Weekly Reporting

agrotech live | wigglelabs

2025-04-09

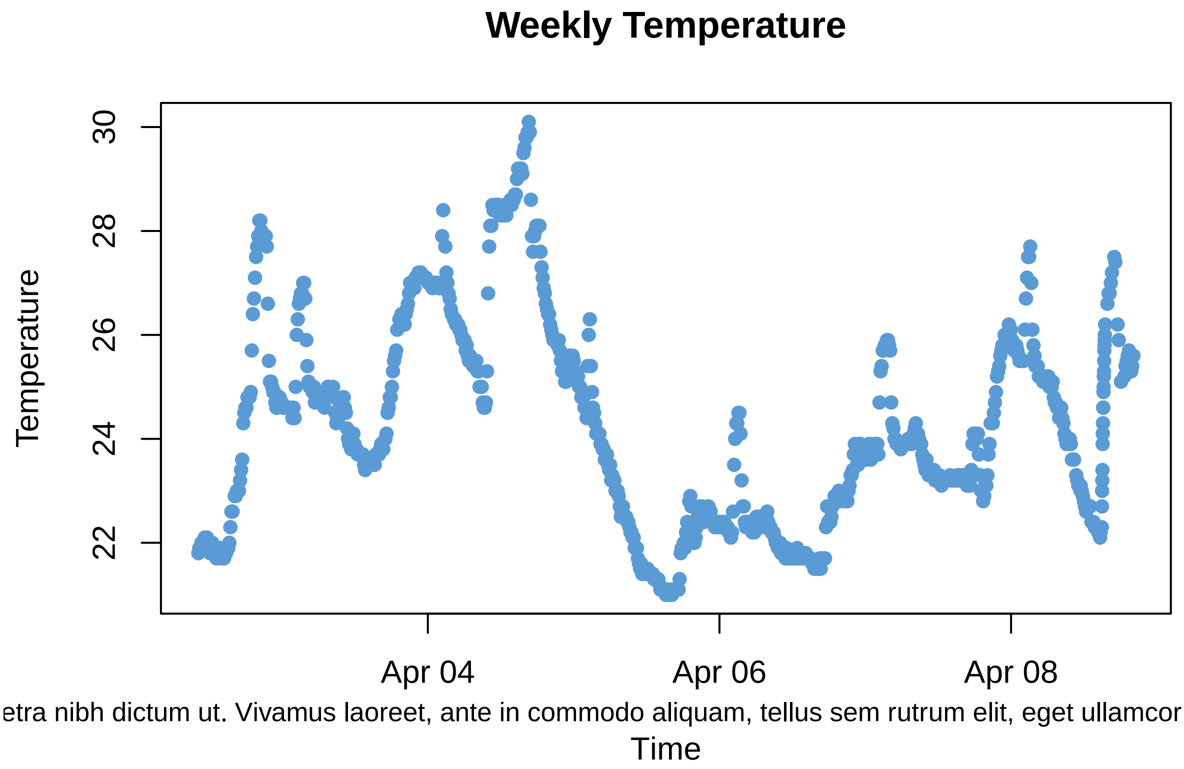
Abstract

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.

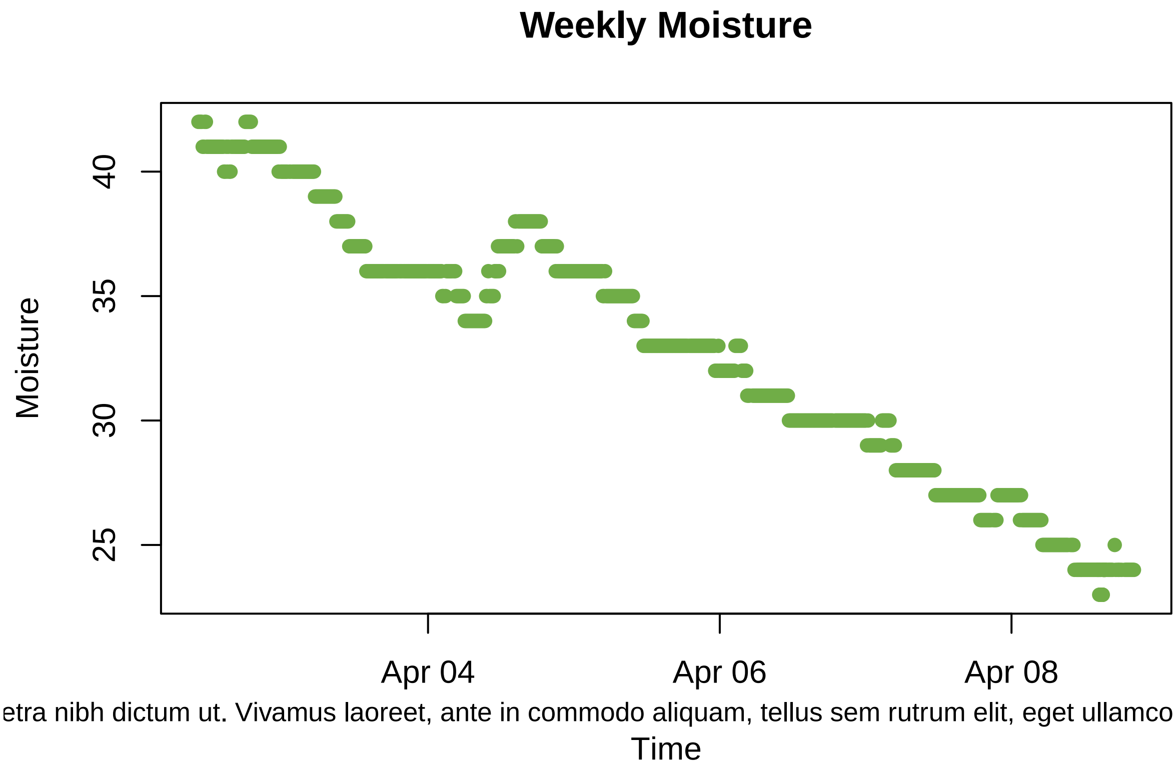
This report covers the past seven days of session data. If there are missing records, then the closest recorded days will be included.

