# 20F-0292

BCS 5E

PROBABILITY AND STATISTICS PROJECT

- **Dataset Name: Crop Recommendation**
- **Dataset Link:** https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset
- **Tools Used: R Programming Language, RStudio, R Markdown, RShiny, flexdashboard**

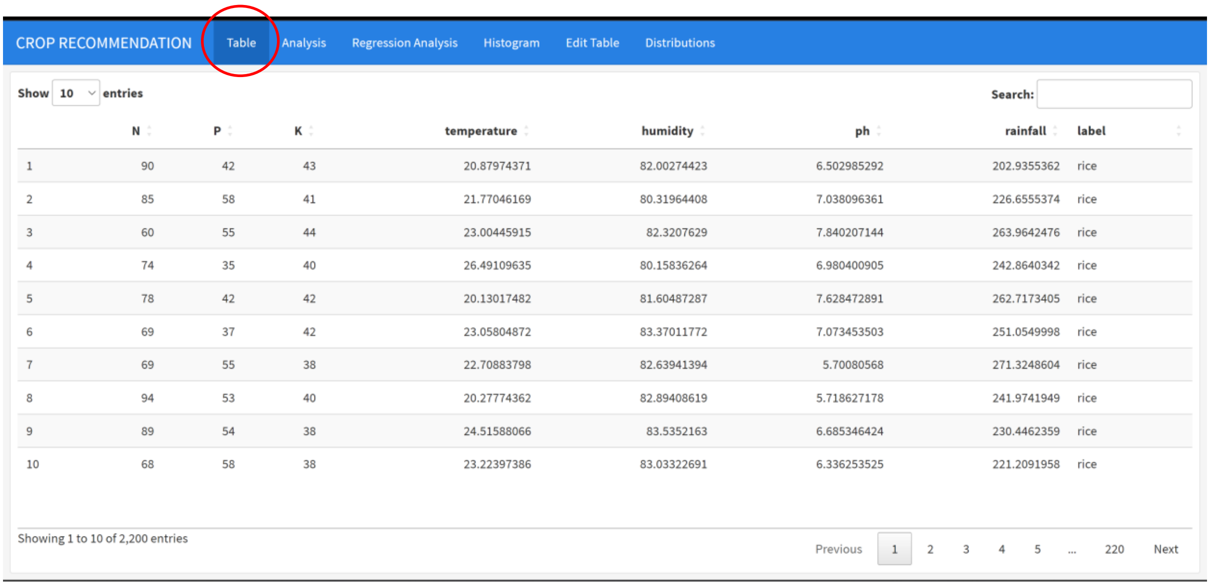
## CODES:

1) Declaration and making of simple dashboard interface

~~~
---

title: "CROP RECOMMENDATION"

output:

  flexdashboard::flex_dashboard

runtime: shiny

---
~~~

2) Importing our excel file into RMarkdown and Displaying Table in Table section

~~~
# Table

```{r}

library(DT)

startData <- read.csv("crop.csv")

DT::datatable(startData)

```
~~~



| | N | P | K | temperature | humidity | ph | rainfall | label |
|---|---|---|---|---|---|---|---|---|
| 1 | 90 | 42 | 43 | 20.87974371 | 82.00274423 | 6.502985292 | 202.9355362 | rice |
| 2 | 85 | 58 | 41 | 21.77046169 | 80.31964408 | 7.038096361 | 226.6555374 | rice |
| 3 | 60 | 55 | 44 | 23.00445915 | 82.3207629 | 7.840207144 | 263.9642476 | rice |
| 4 | 74 | 35 | 40 | 26.49109635 | 80.15836264 | 6.980400905 | 242.8640342 | rice |
| 5 | 78 | 42 | 42 | 20.13017482 | 81.60487287 | 7.628472891 | 262.7173405 | rice |
| 6 | 69 | 37 | 42 | 23.05804872 | 83.37011772 | 7.073453503 | 251.0549998 | rice |
| 7 | 69 | 55 | 38 | 22.70883798 | 82.63941394 | 5.70080568 | 271.3248604 | rice |
| 8 | 94 | 53 | 40 | 20.27774362 | 82.89408619 | 5.718627178 | 241.9741949 | rice |
| 9 | 89 | 54 | 38 | 24.51588066 | 83.5352163 | 6.685346424 | 230.4462359 | rice |
| 10 | 68 | 58 | 38 | 23.22397386 | 83.03322691 | 6.336253525 | 221.2091958 | rice |

Showing 1 to 10 of 2,200 entries

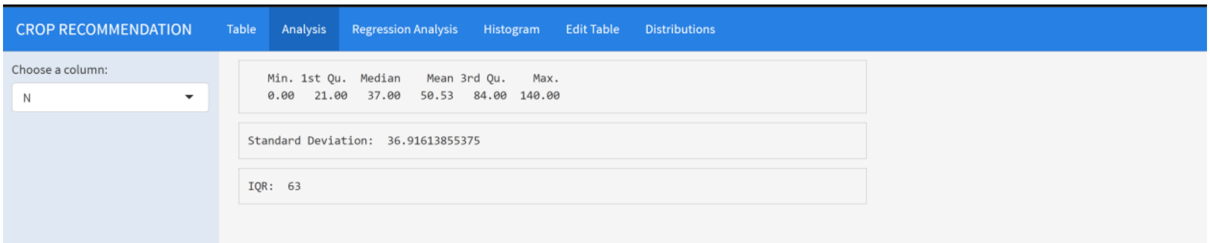
*Figure 1 Table Interface*

You can also filter the searches for the ones you best want

3) Doing statistical analysis on whole data in Analysis section

~~~
# Analysis

## Column {.sidebar}

```{r}


 output$summary1 <- renderPrint({
  if(input$datalist=="All"){
   myVec2=c()
   val <- 2
   while(val <= row_num){
    myVec2 <- append(myVec2,mean(t(data_origin[val,])))
    val<-val+1
   }
   summary(myVec2)
  }
  else{
   label <- input$datalist
   rows <- length(t(data_origin[,2]))
   summary(data_origin[2:rows,label])
  }
 })


 output$standD <- renderText({
  if(input$datalist=="All"){
   myVec2=c()
   val <- 2
   while(val <= row_num){
    myVec2 <- append(myVec2,mean(t(data_origin[val,])))
    val<-val+1
   }
   c("Standard Deviation: ",sd(myVec2))
~~~

~~~
    }
    else{
      label <-input$datalist
      rows <- length(t(data_origin[,2]))
      c("Standard Deviation: ",sd(data_origin[2:rows,label]))
    }
  })

output$IQR <- renderText({
    if(input$datalist=="All"){
      myVec2=c()
      val <- 2
      while(val <= row_num){
        myVec2 <- append(myVec2,mean(t(data_origin[val,])))
        val<-val+1
      }
      c("IQR: ",IQR(myVec2))
    }
    else{
      label <-input$datalist
      rows <- length(t(data_origin[,2]))
      c("IQR: ",IQR(data_origin[2:rows,label]))
    }
  })
data <- read.csv("crop.csv")
data_origin <- subset(data, select = -c(8,8))
selectInput("datalist", "Choose a column:", c(rownames(t(data_origin))))
```
## Column

```{r}
column(8, verbatimTextOutput("minval"))
column(8, verbatimTextOutput("maxval"))
column(8, verbatimTextOutput("summary1"))
column(8, verbatimTextOutput("standD"))
column(8, verbatimTextOutput("IQR"))
```
~~~

