Weekly Reporting

agrotech live | wigglelabs

2025-04-09

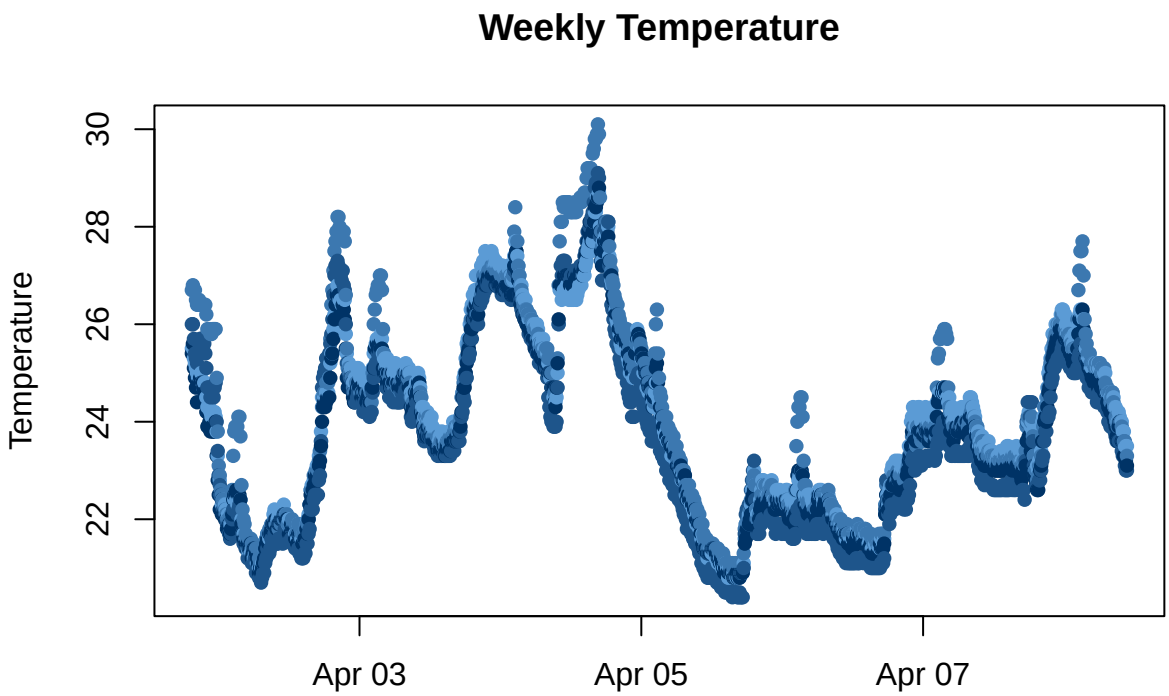
Abstract

In an ongoing effort to achieve food sovereignty, Wiggle Labs has developed Agrotech Live to monitor the soil health of different plants and crops. This tool collects four data points; Temperature, Moisture, Light and Conductivity from sensors placed near a subject. The training data for this program are the ideal conditions for the subject, and the performance of the experiment is based on how close the collected sensor (testing) data is to the input care training data. Input data is identical in structure the testing data (collected during a session), except it represents only perfect conditions for the subject. A score is generated (along with other statistical results) periodically to communicate how well the experiment is performing.

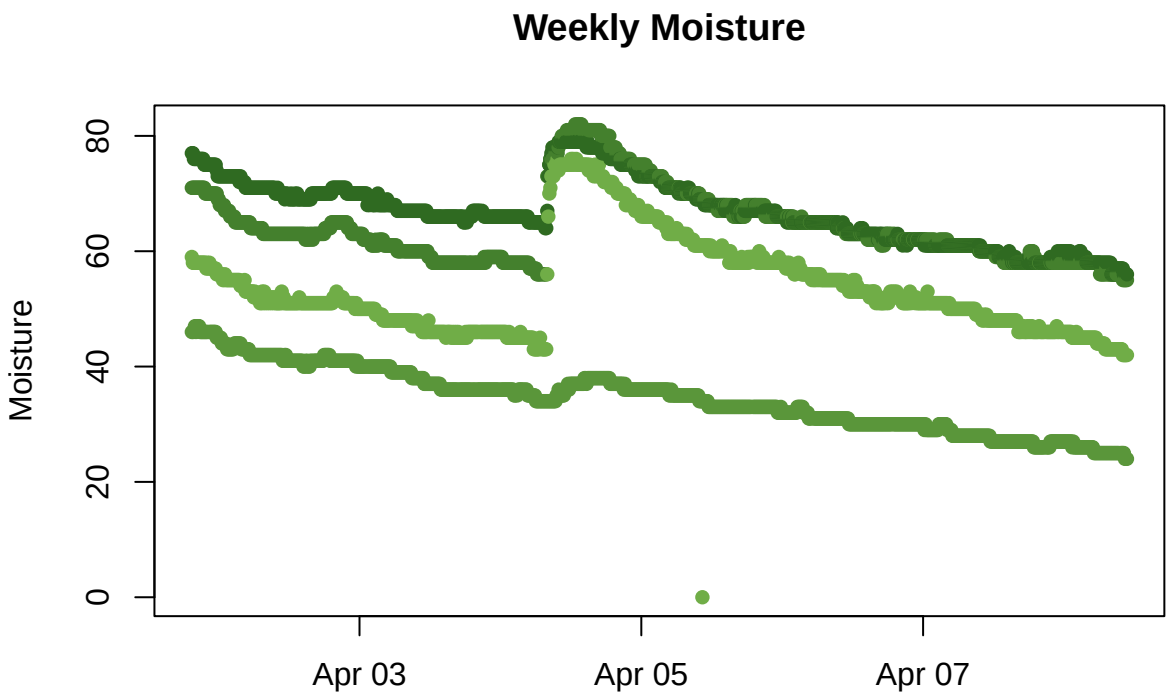
This report covers the past seven days of session data. If there are missing records, then the closest recorded days will be included.

