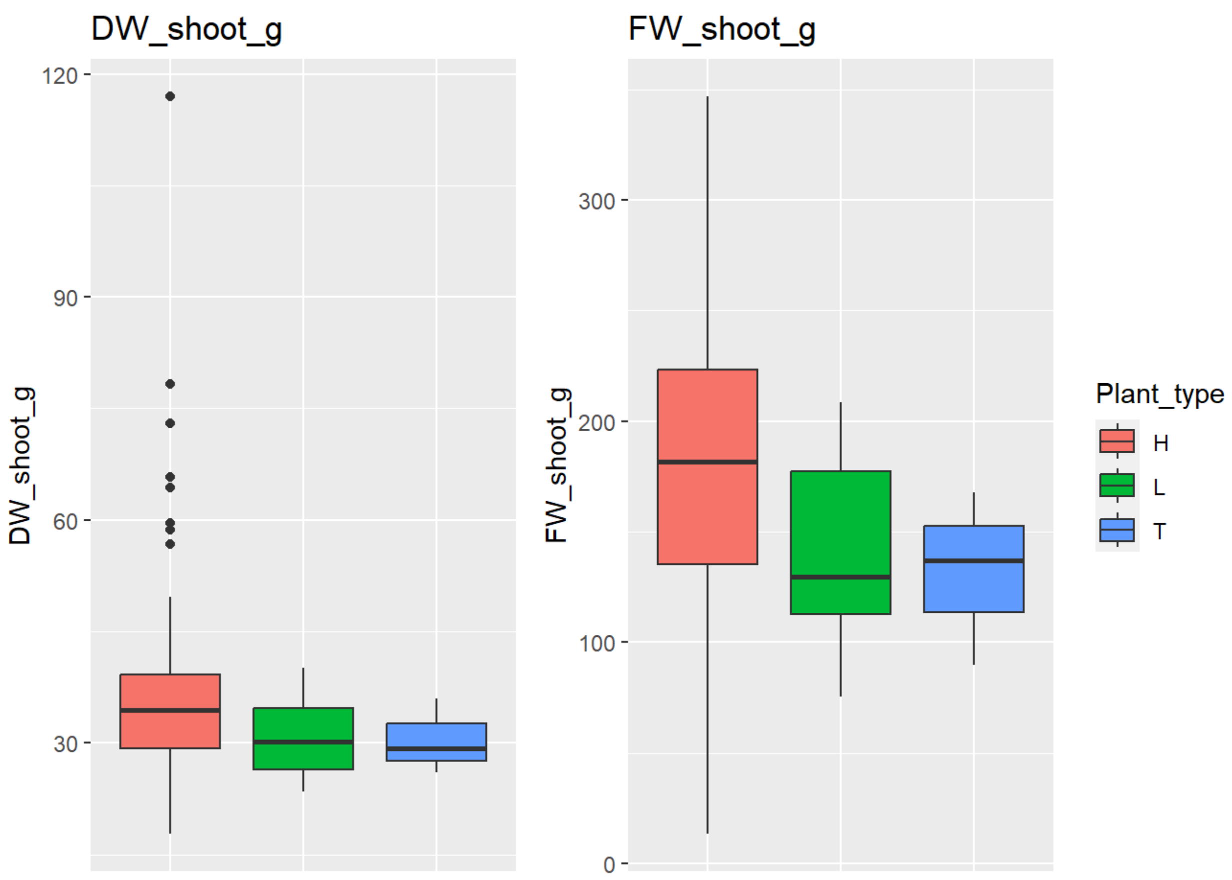
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
## Removed 1 rows containing non-finite values (`stat_boxplot()`).
```



```
create_boxplots(endpoint, variables, "Plant_type")
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
## Removed 1 rows containing non-finite values (`stat_boxplot()`).
```



## Correlation plots

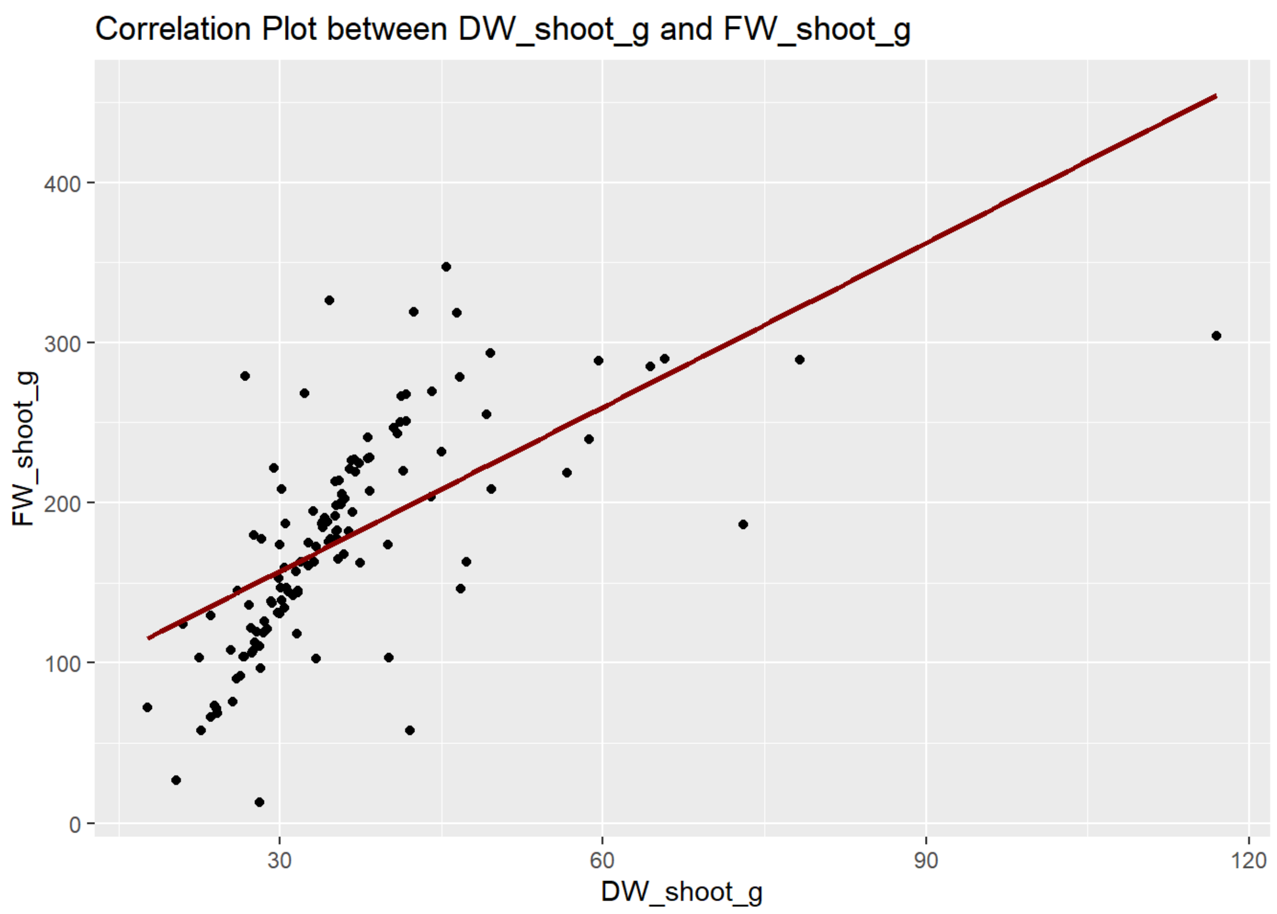
```
for (i in 1:(length(variables) - 1)) {
  for (j in (i + 1):length(variables)) {
    calculate_correlation_plot(endpoint, variables[i], variables[j])
  }
}
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 126 rows containing missing values (`geom_text()`).
```

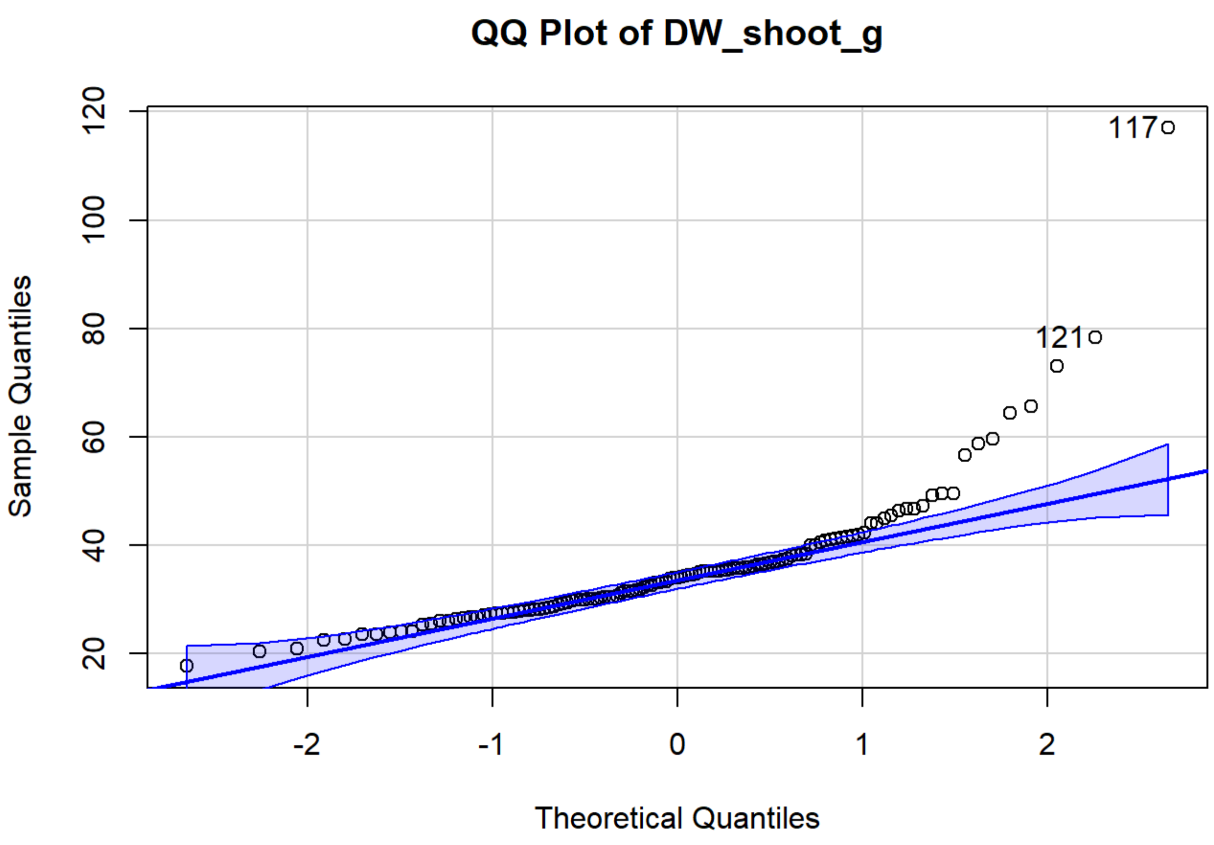


## B. Normality hypothesis and outlier detection

Test for normality hypothesis and plot density histogram. The red curve is the normal distribution, the blue dotted curve is the data density curve.

```
normality_results <- normality_test_histogram(endpoint)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

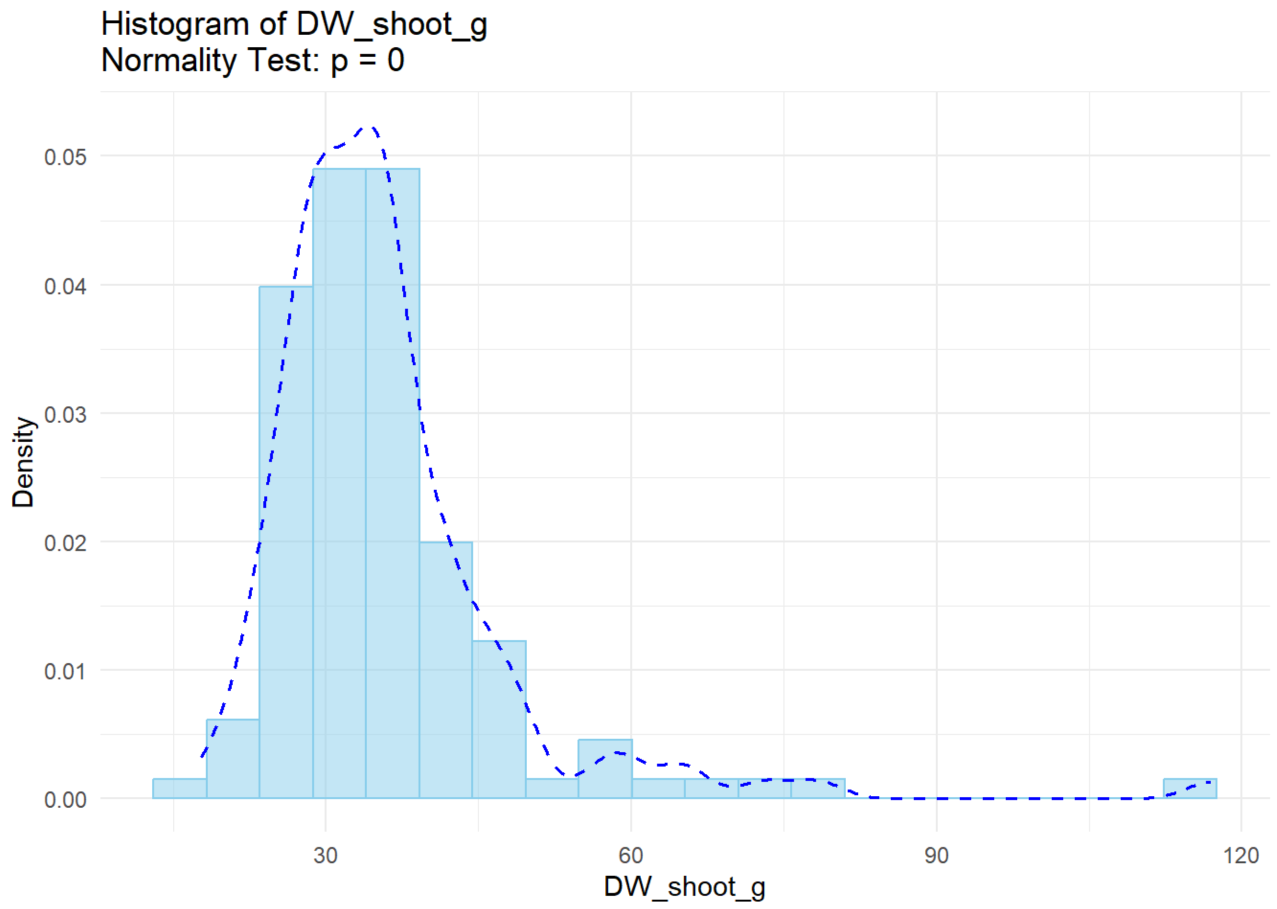


```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

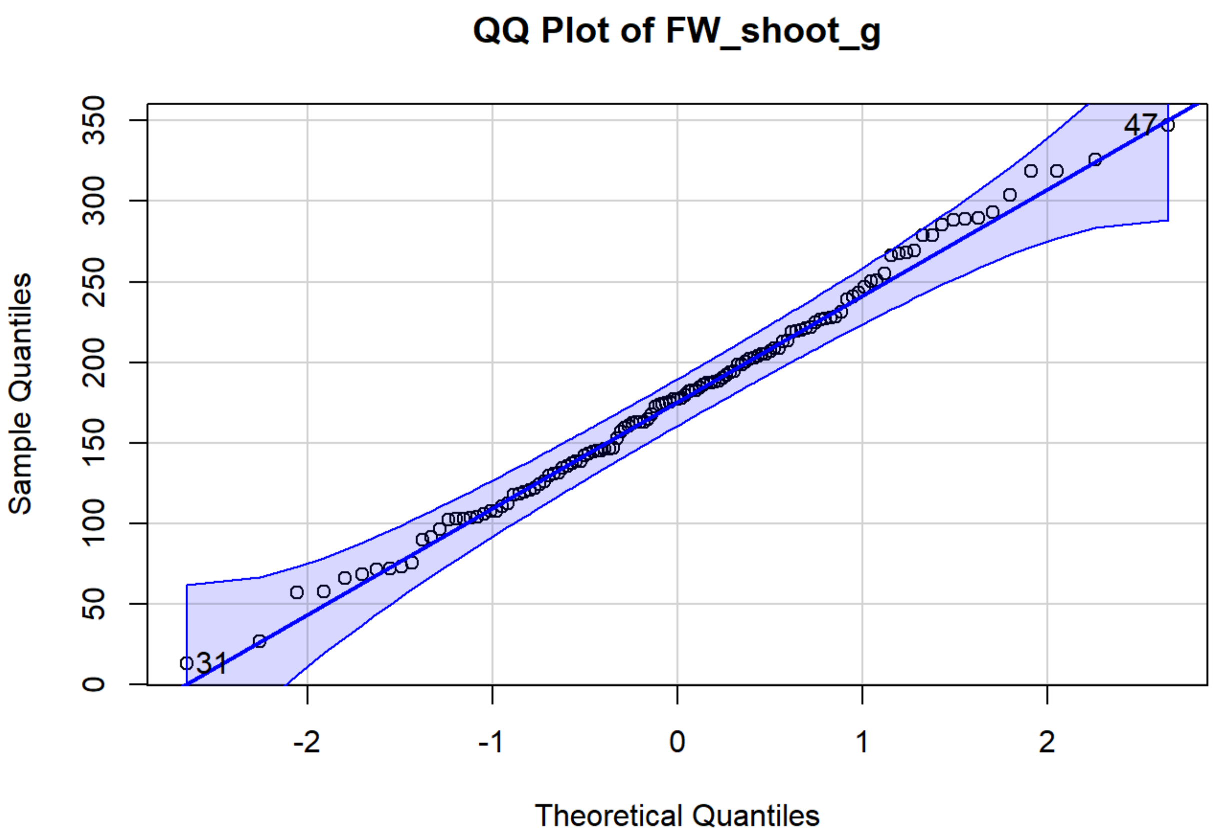
```
## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_density()`).
```