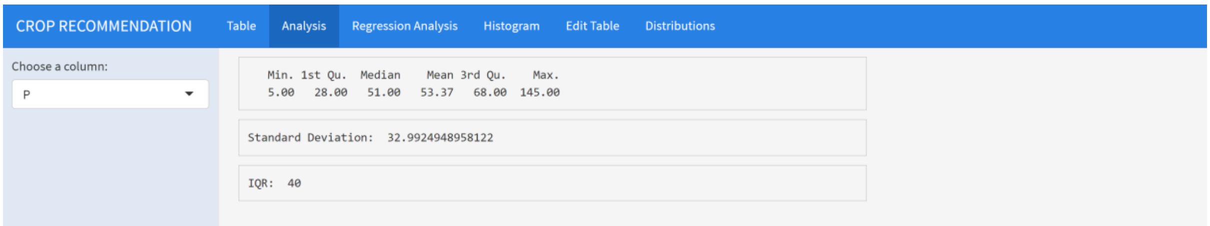


```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00   21.00   37.00   50.53   84.00  140.00 
```

```
Standard Deviation:  36.91613855375
```

```
IQR:  63
```

*Figure 2 Analytics done on Nitrogen*

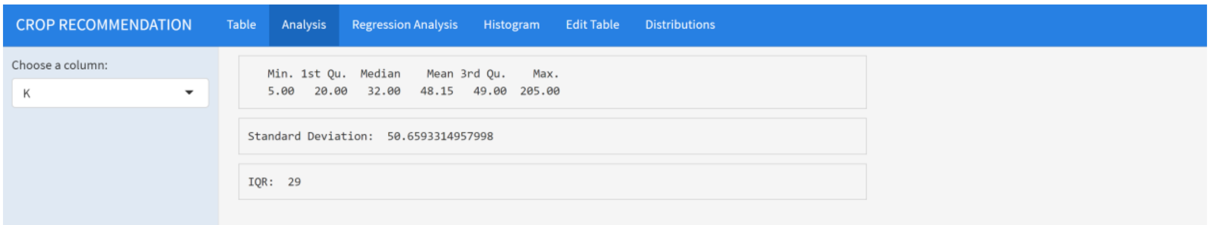


```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   5.00   28.00   51.00   53.37   68.00  145.00 
```

```
Standard Deviation:  32.9924948958122
```

```
IQR:  40
```

*Figure 3 Analytics done for Phosphorus*

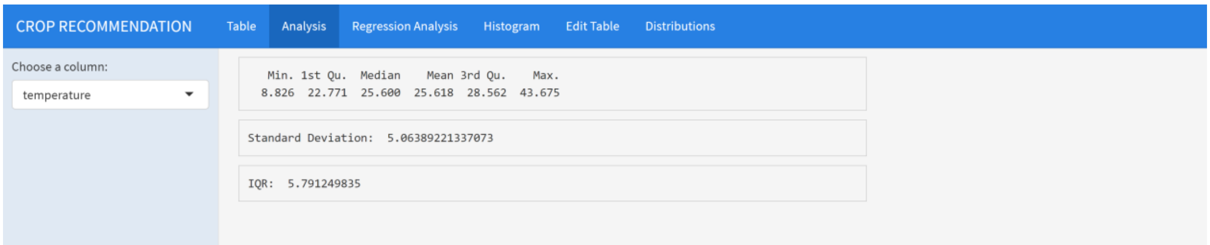


```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   5.00   20.00   32.00   48.15   49.00  205.00 
```

```
Standard Deviation:  50.6593314957998
```

```
IQR:  29
```

*Figure 4 Analysis done for Potassium*

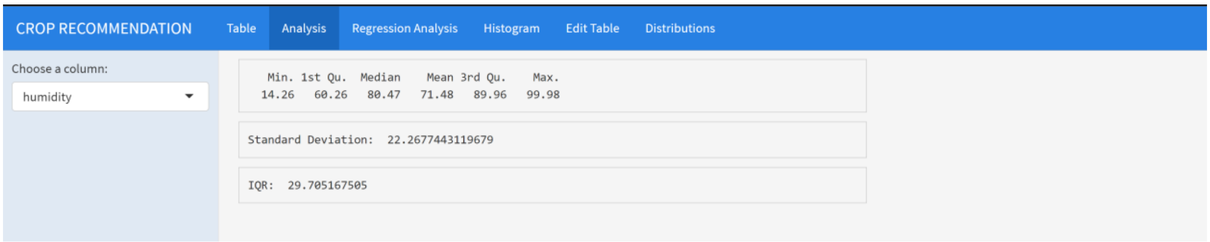


```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  8.826  22.771  25.600  25.618  28.562  43.675 
```

```
Standard Deviation:  5.06389221337073
```

```
IQR:  5.791249835
```

*Figure 5 Analysis Done for temperature*

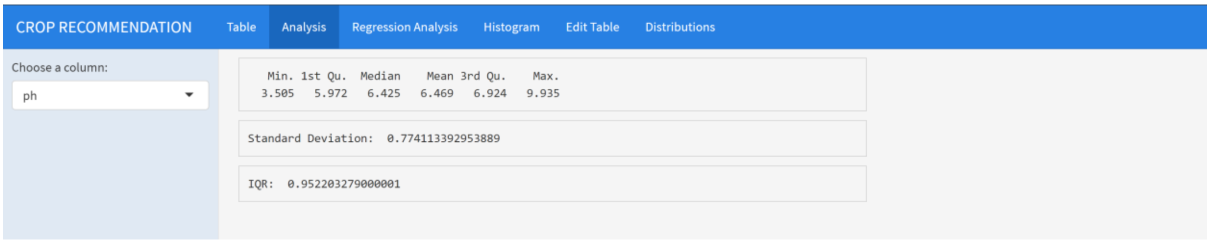


```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  14.26   60.26   80.47   71.48   89.96   99.98 
```

```
Standard Deviation:  22.2677443119679
```

```
IQR:  29.705167505
```

*Figure 6 Analysis done for humidity*



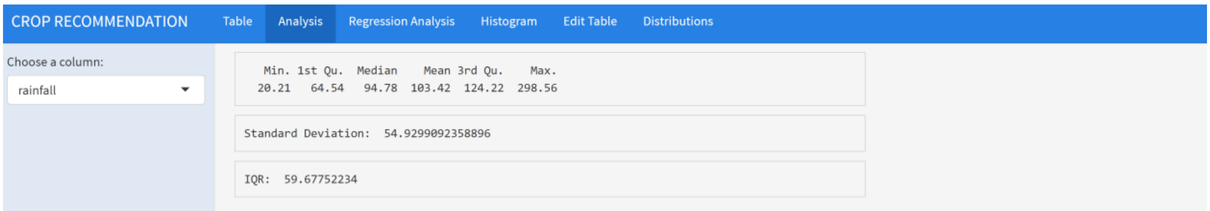
Choose a column: ph

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 3.505   5.972   6.425   6.469   6.924   9.935 
```

```
Standard Deviation:  0.774113392953889
```

```
IQR:  0.952203279000001
```

*Figure 7 Analysis done for ph*



Choose a column: rainfall

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 20.21   64.54   94.78  103.42  124.22  298.56 
```