Time Series

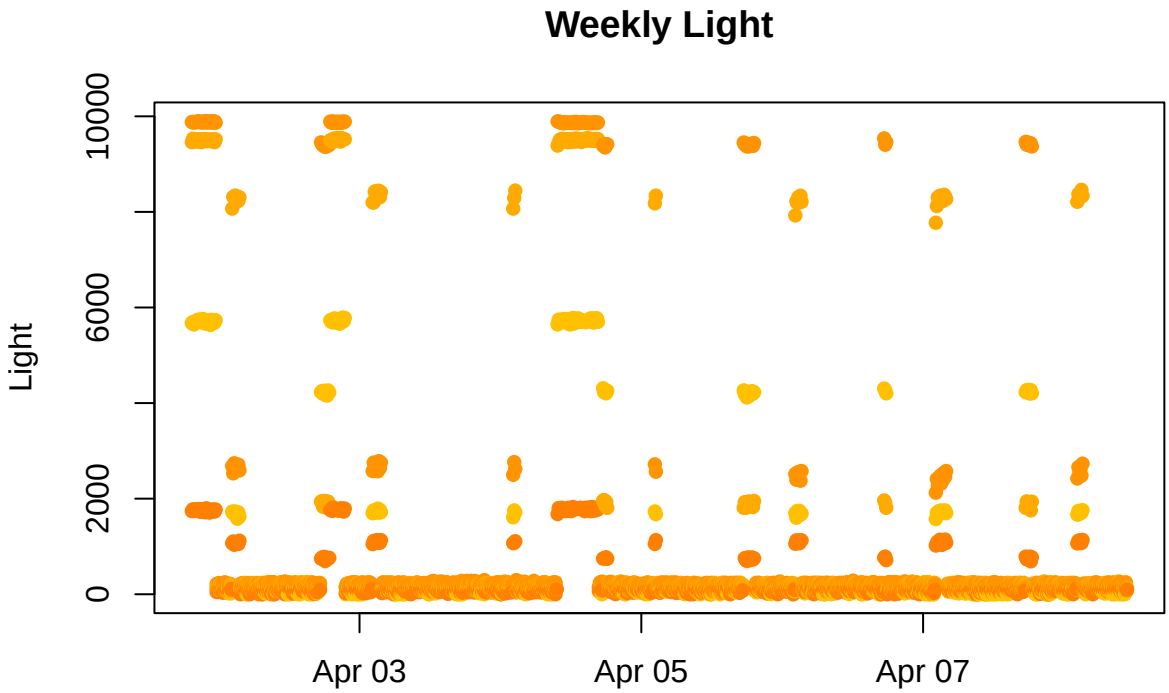
This section examines the relationships between sensor variables (temperature, moisture, light, and conductivity) over the past week. Pairwise scatter plots illustrate how these variables interact, while color coding emphasizes key patterns and potential correlations.



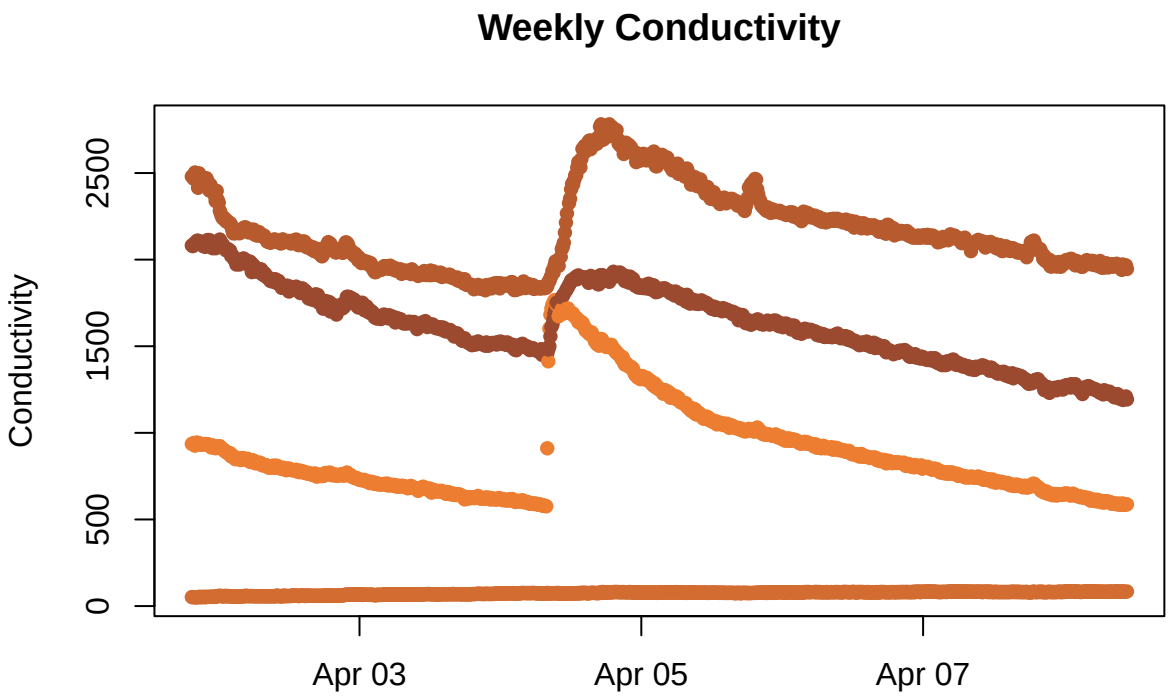
etra nibh dictum ut. Vivamus laoreet, ante in commodo aliquam, tellus sem rutrum elit, eget ullamcorper ju



etra nibh dictum ut. Vivamus laoreet, ante in commodo aliquam, tellus sem rutrum elit, eget ullamcorper ju



etra nibh dictum ut. Vivamus laoreet, ante in commodo aliquam, tellus sem rutrum elit, eget ullamcorper ju



etra nibh dictum ut. Vivamus laoreet, ante in commodo aliquam, tellus sem rutrum elit, eget ullamcorper ju

Matrices
Correlation Matrices

```
  Temperature <- c(agt_data$Temperature)
  Moisture <- c(agt_data$Moisture)
  Light <- c(agt_data$Light)
  Conductivity <- c(agt_data$Conductivity)

  matrix_data <- data.frame(Temperature, Moisture, Light, Conductivity)
  print("Covariance Matrix")
```

```
## [1] "Covariance Matrix"
```

```
  cov_matrix <- cov(matrix_data)
  cov_matrix
```

```
##                Temperature      Moisture       Light Conductivity
## Temperature      3.6528709    -0.7652509    1916.922    -182.4023
## Moisture        -0.7652509   206.6115942    2040.052   10563.0335
## Light         1916.9223345  2040.0521948 5936029.397   34173.5594
## Conductivity -182.4023115 10563.0335350   34173.559  666413.2717
```

```
  print("Correlation Matrix")
```

