Packages importation

1. Data importation

A. Datasets structures

B. Data manipulation

2. Data template

A. Data template: plant_info

B. Data template: endpoint

C. Data template: timeseries

D. SPPU data templates

3. Export the data templates in .txt

SPPU Data Preparation

Elise

2024-06-09

Set the right working directory.

```
setwd("C:/Users/elise/Documents/Mémoire/Main/Data/Drive/SPPU")
```

Packages importation

1. Data importation

The first step in this data preparation process involves importing all the pertinent datasets listed in the Google Sheets “Variables template” document. Fist we find the files, then import them.

```
## [1] "env_multisensor.xlsx"    "ISA_EPPN2020_SPPU.xlsx"
## [3] "Rgb1_Morpho_Plant.xlsx"  "Rgb2_Morpho_Plant.xlsx"
## [5] "SPPU.Rmd"                "SPPU_Initial Code Draft"
```

We can extract the coordinates of each plant with the ISA_EPPN.xlsx dataset, using a made-up function “coordinates_isaTAB”.

```
# Get the coordinates
isaTAB <- read_excel("ISA_EPPN2020_SPPU.xlsx", sheet = "s_exp")
```

```
## New names:
## • `Unit` -> `Unit...9`
## • `Term Source REF` -> `Term Source REF...10`
## • `Term Accession Number` -> `Term Accession Number...11`
## • `Unit` -> `Unit...13`
## • `Term Source REF` -> `Term Source REF...14`
## • `Term Accession Number` -> `Term Accession Number...15`
## • `Unit` -> `Unit...22`
## • `Term Source REF` -> `Term Source REF...23`
## • `Term Accession Number` -> `Term Accession Number...24`
## • `Unit` -> `Unit...26`
## • `Term Source REF` -> `Term Source REF...27`
## • `Term Accession Number` -> `Term Accession Number...28`
```

```
coordinates <- coordinates_isaTAB(isaTAB)
```

A. Datasets structures

We can take a quick look at all the datasets.

- coordinates
- rgb1
- rgb2
- data_environment

```
head(coordinates)
```

```
##    Sample.Name nrow ncol rep
## 1 PS_Tray_4001    1    1   1
## 2 PS_Tray_4002    1    2   1
## 3 PS_Tray_4003    1    3   1
## 4 PS_Tray_4004    1    4   1
## 5 PS_Tray_4005    1    5   1
## 6 PS_Tray_4006    1    6   1
```

```
head(rgb1)
```

```
## # A tibble: 6 × 22
##   `Measuring Date`    `Measuring Time`    `Experiment ID` `Round Order`
##   <dttm>              <dttm>                        <dbl>         <dbl>
## 1 2020-02-17 22:02:34 2020-02-17 22:02:34              53             8
## 2 2020-02-17 22:03:25 2020-02-17 22:03:25              53             8
## 3 2020-02-17 22:05:16 2020-02-17 22:05:16              53             8
## 4 2020-02-17 22:06:08 2020-02-17 22:06:08              53             8
## 5 2020-02-17 22:16:54 2020-02-17 22:16:54              53             8
## 6 2020-02-17 22:17:46 2020-02-17 22:17:46              53             8
## # ℹ 18 more variables: `Tray ID` <chr>, `Tray Info` <chr>, `Plant ID` <chr>,
## #   Position <chr>, `Plant Name` <chr>, `Plant Info` <chr>, PID <chr>,
## #   Angle <dbl>, `Camera Position` <dbl>, AREA_PX <dbl>, AREA_MM <dbl>,
## #   PERIMETER_PX <dbl>, PERIMETER_MM <dbl>, COMPACTNESS <dbl>, WIDTH_PX <dbl>,
## #   WIDTH_MM <dbl>, HEIGHT_PX <dbl>, HEIGHT_MM <dbl>
```

```
head(rgb2)
```

```
## # A tibble: 6 × 23
##   `Measuring Date`    `Measuring Time`    `Experiment ID` `Round Order`
##   <dttm>              <dttm>                        <dbl>         <dbl>
## 1 2020-02-17 22:02:06 2020-02-17 22:02:06              53             8
## 2 2020-02-17 22:04:55 2020-02-17 22:04:55              53             8
## 3 2020-02-17 22:16:36 2020-02-17 22:16:36              53             8
## 4 2020-02-17 22:26:10 2020-02-17 22:26:10              53             8
## 5 2020-02-17 22:37:05 2020-02-17 22:37:05              53             8
## 6 2020-02-17 22:47:36 2020-02-17 22:47:36              53             8
## # ℹ 19 more variables: `Tray ID` <chr>, `Tray Info` <chr>, `Plant ID` <chr>,
## #   Position <chr>, `Plant Name` <chr>, `Plant Info` <chr>, PID <chr>,
## #   `Camera Position` <dbl>, AREA_PX <dbl>, AREA_MM <dbl>, PERIMETER_PX <dbl>,
## #   PERIMETER_MM <dbl>, COMPACTNESS <dbl>, ROUNDNESS <dbl>, ROUNDNESS2 <dbl>,
## #   ISOTROPY <dbl>, ECCENTRICITY <dbl>, RMS <dbl>, SOL <dbl>
```

```
head(data_environment)
```

```
## # A tibble: 6 × 14
##   Date                T_Actual1 Rh_Actual1 T_Actual2 Rh_Actual2 CO2_Actual
##   <dttm>                  <dbl>      <dbl>     <dbl>      <dbl>      <dbl>
## 1 2020-03-02 00:00:00     45097         38     45066         39        300
## 2 2020-03-02 00:10:00     45097         38     45066         39        300
## 3 2020-03-02 00:20:00     45097         38     45066         39        300
## 4 2020-03-02 00:30:00     45127         38     45066         39        300
## 5 2020-03-02 00:40:00     45097         38     45066         39        300
## 6 2020-03-02 00:50:00     45127         38     45066         39        300
## # ℹ 8 more variables: Light_Intensity_Outside <dbl>,
## #   Light_Intensity_Inner <dbl>, L1 <dbl>, L2 <dbl>, L3 <dbl>, L4 <dbl>,
## #   L5 <dbl>, L6 <dbl>
```