```
## Warning: Removed 101 rows containing missing values (`geom_function()`).
```

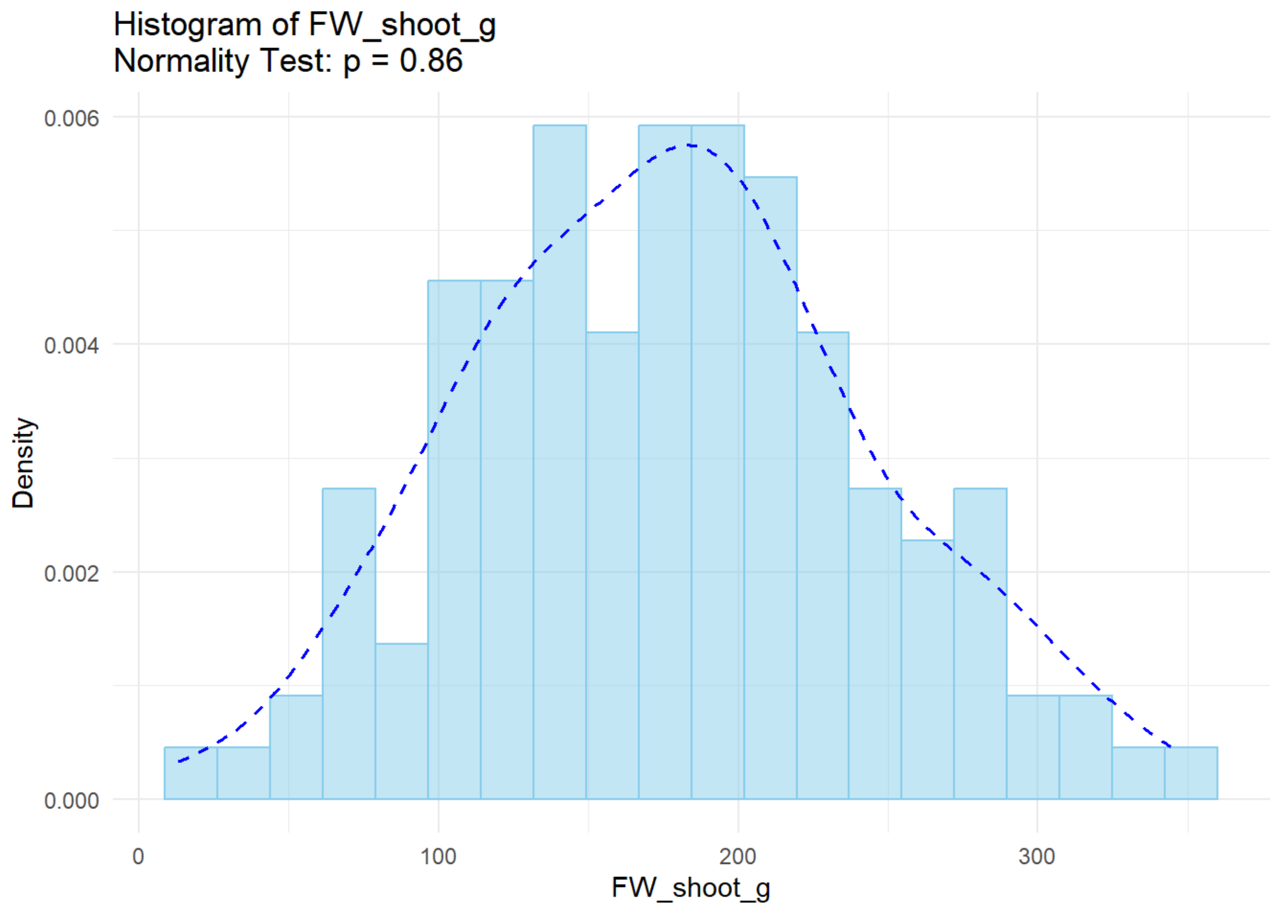


```
## [1] 117 121
```



```
## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
## Warning: Removed 1 rows containing non-finite values (`stat_density()`).
```

```
## Warning: Removed 101 rows containing missing values (`geom_function()`).
```



```
## [1] 47 31
```

Remove the outliers, replacing them with NULL values and normality visual verification.

The function detect_replace_ouliers_by_genotype checks for outlying values, using the Tukey method.

Then run the function on all variables of the dataset.

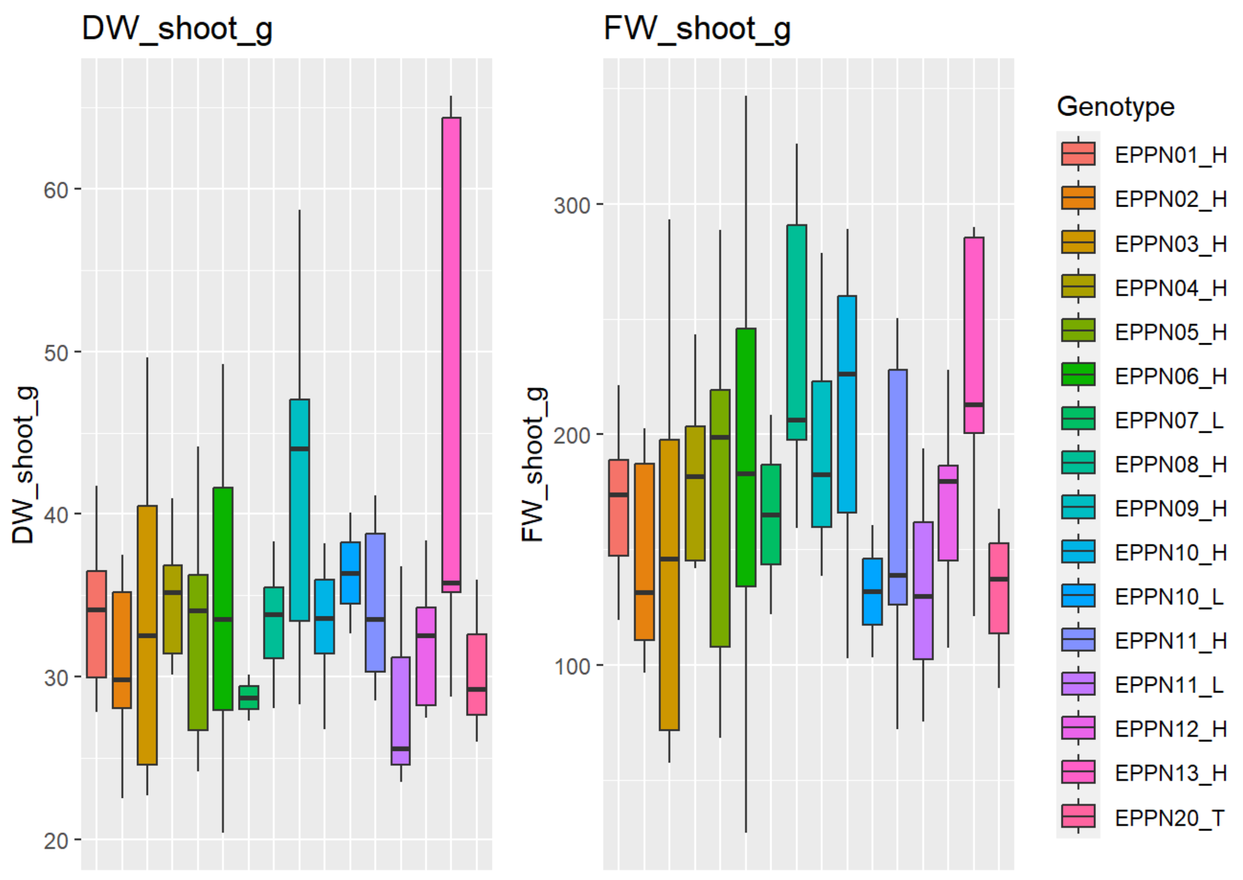
```
endpoint_clean <- endpoint
# Run the function on the dataset for all the variables
endpoint_clean <- detect_replace_outliers_by_genotype(endpoint_clean)
```

## Boxplots after outlier detection

```
create_boxplots(endpoint_clean, variables, "Genotype")
```

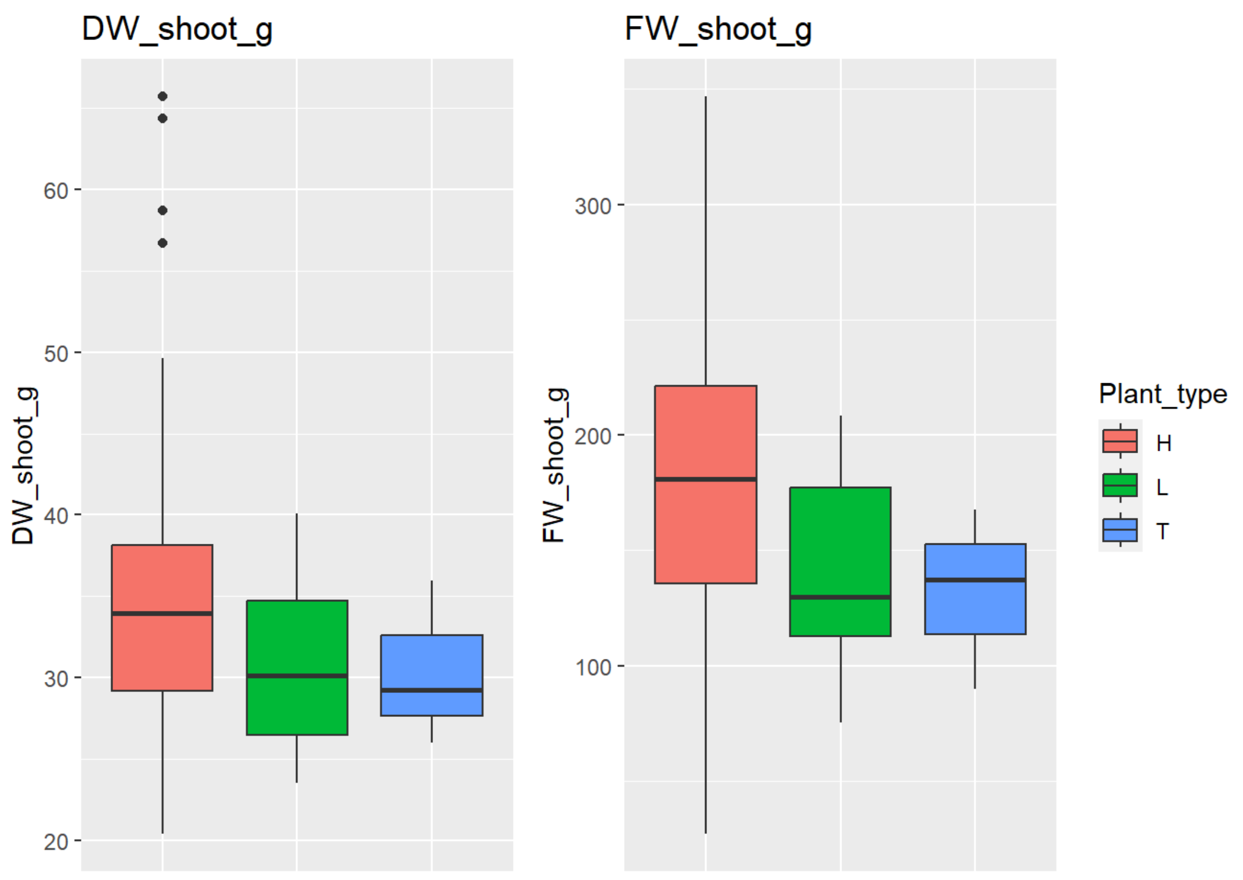
```
## Warning: Removed 7 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 4 rows containing non-finite values (`stat_boxplot()`).
```



```
create_boxplots(endpoint_clean, variables, "Plant_type")
```

```
## Warning: Removed 7 rows containing non-finite values (`stat_boxplot()`).
## Removed 4 rows containing non-finite values (`stat_boxplot()`).
```



## Violin and sina plots after outlier detection

```
create_violin_plots(endpoint_clean, variables, "Genotype")
```

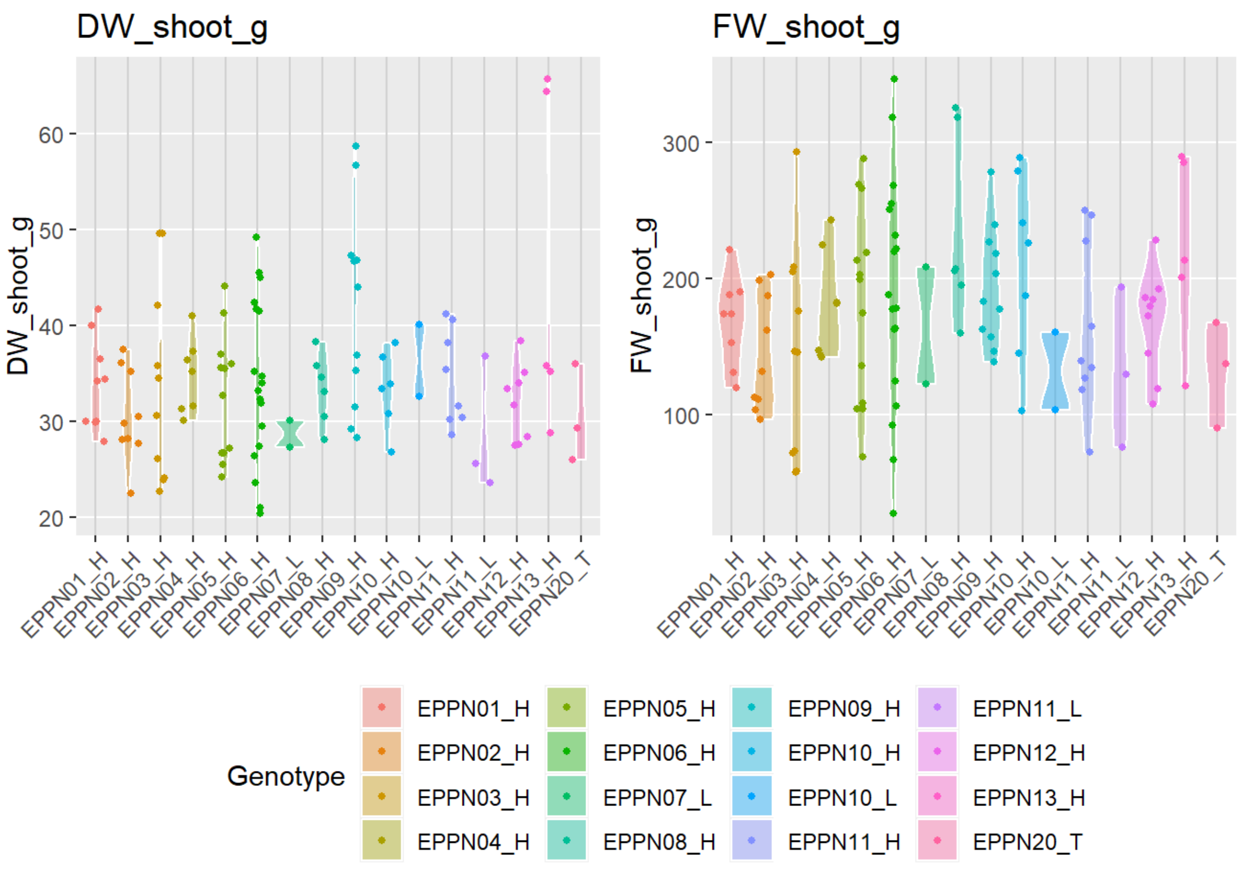
```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## ℹ Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 7 rows containing non-finite values (`stat_ydensity()`).
```

```
## Warning: Removed 7 rows containing non-finite values (`stat_sina()`).
```

```
## Warning: Removed 4 rows containing non-finite values (`stat_ydensity()`).
```

```
## Warning: Removed 4 rows containing non-finite values (`stat_sina()`).
```

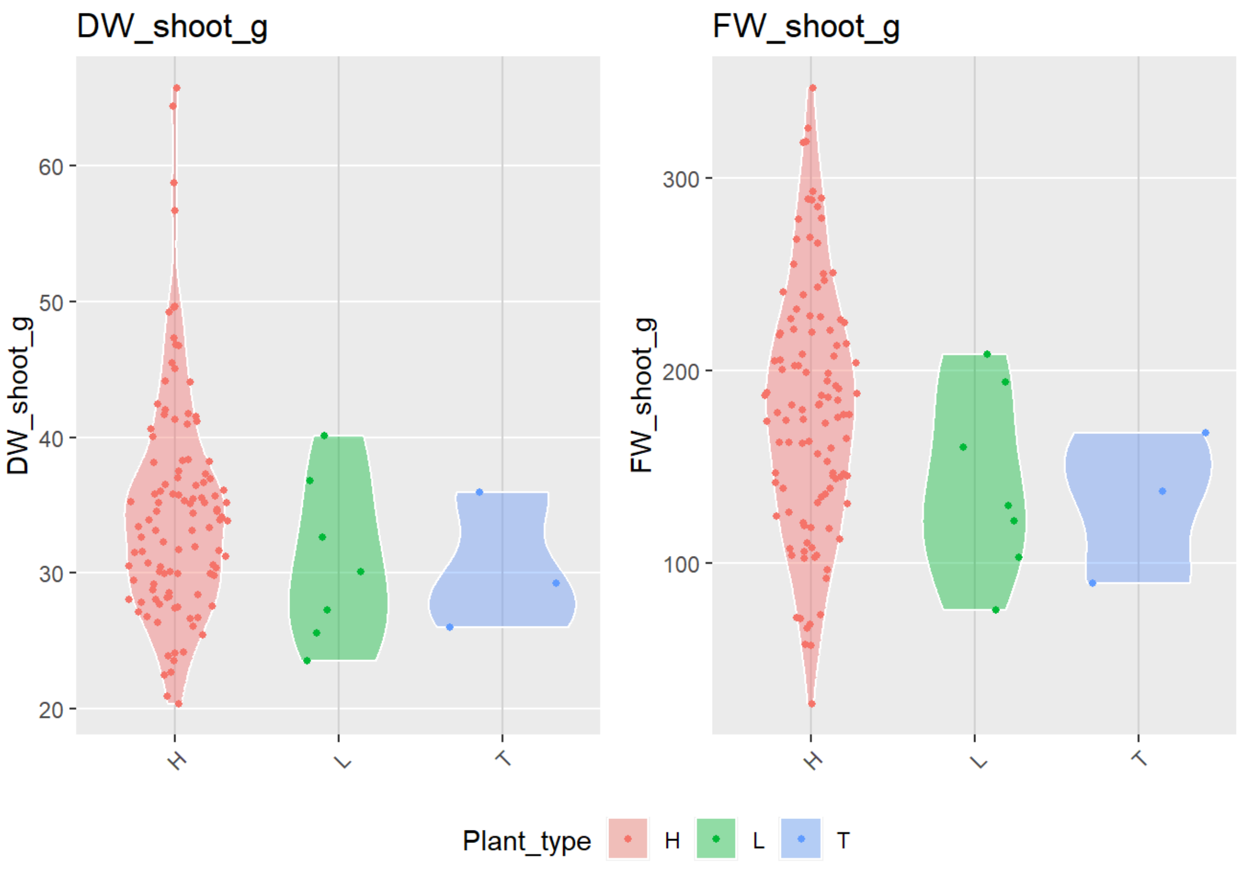