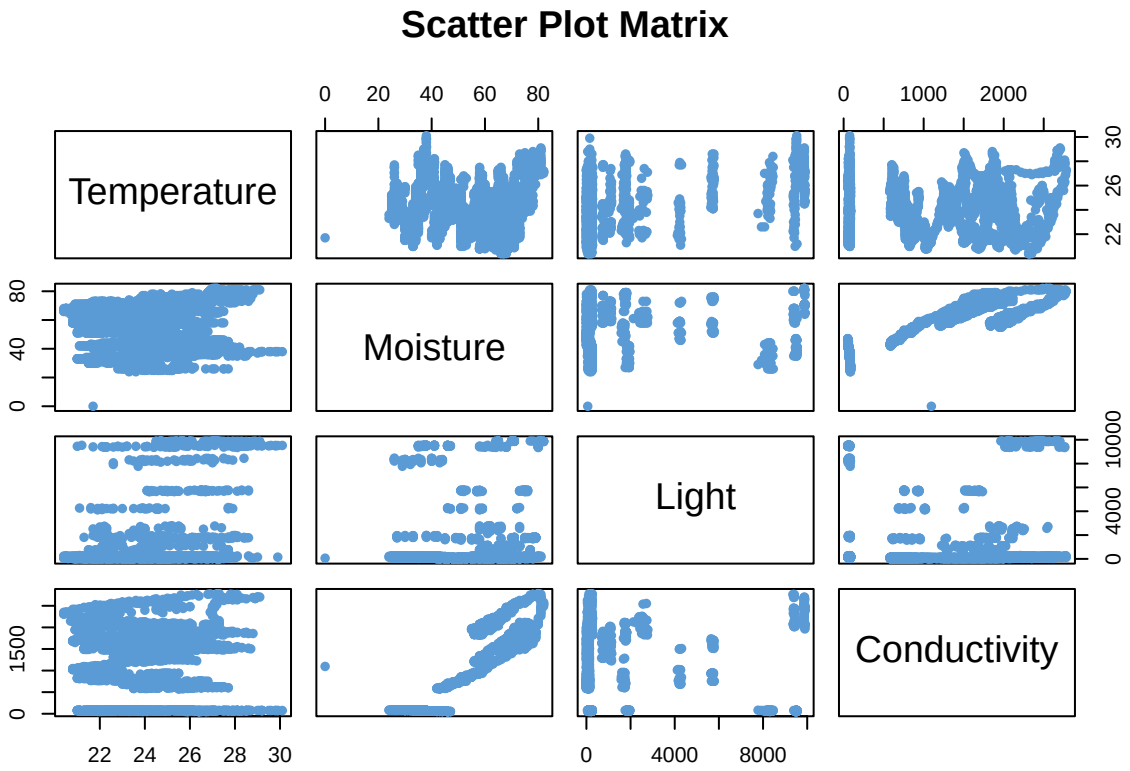
```
## [1] "Correlation Matrix"
```

```r
  cor_matrix <- cor(matrix_data)
  cor_matrix
```

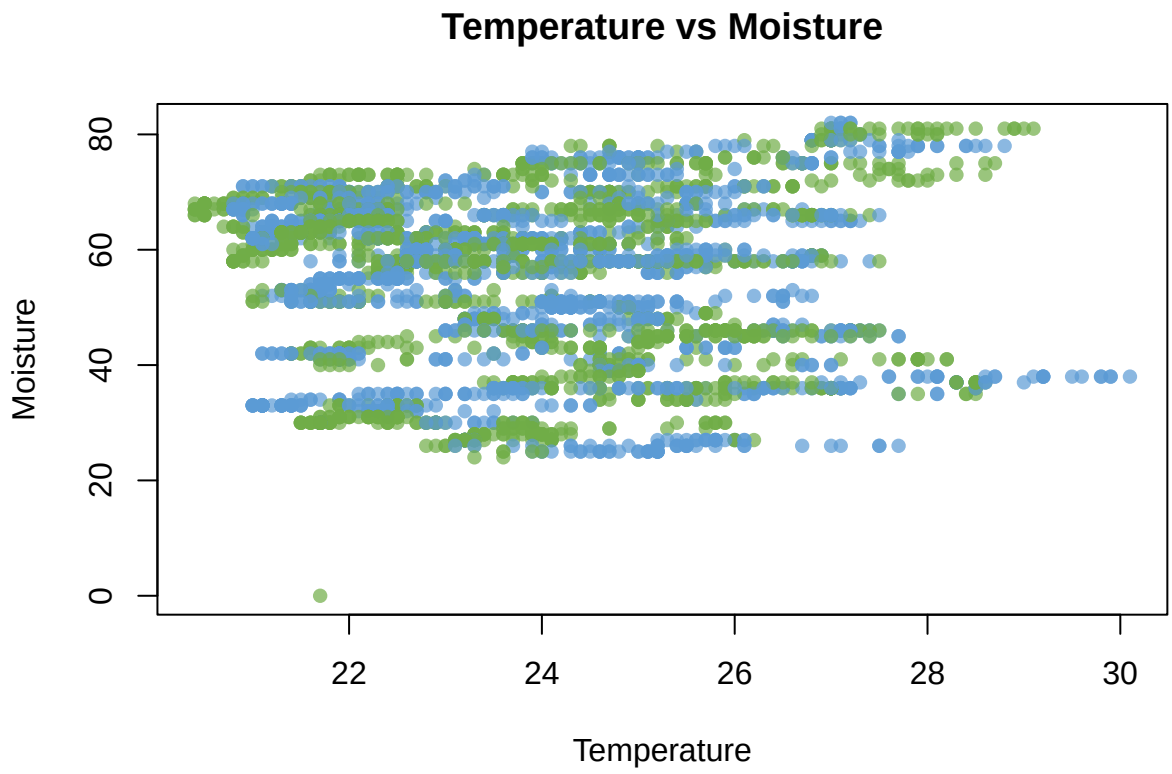
```
##                Temperature    Moisture      Light Conductivity
## Temperature     1.0000000 -0.02785540 0.41166063  -0.11690723
## Moisture       -0.0278554  1.00000000 0.05825268   0.90020101
## Light           0.4116606  0.05825268 1.00000000   0.01718187
## Conductivity   -0.1169072  0.90020101 0.01718187   1.00000000
```

Plotting variables against each other.

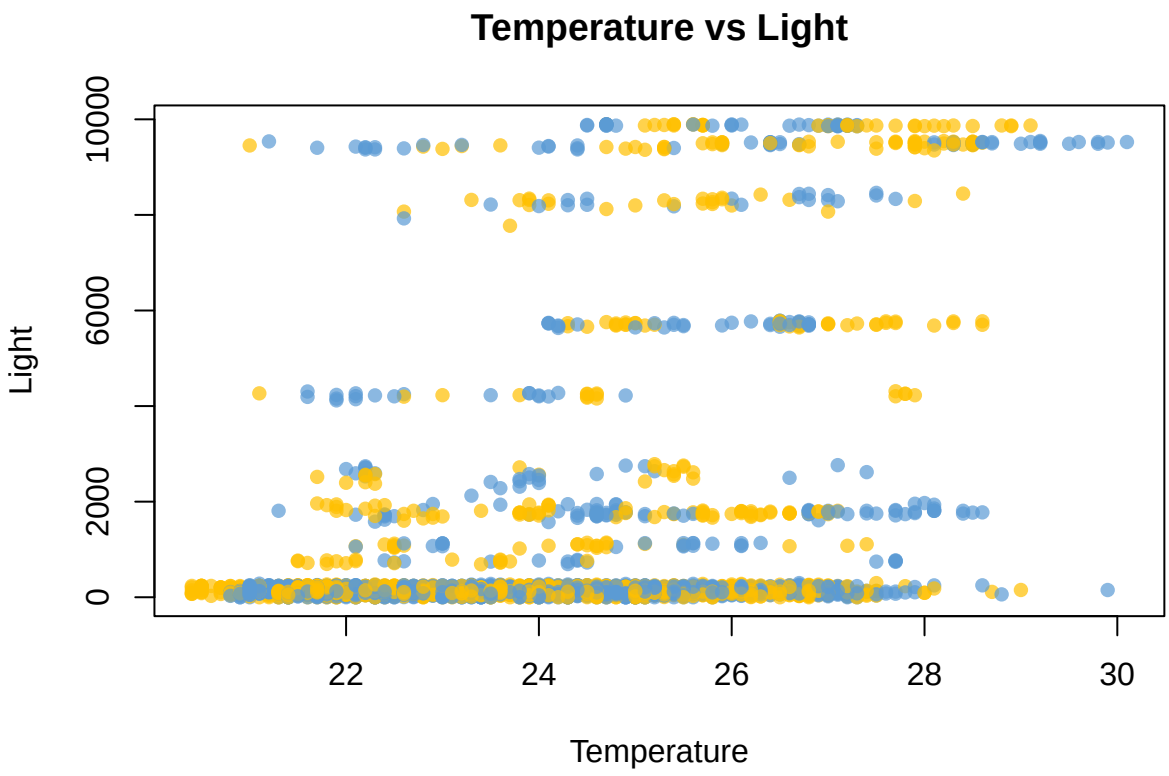
```r
# Define the color palette
colors <- c(
    temperature = "#5B9BD5",  # Soft blue
    moisture = "#70AD47",     # Sage green
    light = "#FFC000",        # Warm yellow
    conductivity = "#ED7D31"  # Muted orange
)

# Assuming agt_data is your data frame
pairs(agt_data[, c("Temperature", "Moisture", "Light", "Conductivity")],
      col = colors["temperature"],
      pch = 16,
      main = "Scatter Plot Matrix")
```



