Project

Supervised Machine Learning: Plant Health

Reading in the dataset

```
health_data <- "data/plant_moniter_health_data.csv"
data <- read.csv(health_data,
                 header = TRUE, sep = ",")
```

```
data$Health_Status = as.factor(data$Health_Status)
summary(data)
```

```
##    Plant_ID         Temperature_C     Humidity_.    Soil_Moisture_.
##  Length:1000        Min.   :15.28   Min.   :30.60   Min.   : -0.2927
##  Class  :character   1st Qu.:23.06   1st Qu.:53.94   1st Qu.: 35.2800
##  Mode   :character   Median :25.08   Median :60.63   Median : 44.9962
##                      Mean   :25.06   Mean   :60.71   Mean   : 45.0875
##                      3rd Qu.:26.94   3rd Qu.:67.29   3rd Qu.: 54.9137
##                      Max.   :36.56   Max.   :91.93   Max.   :103.8936
##     Soil_pH      Nutrient_Level  Light_Intensity_lux  Health_Score
##  Min.   :5.035   Min.   :18.23   Min.   :11301       Min.   : 52.87
##  1st Qu.:6.131   1st Qu.:43.17   1st Qu.:17919       1st Qu.: 72.45
##  Median :6.500   Median :49.82   Median :19872       Median : 79.45
##  Mean   :6.491   Mean   :49.51   Mean   :19860       Mean   : 79.72
##  3rd Qu.:6.833   3rd Qu.:56.39   3rd Qu.:21837       3rd Qu.: 87.00
##  Max.   :8.122   Max.   :81.13   Max.   :29295       Max.   :115.29
##  Health_Status
##  0:174
##  1:826
##
##
##
##
```

Exploratory Data Analysis

To check if there are duplicate plant observations recorded, we check if the distinct count of the `Plant_ID` is the same as the number of rows.

```
data %>% summarise(count = n_distinct(Plant_ID))
```