```
create_violin_plots(endpoint_clean, variables, "Plant_type")
```

```
## Warning: Removed 7 rows containing non-finite values (`stat_ydensity()`).
```

```
## Warning: Removed 7 rows containing non-finite values (`stat_sina()`).
```

```
## Warning: Removed 4 rows containing non-finite values (`stat_ydensity()`).
## Warning: Removed 4 rows containing non-finite values (`stat_sina()`).
```



## Exploration statistics for the variables after outlier detection

```
skim(endpoint_clean[variables])
```

Data summary

| | |
|---|---|
| Name | endpoint_clean[variables] |
| Number of rows | 126 |
| Number of columns | 2 |
| _______________________ | |
| Column type frequency: | |
| numeric | 2 |
| ________________________ | |
| Group variables | None |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| DW_shoot_g | 7 | 0.94 | 34.45 | 8.18 | 20.37 | 28.67 | 33.87 | 37.41 | 65.73 | ▃▇▃▁▁ |
| FW_shoot_g | 4 | 0.97 | 177.12 | 64.78 | 26.89 | 130.97 | 177.24 | 219.18 | 346.90 | ▂▆▇▃▂ |

```
for (var in variables) {
  cat("\nSummary for:", var, "\n")
  endpoint_clean %>%
    group_by(Genotype) %>%
    summarize(mean    = mean(get(var), na.rm = TRUE),
              std.dev = sd(get(var), na.rm = TRUE),
              n_missing  = sum(is.na(get(var)))) %>%
    arrange(desc(mean)) %>%
    print(n = Inf)
}
```

```
##
## Summary for: DW_shoot_g
## # A tibble: 16 × 4
##    Genotype  mean std.dev n_missing
##    <fct>    <dbl>   <dbl>     <int>
##  1 EPPN13_H  46.0   17.6          0
##  2 EPPN09_H  41.9   10.5          0
##  3 EPPN10_L  36.4    5.29         0
##  4 EPPN04_H  34.7    3.91         0
##  5 EPPN11_H  34.5    5.01         2
##  6 EPPN06_H  34.1    8.64         0
##  7 EPPN03_H  33.9   10.3          0
##  8 EPPN01_H  33.8    4.87         0
##  9 EPPN08_H  33.4    3.70         1
## 10 EPPN10_H  33.3    4.11         1
## 11 EPPN05_H  32.7    6.61         1
## 12 EPPN12_H  32.0    3.94         2
## 13 EPPN02_H  30.6    4.81         0
## 14 EPPN20_T  30.4    5.07         0
## 15 EPPN07_L  28.7    2.00         0
## 16 EPPN11_L  28.6    7.13         0
##
## Summary for: FW_shoot_g
## # A tibble: 16 × 4
##    Genotype  mean std.dev n_missing
##    <fct>    <dbl>   <dbl>     <int>
##  1 EPPN08_H  235.    69.6         1
##  2 EPPN13_H  222.    69.5         0
##  3 EPPN10_H  210.    69.0         0
##  4 EPPN09_H  194.    43.7         0
##  5 EPPN06_H  189.    85.2         0
##  6 EPPN05_H  181.    71.8         0
##  7 EPPN04_H  181.    40.6         0
##  8 EPPN01_H  169.    33.3         1
##  9 EPPN12_H  168.    38.0         1
## 10 EPPN07_L  165.    61.2         0
## 11 EPPN11_H  164.    63.1         1
## 12 EPPN02_H  145.    43.0         0
## 13 EPPN03_H  143.    79.2         0
## 14 EPPN11_L  133.    59.3         0
## 15 EPPN10_L  132.    40.5         0
## 16 EPPN20_T  132.    39.3         0
```

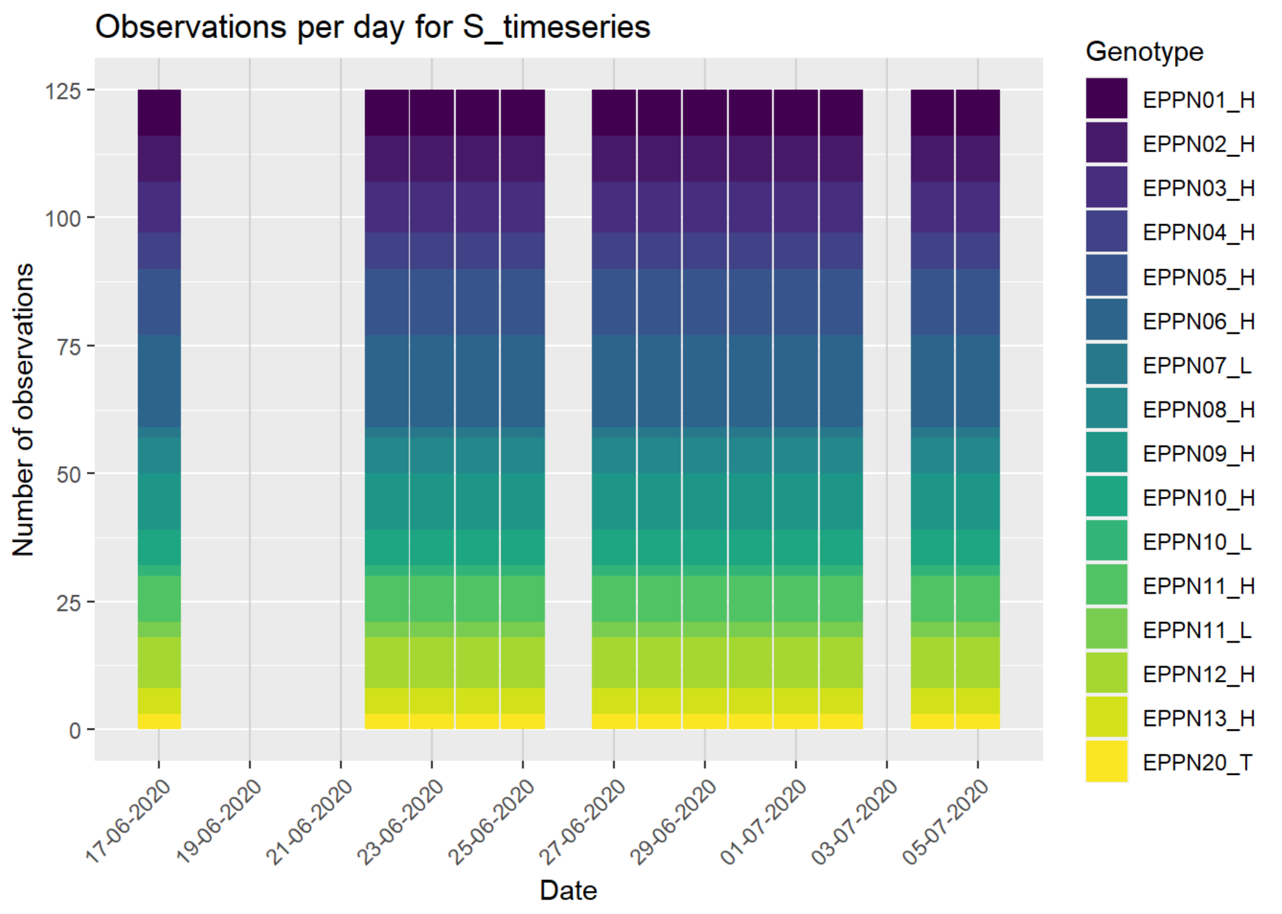
# 2. Exploration of the timeseries data

In this part, we look at the timeseries, S_timeseries and T_timeseries datasets, also using several functions, located in the functions.R script.

## Number of data observations per day for the traits of the timeseries datasets

```
h2 <- ggplot(S_timeseries, aes(x = Date)) +
  geom_bar(aes(fill = Genotype), position = "stack", width = 0.96) +
  scale_fill_viridis_d(option = "D") +
  labs(x = "Date", y = "Number of observations", title = "Observations per day for S_timeser
ies") +
  scale_y_continuous(breaks = seq(from = 0, to = 325, by = 25)) +
  scale_x_date(date_breaks = "2 days", date_labels = "%d-%m-%Y") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid.major.x = element_line(color = "lightgray", size = 0.5),
        panel.grid.minor.x = element_blank())