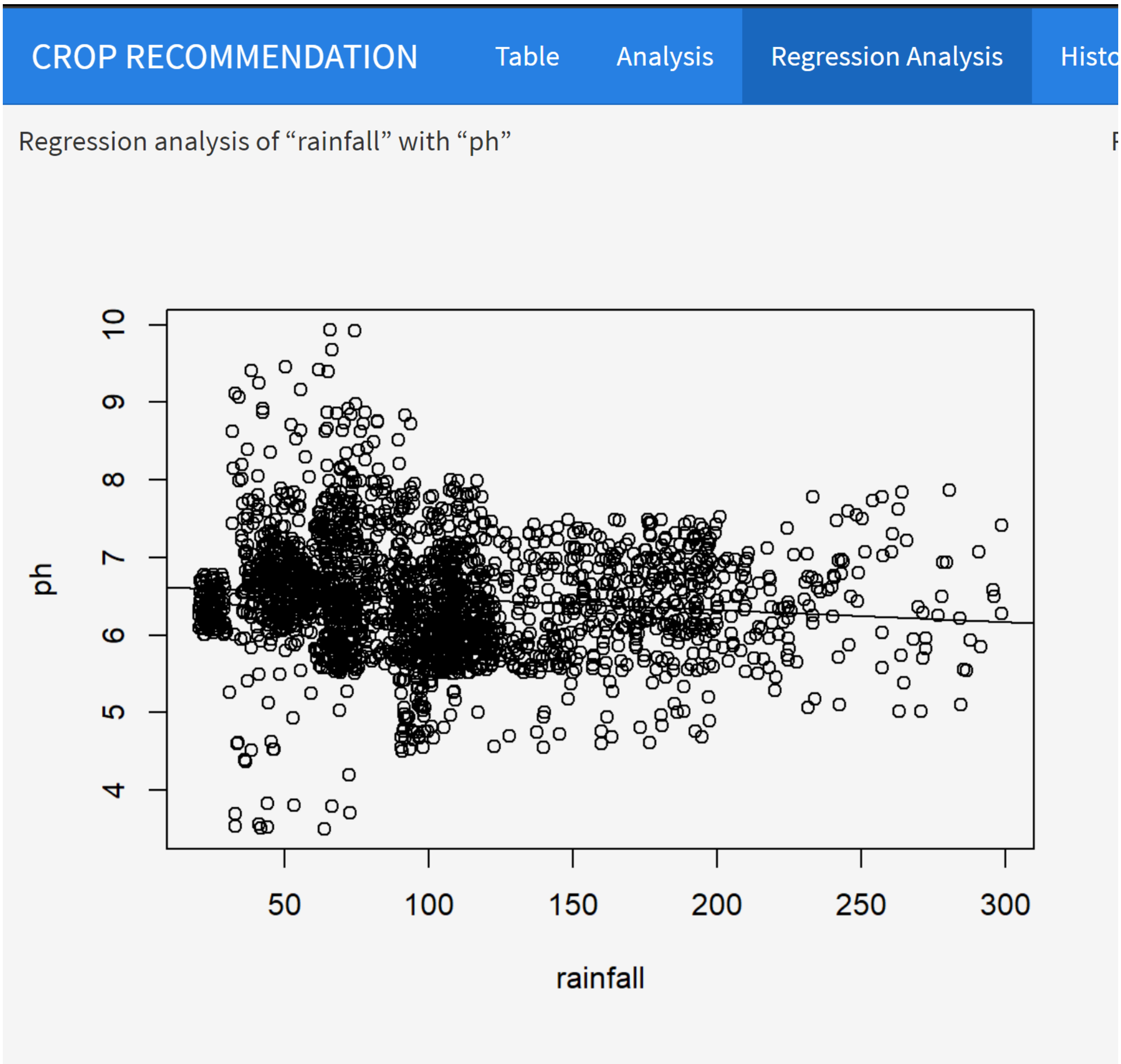
```
Standard Deviation:  54.9299092358896
```

```
IQR:  59.67752234
```

*Figure 8 Analysis Done for rainfall*

## 4) Regression Analysis on Each Data

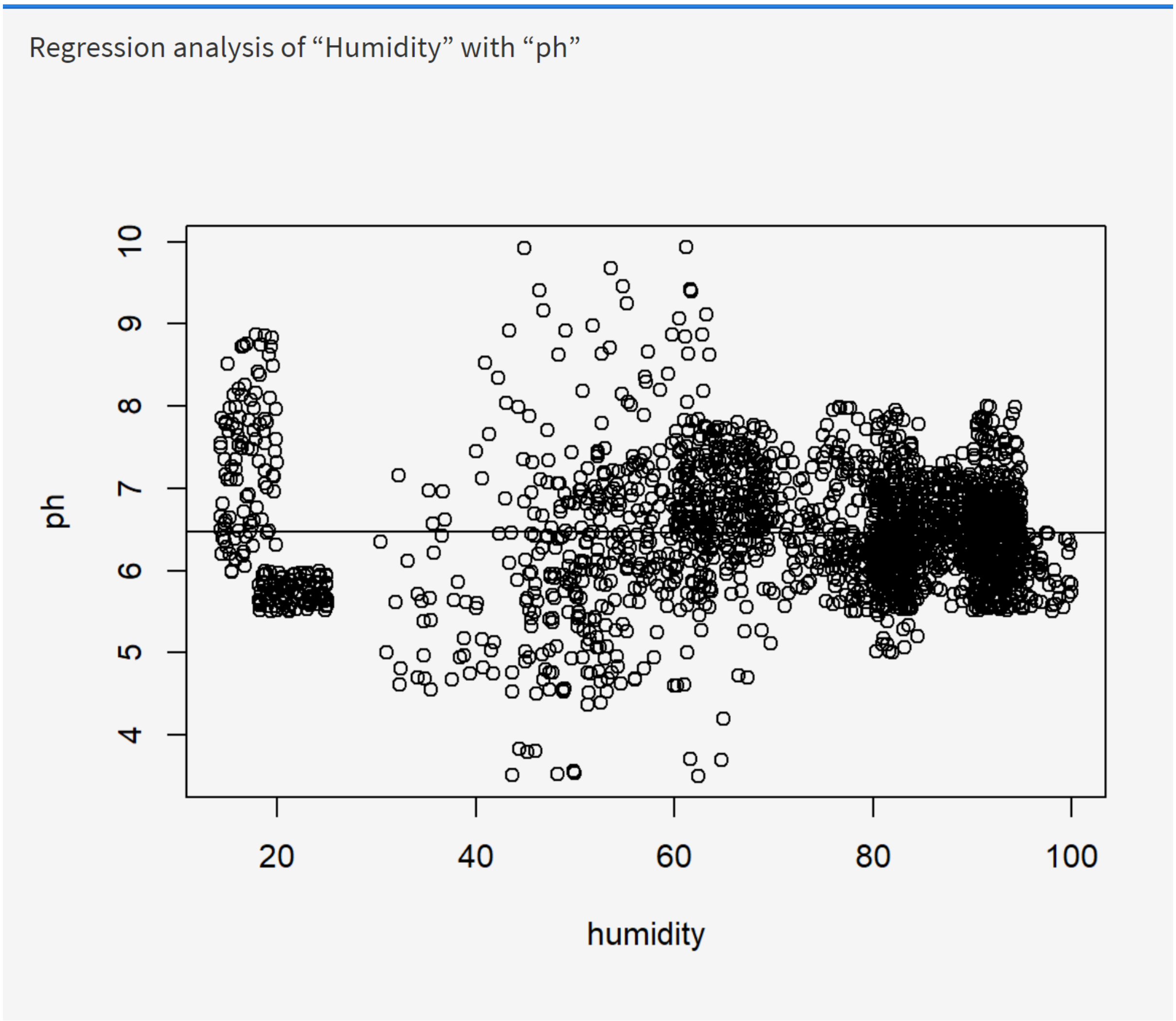
~~~
# Regression Analysis

## ph~Rainfall

Regression analysis of "rainfall" with "ph"

```{r}


library(datasets)

mydata=as.data.frame(read.csv("crop.csv"))

colnames(mydata)=c("N", "P", "K", "temperature", "humidity", "ph", "rainfall")

model1=lm(ph~rainfall, data=mydata)

plot(ph~rainfall, data=mydata,

     xlab="rainfall", ylab="ph")

abline(model1)

```
~~~



Regression analysis of “rainfall” with “ph”