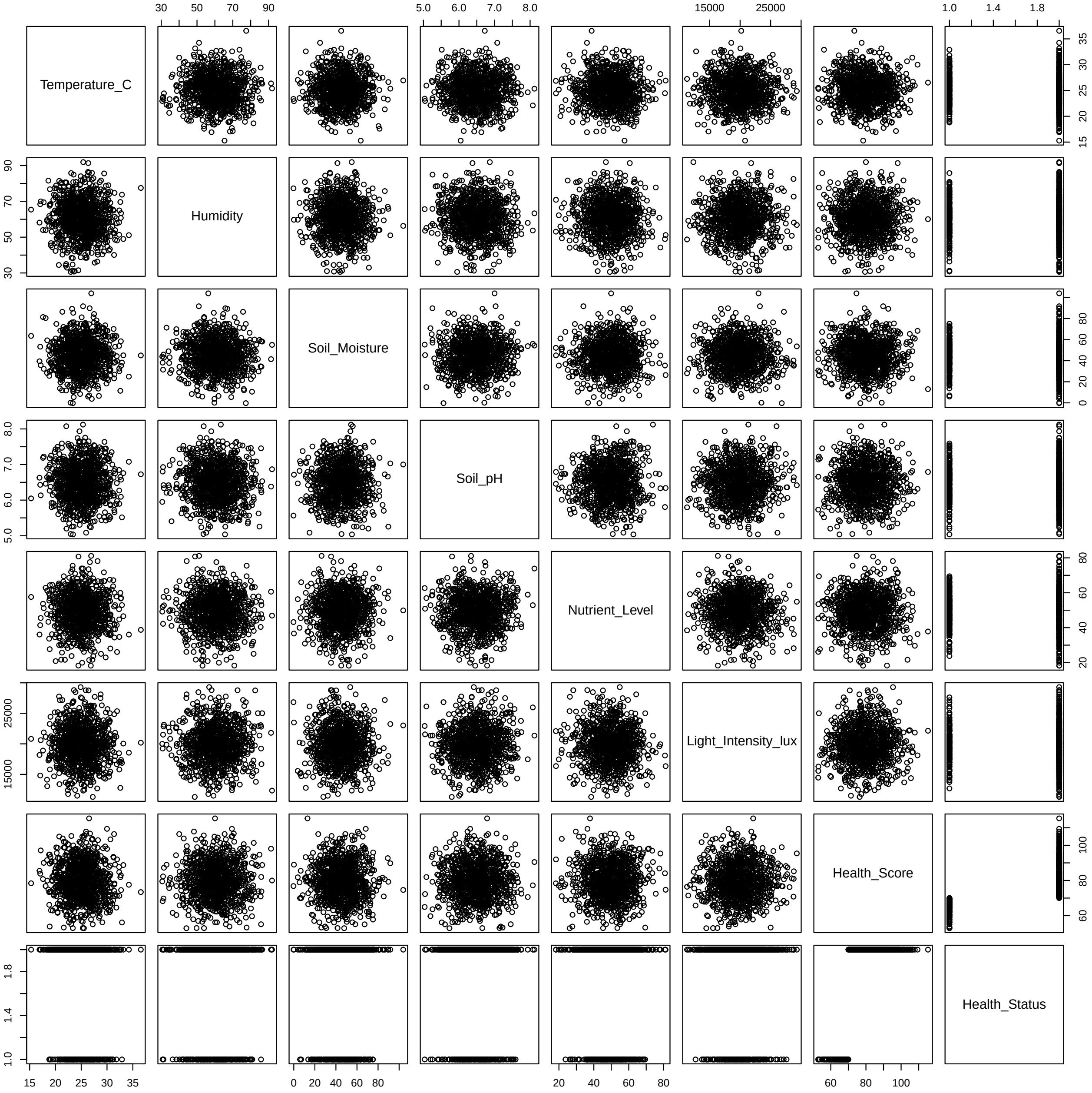
```
##   count
## 1  1000
```

Perfect, this field can be dropped. We also renamed the other variables for simplicity.

```
data <- data[,-c(1)]
data <- data %>%
  rename(
    Humidity = Humidity_.,
    Soil_Moisture = Soil_Moisture_.
    )
```

Looking at all the variables, we use a pair plot:

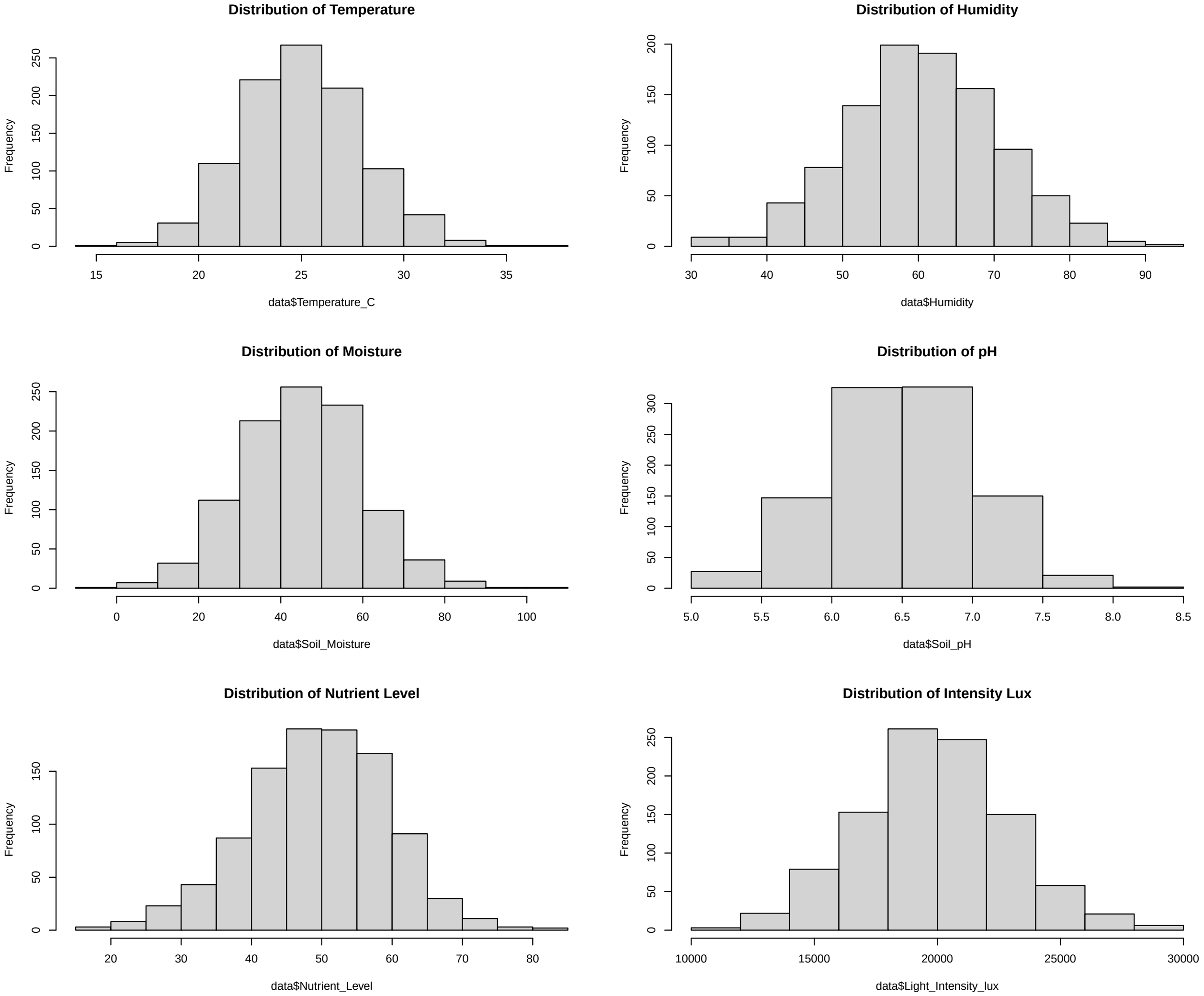
```
pairs(data)
```



These plots show that the variables are not correlated. If so, very weekly. The Health Score, looks like a continuous variable that is used to create the Health Status values.