B. Data manipulation

This next step standardizes diverse datasets by renaming variables for consistency, converting data into appropriate units, adding necessary columns, and merging the datasets.

```
################################################################################
# COORDINATES
################################################################################
# Unit.ID
coordinates$Unit.ID <- seq_len(nrow(coordinates))
# Reference for Sample.Name et Unit.ID
reference <- coordinates[, c("Sample.Name", "Unit.ID")]
## We can then copy dataset2$Unit.ID <- reference$Unit.ID[match(dataset2$Sample.Name, r
eference$Sample.Name)]

################################################################################
# rgb 1
################################################################################
# Time, Date and Timestamp
rgb1$Timestamp <- rgb1$`Measuring Time`

rgb1$Date <- sapply(strsplit(as.character(rgb1$Timestamp), split = " "), '[', 1)
rgb1$Time <- sapply(strsplit(as.character(rgb1$Timestamp), split = " "), '[', 2)

# Name of the platform
rgb1$Platform <- "SPPU"

# Unit.ID
rgb1$Unit.ID <- reference$Unit.ID[match(rgb1$`Plant ID`, reference$Sample.Name)]

# Rename the columns for the template
rgb1 <- rename(rgb1,
                           S_Area_pixel = AREA_PX,
                           S_Area_mm_squared = AREA_MM,
                           S_Perimeter_pixel = PERIMETER_PX,
                           S_Perimeter_mm = PERIMETER_MM,
                           S_Compactness = COMPACTNESS,
                           S_Width_pixel = WIDTH_PX,
                           S_Width_mm = WIDTH_MM,
                           S_Height_pixel = HEIGHT_PX,
                           S_Height_mm = HEIGHT_MM
                           )


################################################################################
# rgb2
################################################################################
# Time, Date and Timestamp
rgb2$Timestamp <- rgb2$`Measuring Time`

rgb2$Date <- sapply(strsplit(as.character(rgb2$Timestamp), split = " "), '[', 1)
rgb2$Time <- sapply(strsplit(as.character(rgb2$Timestamp), split = " "), '[', 2)

# Name of the platform
rgb2$Platform <- "SPPU"

# Unit.ID
rgb2$Unit.ID <- reference$Unit.ID[match(rgb2$`Plant ID`, reference$Sample.Name)]

# Rename the columns for the template
```

```
rgb2 <- rename(rgb2,
                           T_Area_pixel = AREA_PX,
                           T_Area_mm_squared = AREA_MM,
                           T_Perimeter_pixel = PERIMETER_PX,
                           T_Perimeter_mm = PERIMETER_MM,
                           T_Compactness = COMPACTNESS,
                           T_Roundness = ROUNDNESS,
                           T_Roundness2 = ROUNDNESS2,
                           T_Isotropy = ISOTROPY,
                           T_Eccentricity = ECCENTRICITY,
                           )
rgb2$Rms <- rgb2$RMS
rgb2$Sol <- rgb2$SOL
```