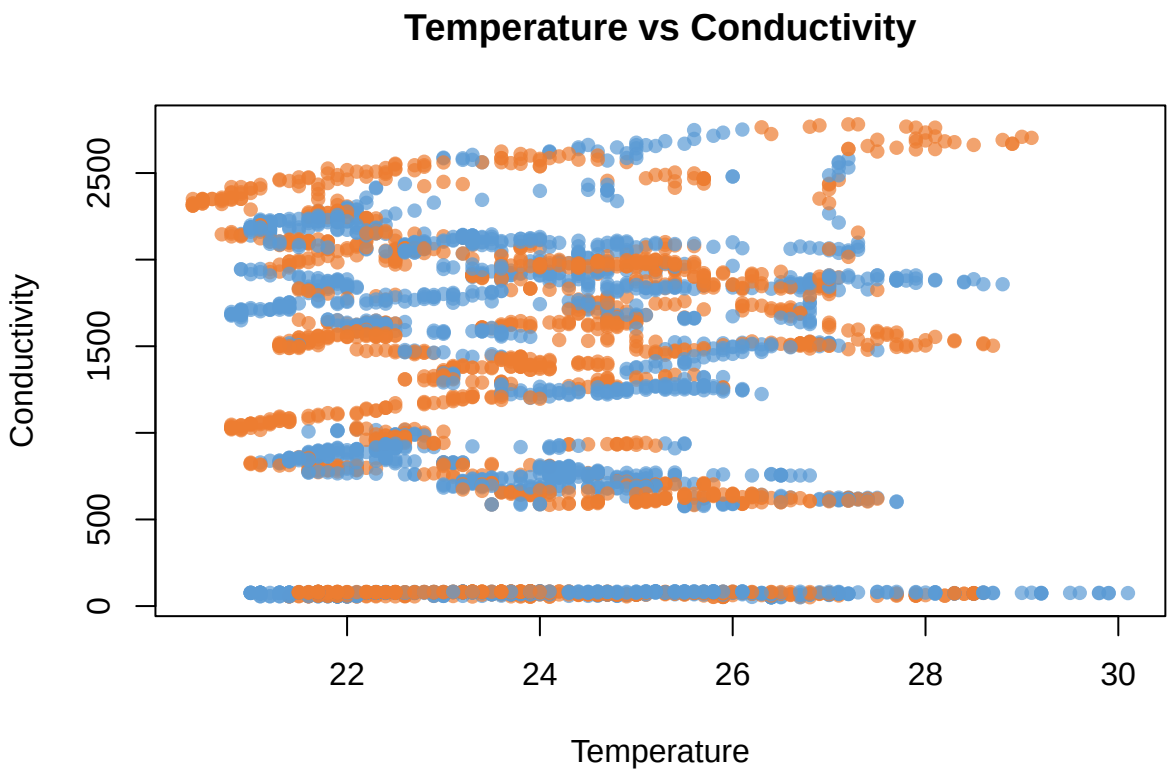
```r
# Plot Temperature vs Moisture
plot(agt_data$Temperature, agt_data$Moisture,
     col = adjustcolor(colors[c("temperature", "moisture")], alpha=0.7),
     xlab = "Temperature",
     ylab = "Moisture",
     main = "Temperature vs Moisture",
     pch = 16)
```



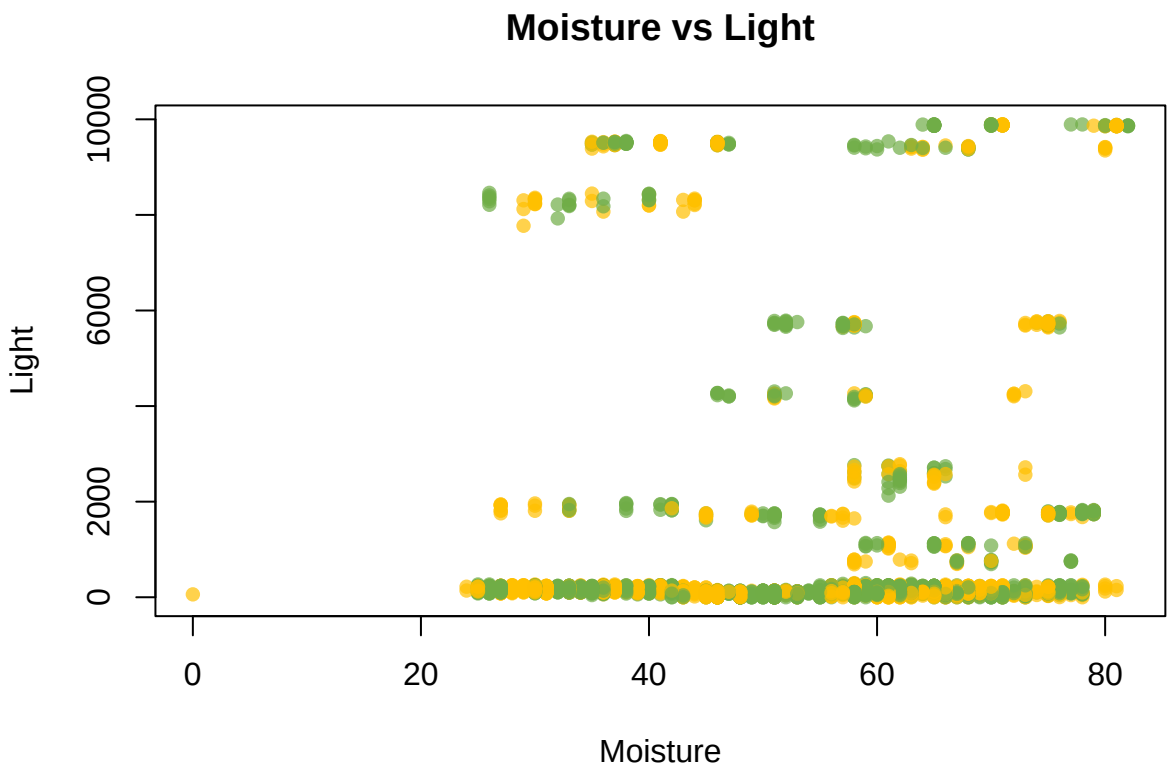

```r
# Plot Temperature vs Light
plot(agt_data$Temperature, agt_data$Light,
     col = adjustcolor(colors[c("temperature", "light")], alpha=0.7),
     xlab = "Temperature",
     ylab = "Light",
     main = "Temperature vs Light",
     pch = 16)
```