h2
```



## A. Exploration of the timeseries dataframe

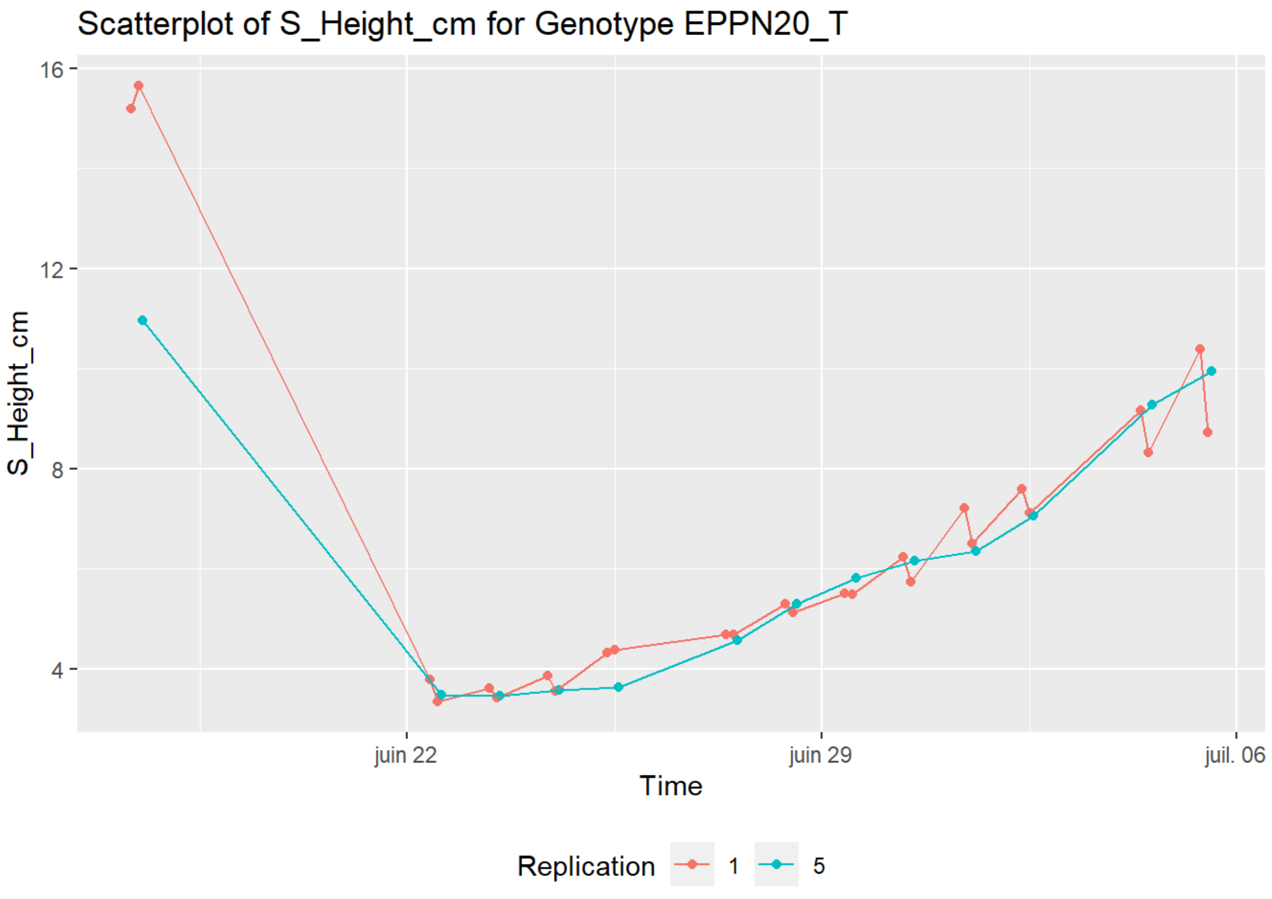
```
print(paste0("No data for", platform))
```

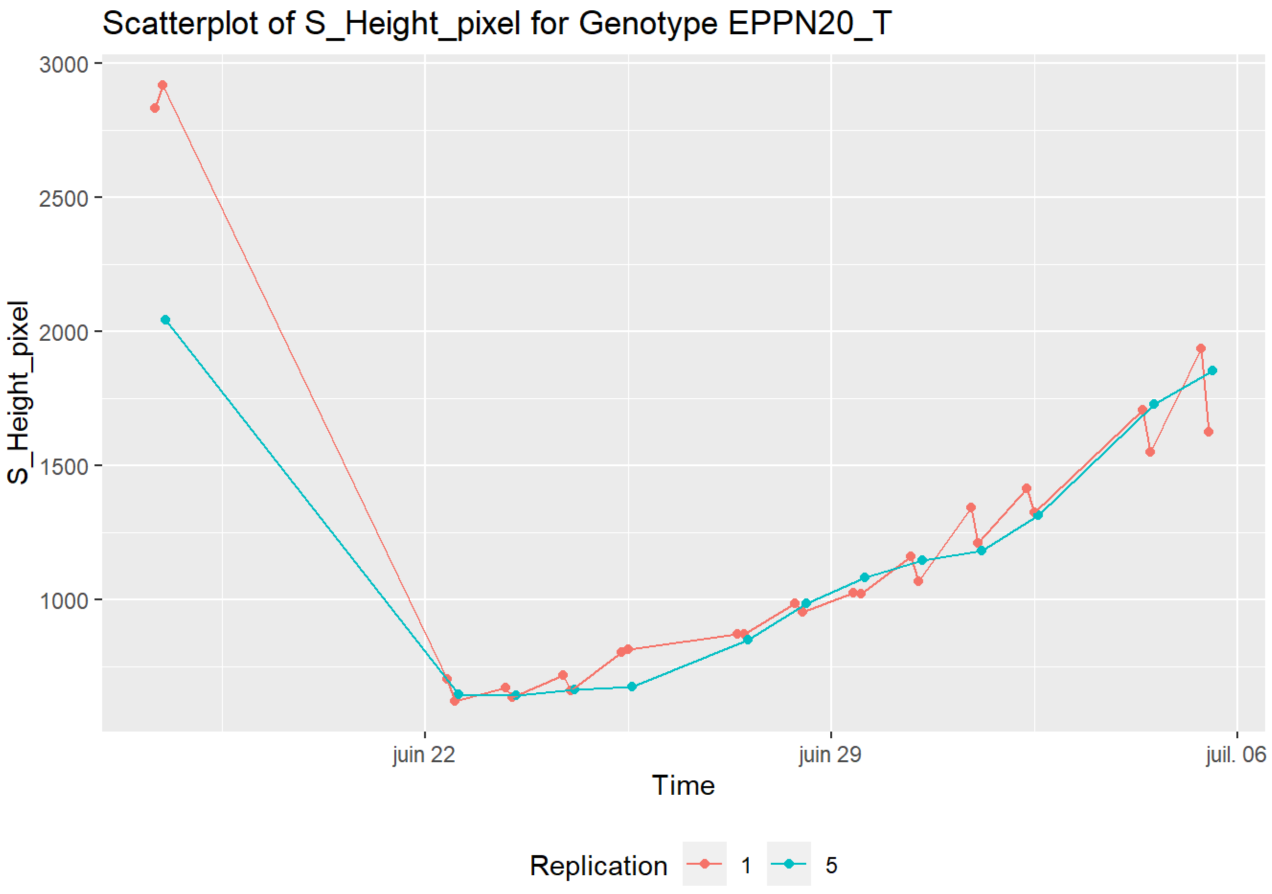
```
## [1] "No data forNaPPI"
```

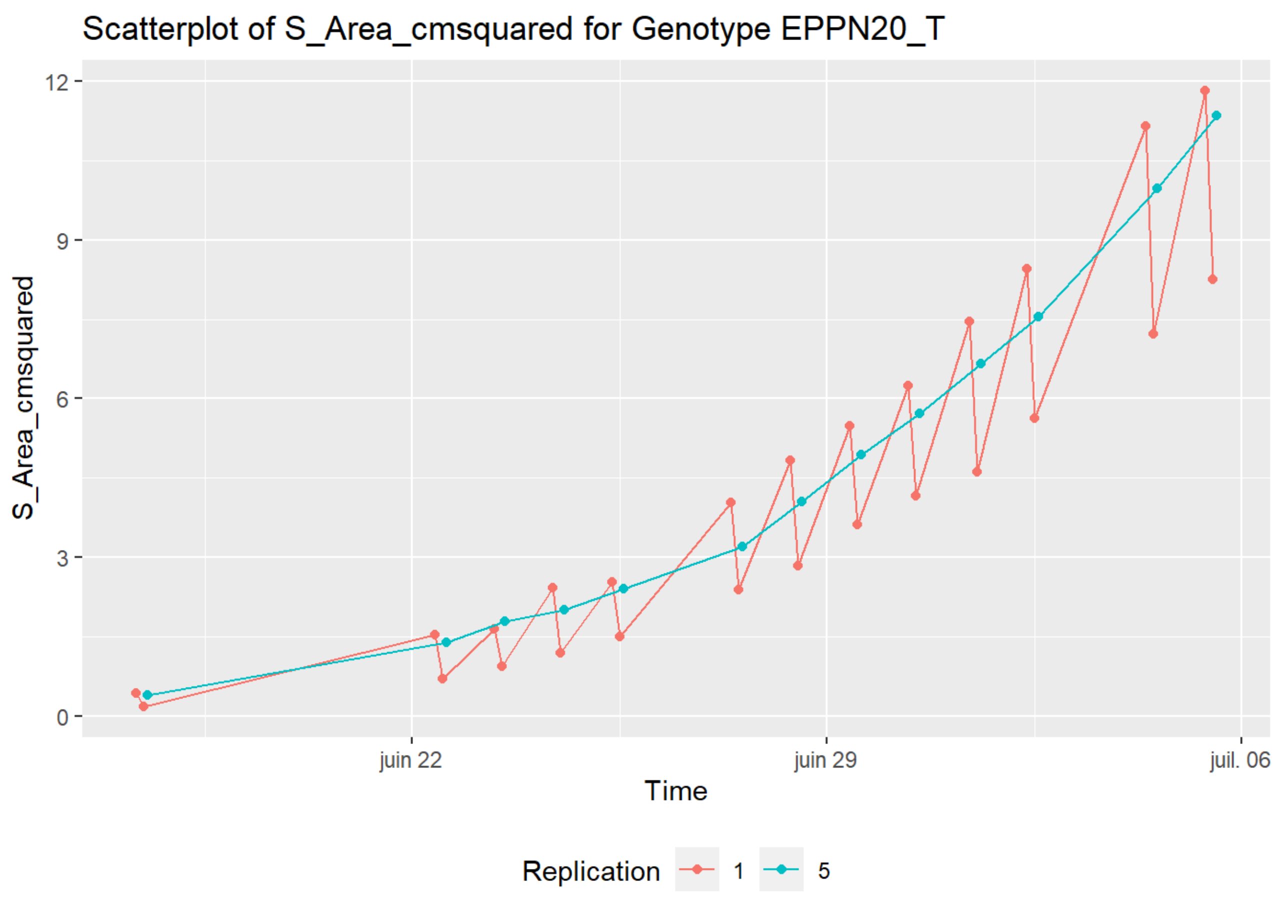
## B. Exploration of the S_timeseries dataframe

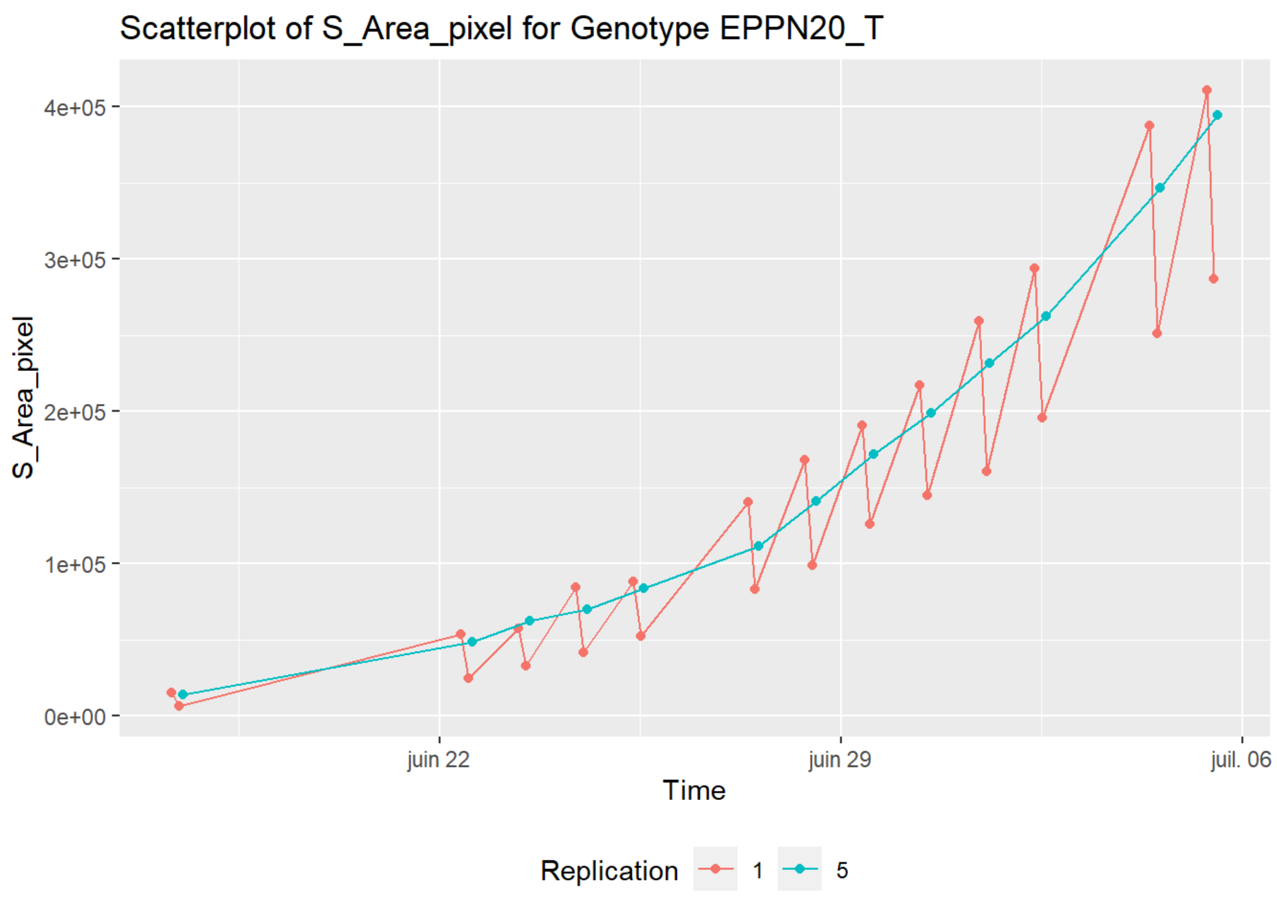
Scatter plots by Genotype

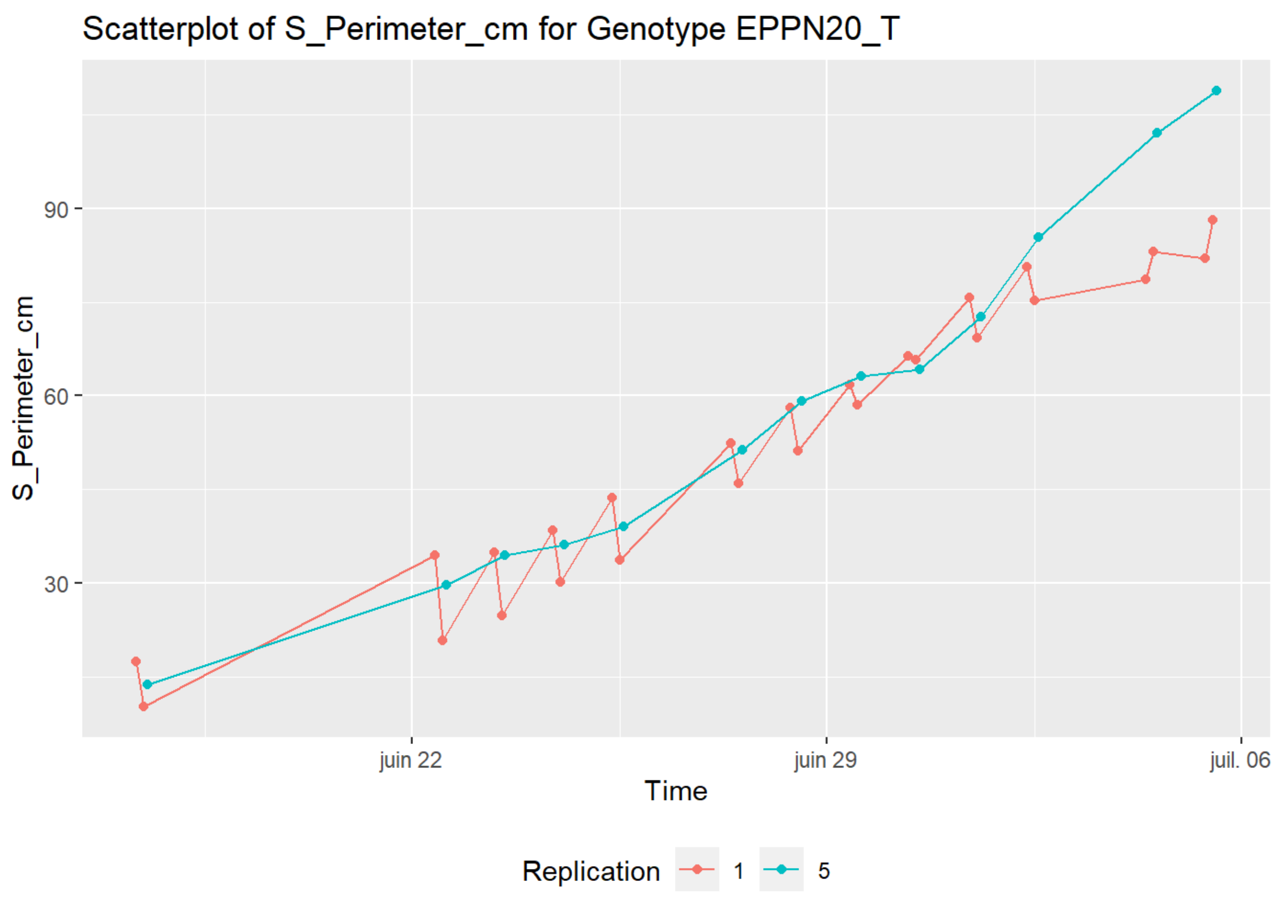
```
plot_scatter_by_genotype(S_timeseries, variables_S, "EPPN20_T")
```

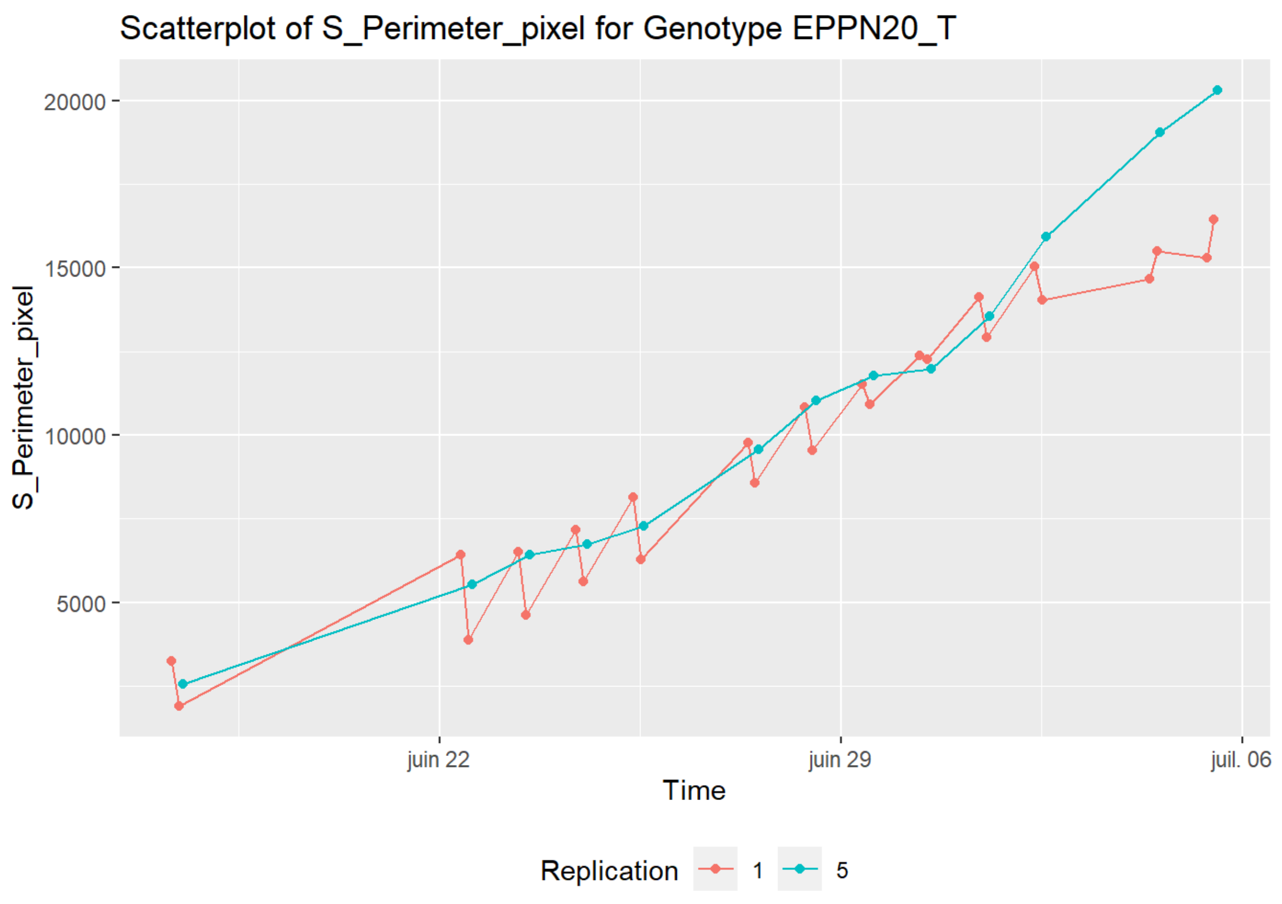

Scatterplot of S_Height_cm for Genotype EPPN20_T
S_Height_cm
16
12
8
4
juin 22
juin 29
juil. 06
Time
Replication
1
5

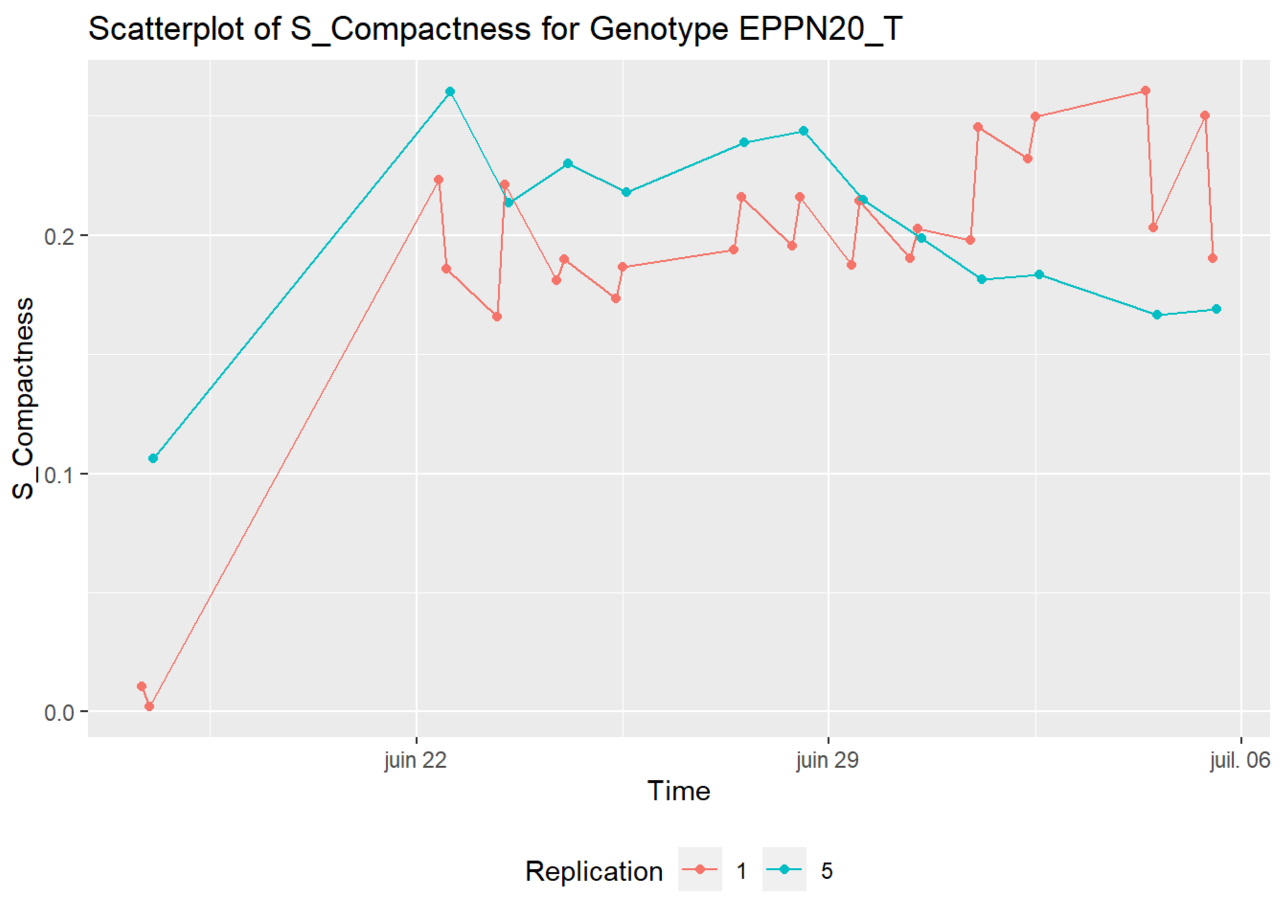

Scatterplot of S_Height_pixel for Genotype EPPN20_T
S_Height_pixel
3000
2500
2000
1500
1000
juin 22
juin 29
juil. 06
Time
Replication
1
5

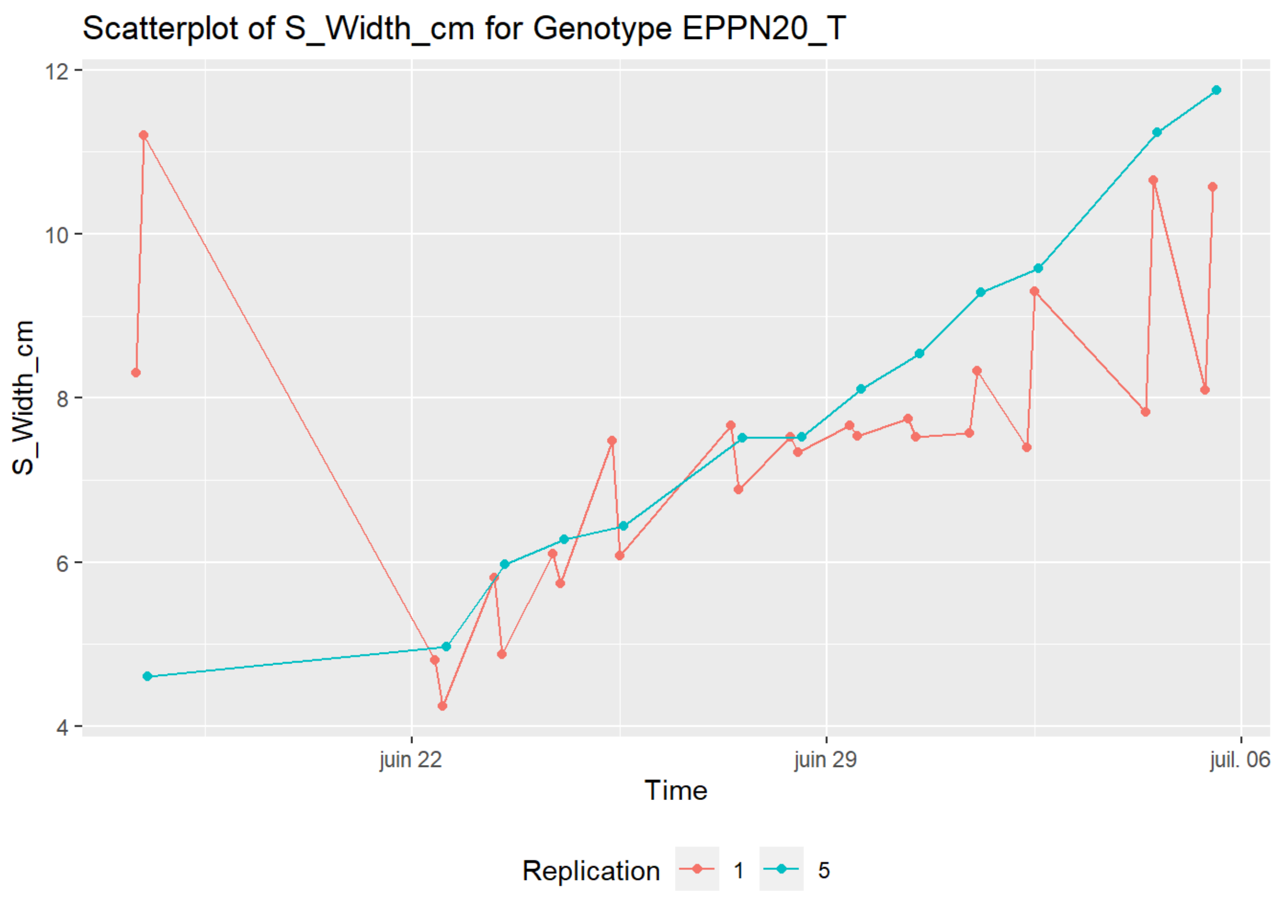

Scatterplot of S_Area_cmsquared for Genotype EPPN20_T
S_Area_cmsquared
12
9
6
3