*Figure 9 Rainfall ph regression analysis*

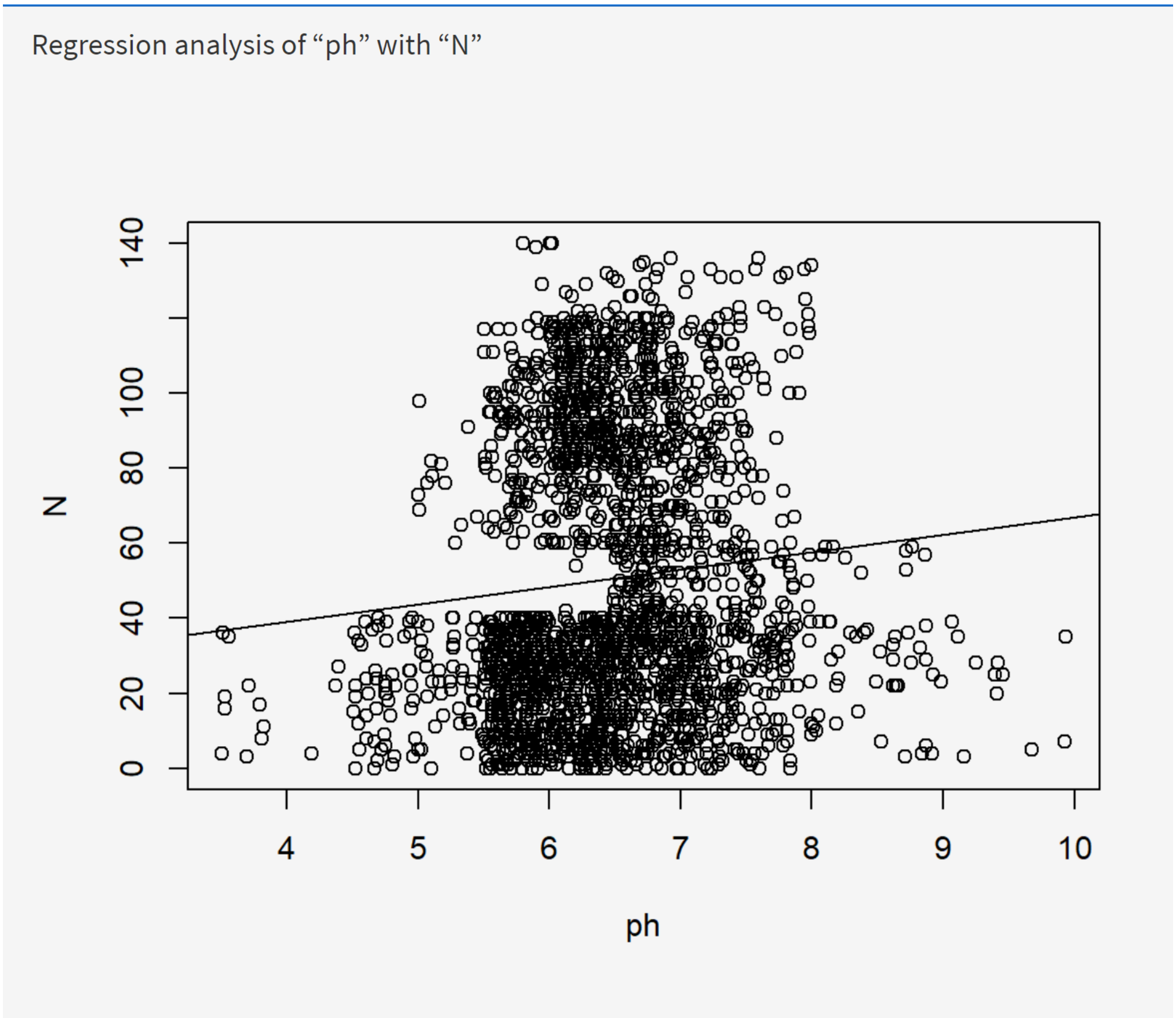
~~~
## ph~humidity

Regression analysis of "Humidity" with "ph"

```{r}


library(datasets)

mydata=as.data.frame(read.csv("crop.csv"))

colnames(mydata)=c("N", "P", "K", "temperature", "humidity", "ph", "rainfall")

model1=lm(ph~humidity, data=mydata)

plot(ph~humidity, data=mydata,

     xlab="humidity", ylab="ph")

abline(model1)

```
~~~



Regression analysis of “Humidity” with “ph”

*Figure 10 Regression analysis for ph and humidity*

~~~
## N~ph

Regression analysis of "ph" with "N"

```{r}


library(datasets)

mydata=as.data.frame(read.csv("crop.csv"))

colnames(mydata)=c("N", "P", "K", "temperature", "humidity", "ph", "rainfall")

model1=lm(N~ph, data=mydata)

plot(N~ph, data=mydata,

     xlab="ph", ylab="N")

abline(model1)

```
~~~



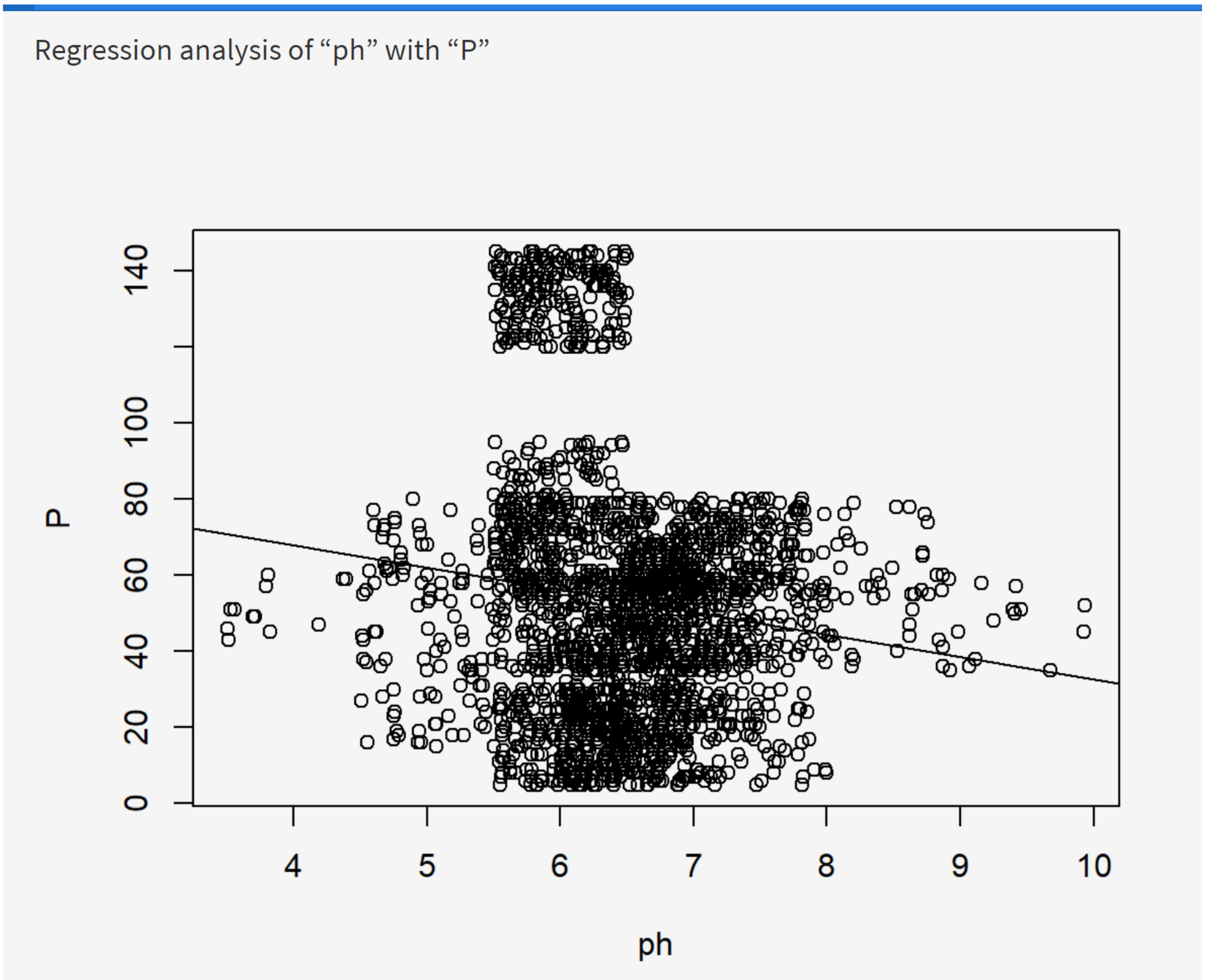
Regression analysis of “ph” with “N”