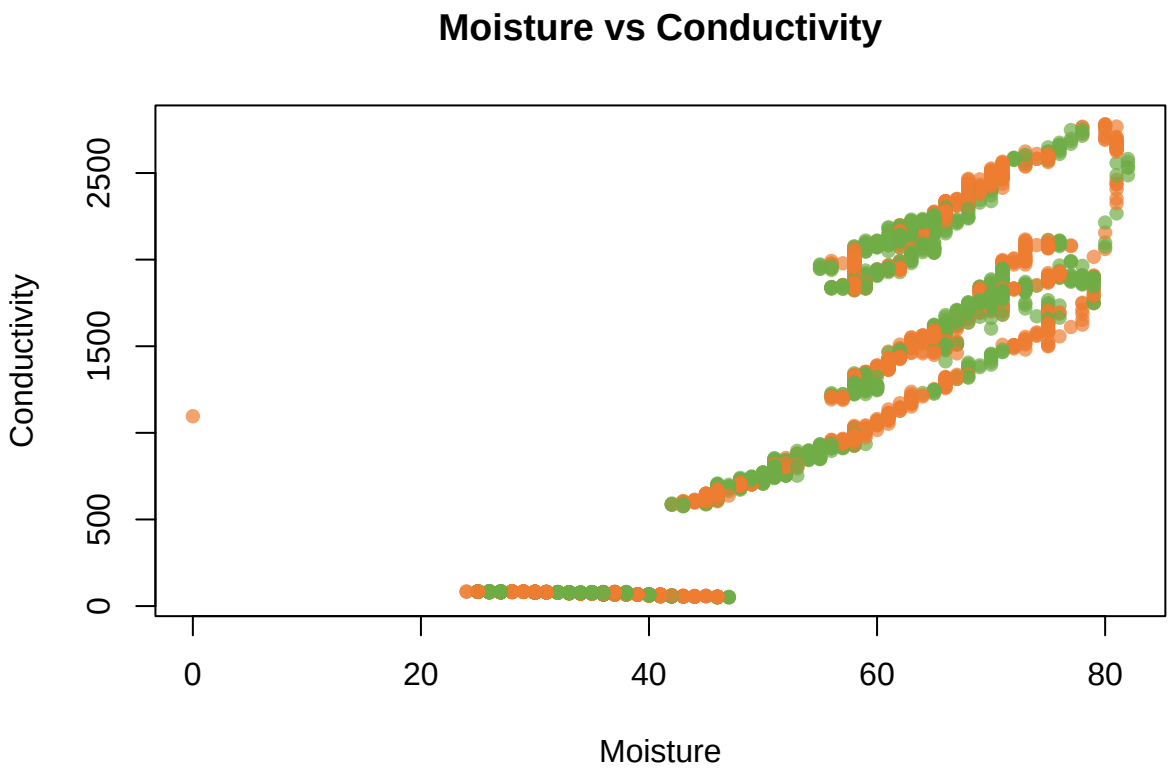

```r
# Plot Temperature vs Conductivity
plot(agt_data$Temperature, agt_data$Conductivity,
     col = adjustcolor(colors[c("temperature", "conductivity")], alpha=0.7),
     xlab = "Temperature",
     ylab = "Conductivity",
     main = "Temperature vs Conductivity",
     pch = 16)
```



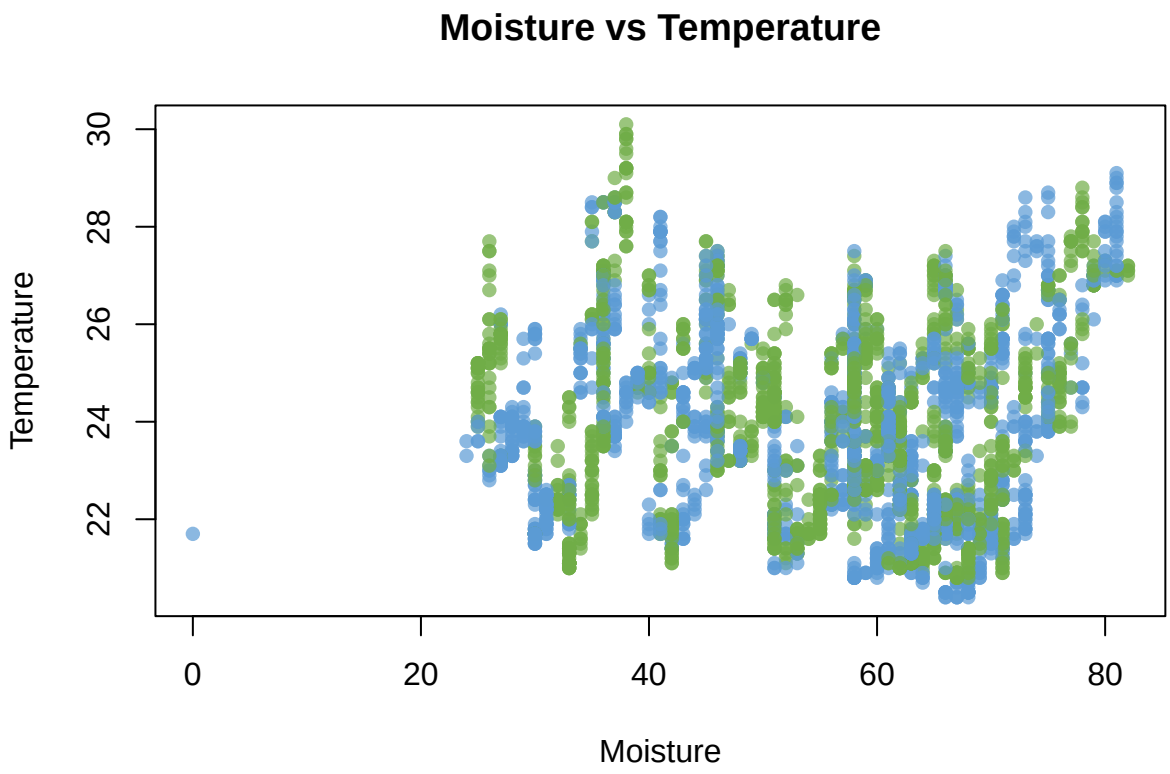

```
# Plot Moisture vs Light
plot(agt_data$Moisture, agt_data$Light,
     col = adjustcolor(colors[c("moisture", "light")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Light",
     main = "Moisture vs Light",
     pch = 16)
```