0
juin 22
juin 29
juil. 06
Time
Replication
1
5

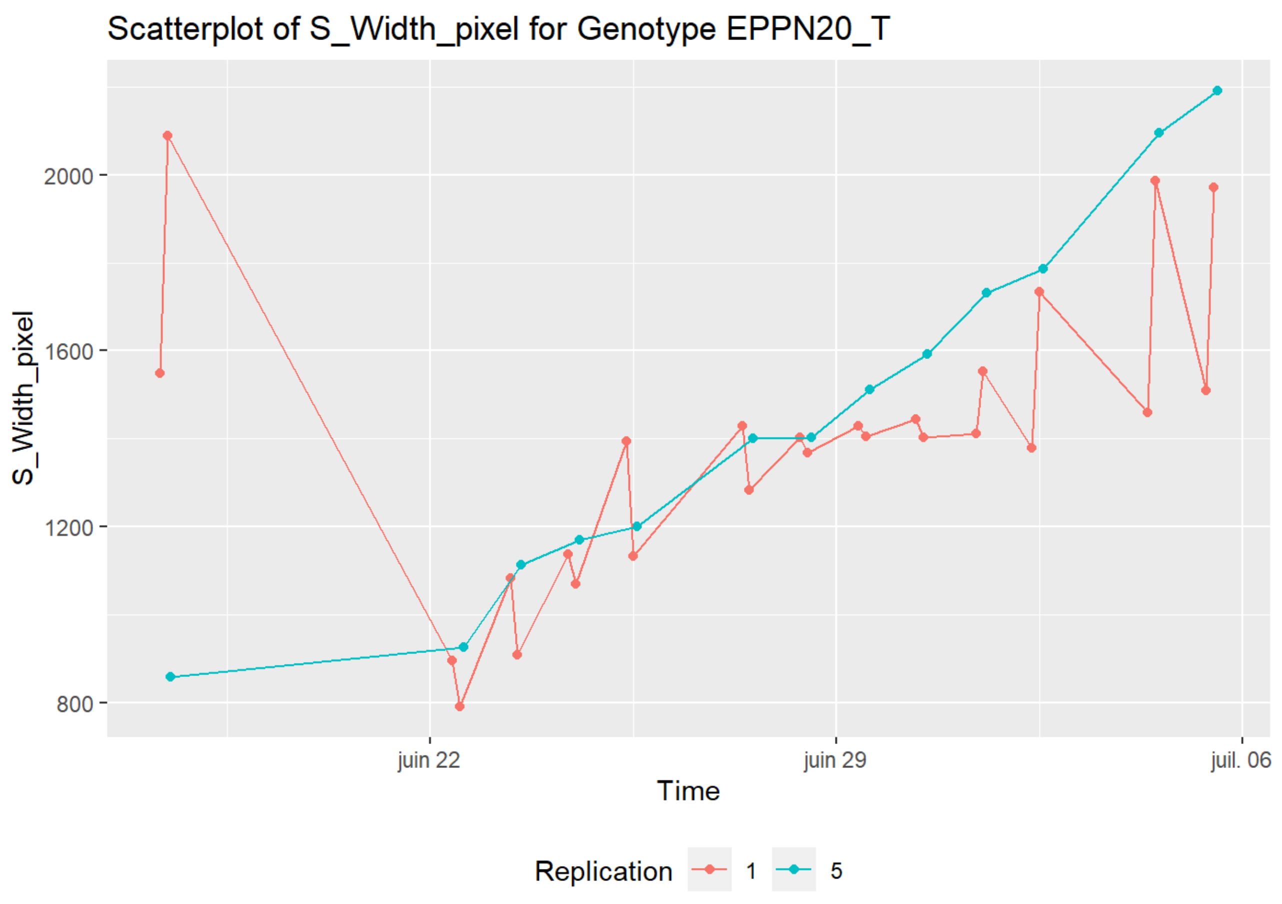

Scatterplot of S_Area_pixel for Genotype EPPN20_T
S_Area_pixel
4e+05
3e+05
2e+05
1e+05
0e+00
juin 22
juin 29
juil. 06
Time
Replication
1
5


Scatterplot of S_Perimeter_cm for Genotype EPPN20_T
S_Perimeter_cm
90
60
30
juin 22
juin 29
juil. 06
Time
Replication
1
5


Scatterplot of S_Perimeter_pixel for Genotype EPPN20_T
S_Perimeter_pixel
20000
15000
10000
5000
juin 22
juin 29
juil. 06
Time
Replication
1
5

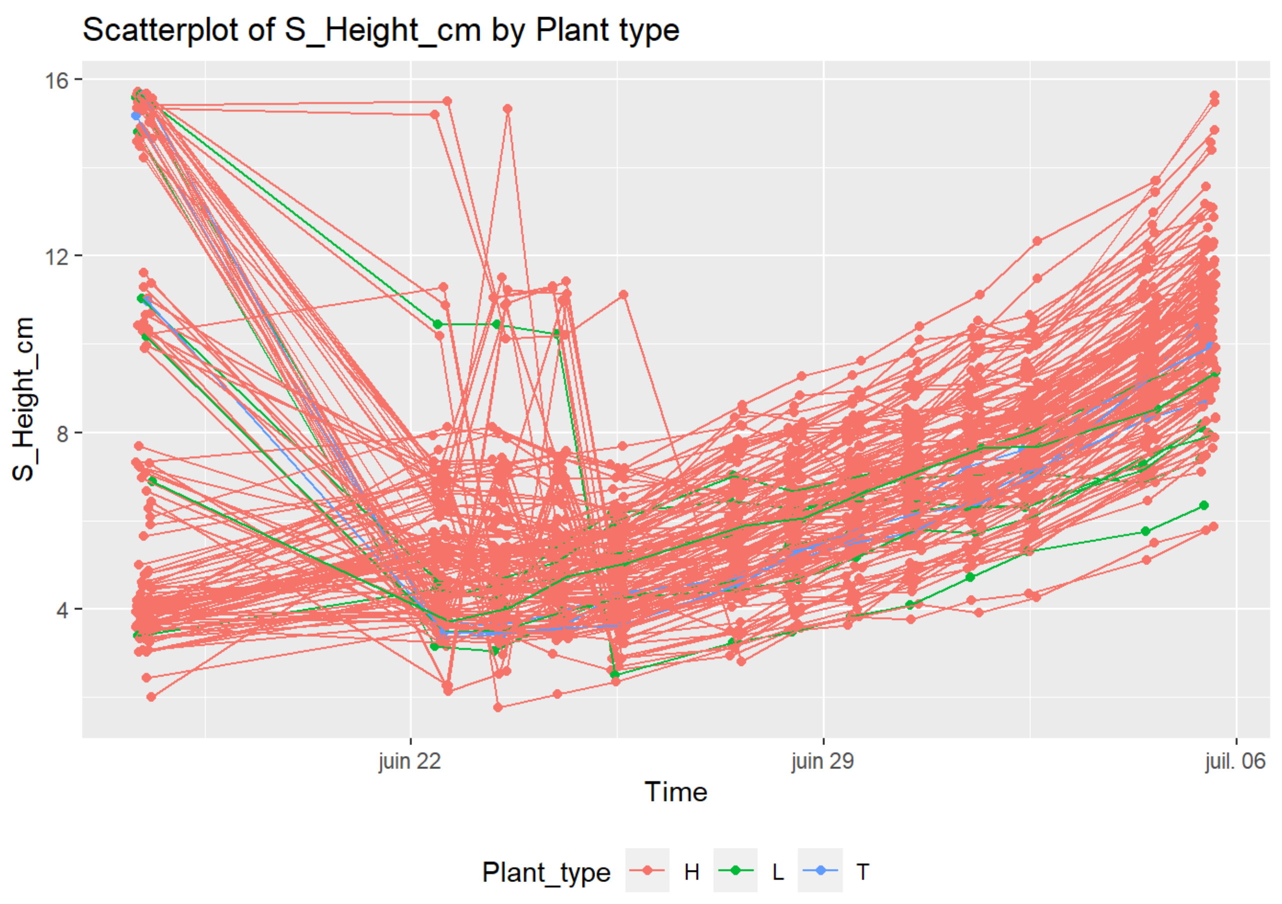

Scatterplot of S_Compactness for Genotype EPPN20_T
S_Compactness
0.0
0.1
0.2
juin 22
juin 29
juil. 06
Time
Replication
1
5


Scatterplot of S_Width_cm for Genotype EPPN20_T
S_Width_cm
4
6
8
10
12
juin 22
juin 29
juil. 06
Time
Replication
1
5

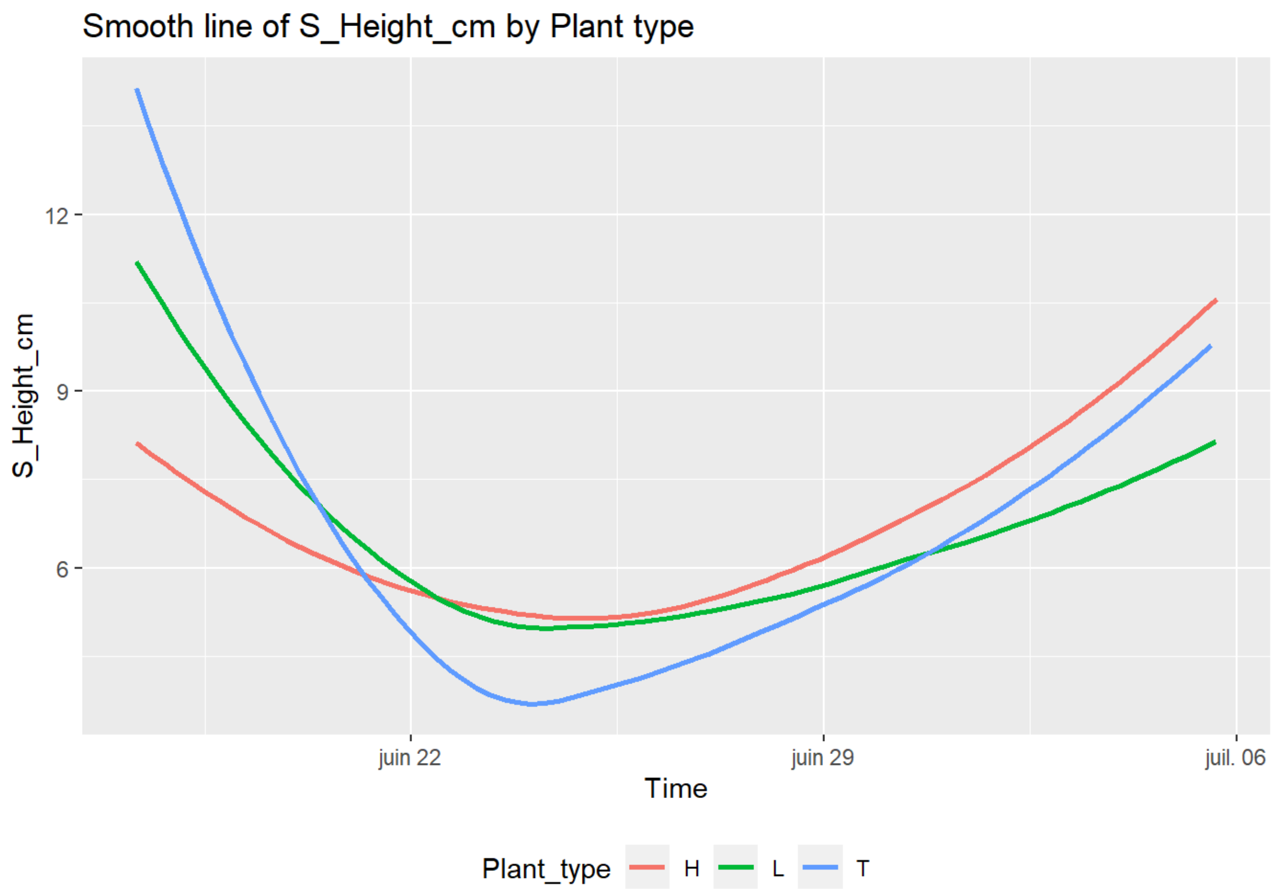


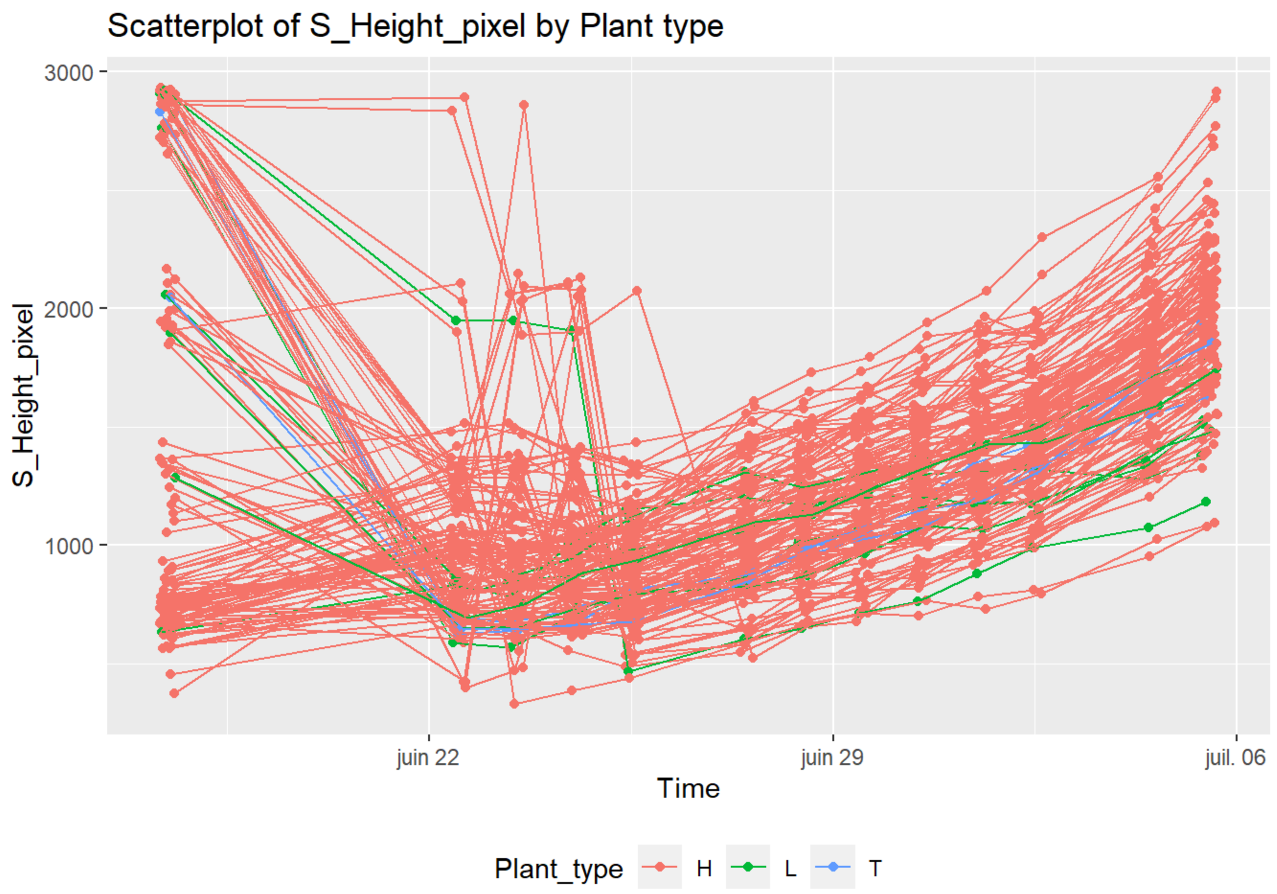
Scatterplots for all genotypes by Plant type (Hybride, Line, EPPN20_T) with smooth line.