*Figure 11 Regression analysis for ph and N*

~~~~
## P~ph

Regression analysis of "ph" with "P"

```{r}


library(datasets)

mydata=as.data.frame(read.csv("crop.csv"))

colnames(mydata)=c("N", "P", "K", "temperature", "humidity", "ph", "rainfall")

model1=lm(P~ph, data=mydata)

plot(P~ph, data=mydata,

    xlab="ph", ylab="P")

abline(model1)

```
~~~~



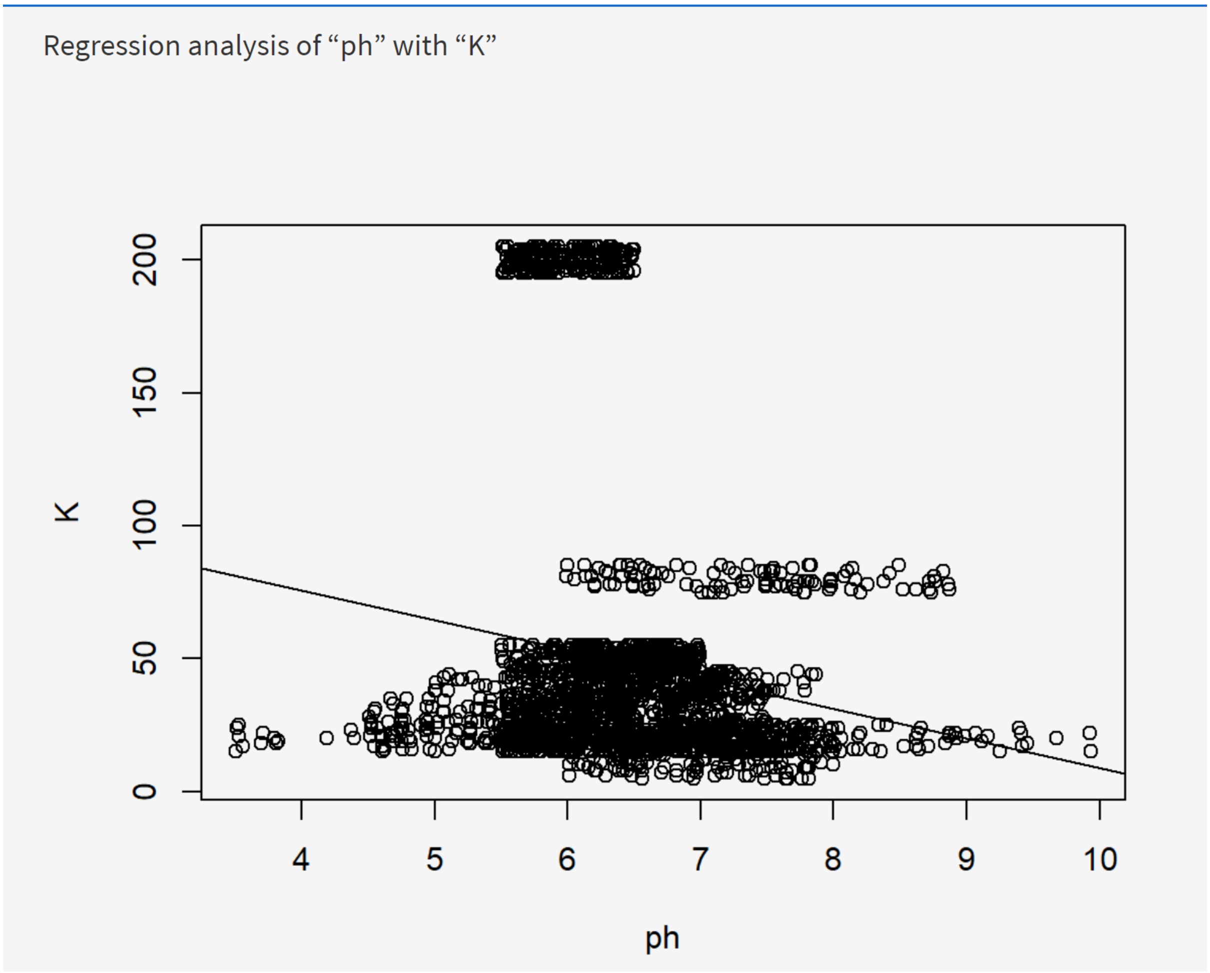
Regression analysis of “ph” with “P”

*Figure 12 Regression analysis for ph and P*

~~~~
## K~ph

Regression analysis of "ph" with "K"

```{r}


library(datasets)

mydata=as.data.frame(read.csv("crop.csv"))

colnames(mydata)=c("N", "P", "K", "temperature", "humidity", "ph", "rainfall")

model1=lm(K~ph, data=mydata)

plot(K~ph, data=mydata,

    xlab="ph", ylab="K")

abline(model1)

```
~~~~



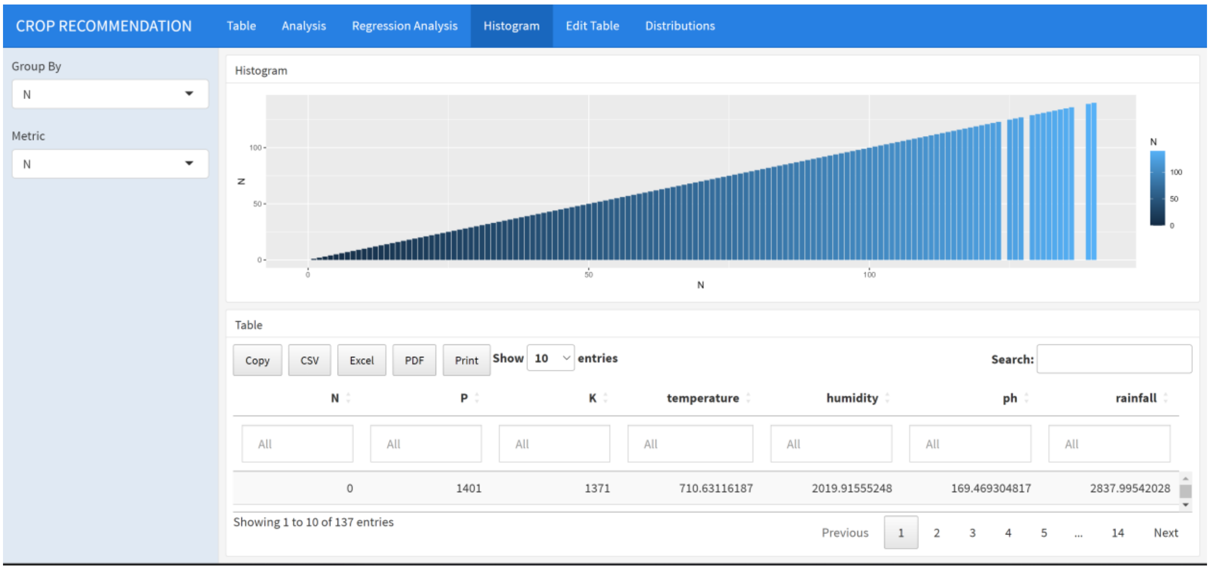
Regression analysis of “ph” with “K”