Time Series Comparison

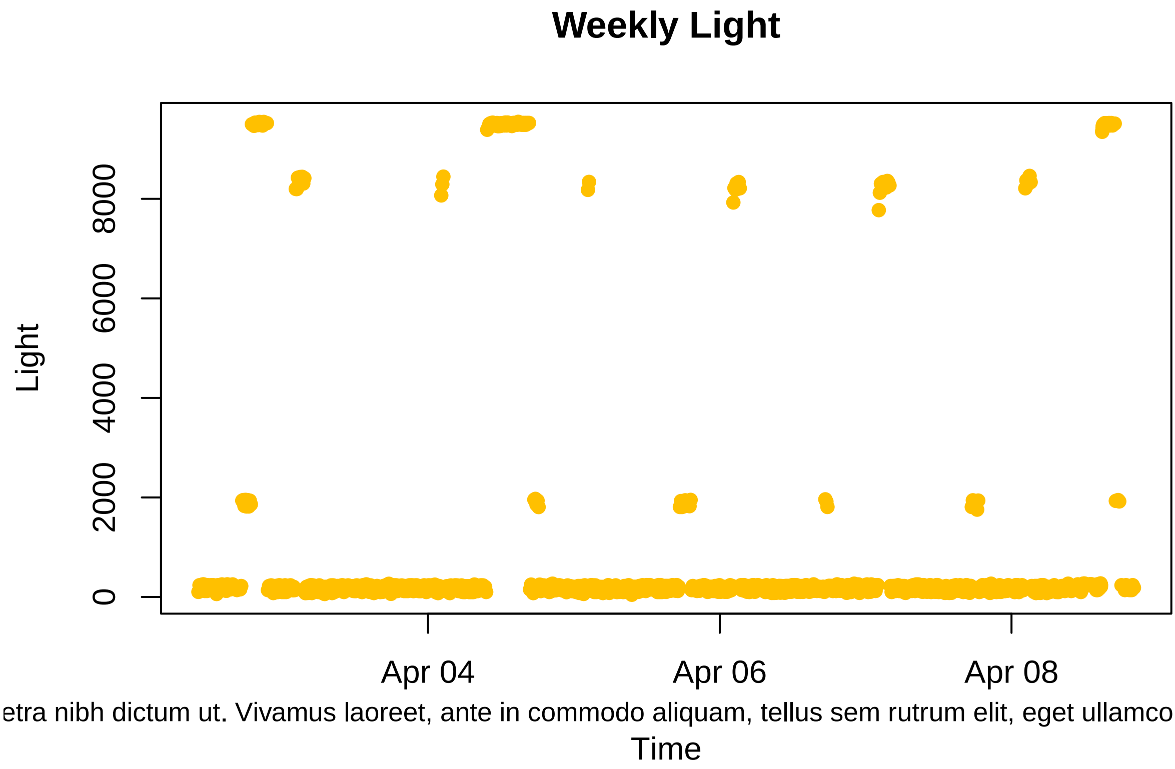
This section examines the relationships between sensor variables from the past week and the initial input data we're comparing the session data to.



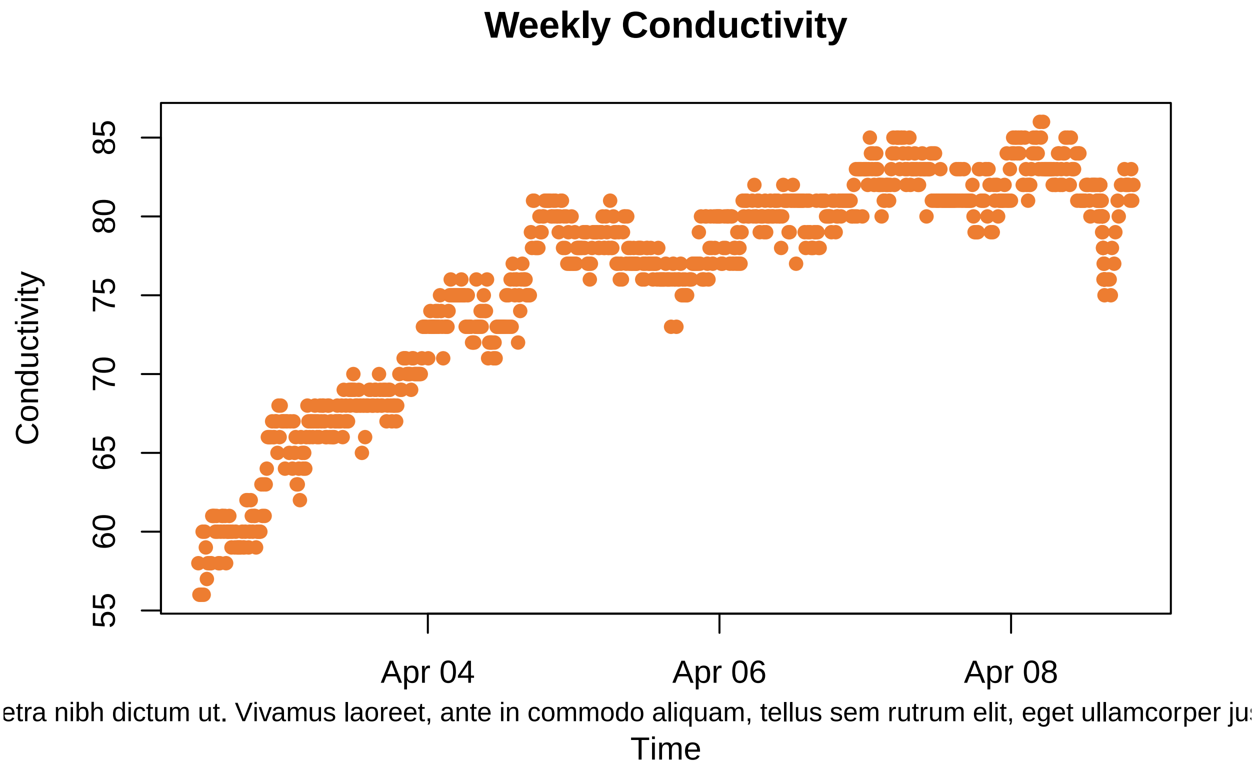
etra nibh dictum ut. Vivamus laoreet, ante in commodo aliquam, tellus sem rutrum elit, eget ullamcorper ju