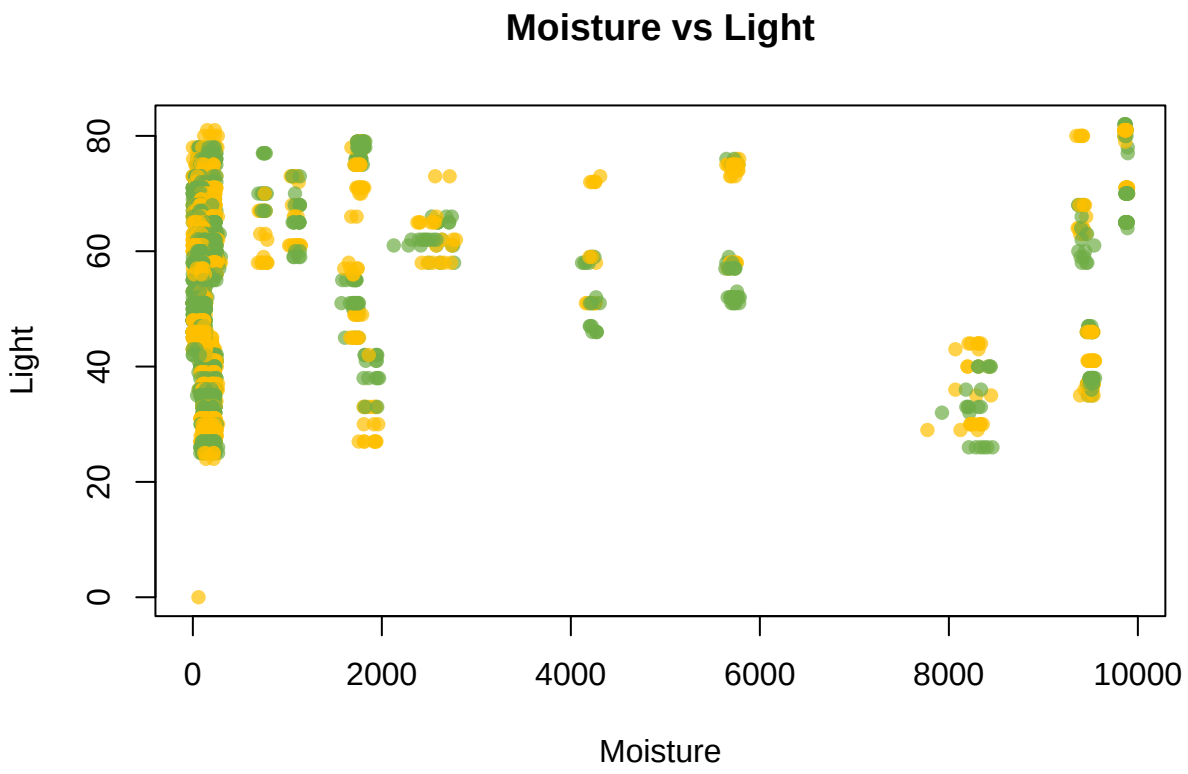

```
# Plot Moisture vs Conductivity
plot(agt_data$Moisture, agt_data$Conductivity,
     col = adjustcolor(colors[c("moisture", "conductivity")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Conductivity",
     main = "Moisture vs Conductivity",
     pch = 16)
```



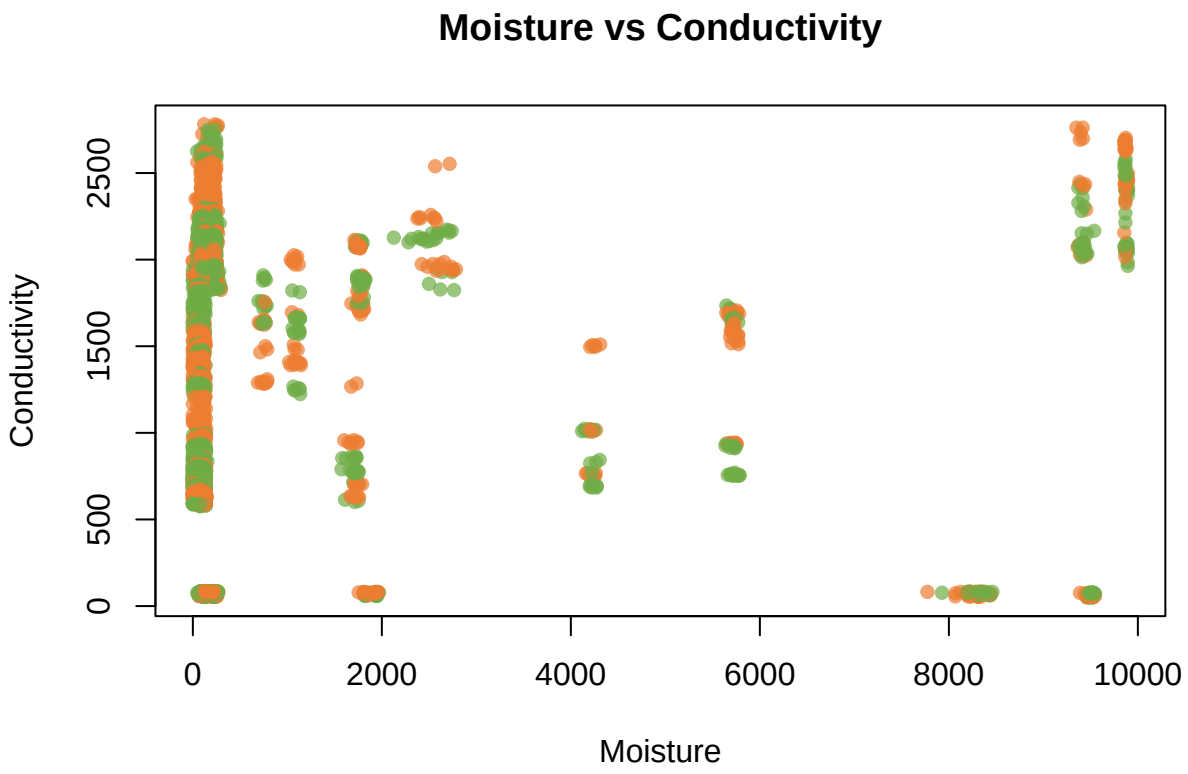

```
# Plot Light vs Conductivity
plot(agt_data$Moisture, agt_data$Temperature,
     col = adjustcolor(colors[c("moisture", "temperature")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Temperature",
     main = "Moisture vs Temperature",
     pch = 16)
```