Camera angles

For the SPPU platform, the variables in the data_imaging are measured form 2 different camera angles. It is neccessary to consolidate theses measurements into a single value for each variable for the data analysis steps. In this code block, it is done by either taking the maximum of the 2 values or by taking the mean of the 2 values.

Depending on the variable, the mean or the maximum is taken. The result is stocked in the dataset rgb1_2 and rgb2_2.

Variable	Mean or maximum
Height	Mean
Area	Maximum
Perimeter	Maximum
Width	Maximum
Convex hull	Maximum
Solidity	Maximum
Compactness	Maximum

```
# Data frame containing the results
rgb1_2 <- data.frame()

for (i in seq(1, nrow(rgb1), by = 2)) {
  if (i + 1 <= nrow(rgb1)) {
    row1 <- rgb1[i, ]
    row2 <- rgb1[i + 1, ]

    # Compute the mean or the maximum of the 2 camera angles values
    mean_and_max_row <- data.frame(
      Date = row2$Date, # We keep the important columns
      Time = row2$Time, # We keep the important columns
      Unit.ID = row2$Unit.ID, # We keep the important columns
      Timestamp = row2$Timestamp, # We keep the important columns
      Platform = row2$Platform, # We keep the important columns

      S_Area_mm_squared = max(c(as.numeric(row1$S_Area_mm_squared), as.numeric(row2$S_A
rea_mm_squared))),
      S_Area_pixel = max(c(as.numeric(row1$S_Area_pixel), as.numeric(row2$S_Area_pixe
l))),

      S_Perimeter_mm = max(c(as.numeric(row1$S_Perimeter_mm), as.numeric(row2$S_Perimet
er_mm))),
      S_Perimeter_pixel = max(c(as.numeric(row1$S_Perimeter_pixel), as.numeric(row2$S_P
erimeter_pixel))),


      S_Compactness = max(c(as.numeric(row1$S_Compactness), as.numeric(row2$S_Compactne
ss))),

      S_Width_mm = max(c(as.numeric(row1$S_Width_mm), as.numeric(row2$S_Width_mm))),
      S_Width_pixel = max(c(as.numeric(row1$S_Width_pixel), as.numeric(row2$S_Width_pix
el))),


      S_Height_mm = mean(c(as.numeric(row1$S_Height_mm), as.numeric(row2$S_Height_m
m))),
      S_Height_pixel = mean(c(as.numeric(row1$S_Height_pixel), as.numeric(row2$S_Height
_pixel)))
    )

    # Ajouter les résultats au dataframe final
    rgb1_2 <- rbind(rgb1_2, mean_and_max_row)
  }
}

# Afficher le dataframe final
head(rgb1_2)
```

```
##         Date     Time Unit.ID           Timestamp Platform S_Area_mm_squared
## 1 2020-02-17 22:03:25       1 2020-02-17 22:03:25     SPPU        121.947012
## 2 2020-02-17 22:06:08       2 2020-02-17 22:06:08     SPPU        210.572309
## 3 2020-02-17 22:17:46       3 2020-02-17 22:17:46     SPPU        364.619823
## 4 2020-02-17 22:27:03       4 2020-02-17 22:27:03     SPPU         57.571551
## 5 2020-02-17 22:38:22       5 2020-02-17 22:38:22     SPPU          7.676207
## 6 2020-02-17 22:48:56       6 2020-02-17 22:48:56     SPPU         16.224710
##   S_Area_pixel S_Perimeter_mm S_Perimeter_pixel S_Compactness S_Width_mm
## 1          699       77.49656         185.53911     0.8228228  11.430570
## 2         1207       91.68040         219.49747     0.8852972  13.063508
## 3         2090      147.11142         352.20815     0.7856250  18.778793
## 4          330       54.30900         130.02439     0.8847185  12.655274
## 5           44       13.66990          32.72792     0.7045455  16.329385
## 6           93       18.19275          43.55635     0.9649123   3.265877
##   S_Width_pixel S_Height_mm S_Height_pixel
## 1            28   33.119658           77.5
## 2            32   58.333333          136.5
## 3            46   62.606838          146.5
## 4            31   21.153846           49.5
## 5            40    1.068376            2.5
## 6             8    5.769231           13.5
```