```
plot_scatter_with_smooth(S_timeseries, variables_S)
```

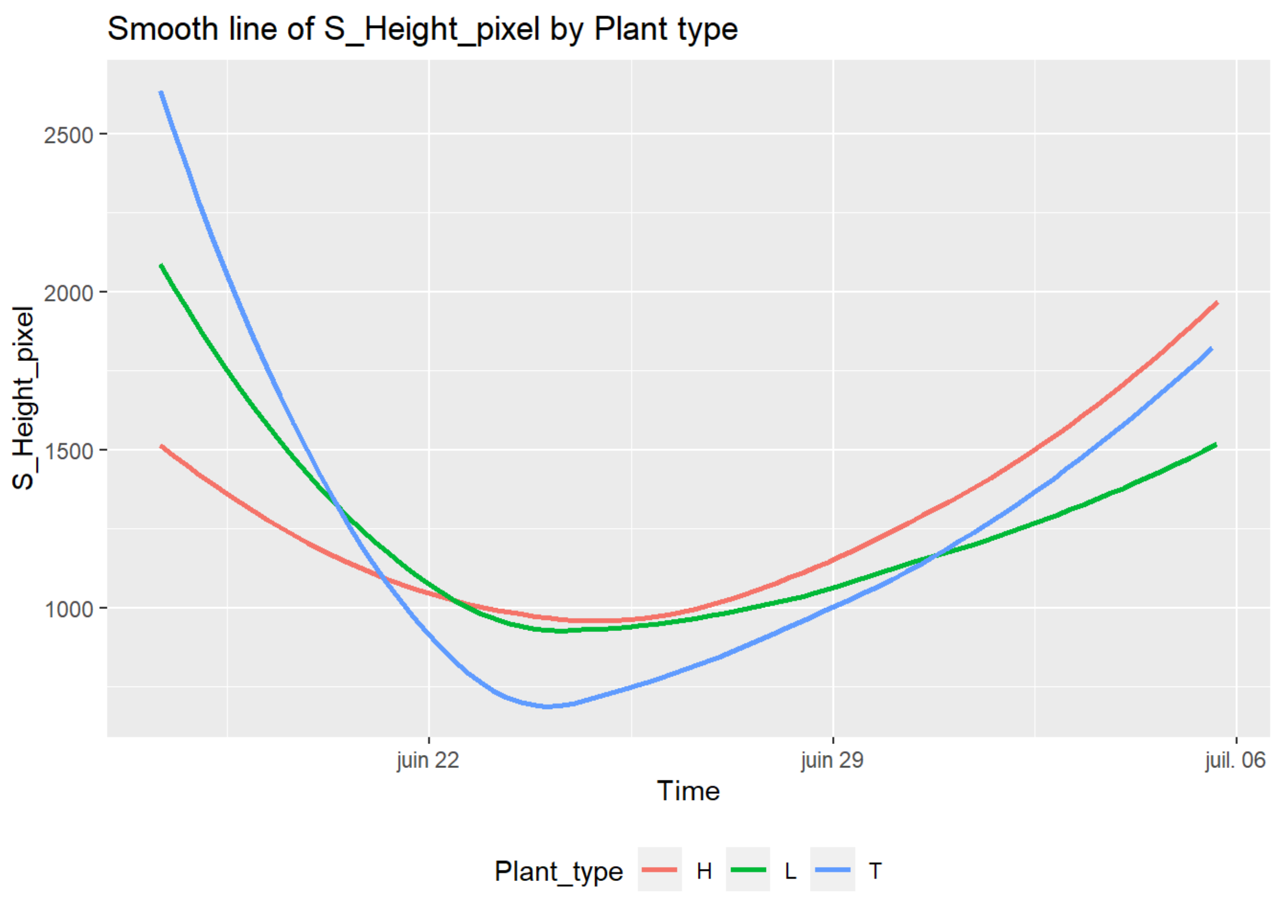


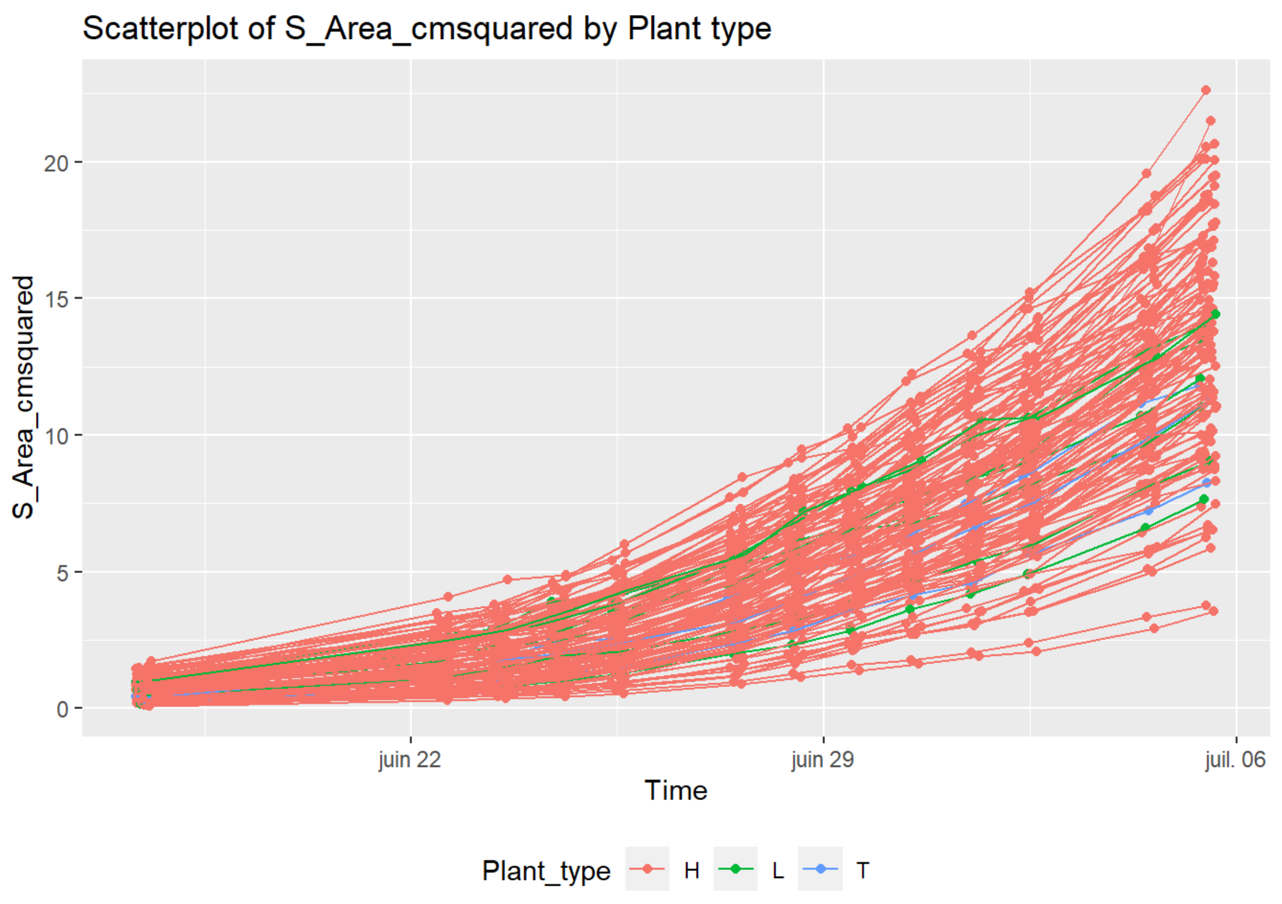
```
## `geom_smooth()` using formula = 'y ~ x'
```



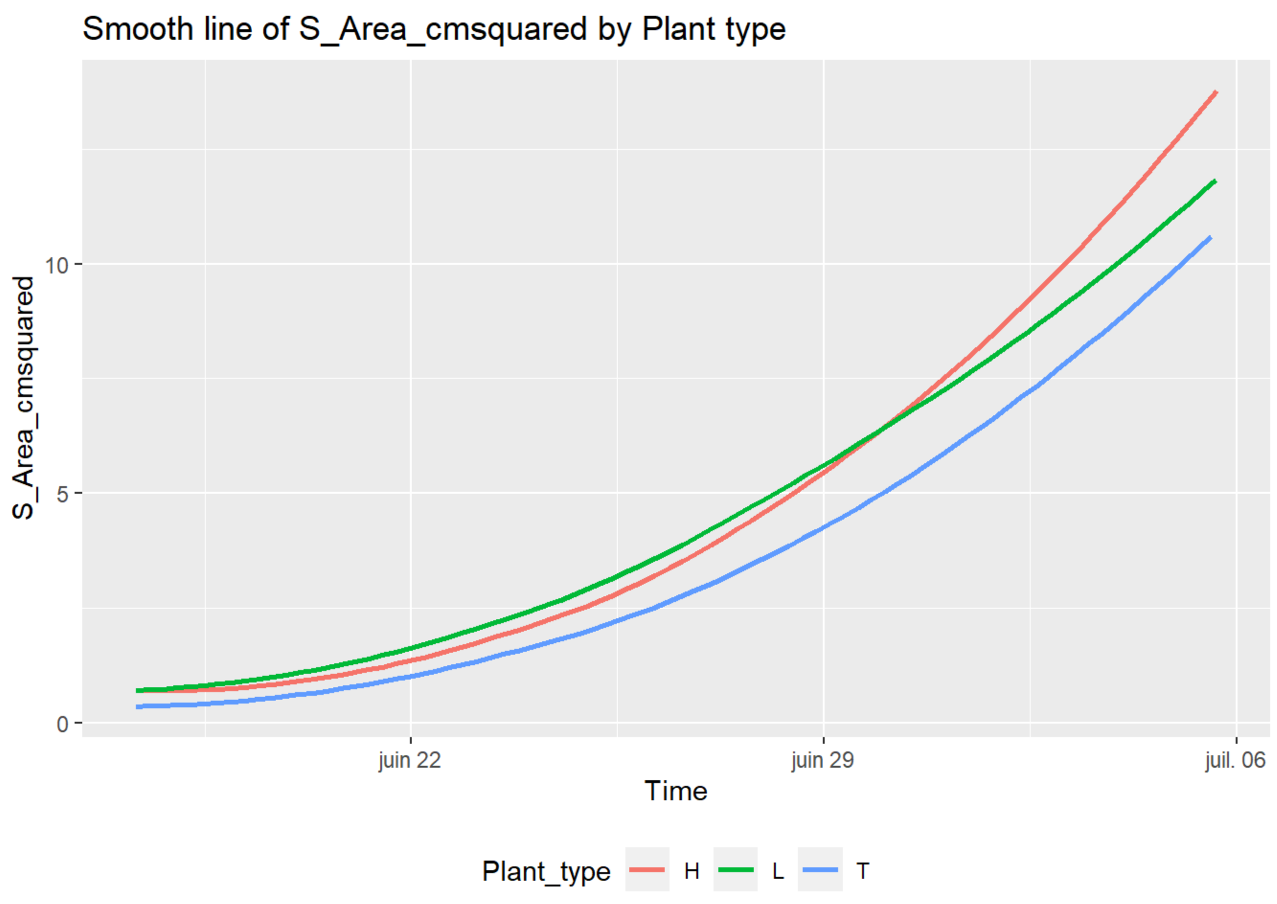


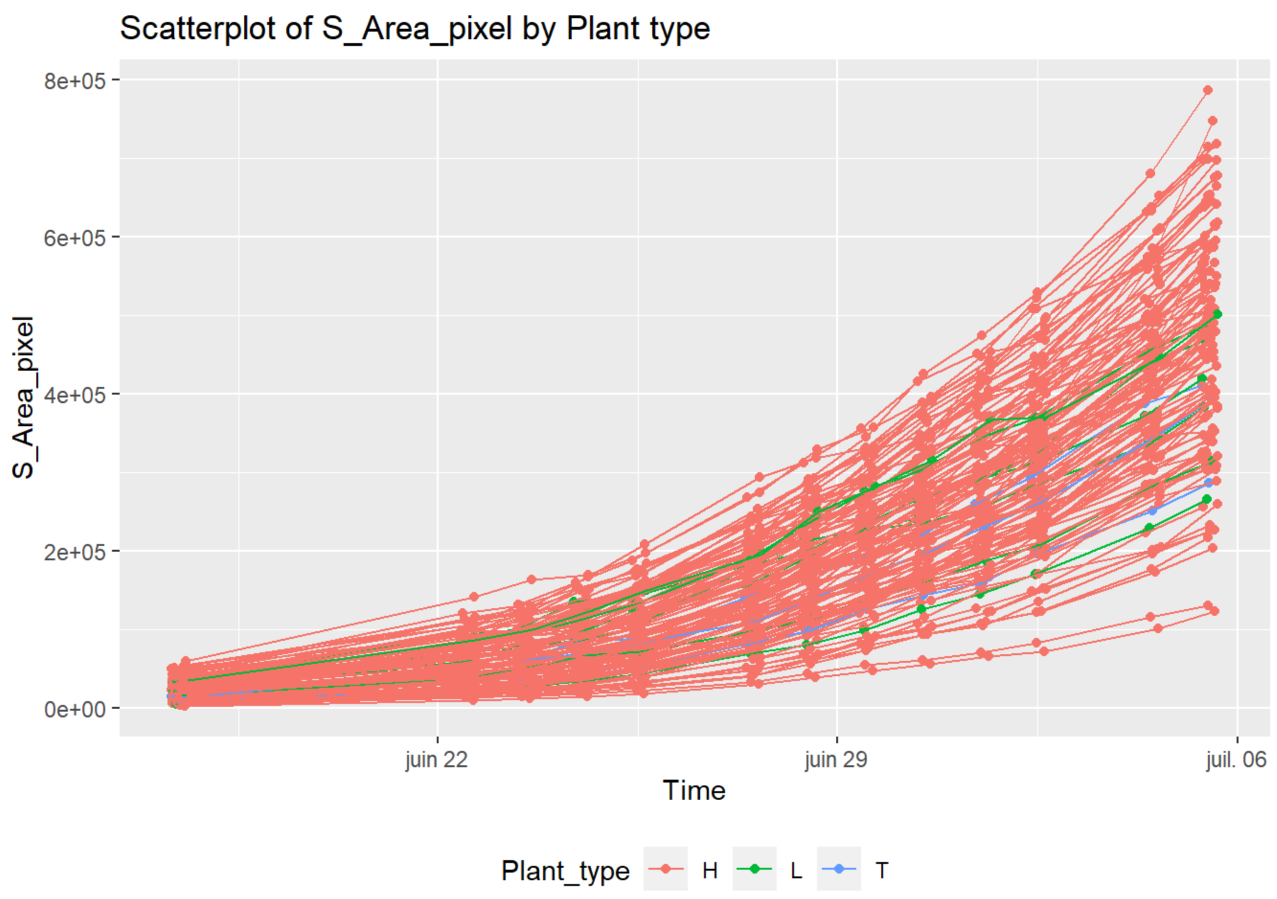
```
## `geom_smooth()` using formula = 'y ~ x'
```



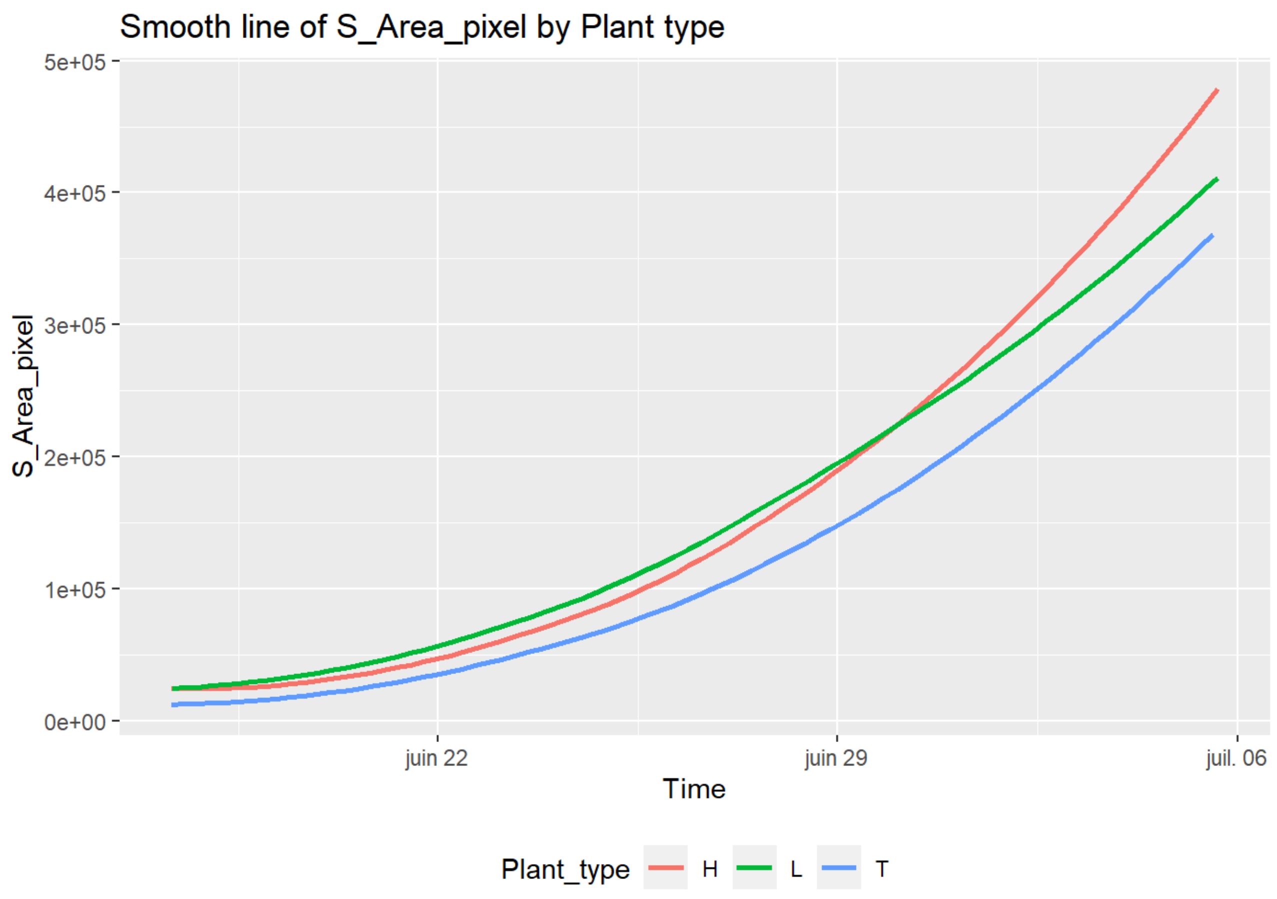


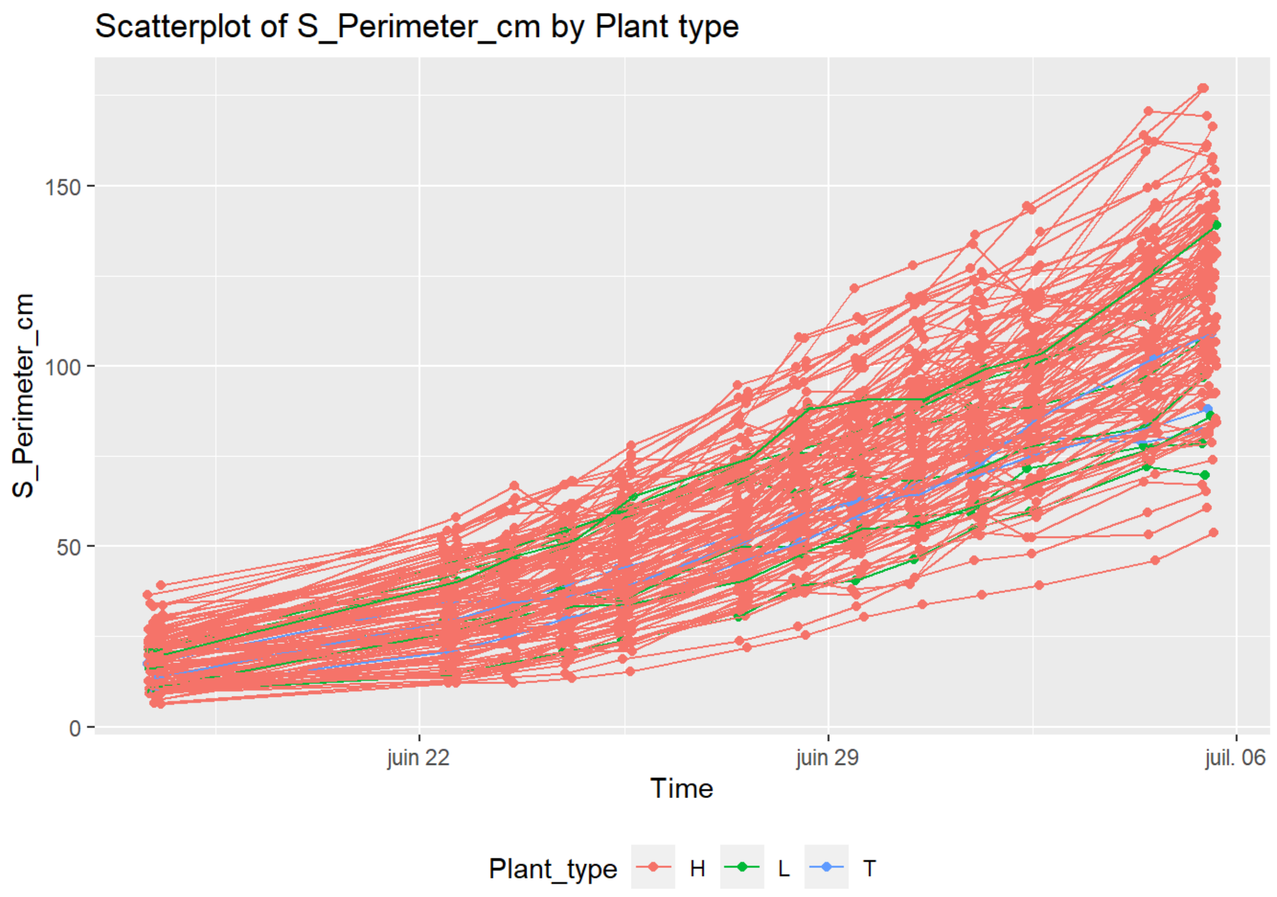
```
## `geom_smooth()` using formula = 'y ~ x'
```