*Figure 13 Regression Analysis for ph and K*

5) Histogram on Each Data (Most effort put on this one)

~~~
# Histogram

```{r setup, include=FALSE}

library(tidyverse)

library(DT)

startData <- read.csv("crop.csv")

updateData <- reactive(

startData %>% group_by(!!! rlang::syms(input$GB)) %>%

summarise_if(is.numeric,sum,na.rm=T))

```

## Column {.sidebar}
~~~

~~~
```{r}
selectInput(inputId = "GB",label= "Group By", choices = names(startData))
selectInput(inputId = "Metric",label= "Metric", choices =
names (select_if(startData, is.numeric)))
```

## Column

### Histogram

```{r}
renderPlot({
  updateData()%>%
    ggplot(aes(x=!! rlang::sym(input$GB),y=!! rlang::sym(input$Metric),fill=!!
rlang::sym(input$GB))) +
    geom_col()
})
```

### Table

```{r}
renderDT(
  updateData(), rownames = F, extensions = 'Buttons', filter="top", editable=T,
  options = list(
    dom = 'Blfrtip',
    buttons = c('copy', 'csv', 'excel', 'pdf', 'print'),
    lengthMenu = list(c(10,50, 100, -1), c(10,50, 100, "All"))
  )
)
```
~~~



Group By: N

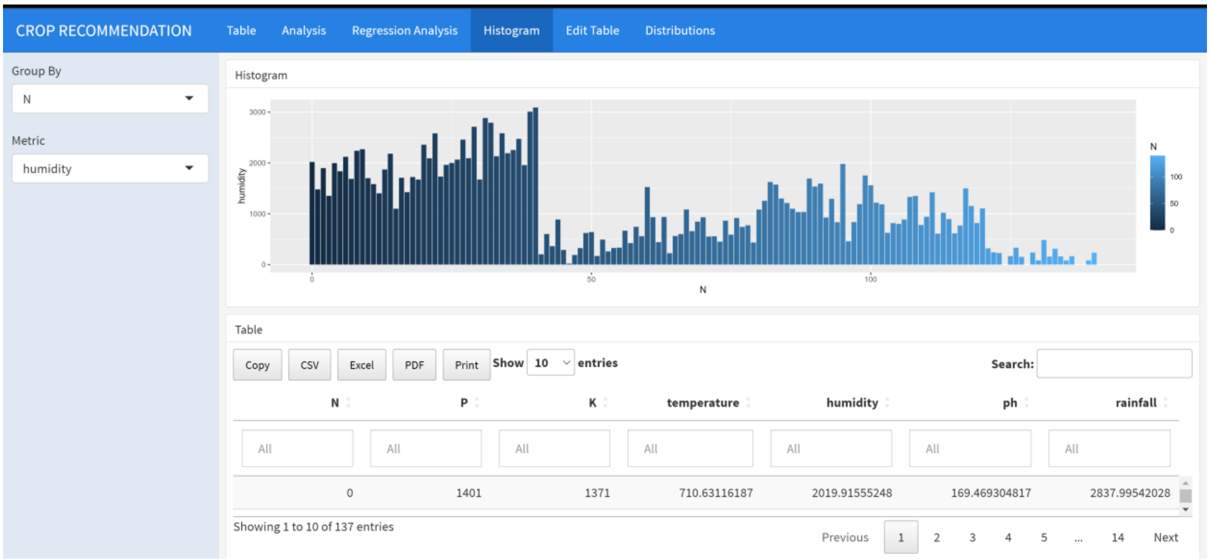
Metric: N

Show 10 entries

| N | P | K | temperature | humidity | ph | rainfall |
|---|---|---|---|---|---|---|
| All | All | All | All | All | All | All |
| 0 | 1401 | 1371 | 710.63116187 | 2019.91555248 | 169.469304817 | 2837.99542028 |

Showing 1 to 10 of 137 entries

*Figure 14 Histogram for N and N*



Group By: N

Metric: humidity

Show 10 entries

| N | P | K | temperature | humidity | ph | rainfall |
|---|---|---|---|---|---|---|
| All | All | All | All | All | All | All |
| 0 | 1401 | 1371 | 710.63116187 | 2019.91555248 | 169.469304817 | 2837.99542028 |

Showing 1 to 10 of 137 entries

*Figure 15 Histogram For N and Humidity*

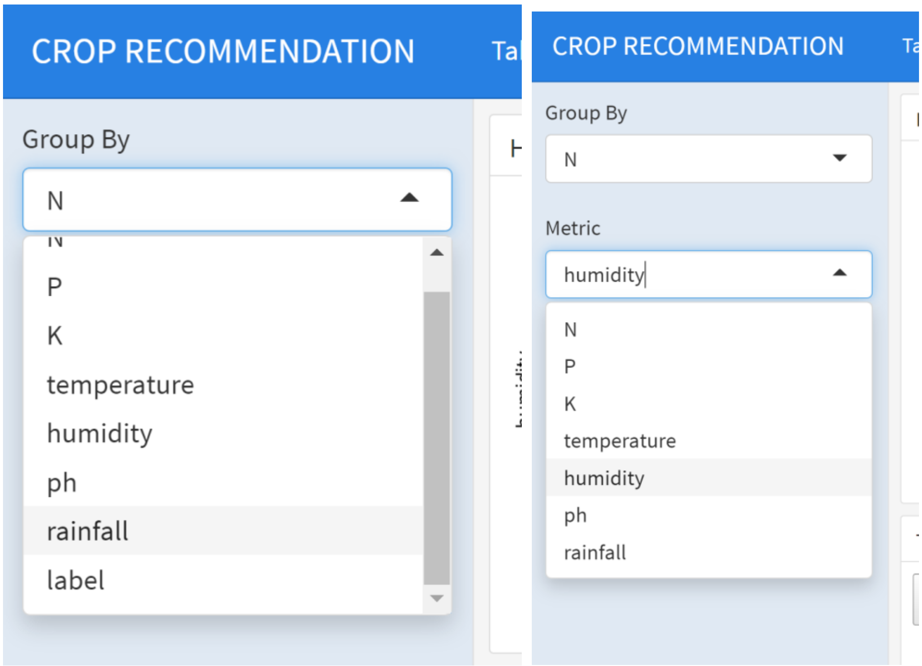
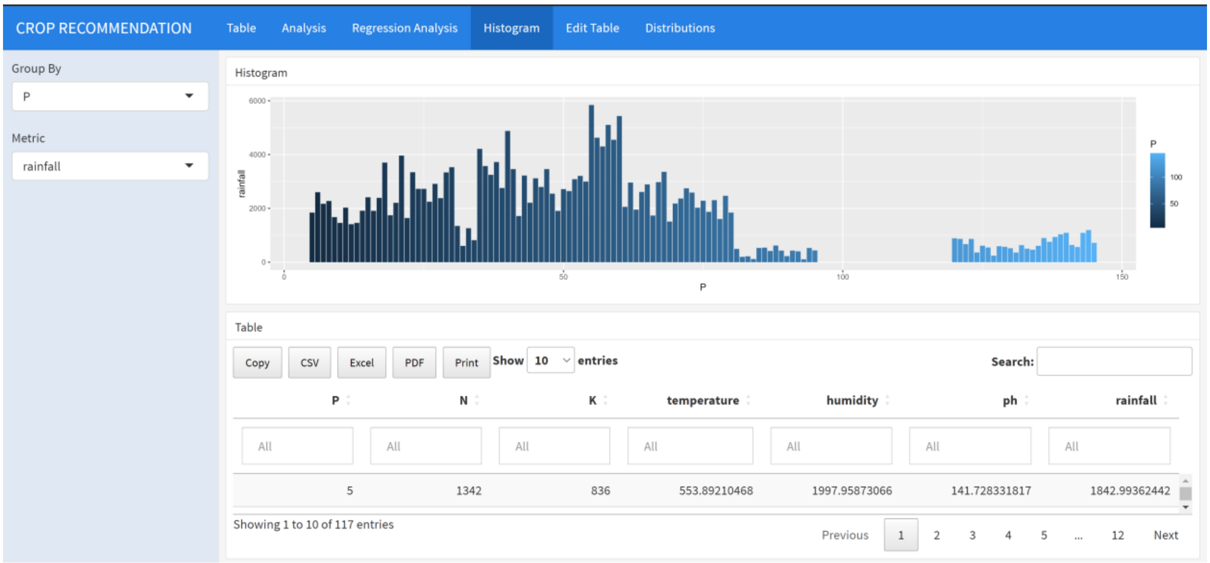