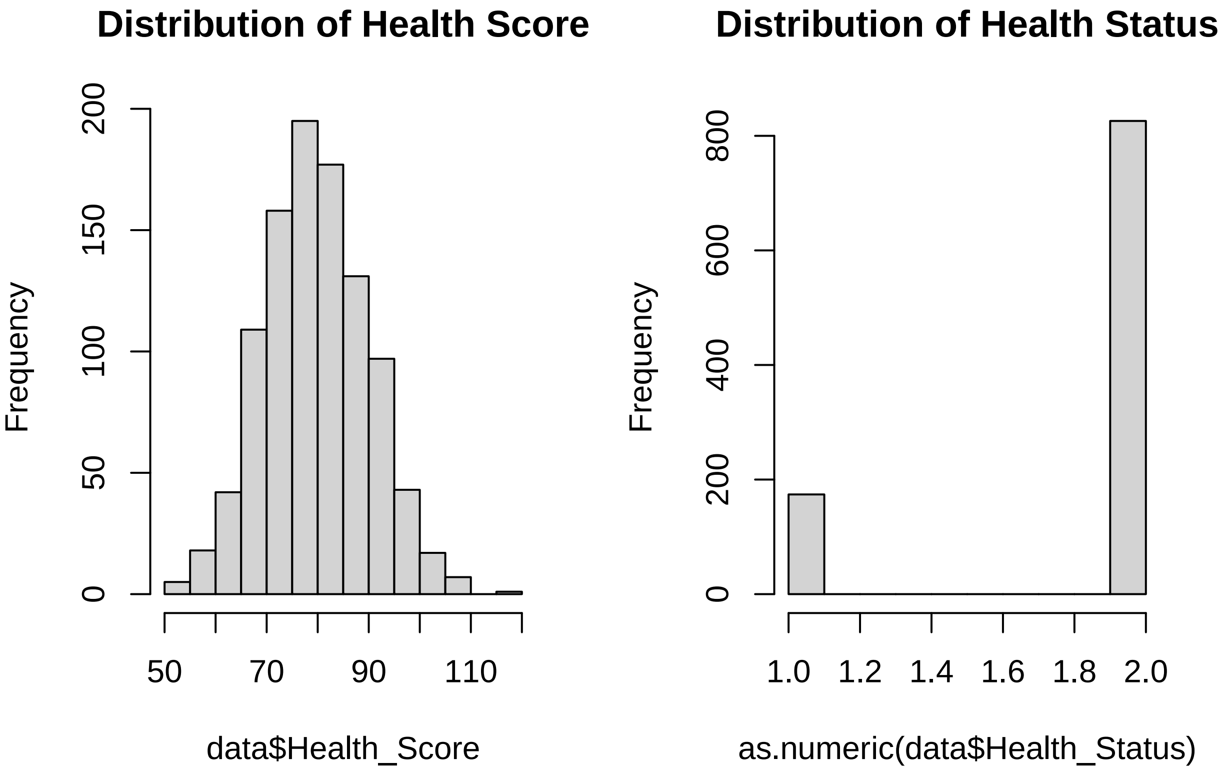
All variables look roughly normally distributed - we should confirm this:

```r
par(mfrow=c(3,2))

hist(data$Temperature_C, main="Distribution of Temperature")
hist(data$Humidity, main="Distribution of Humidity")
hist(data$Soil_Moisture, main="Distribution of Moisture")
hist(data$Soil_pH, main="Distribution of pH")
hist(data$Nutrient_Level, main="Distribution of Nutrient Level")
hist(data$Light_Intensity_lux, main="Distribution of Intensity Lux")
```



These do all look pretty close to a normal distribution. We need to also see the distribution of the two response variables. One of which is a categorical with 2 classes (0 and 1).

```r
par(mfrow=c(1,2))

hist(data$Health_Score, main = "Distribution of Health Score")
hist(as.numeric(data$Health_Status), main = "Distribution of Health Status")
```



Whilst the health score is normally distributed, the health status is very imbalanced - we will have to take this into consideration when we explore modeling plant health.

Modeling the health status

First we can try model the health status using all variables with logistic regression.

```
model <- glm(Health_Status ~ Temperature_C + Humidity + Soil_Moisture + Soil_pH + Nutrient_Level + Ligh
             family='binomial',
             data=data)

summary(model)
```

```
##
## Call:
## glm(formula = Health_Status ~ Temperature_C + Humidity + Soil_Moisture +
##     Soil_pH + Nutrient_Level + Light_Intensity_lux, family = "binomial",
##     data = data)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.200e+00  1.586e+00   1.387   0.1653
## Temperature_C       -5.000e-02  2.847e-02  -1.756   0.0791 .
## Humidity             5.100e-03  8.412e-03   0.606   0.5443
## Soil_Moisture       -6.615e-03  5.667e-03  -1.167   0.2431
## Soil_pH              5.156e-02  1.631e-01   0.316   0.7520
## Nutrient_Level       2.253e-03  8.447e-03   0.267   0.7896
## Light_Intensity_lux  8.306e-06  2.781e-05   0.299   0.7651
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 924.34  on 999  degrees of freedom
## Residual deviance: 919.08  on 993  degrees of freedom
## AIC: 933.08
##
## Number of Fisher Scoring iterations: 4
```