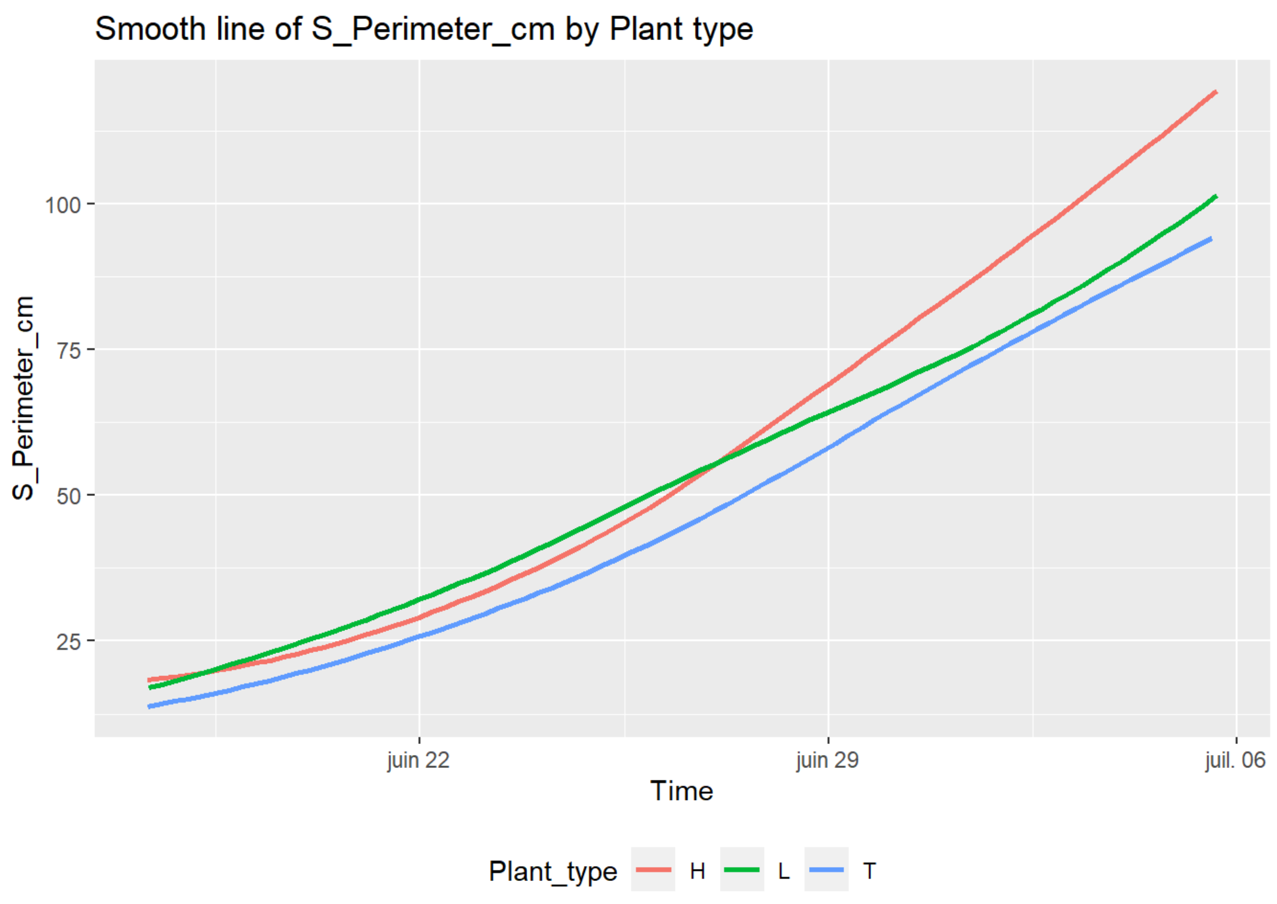


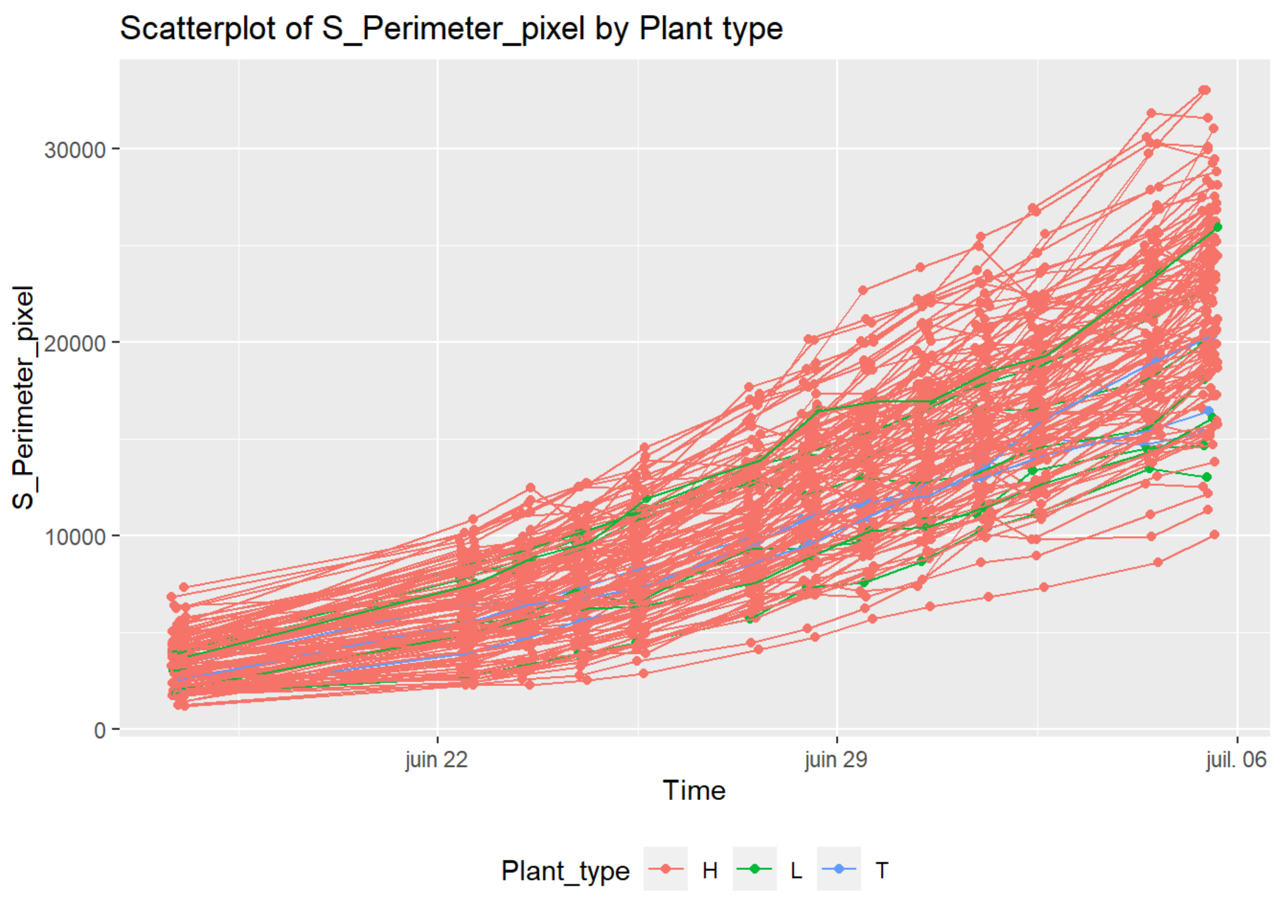
```
## `geom_smooth()` using formula = 'y ~ x'
```



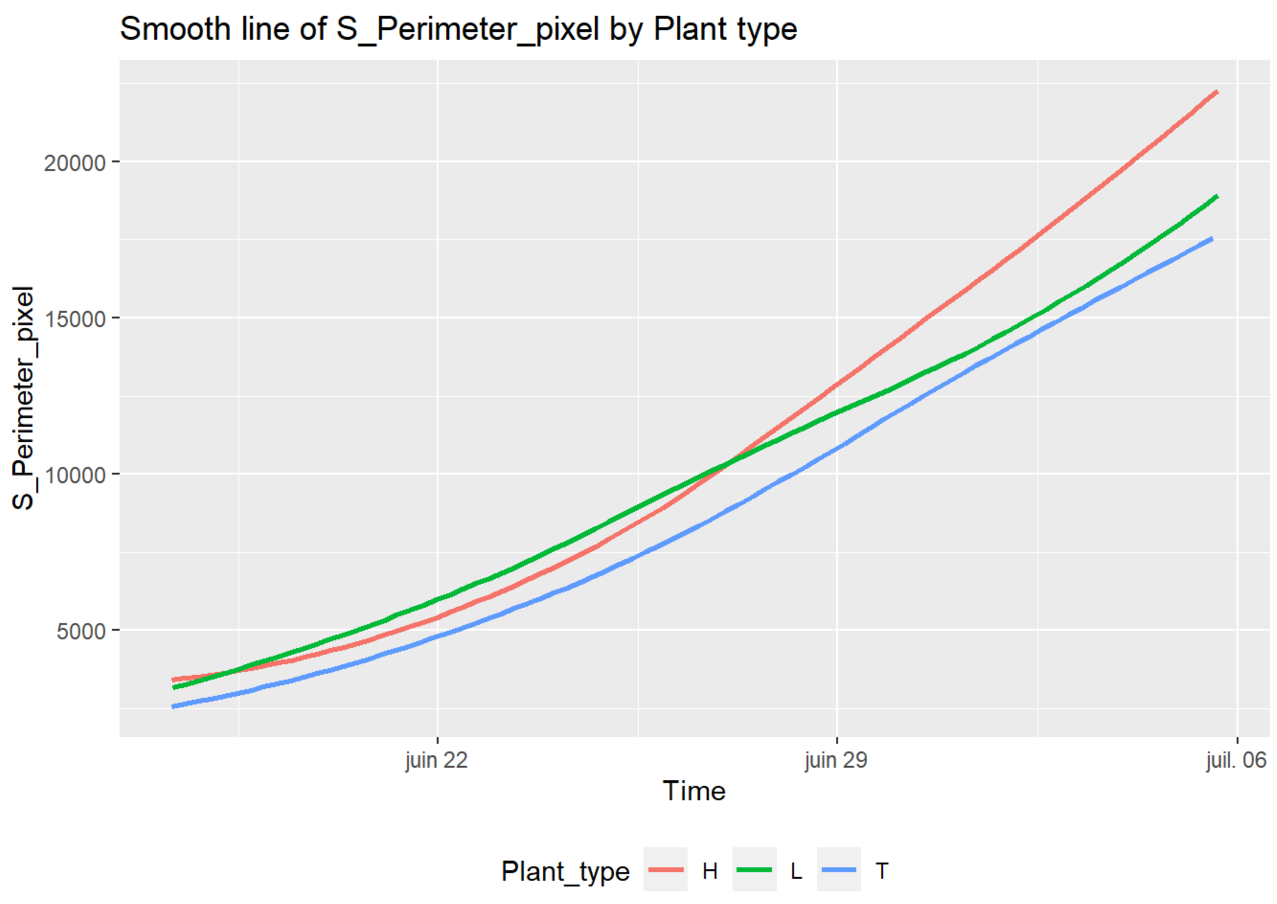


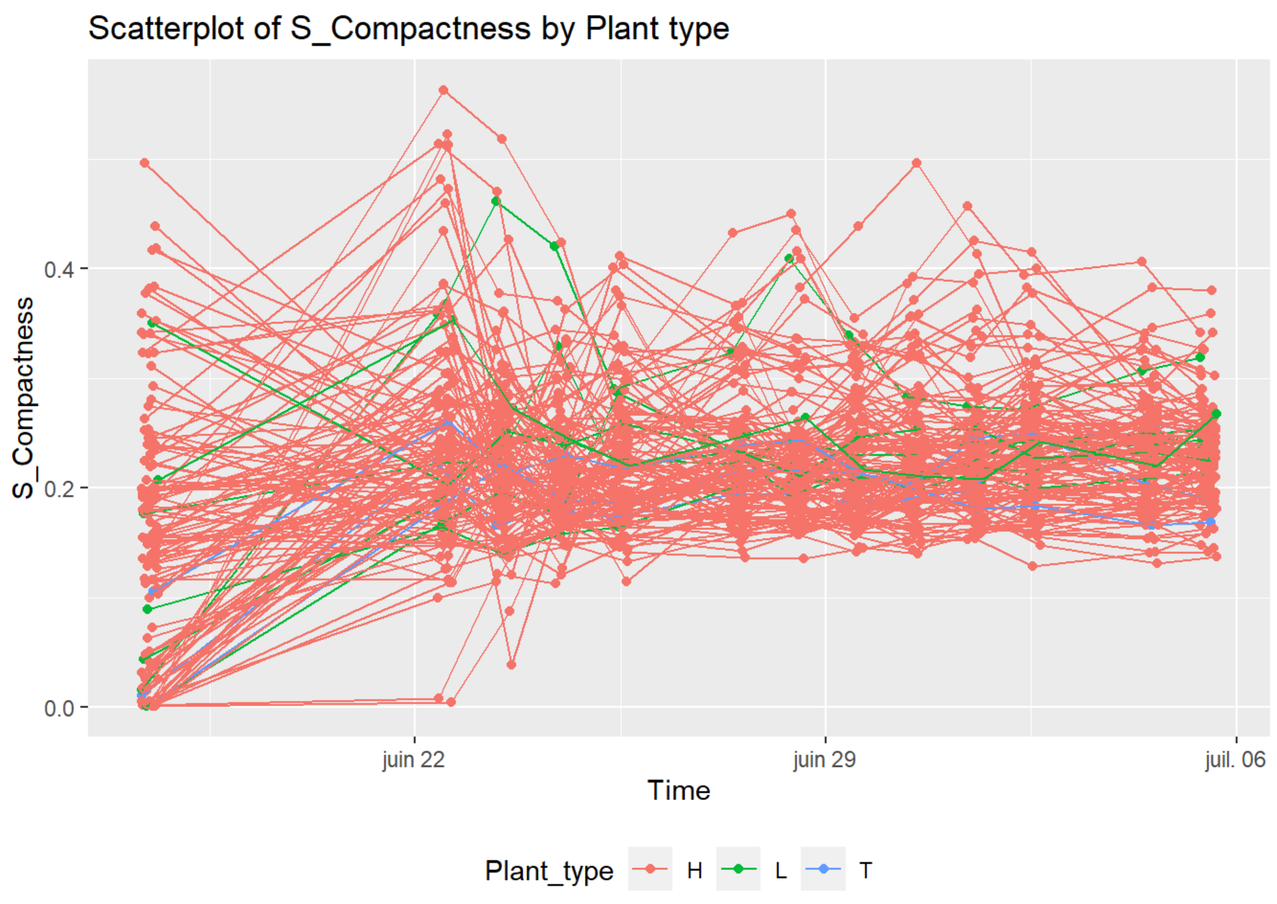
```
## `geom_smooth()` using formula = 'y ~ x'
```



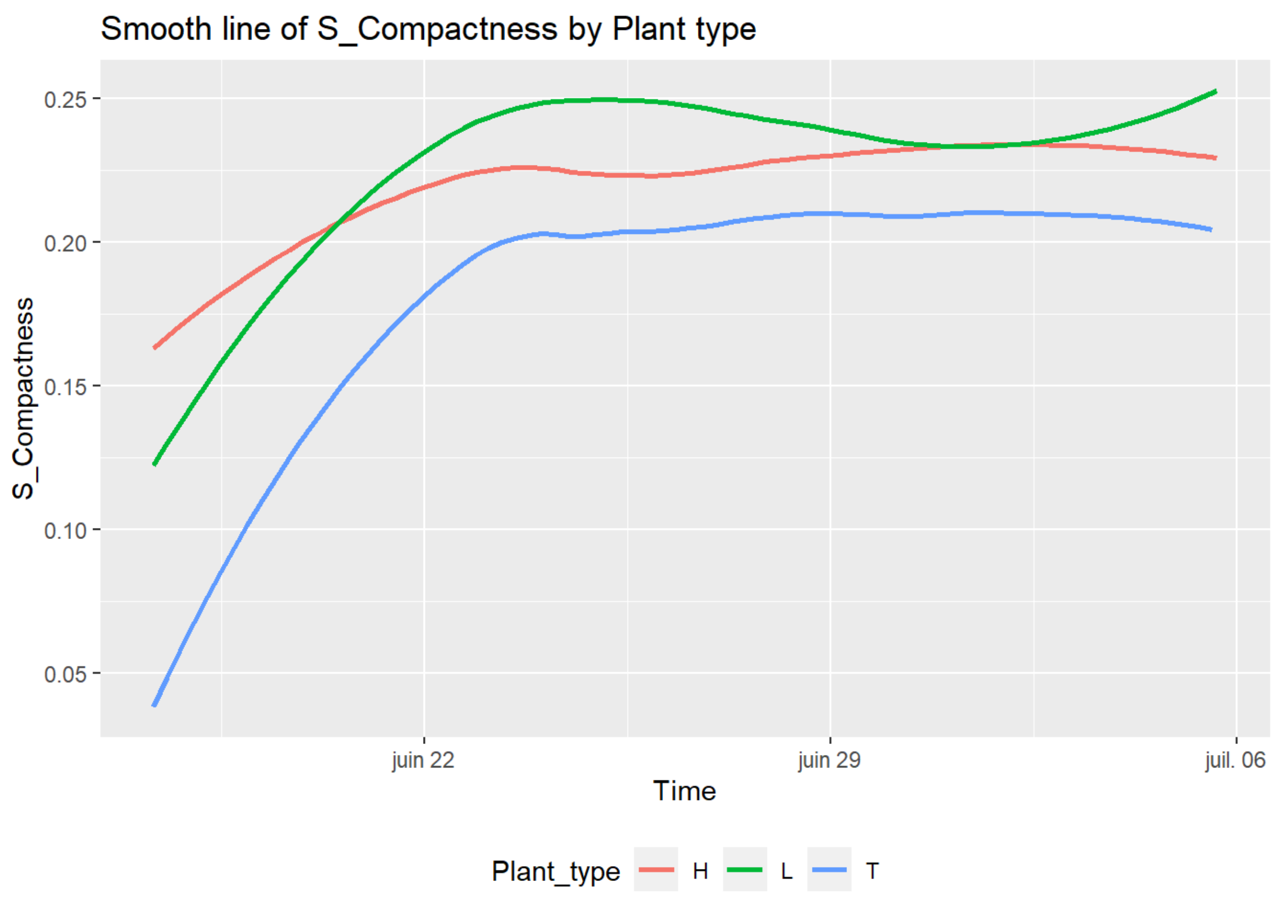


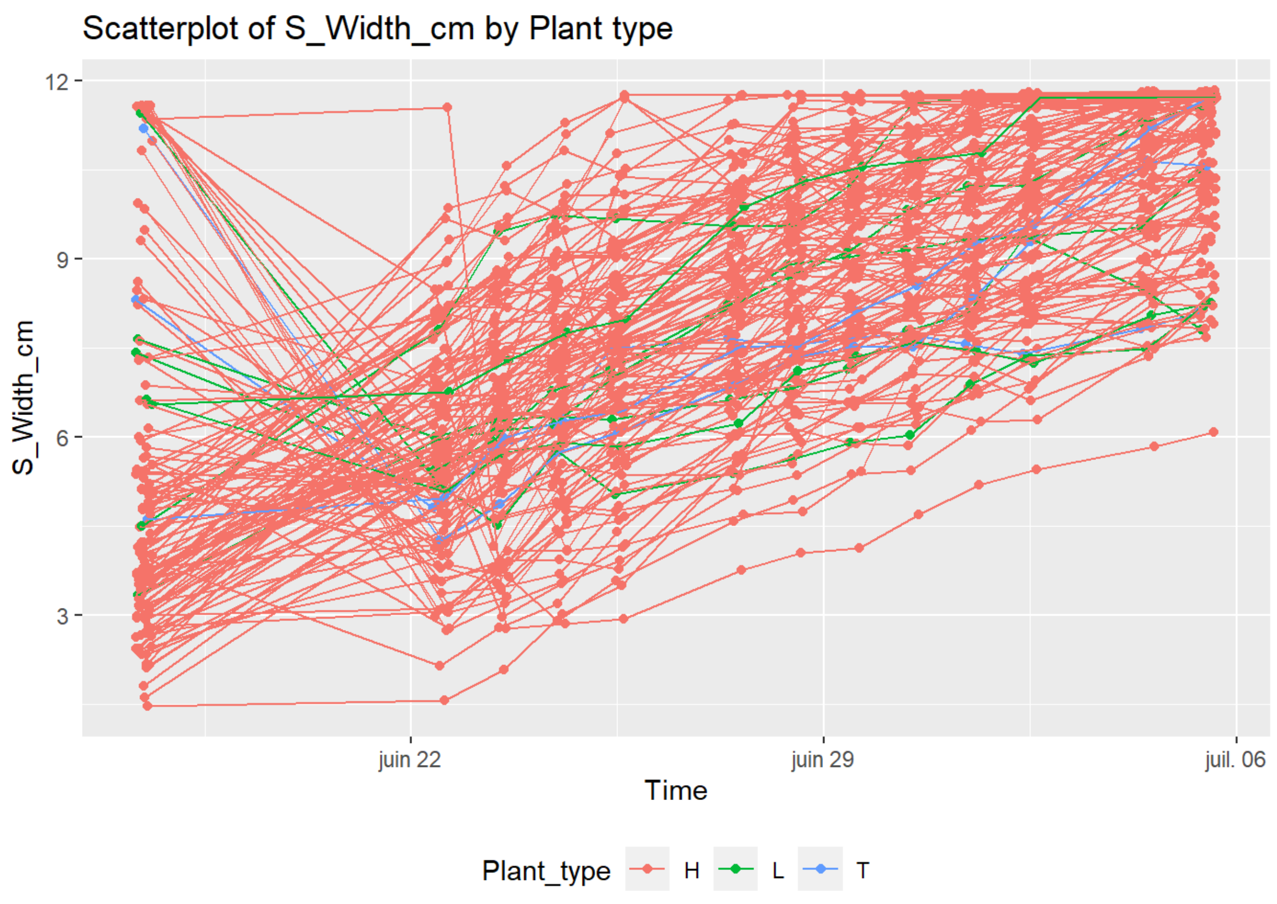
```
## `geom_smooth()` using formula = 'y ~ x'
```



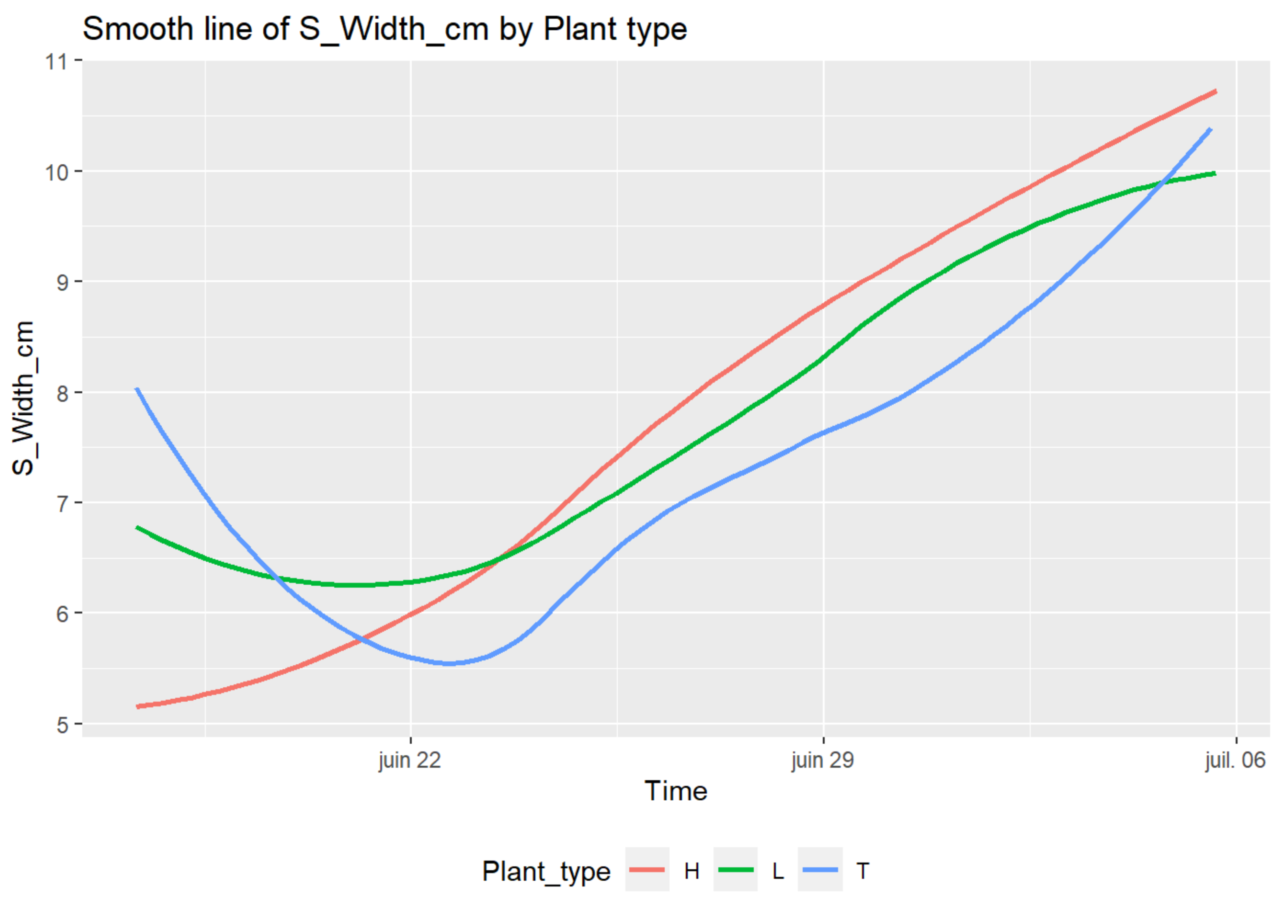


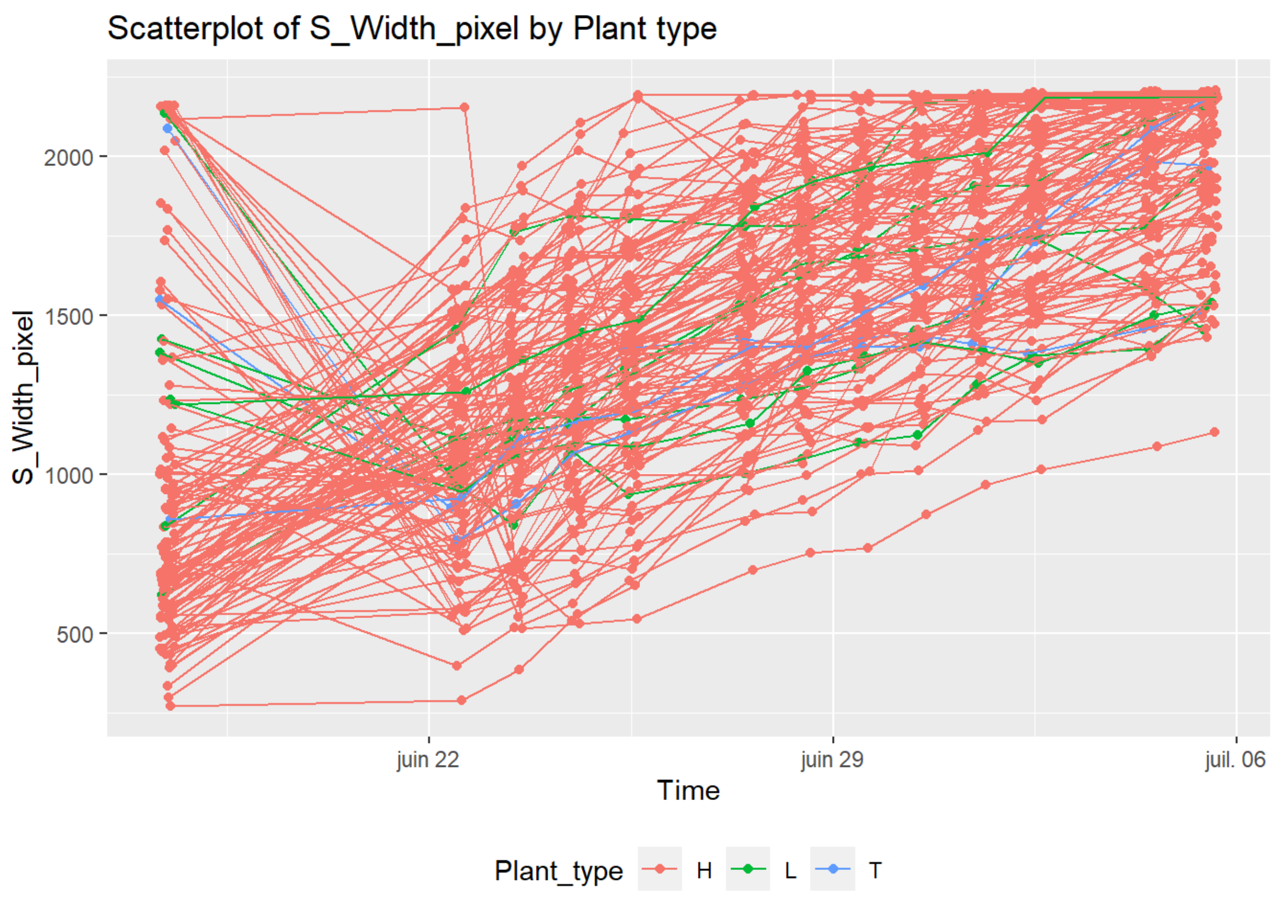