*Figure 16 Special thing about this Histogram is that it is dynamic and you can choose metric and group by by your self on which you want to apply methods*



| P | N | K | temperature | humidity | ph | rainfall |
|---|---|---|---|---|---|---|
| All | All | All | All | All | All | All |
| 5 | 1342 | 836 | 553.89210468 | 1997.95873066 | 141.728331817 | 1842.99362442 |

Showing 1 to 10 of 117 entries

*Figure 17 Histogram of P and Rainfall*

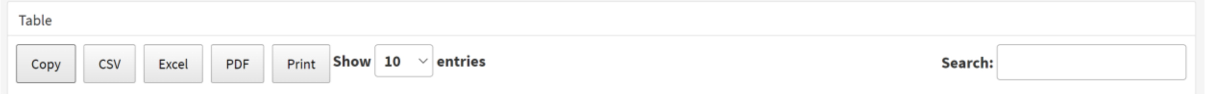
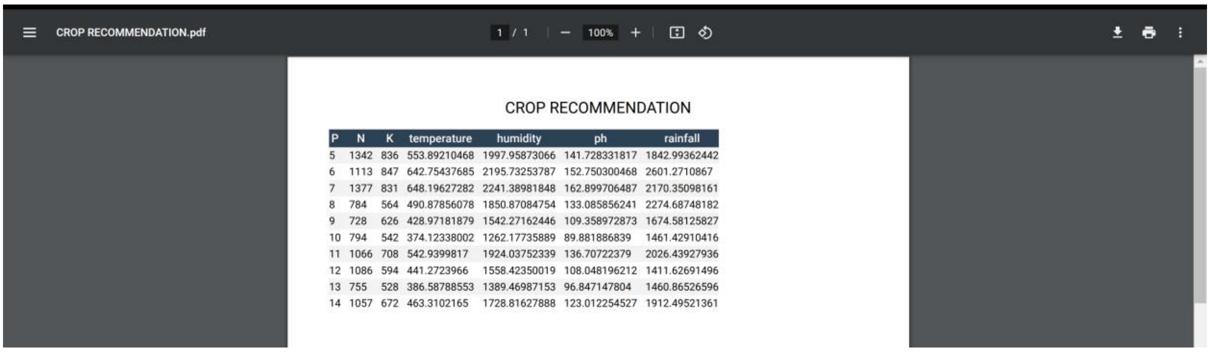


*Figure 18 You can also print data to your favourite form*



CROP RECOMMENDATION

| P | N | K | temperature | humidity | ph | rainfall |
|---|---|---|---|---|---|---|
| 5 | 1342 | 836 | 553.89210468 | 1997.95873066 | 141.728331817 | 1842.99362442 |
| 6 | 1113 | 847 | 642.75437685 | 2195.73253787 | 152.750300468 | 2601.2710867 |
| 7 | 1377 | 831 | 648.19627282 | 2241.38981848 | 162.899706487 | 2170.35098161 |
| 8 | 784 | 564 | 490.87856078 | 1850.87084754 | 133.085856241 | 2274.68748182 |
| 9 | 728 | 626 | 428.97181879 | 1542.27162446 | 109.358972873 | 1674.58125827 |
| 10 | 794 | 542 | 374.12338002 | 1262.17735889 | 89.881886839 | 1461.42910416 |
| 11 | 1066 | 708 | 542.9399817 | 1924.03752339 | 136.70722379 | 2026.43927936 |
| 12 | 1086 | 594 | 441.2723966 | 1558.42350019 | 108.048196212 | 1411.62691496 |
| 13 | 755 | 528 | 386.58788553 | 1389.46987153 | 96.847147804 | 1460.86526596 |
| 14 | 1057 | 672 | 463.3102165 | 1728.81627888 | 123.012254527 | 1912.49521361 |

*Figure 19 Pdf of 10 datas*

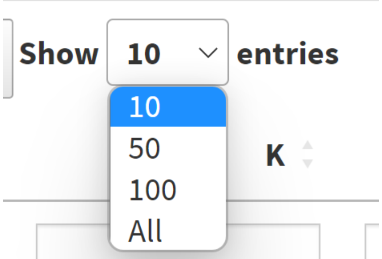
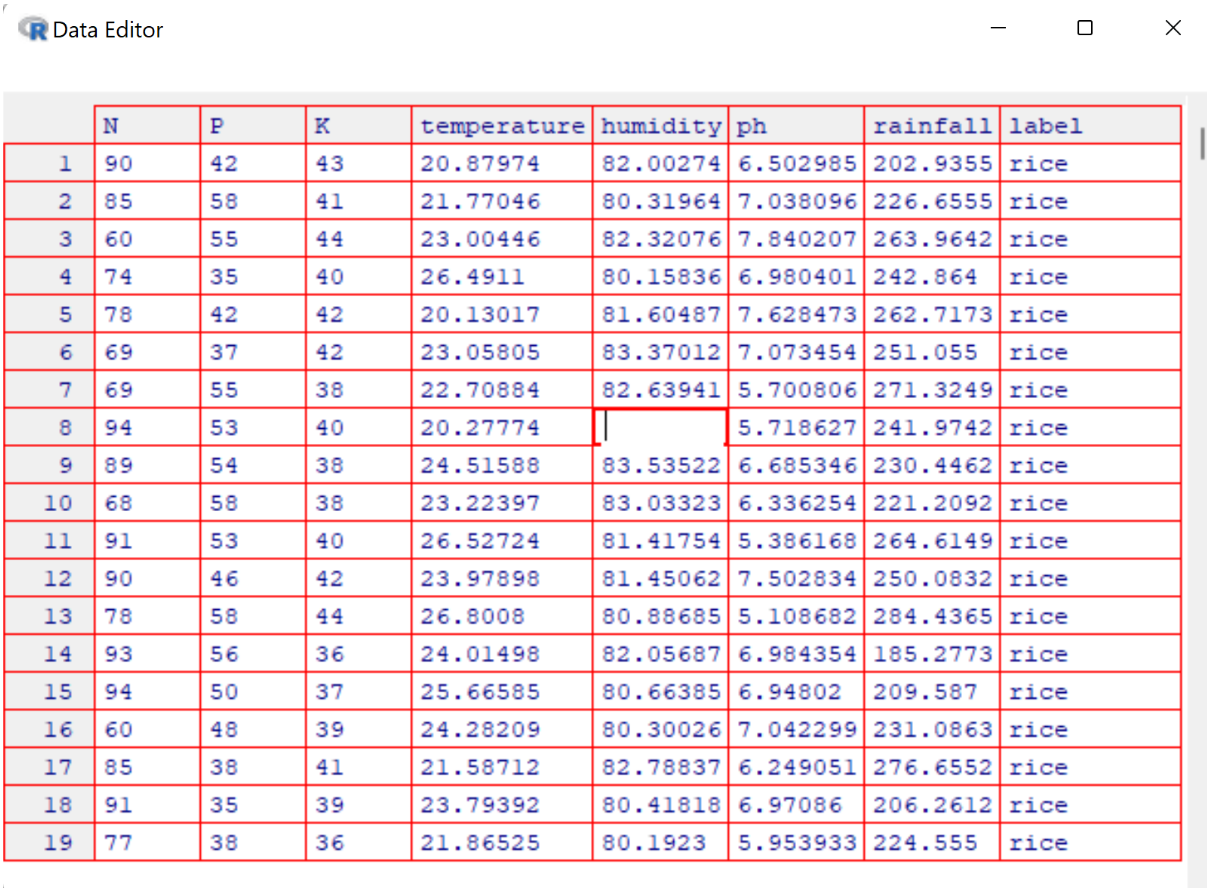