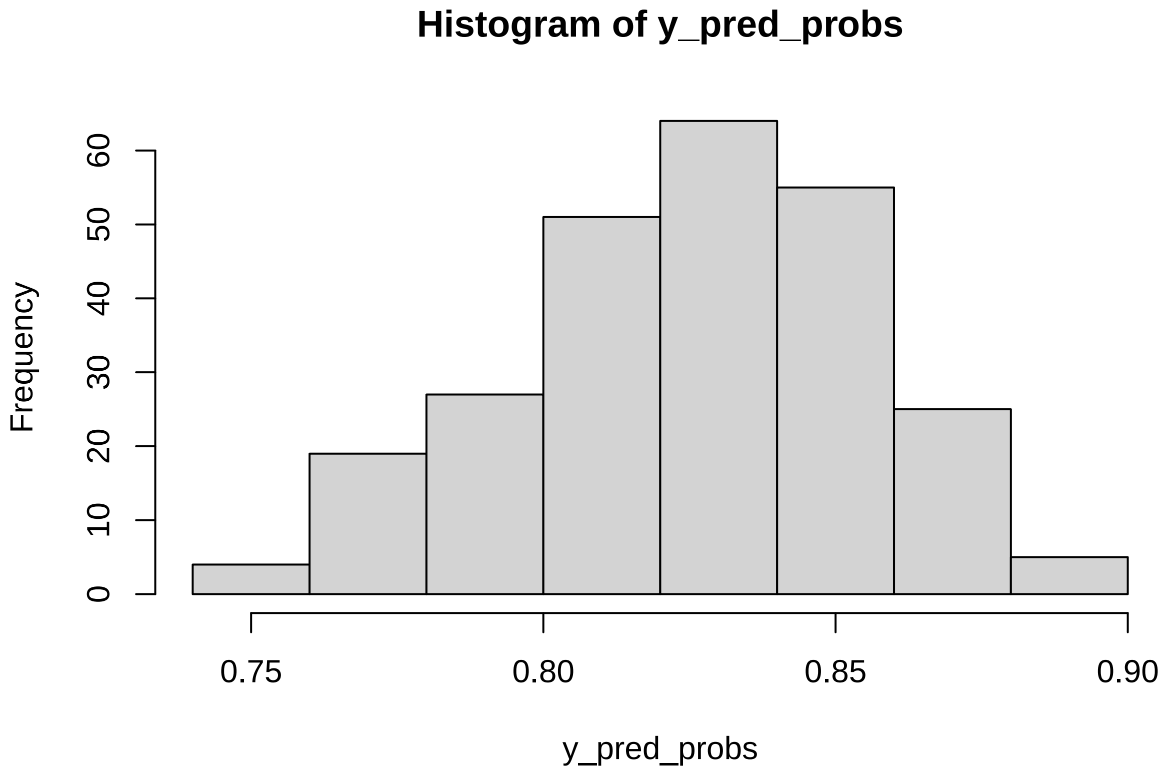
This model does not find any of the variables to be statistically significant (except temperature). We should run this model with a train test split to see how well it predicts the health score.

```
train_size <- 0.75*nrow(data)
training_sample <- sample(1:nrow(data), train_size)
training_dataset <- data[training_sample, ]
testing_dataset <- data[-training_sample, ]

model_validate <- glm(
  Health_Status ~ Temperature_C + Humidity + Soil_Moisture + Soil_pH + Nutrient_Level + Light_Intensity
  data=training_dataset,
  family='binomial'
)

y_pred_probs <- predict(
  model_validate,
  newdata=testing_dataset,
  type='response'
)

y_pred <- ifelse(y_pred_probs > 0.5, 1, 0)

table(y_pred, testing_dataset$Health_Status)
```

```
##
## y_pred   0   1
##      1  43 207
```

Unsurprisingly its predicting the 1 every time. This could be due to the unbalanced nature of the response variable. The distribution of the predicted values might be interesting:

```
hist(y_pred_probs)
```


Histogram of y_pred_probs
Frequency
0
10
20
30
40
50
60
0.75
0.80
0.85
0.90
y_pred_probs