```
## `geom_smooth()` using formula = 'y ~ x'
```



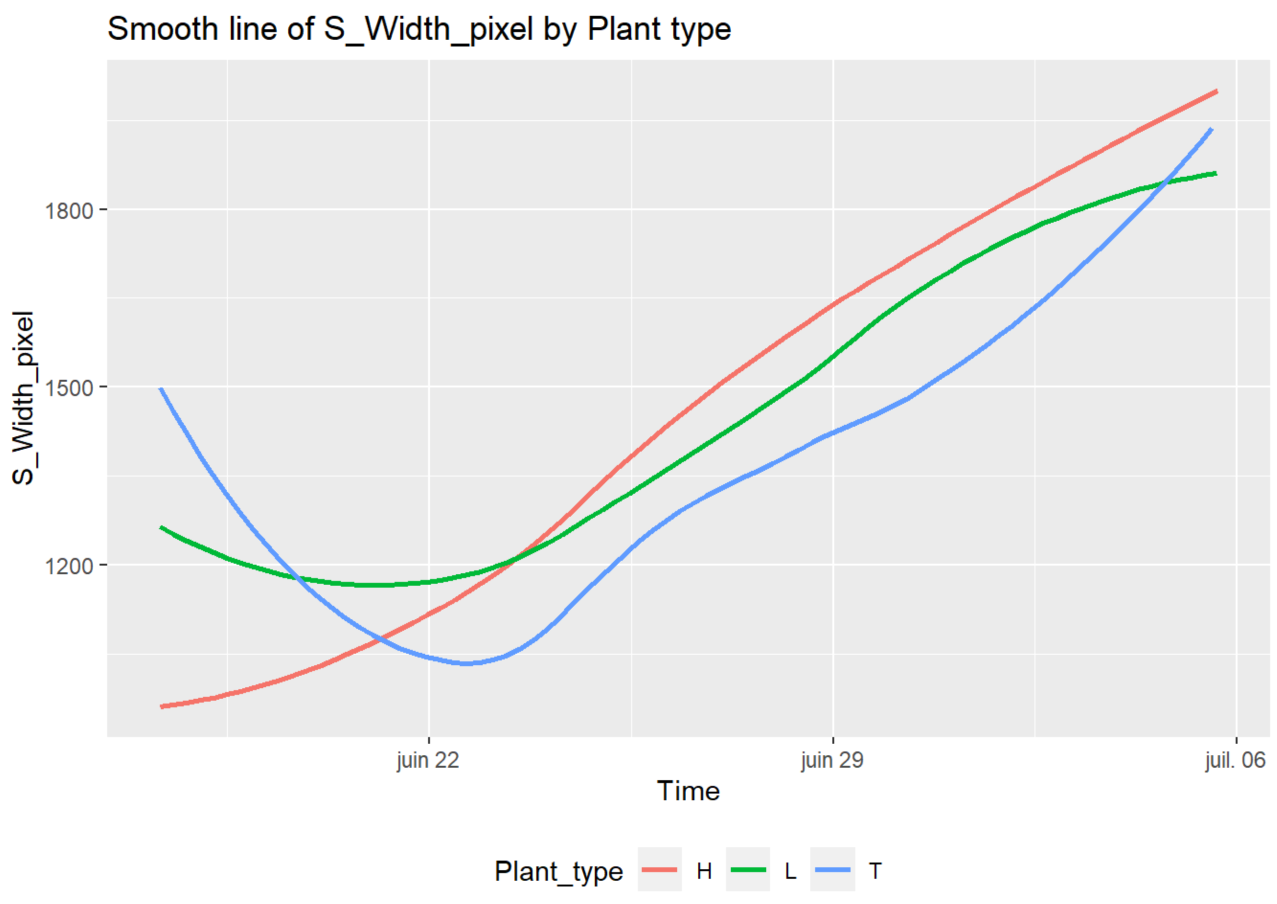


```
## `geom_smooth()` using formula = 'y ~ x'
```





```
## `geom_smooth()` using formula = 'y ~ x'
```



Scatter plots for all genotypes by water treatment

```
print(paste0("No data for", platform))
```

```
## [1] "No data forNaPPI"
```

## C. Exploration of the T_timeseries dataframe

```
print(paste0("No data for", platform))
```

```
## [1] "No data forNaPPI"
```