```
# Data frame containing the results
rgb2_2 <- data.frame()

for (i in seq(1, nrow(rgb2), by = 2)) {
  if (i + 1 <= nrow(rgb2)) {
    row1 <- rgb2[i, ]
    row2 <- rgb2[i + 1, ]

    # Compute the mean or the maximum of the 2 camera angles values
    mean_and_max_row <- data.frame(
      Date = row2$Date, # We keep the important columns
      Time = row2$Time, # We keep the important columns
      Unit.ID = row2$Unit.ID, # We keep the important columns
      Timestamp = row2$Timestamp, # We keep the important columns
      Platform = row2$Platform, # We keep the important columns

      T_Area_mm_squared = max(c(as.numeric(row1$T_Area_mm_squared), as.numeric(row2$T_A
rea_mm_squared))),
      T_Area_pixel = max(c(as.numeric(row1$T_Area_pixel), as.numeric(row2$T_Area_pixe
l))),

      T_Perimeter_mm = max(c(as.numeric(row1$T_Perimeter_mm), as.numeric(row2$T_Perimet
er_mm))),
      T_Perimeter_pixel = max(c(as.numeric(row1$T_Perimeter_pixel), as.numeric(row2$T_P
erimeter_pixel))),


      T_Compactness = max(c(as.numeric(row1$T_Compactness), as.numeric(row2$T_Compactne
ss))),

      T_Roundness = max(c(as.numeric(row1$T_Roundness), as.numeric(row2$T_Roundness))),
      T_Roundness2 = max(c(as.numeric(row1$T_Roundness2), as.numeric(row2$T_Roundness
2))),


      T_Isotropy = mean(c(as.numeric(row1$T_Isotropy), as.numeric(row2$T_Isotropy))),
      T_Eccentricity = mean(c(as.numeric(row1$T_Eccentricity), as.numeric(row2$T_Eccent
ricity))),

      Rms = mean(c(as.numeric(row1$Rms), as.numeric(row2$Rms))),
      Sol = mean(c(as.numeric(row1$Sol), as.numeric(row2$Sol)))
    )

    # Ajouter les résultats au dataframe final
    rgb2_2 <- rbind(rgb2_2, mean_and_max_row)
  }
}

# Afficher le dataframe final
head(rgb2_2)
```

```
##         Date     Time Unit.ID           Timestamp Platform T_Area_mm_squared
## 1 2020-02-17 22:04:55       2 2020-02-17 22:04:55     SPPU        117.817448
## 2 2020-02-17 22:26:10       4 2020-02-17 22:26:10     SPPU         61.219355
## 3 2020-02-17 22:47:36       6 2020-02-17 22:47:36     SPPU               NaN
## 4 2020-02-17 23:09:45       8 2020-02-17 23:09:45     SPPU         90.138381
## 5 2020-02-17 23:29:53      10 2020-02-17 23:29:53     SPPU          8.667998
## 6 2020-02-17 23:48:49      12 2020-02-17 23:48:49     SPPU               NaN
##   T_Area_pixel T_Perimeter_mm T_Perimeter_pixel T_Compactness T_Roundness
## 1         1400       44.03118         151.78175     0.9936125   0.7636578
## 2          688       47.66255         159.78175     0.8566610   0.3386445
## 3          NaN            NaN               NaN           NaN         NaN
## 4         1013       48.85575         163.78175     0.8924528   0.4970396
## 5          103       10.62544          36.62742     1.0098039   1.0349962
## 6          NaN            NaN               NaN           NaN         NaN
##   T_Roundness2 T_Isotropy T_Eccentricity       Rms       Sol
## 1    0.8533225          0      1.0623059 0.9169442 2.8652674
## 2    0.5167950          0      1.9424430 0.9798271 4.3768998
## 3          NaN        NaN            NaN       NaN       NaN
## 4    0.6412938          0      1.2360120 0.9576695 2.7342152
## 5    1.0687747        NaN      0.3834559 0.6976545 0.1449551
## 6          NaN        NaN            NaN       NaN       NaN
```

Unit conversions

The data template is only in cm, cm² and g. This step converts the data in the right units.