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(session_data$Temperature)
Moisture <- c(session_data$Moisture)
Light <- c(session_data$Light)
Conductivity <- c(session_data$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     3.934553   2.490763    3190.5781    -2.010008
## Moisture        2.490763  27.523300     926.2374   -29.475269
## Light        3190.578127 926.237412 9349112.3256 -3105.184678
## Conductivity   -2.010008 -29.475269   -3105.1847    47.738555
```

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

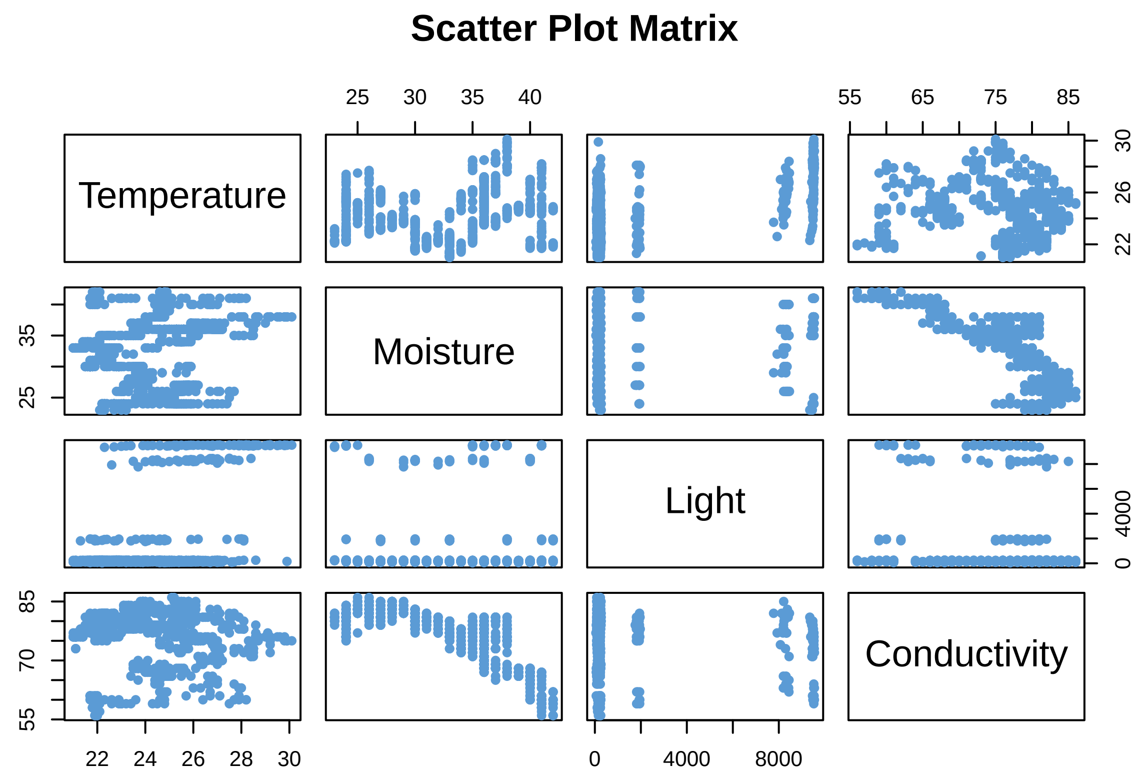
```
## [1] "Correlation Matrix"
```

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

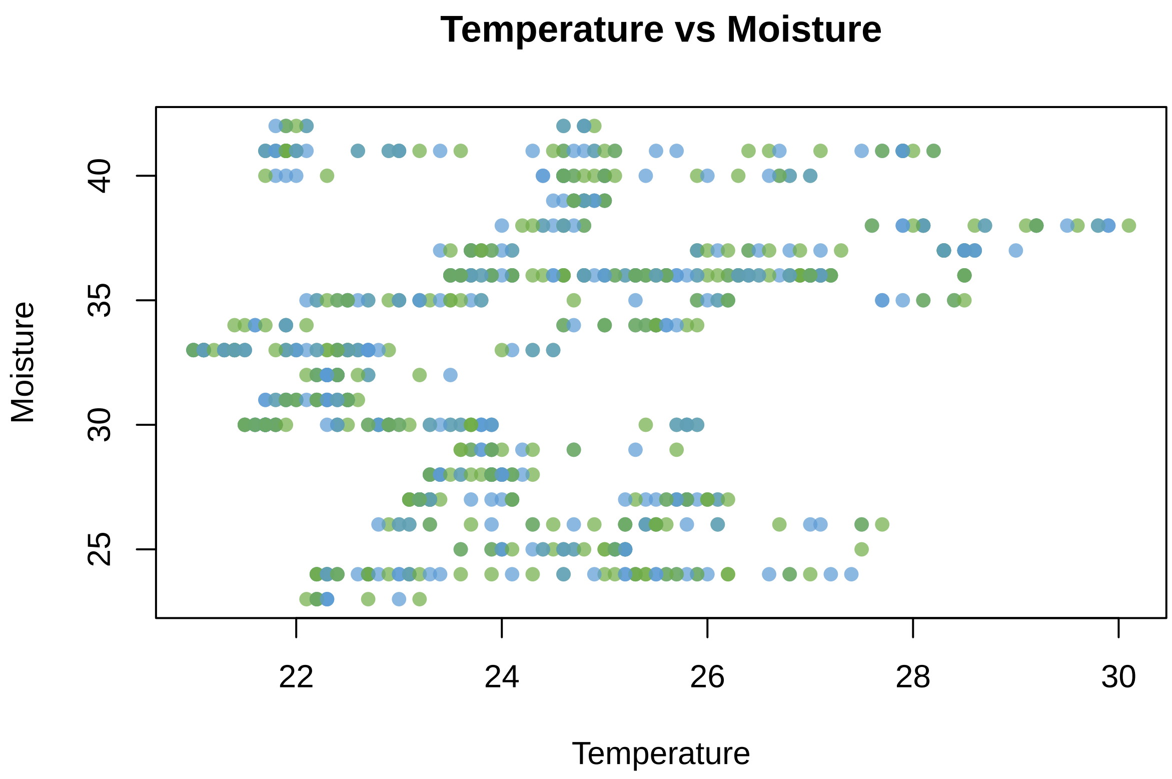
```
##              Temperature    Moisture       Light Conductivity
## Temperature    1.0000000  0.23935055  0.52606149   -0.1466613
## Moisture       0.2393506  1.00000000  0.05774134   -0.8131542
## Light          0.5260615  0.05774134  1.00000000   -0.1469832
## Conductivity  -0.1466613 -0.81315416 -0.14698317    1.0000000
```

Plotting variables against each other.

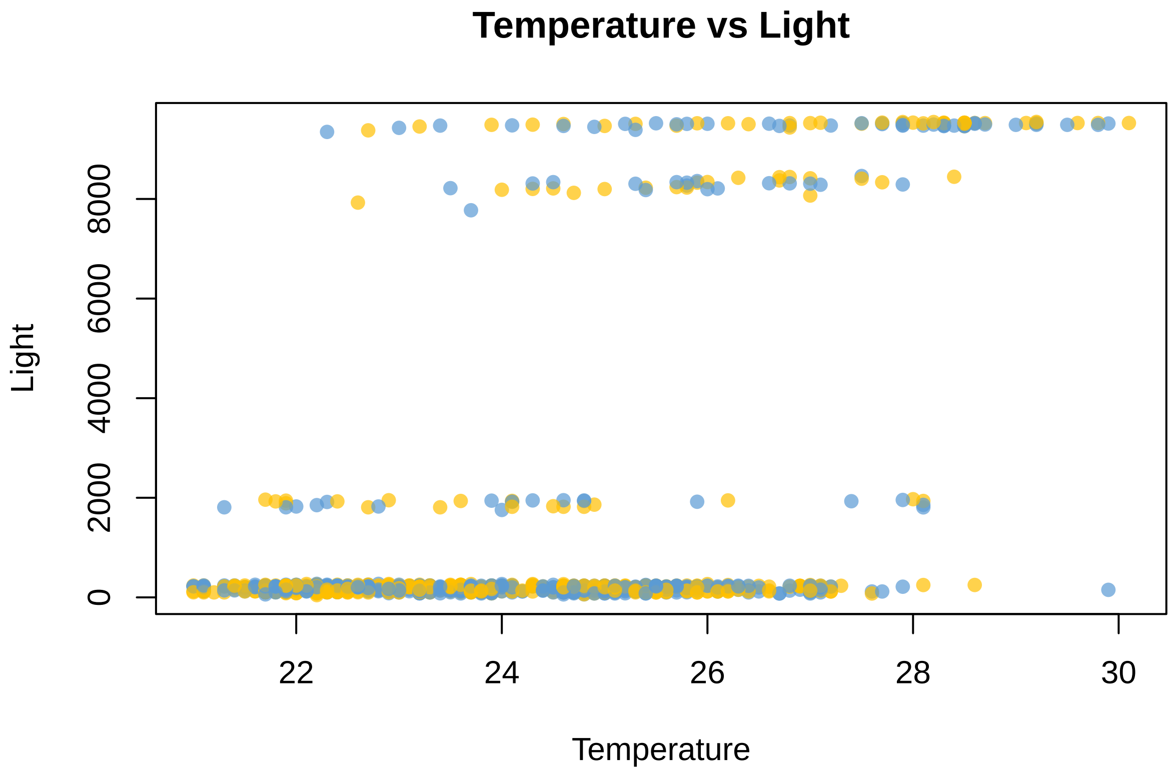
```r
# Define the color palette
colors <- c(
    temperature = "#5B9BD5",   # Soft blue
    moisture = "#70AD47",      # Sage green
    light = "#FFC000",         # Warm yellow
    conductivity = "#ED7D31"  # Muted orange
)