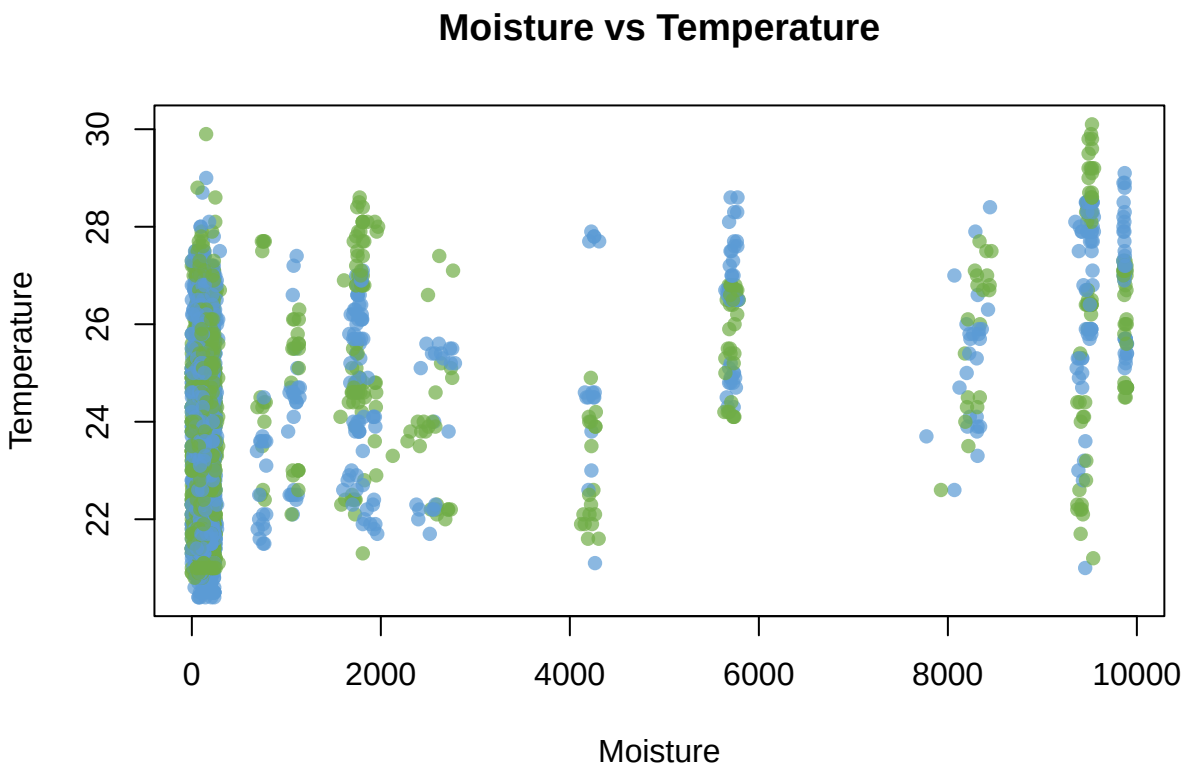

```r
# Plot Moisture vs Light
plot(agt_data$Light, agt_data$Moisture,
     col = adjustcolor(colors[c("moisture", "light")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Light",
     main = "Moisture vs Light",
     pch = 16)
```



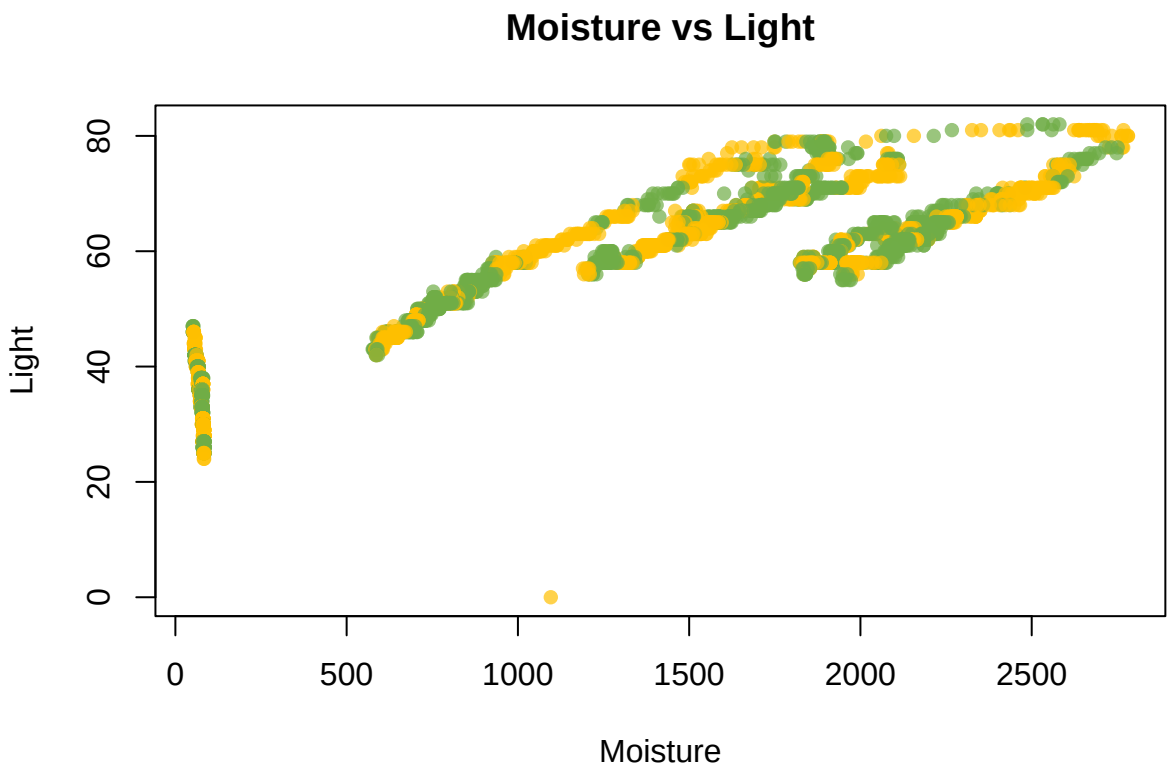

```r
# Plot Moisture vs Conductivity
plot(agt_data$Light, agt_data$Conductivity,
     col = adjustcolor(colors[c("moisture", "conductivity")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Conductivity",
     main = "Moisture vs Conductivity",
     pch = 16)
```