For the SPPU platform, 6 variables are in mm.

```
rgb1_2$S_Height_cm <- 0.01 * rgb1_2$S_Height_mm
rgb1_2$S_Area_cmsquared <- 0.01 * 0.01 * rgb1_2$S_Area_mm_squared
rgb1_2$S_Perimeter_cm <- 0.01 * rgb1_2$S_Perimeter_mm
rgb1_2$S_Width_cm <- 0.01 * rgb1_2$S_Width_mm

rgb2_2$T_Area_cmsquared <- 0.01 * 0.01 * rgb2_2$T_Area_mm_squared
rgb2_2$T_Perimeter_cm <- 0.01 * rgb2_2$T_Perimeter_mm
```

2. Data template

A. Data template: plant_info

This dataset contains information about the plant: Unit.ID, genotype, replication, row and column location in the greenhouse, and soil treatment.

B. Data template: endpoint

This datasets contains information of the end of the experiment (variables at harvest). It is then linked by the Unit.ID to the plant_info data template.

C. Data template: timeseries

This section in divided in three data templates:

- timeseries

- S_timeseries (variables computed from sideview imaging or image processing)
- T_timeseries (variables computed from topview imaging or image processing)

The time interval between data timestamps varies in each platform. They are then linked by the Unit.ID to the plant_info data template.