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



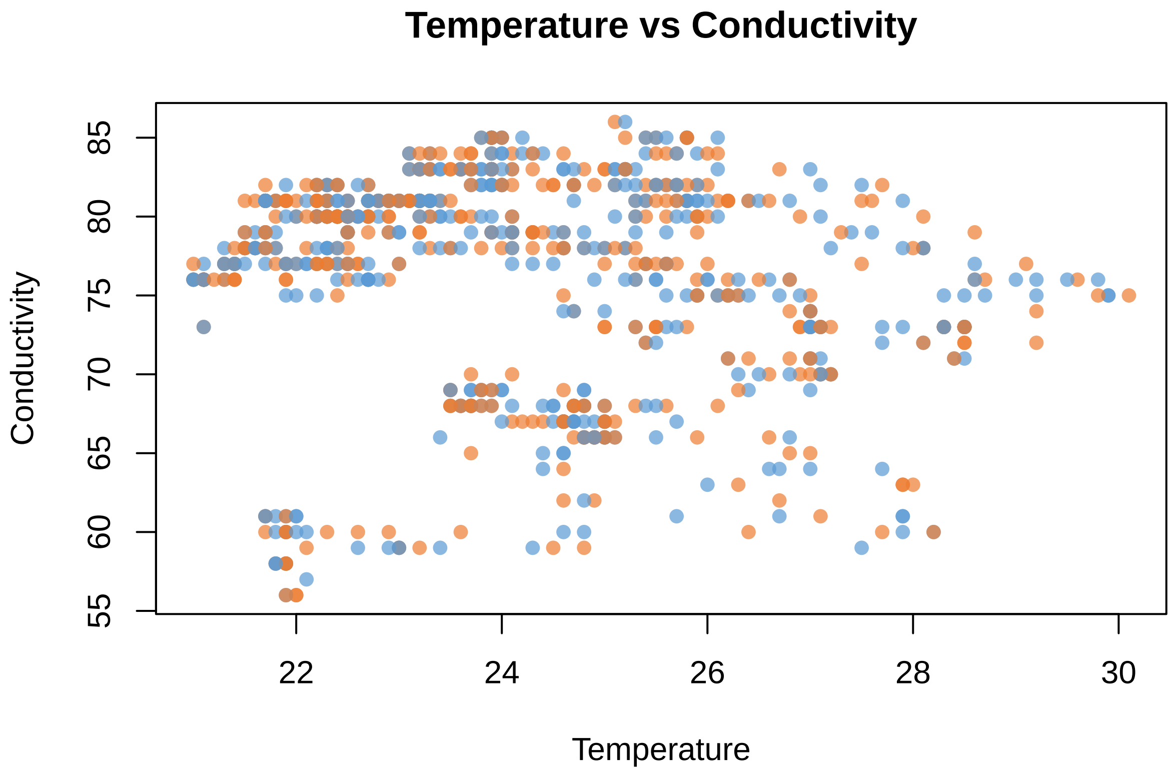
```
# Plot Temperature vs Moisture
plot(session_data$Temperature, session_data$Moisture,
     col = adjustcolor(colors[c("temperature", "moisture")], alpha=0.7),
     xlab = "Temperature",
     ylab = "Moisture",
     main = "Temperature vs Moisture",
     pch = 16)
```



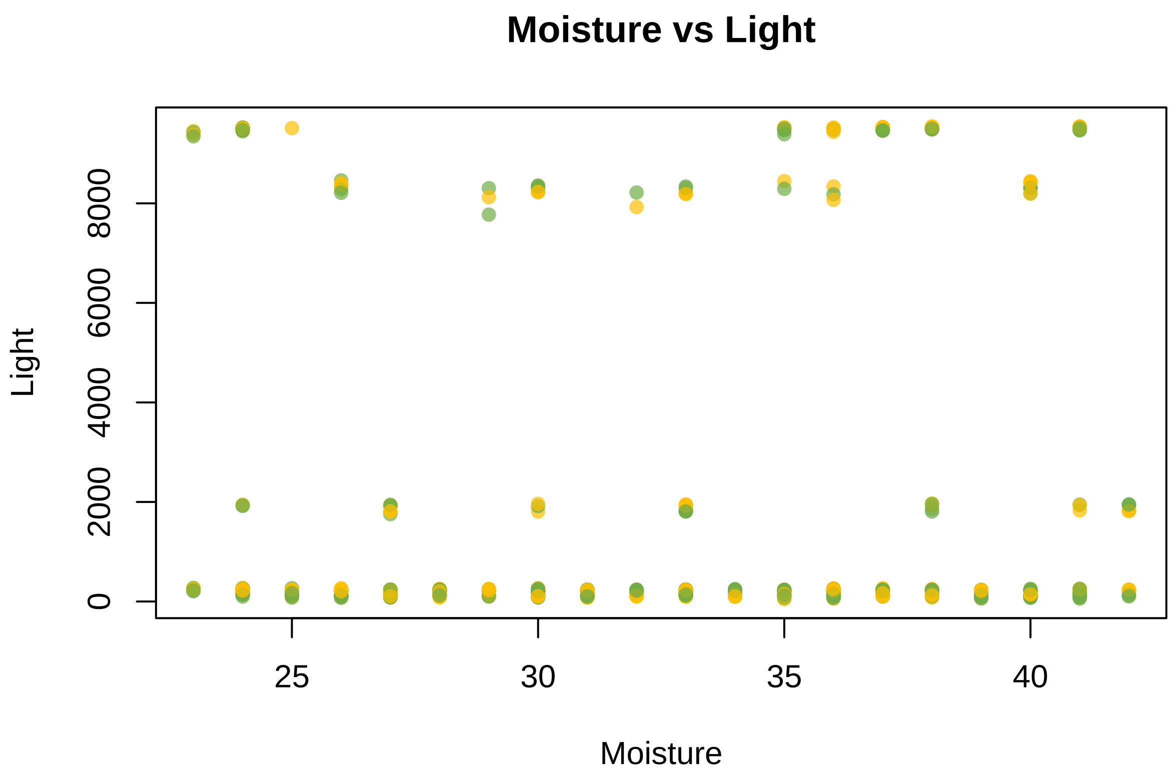
```
# Plot Temperature vs Light
plot(session_data$Temperature, session_data$Light,
     col = adjustcolor(colors[c("temperature", "light")], alpha=0.7),
     xlab = "Temperature",
     ylab = "Light",
     main = "Temperature vs Light",
     pch = 16)
```