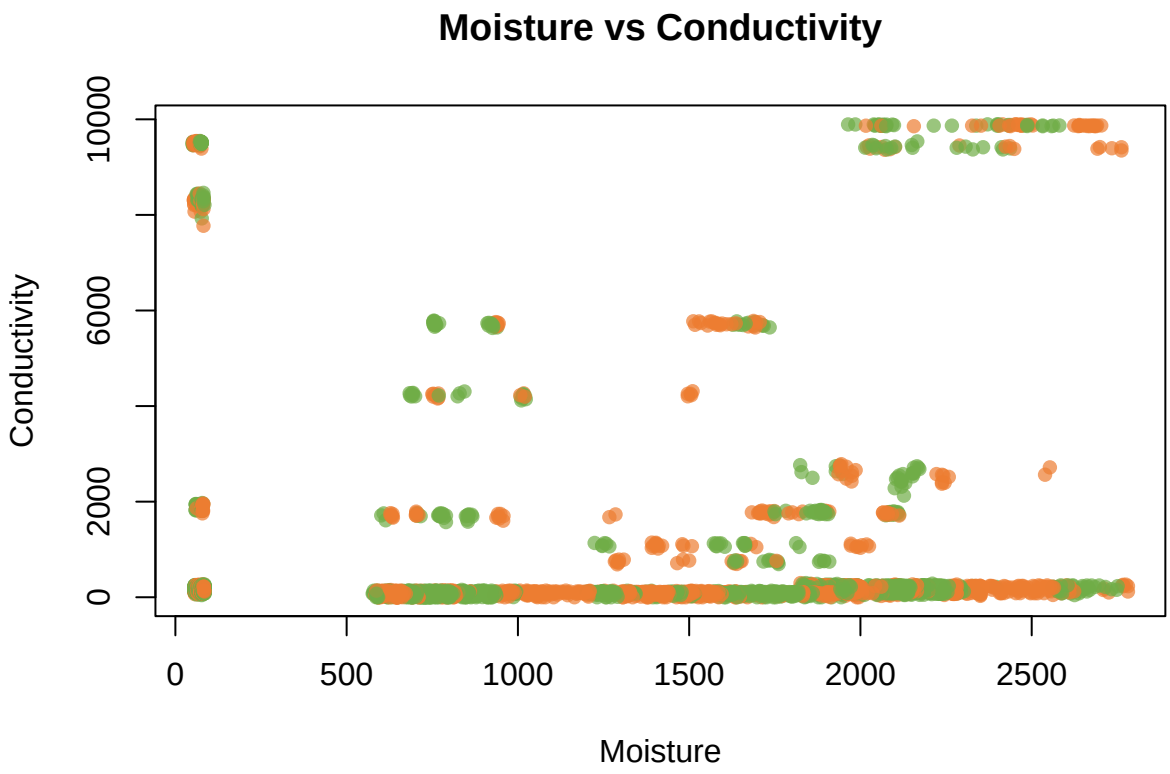

```r
# Plot Light vs Conductivity
plot(agt_data$Light, agt_data$Temperature,
     col = adjustcolor(colors[c("moisture", "temperature")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Temperature",
     main = "Moisture vs Temperature",
     pch = 16)
```



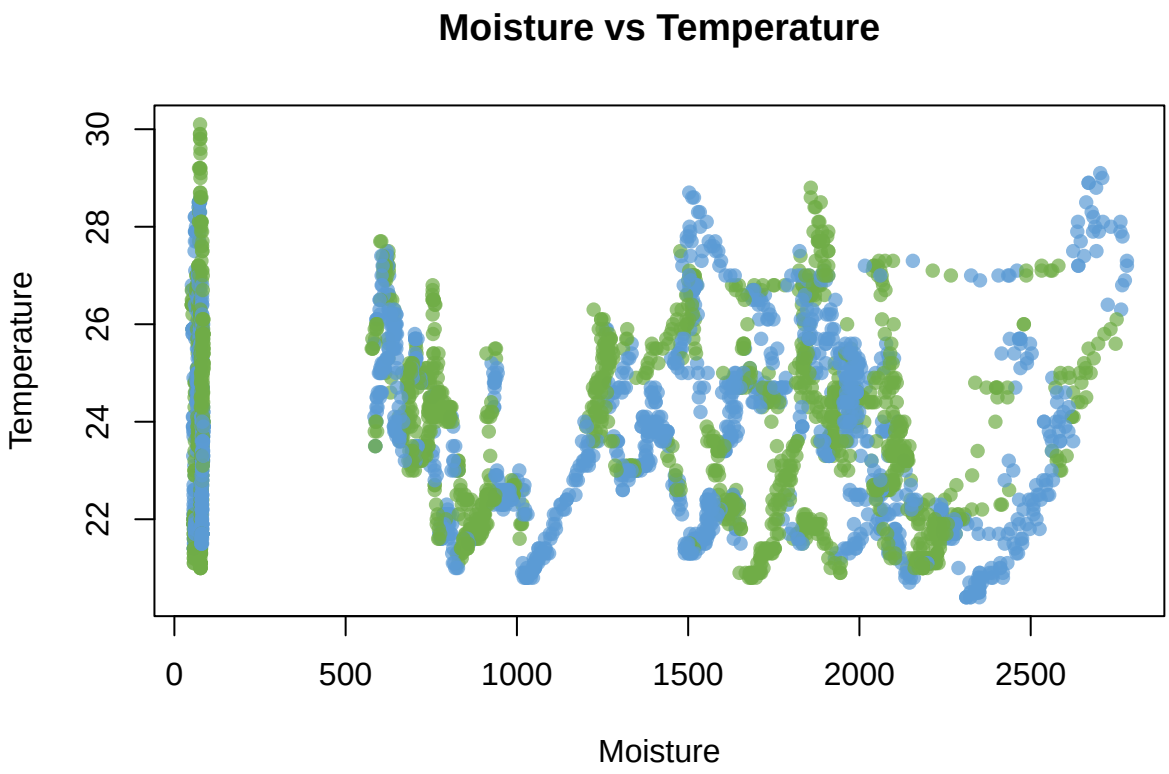

```r
# Plot Moisture vs Light
plot(agt_data$Conductivity, agt_data$Moisture,
     col = adjustcolor(colors[c("moisture", "light")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Light",
     main = "Moisture vs Light",
     pch = 16)
```




```
# Plot Moisture vs Conductivity
plot(agt_data$Conductivity, agt_data$Light,
     col = adjustcolor(colors[c("moisture", "conductivity")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Conductivity",
     main = "Moisture vs Conductivity",
     pch = 16)
```




```r
# Plot Light vs Conductivity
plot(agt_data$Conductivity, agt_data$Temperature,
     col = adjustcolor(colors[c("moisture", "temperature")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Temperature",
     main = "Moisture vs Temperature",
     pch = 16)
```



Heatmap

```r
# Load required library
library(ggplot2)

# Select the relevant variables for correlation
matrix_data <- agt_data[, c("Temperature", "Moisture", "Light", "Conductivity")]

# Calculate the correlation matrix
cor_matrix <- cor(matrix_data, use = "complete.obs")

# Convert the correlation matrix into a format suitable for ggplot
cor_data <- as.data.frame(as.table(cor_matrix))
colnames(cor_data) <- c("Variable1", "Variable2", "Correlation")

# Create the heatmap using ggplot2
ggplot(cor_data, aes(x = Variable1, y = Variable2, fill = Correlation)) +
  geom_tile(color = "white") + # Heatmap tiles
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1),
                       name = "Correlation") + # Color scale
  theme_minimal() + # Clean theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) + # Rotate x-axis labels
```

```
labs(title = "Heatmap of Variable Correlations", x = "Variables", y = "Variables")
```