D. SPPU data templates

- plant_info
- endpoint
- timeseries
- S_timeseries
- T_timeseries

```
##   Unit.ID Genotype Soil Replication Row Column Platform
## 1       1  EPPN1_L   S2           1   1      1     SPPU
## 2       2   EPPN_T   S2           1   1      2     SPPU
## 3       3  EPPN2_H   S2           1   1      3     SPPU
## 4       4  EPPN3_H   S2           1   1      4     SPPU
## 5       5  EPPN4_H   S2           1   1      5     SPPU
## 6       6  EPPN3_L   S2           1   1      6     SPPU
```

```
##   Unit.ID Time Date Timestamp DW_shoot_g FW_shoot_g DW_root_g FW_root_g
## 1    <NA>   NA   NA        NA         NA         NA        NA        NA
##   Leaf_number Plant_height_cm DW_plant_g Root_length_cm Root_number Root_angle
## 1          NA              NA         NA             NA          NA         NA
##   Total_wu DW_seed_g FW_seed_g Leaf_area_cmsquared Genotype Soil Replication
## 1       NA        NA        NA                  NA     <NA> <NA>        <NA>
##    Row Column Platform
## 1 <NA>   <NA>     <NA>
```

```
##   Unit.ID Time Date Timestamp Manual_Plant_height_cm Leaf_number Wue
## 1    <NA>   NA   NA        NA                     NA          NA  NA
##   Plant_biomass Ligulated_leaf_number Plant_emergence Plant_transpiration
## 1            NA                    NA              NA                  NA
##   Daily_wu Soil_water_potential Genotype Soil Replication  Row Column Platform
## 1       NA                   NA     <NA> <NA>        <NA> <NA>   <NA>     <NA>
```

```
##   Unit.ID           Timestamp       Date     Time S_Height_cm S_Height_pixel
## 1       1 2020-02-17 22:03:25 2020-02-17 22:03:25  0.33119658           77.5
## 2       2 2020-02-17 22:06:08 2020-02-17 22:06:08  0.58333333          136.5
## 3       3 2020-02-17 22:17:46 2020-02-17 22:17:46  0.62606838          146.5
## 4       4 2020-02-17 22:27:03 2020-02-17 22:27:03  0.21153846           49.5
## 5       5 2020-02-17 22:38:22 2020-02-17 22:38:22  0.01068376            2.5
## 6       6 2020-02-17 22:48:56 2020-02-17 22:48:56  0.05769231           13.5
##   S_Area_cmsquared S_Area_pixel S_Perimeter_cm S_Perimeter_pixel
## 1     0.0121947012          699      0.7749656         185.53911
## 2     0.0210572309         1207      0.9168040         219.49747
## 3     0.0364619823         2090      1.4711142         352.20815
## 4     0.0057571551          330      0.5430900         130.02439
## 5     0.0007676207           44      0.1366990          32.72792
## 6     0.0016224710           93      0.1819275          43.55635
##   S_Convex_hull_area_cmsquared S_Solidity S_Compactness S_Width_cm
## 1                           NA         NA     0.8228228 0.11430570
## 2                           NA         NA     0.8852972 0.13063508
## 3                           NA         NA     0.7856250 0.18778793
## 4                           NA         NA     0.8847185 0.12655274
## 5                           NA         NA     0.7045455 0.16329385
## 6                           NA         NA     0.9649123 0.03265877
##   S_Width_pixel S_Leaf_area_cmsquared Genotype Soil Replication Row Column
## 1            28                    NA  EPPN1_L   S2           1   1      1
## 2            32                    NA   EPPN_T   S2           1   1      2
## 3            46                    NA  EPPN2_H   S2           1   1      3
## 4            31                    NA  EPPN3_H   S2           1   1      4
## 5            40                    NA  EPPN4_H   S2           1   1      5
## 6             8                    NA  EPPN3_L   S2           1   1      6
##   Platform
## 1     SPPU
## 2     SPPU
## 3     SPPU
## 4     SPPU
## 5     SPPU
## 6     SPPU
```

```
##   Unit.ID           Timestamp     Time       Date T_Area_cm_squared
## 1       2 2020-02-17 22:04:55 22:04:55 2020-02-17      0.0117817448
## 2       4 2020-02-17 22:26:10 22:26:10 2020-02-17      0.0061219355
## 3       6 2020-02-17 22:47:36 22:47:36 2020-02-17               NaN
## 4       8 2020-02-17 23:09:45 23:09:45 2020-02-17      0.0090138381
## 5      10 2020-02-17 23:29:53 23:29:53 2020-02-17      0.0008667998
## 6      12 2020-02-17 23:48:49 23:48:49 2020-02-17               NaN
##   T_Area_pixel T_Perimeter_cm T_Perimeter_pixel T_Convex_hull_area_cmsquared
## 1         1400      0.4403118         151.78175                           NA
## 2          688      0.4766255         159.78175                           NA
## 3          NaN            NaN               NaN                           NA
## 4         1013      0.4885575         163.78175                           NA
## 5          103      0.1062544          36.62742                           NA
## 6          NaN            NaN               NaN                           NA
##   T_Solidity T_Compactness T_Roundness T_Roundness2 T_Isotropy T_Eccentricity
## 1         NA     0.9936125   0.7636578    0.8533225          0      1.0623059
## 2         NA     0.8566610   0.3386445    0.5167950          0      1.9424430
## 3         NA           NaN         NaN          NaN        NaN            NaN
## 4         NA     0.8924528   0.4970396    0.6412938          0      1.2360120
## 5         NA     1.0098039   1.0349962    1.0687747        NaN      0.3834559
## 6         NA           NaN         NaN          NaN        NaN            NaN
##       T_Rms     T_Sol Genotype Soil Replication Row Column Platform
## 1 0.9169442 2.8652674   EPPN_T   S2           1   1      2     SPPU
## 2 0.9798271 4.3768998  EPPN3_H   S2           1   1      4     SPPU
## 3       NaN       NaN  EPPN3_L   S2           1   1      6     SPPU
## 4 0.9576695 2.7342152  EPPN2_L   S2           1   1      8     SPPU
## 5 0.6976545 0.1449551  EPPN3_H   S1           1   1     10     SPPU
## 6       NaN       NaN  EPPN1_H   S1           1   1     12     SPPU
```

```
# Remplace characters "NaN" by NA values

S_timeseries <- S_timeseries %>%
  mutate_at(vars(4:ncol(S_timeseries)), ~ ifelse(. == "NaN", NA, .))
T_timeseries <- T_timeseries %>%
  mutate_at(vars(5:ncol(T_timeseries)), ~ ifelse(. == "NaN", NA, .))
```

3. Export the data templates in .txt

Stock the new data sets in a new folder.

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

write.table(plant_info, file = "plant_info.txt", sep = "\t", row.names = FALSE, quote = 
FALSE)
write.table(endpoint, file = "endpoint.txt", sep = "\t", row.names = FALSE, quote = FAL
SE)
write.table(timeseries, file = "timeseries.txt", sep = "\t", row.names = FALSE, quote = 
FALSE)
write.table(S_timeseries, file = "S_timeseries.txt", sep = "\t", row.names = FALSE, quo
te = FALSE)
write.table(T_timeseries, file = "T_timeseries.txt", sep = "\t", row.names = FALSE, quo
te = FALSE)
```