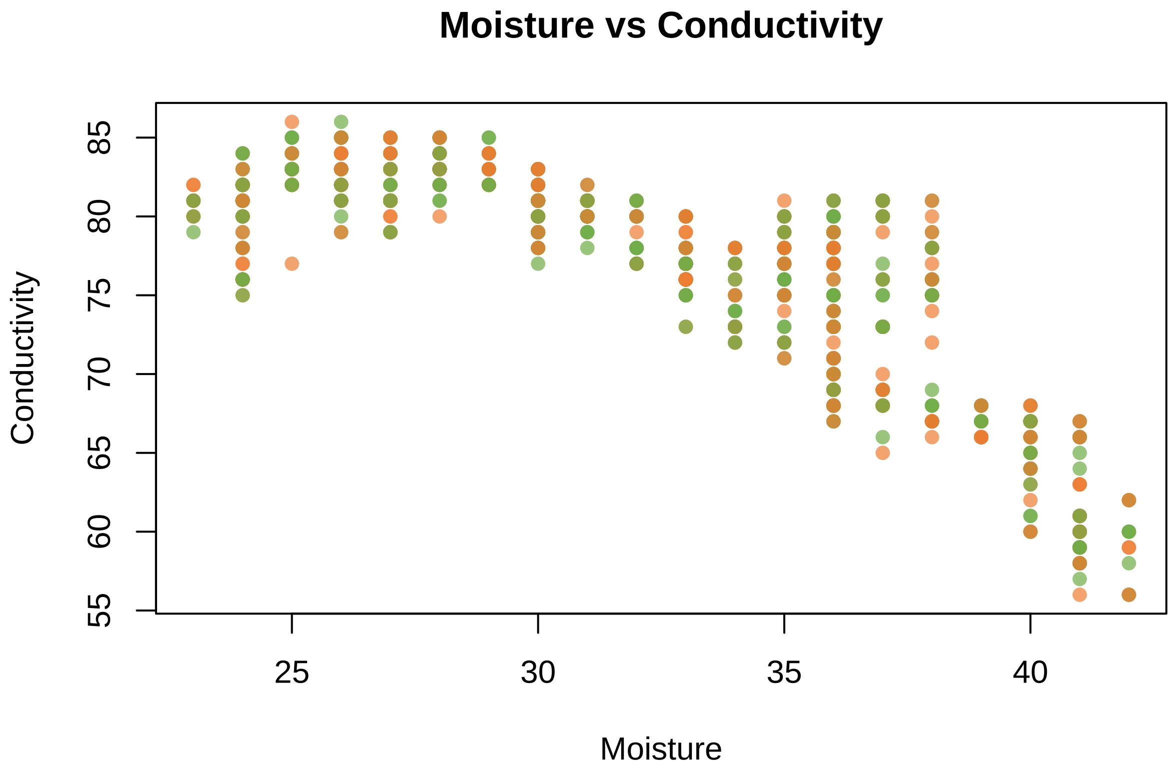
```r
# Plot Temperature vs Conductivity
plot(session_data$Temperature, session_data$Conductivity,
     col = adjustcolor(colors[c("temperature", "conductivity")], alpha=0.7),
     xlab = "Temperature",
     ylab = "Conductivity",
     main = "Temperature vs Conductivity",
     pch = 16)
```



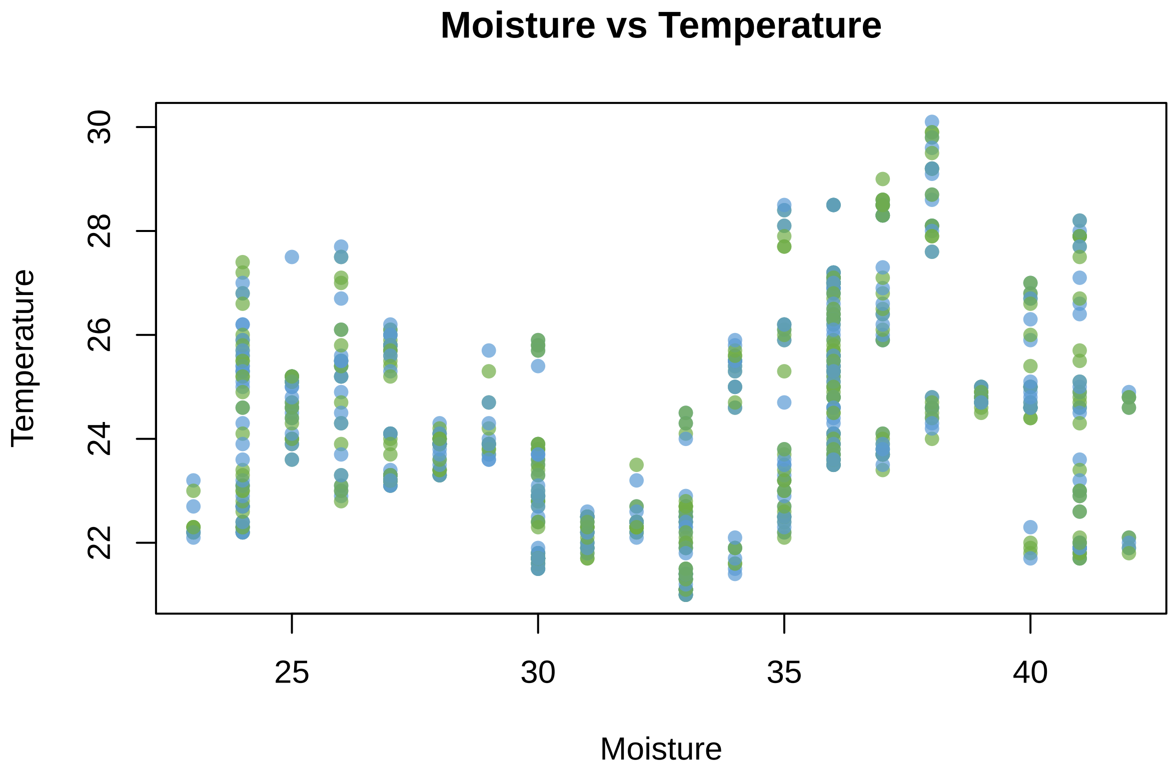
```r
# Plot Moisture vs Light
plot(session_data$Moisture, session_data$Light,
     col = adjustcolor(colors[c("moisture", "light")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Light",
     main = "Moisture vs Light",
     pch = 16)
```



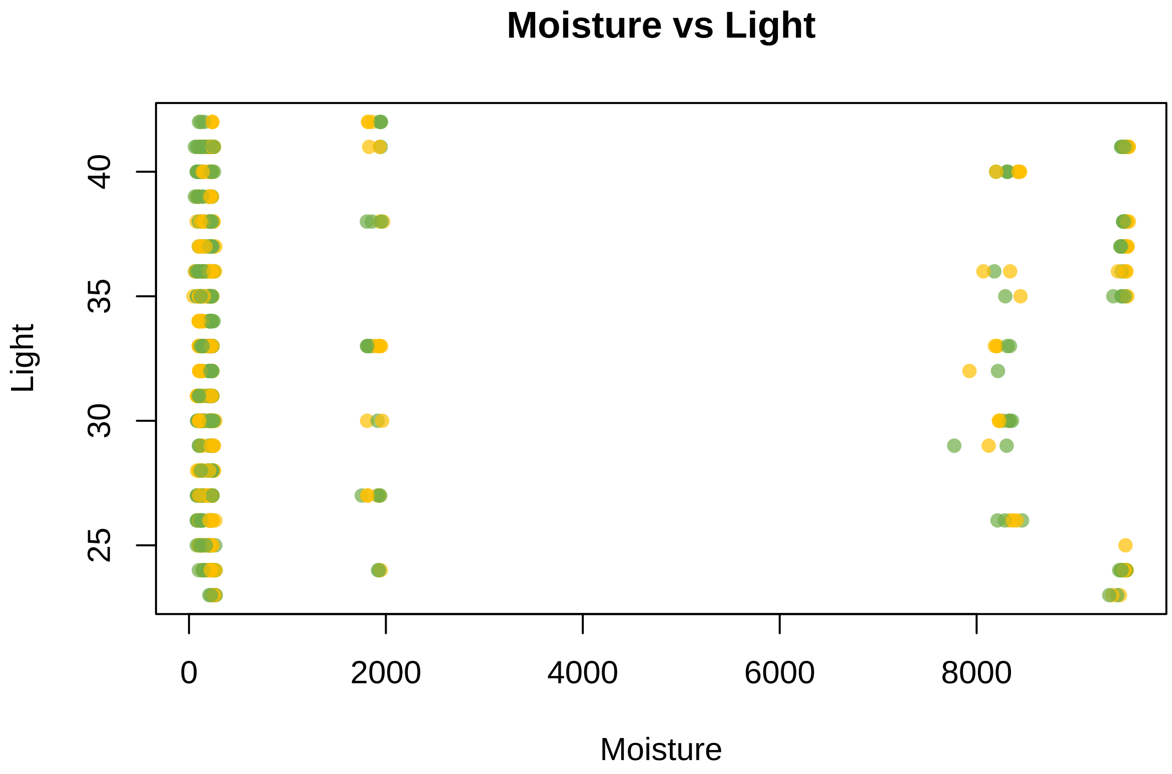
```
# Plot Moisture vs Conductivity
plot(session_data$Moisture, session_data$Conductivity,
     col = adjustcolor(colors[c("moisture", "conductivity")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Conductivity",
     main = "Moisture vs Conductivity",
     pch = 16)
```



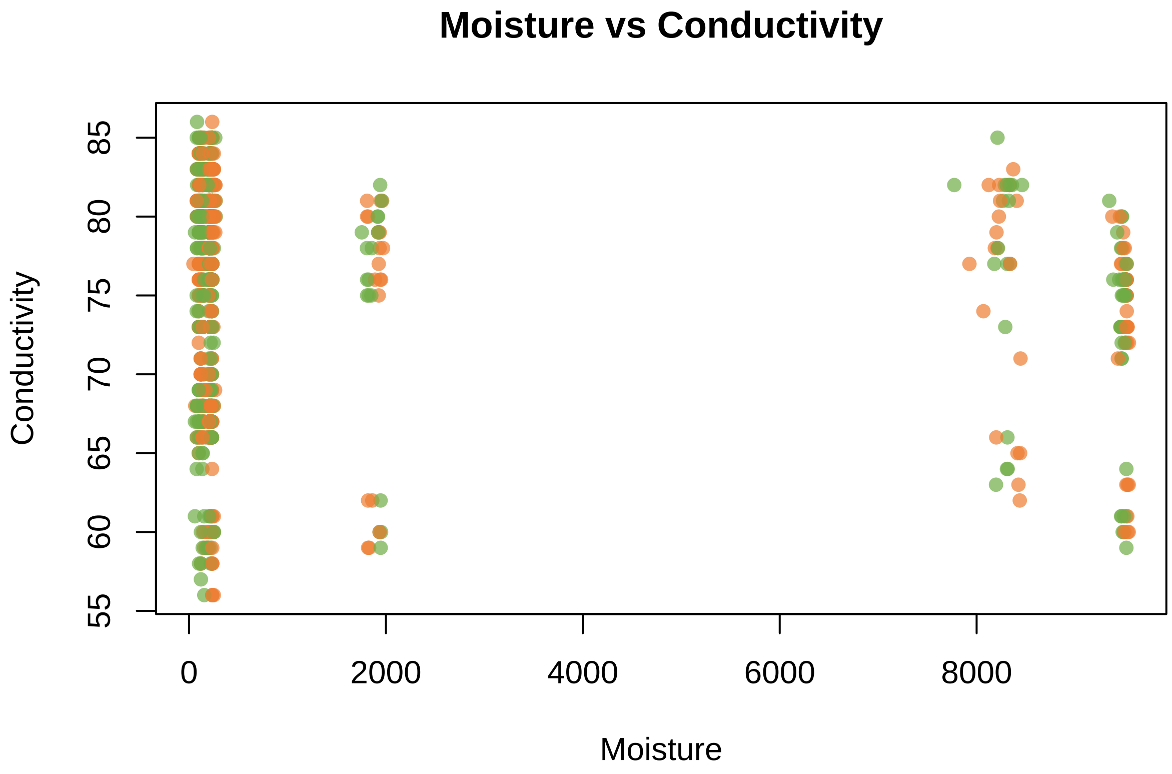
```
# Plot Light vs Conductivity
plot(session_data$Moisture, session_data$Temperature,
     col = adjustcolor(colors[c("moisture", "temperature")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Temperature",
     main = "Moisture vs Temperature",
     pch = 16)
```



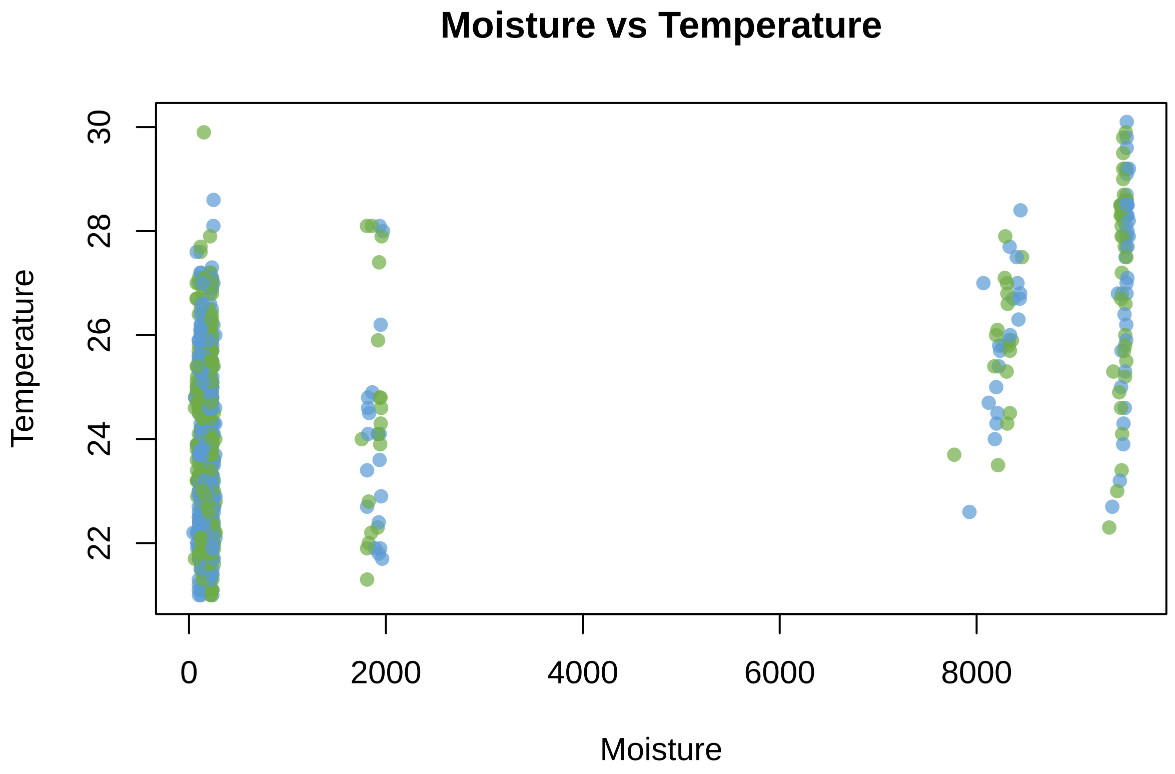
```
# Plot Moisture vs Light
plot(session_data$Light, session_data$Moisture,
     col = adjustcolor(colors[c("moisture", "light")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Light",
     main = "Moisture vs Light",
     pch = 16)
```



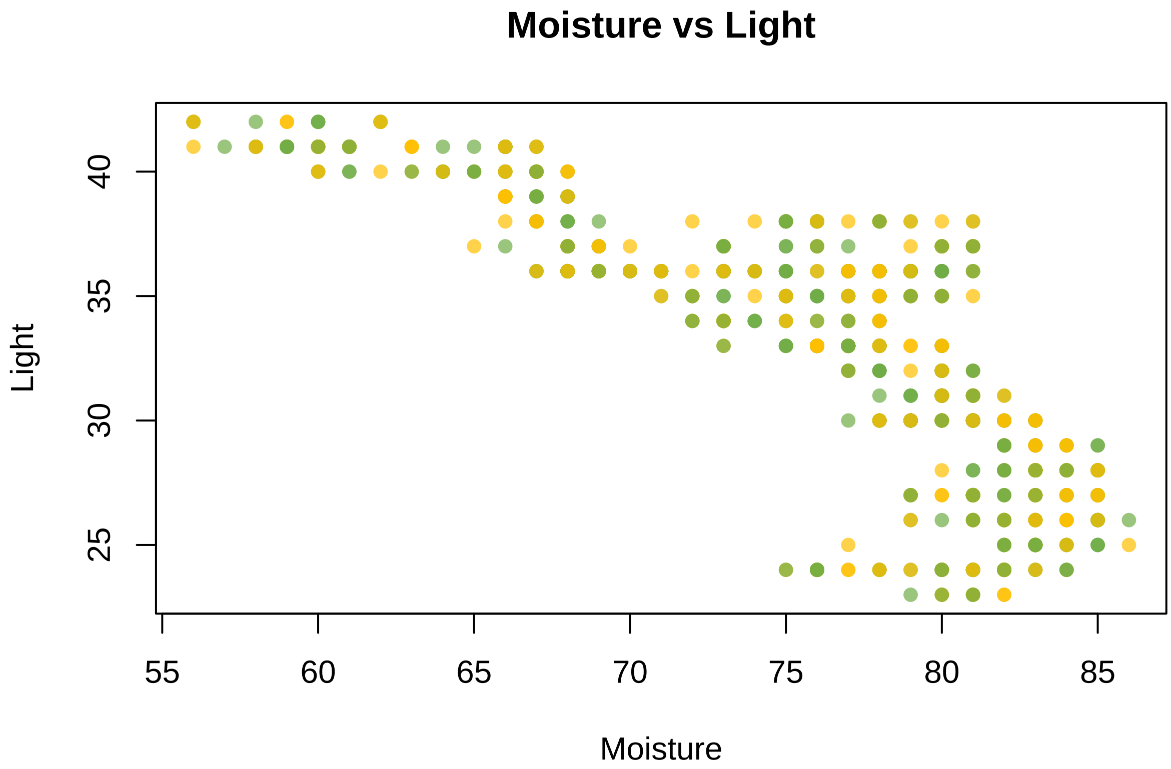
```
# Plot Moisture vs Conductivity
plot(session_data$Light, session_data$Conductivity,
     col = adjustcolor(colors[c("moisture", "conductivity")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Conductivity",
     main = "Moisture vs Conductivity",
     pch = 16)
```



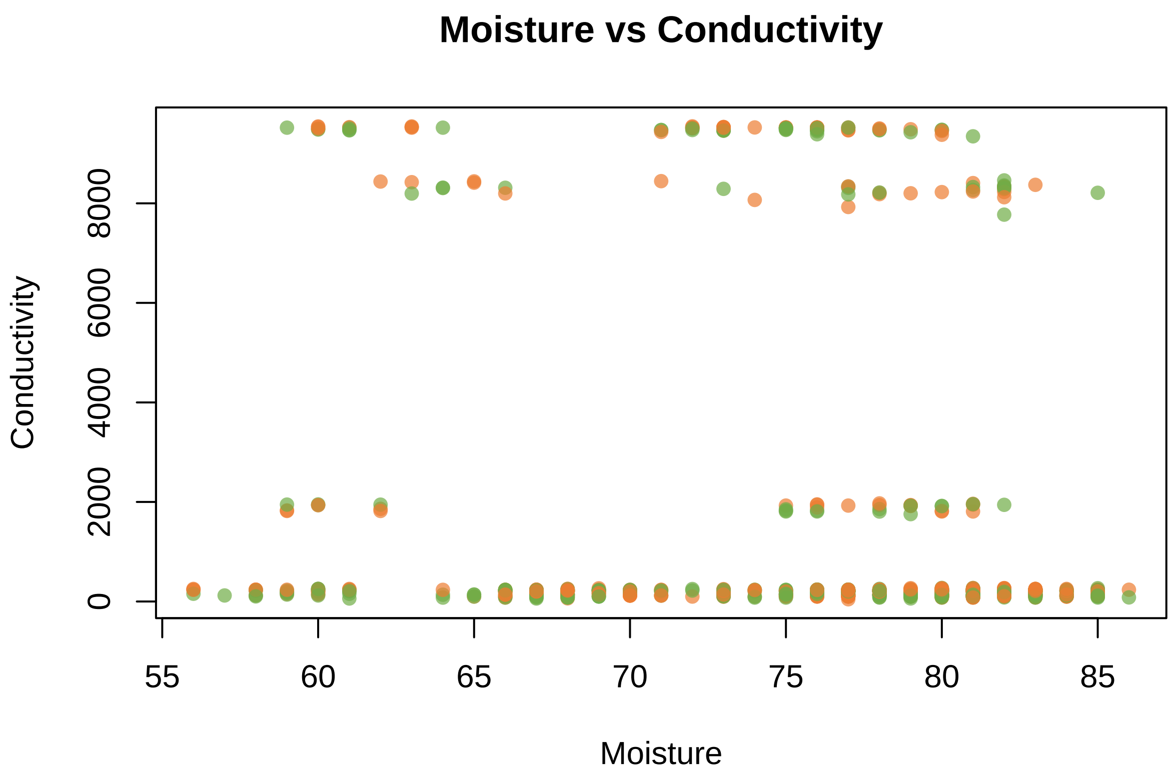
```
# Plot Light vs Conductivity
plot(session_data$Light, session_data$Temperature,
     col = adjustcolor(colors[c("moisture", "temperature")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Temperature",
     main = "Moisture vs Temperature",
     pch = 16)
```



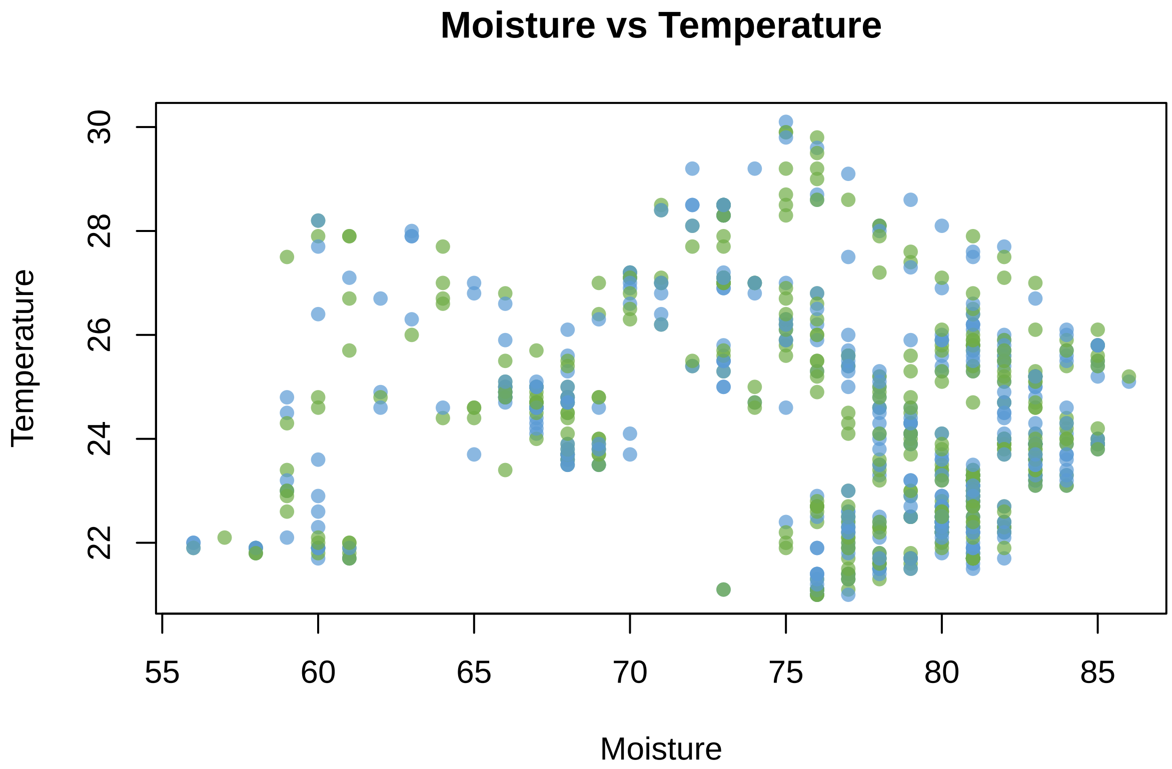
```r
# Plot Moisture vs Light
plot(session_data$Conductivity, session_data$Moisture,
     col = adjustcolor(colors[c("moisture", "light")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Light",
     main = "Moisture vs Light",
     pch = 16)
```



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



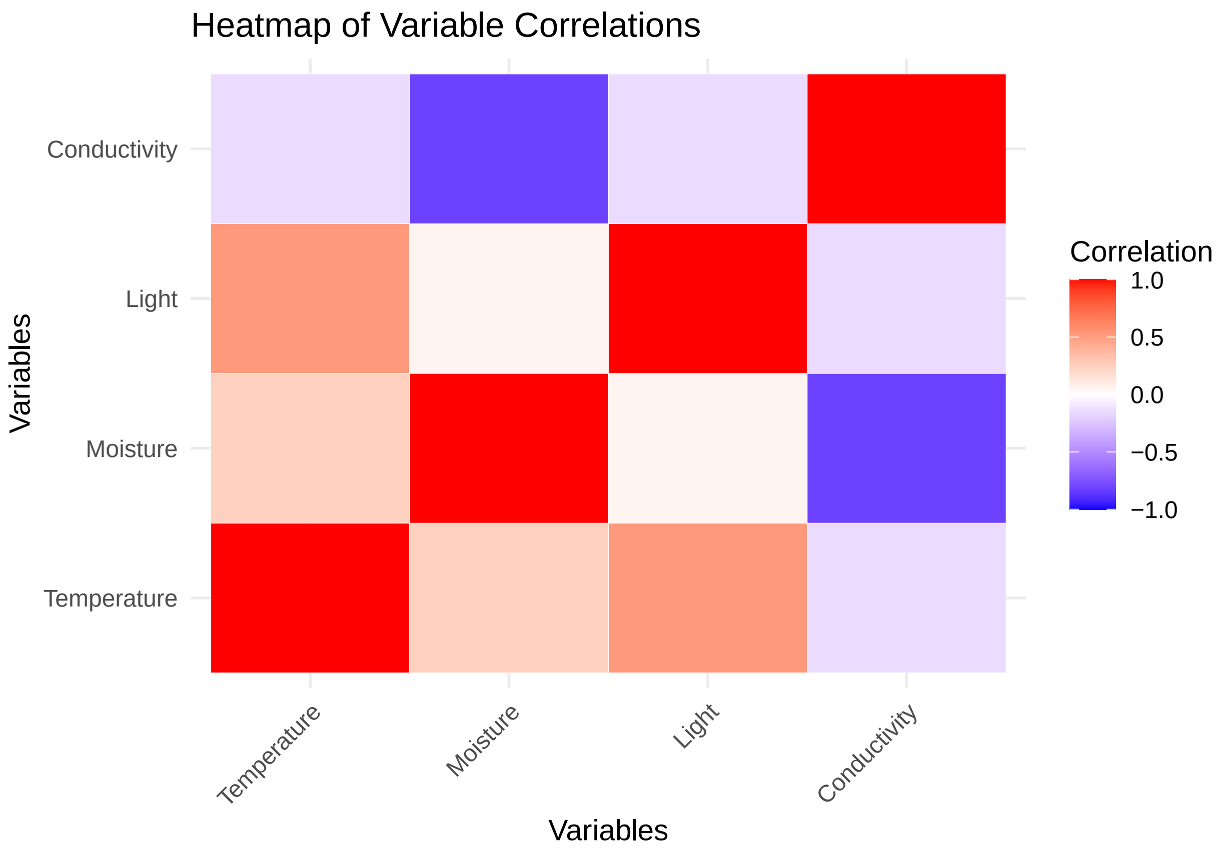
```r
# Plot Light vs Conductivity
plot(session_data$Conductivity, session_data$Temperature,
     col = adjustcolor(colors[c("moisture", "temperature")], alpha=0.7),
     xlab = "Moisture",
     ylab = "Temperature",
     main = "Moisture vs Temperature",
     pch = 16)
```



Heatmap

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

# Select the relevant variables for correlation
matrix_data <- session_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")
```


Heatmap of Variable Correlations
Conductivity
Light
Moisture
Temperature
Variables
Temperature
Moisture
Light
Conductivity
Variables
Correlation