*Figure 20 Also you can choose how much data you want to print in which format*

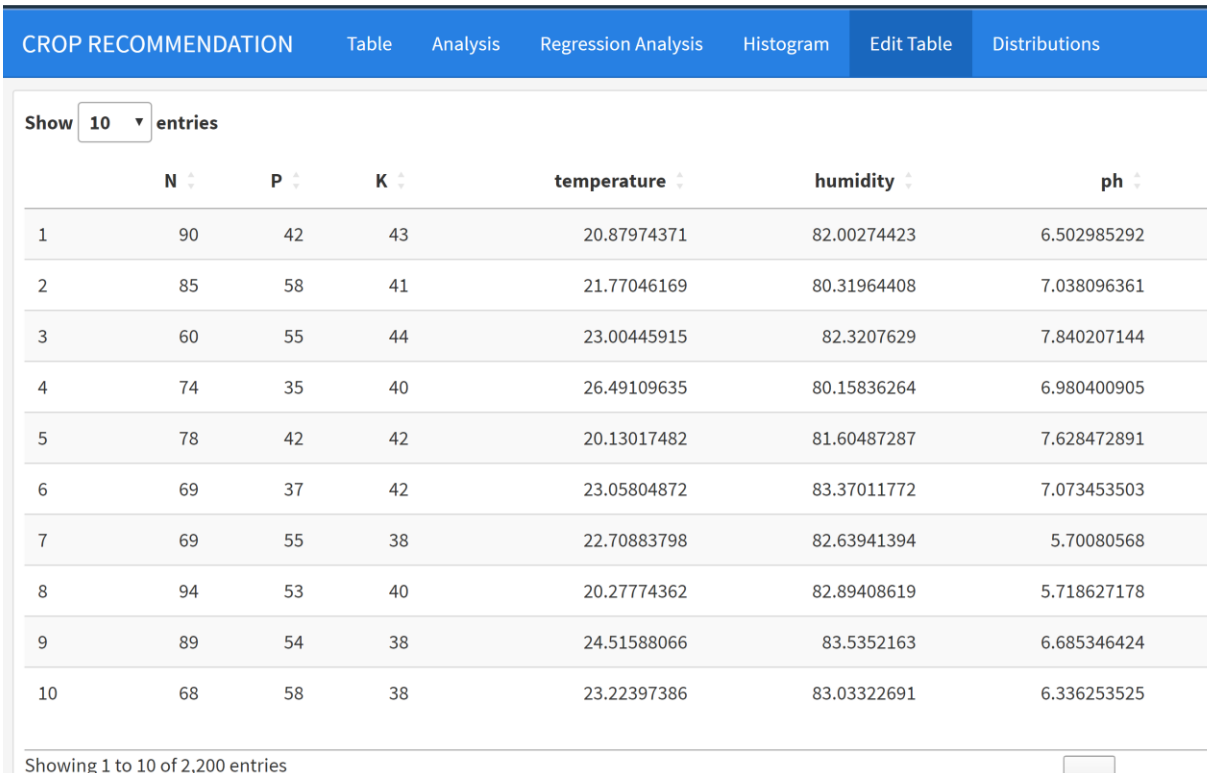
## 6) Editing the data

~~~
# Edit Table



```{r}
library(DT)
startData <- read.csv("crop.csv", header = TRUE, sep = ",")
DT::datatable(startData)
var1 <- renderDT(
  updateData(), rownames = F, extensions = 'Buttons', filter="top", editable=T,
  options = list(
    dom = 'Blfrtip',
    buttons = c('Insert', 'Delete'),
    lengthMenu = list(c(10,50), c(10,50))
  )
)
```
~~~



|  | N | P | K | temperature | humidity | ph | rainfall | label |
|---|---|---|---|---|---|---|---|---|
| 1 | 90 | 42 | 43 | 20.87974 | 82.00274 | 6.502985 | 202.9355 | rice |
| 2 | 85 | 58 | 41 | 21.77046 | 80.31964 | 7.038096 | 226.6555 | rice |
| 3 | 60 | 55 | 44 | 23.00446 | 82.32076 | 7.840207 | 263.9642 | rice |
| 4 | 74 | 35 | 40 | 26.4911 | 80.15836 | 6.980401 | 242.864 | rice |
| 5 | 78 | 42 | 42 | 20.13017 | 81.60487 | 7.628473 | 262.7173 | rice |
| 6 | 69 | 37 | 42 | 23.05805 | 83.37012 | 7.073454 | 251.055 | rice |
| 7 | 69 | 55 | 38 | 22.70884 | 82.63941 | 5.700806 | 271.3249 | rice |
| 8 | 94 | 53 | 40 | 20.27774 |  | 5.718627 | 241.9742 | rice |
| 9 | 89 | 54 | 38 | 24.51588 | 83.53522 | 6.685346 | 230.4462 | rice |
| 10 | 68 | 58 | 38 | 23.22397 | 83.03323 | 6.336254 | 221.2092 | rice |
| 11 | 91 | 53 | 40 | 26.52724 | 81.41754 | 5.386168 | 264.6149 | rice |
| 12 | 90 | 46 | 42 | 23.97898 | 81.45062 | 7.502834 | 250.0832 | rice |
| 13 | 78 | 58 | 44 | 26.8008 | 80.88685 | 5.108682 | 284.4365 | rice |
| 14 | 93 | 56 | 36 | 24.01498 | 82.05687 | 6.984354 | 185.2773 | rice |
| 15 | 94 | 50 | 37 | 25.66585 | 80.66385 | 6.94802 | 209.587 | rice |
| 16 | 60 | 48 | 39 | 24.28209 | 80.30026 | 7.042299 | 231.0863 | rice |
| 17 | 85 | 38 | 41 | 21.58712 | 82.78837 | 6.249051 | 276.6552 | rice |
| 18 | 91 | 35 | 39 | 23.79392 | 80.41818 | 6.97086 | 206.2612 | rice |
| 19 | 77 | 38 | 36 | 21.86525 | 80.1923 | 5.953933 | 224.555 | rice |

*Figure 21 You can also edit and save the data*



Show 10 entries

|  | N | P | K | temperature | humidity | ph |
|---|---|---|---|---|---|---|
| 1 | 90 | 42 | 43 | 20.87974371 | 82.00274423 | 6.502985292 |
| 2 | 85 | 58 | 41 | 21.77046169 | 80.31964408 | 7.038096361 |
| 3 | 60 | 55 | 44 | 23.00445915 | 82.3207629 | 7.840207144 |
| 4 | 74 | 35 | 40 | 26.49109635 | 80.15836264 | 6.980400905 |
| 5 | 78 | 42 | 42 | 20.13017482 | 81.60487287 | 7.628472891 |
| 6 | 69 | 37 | 42 | 23.05804872 | 83.37011772 | 7.073453503 |
| 7 | 69 | 55 | 38 | 22.70883798 | 82.63941394 | 5.70080568 |
| 8 | 94 | 53 | 40 | 20.27774362 | 82.89408619 | 5.718627178 |
| 9 | 89 | 54 | 38 | 24.51588066 | 83.5352163 | 6.685346424 |
| 10 | 68 | 58 | 38 | 23.22397386 | 83.03322691 | 6.336253525 |

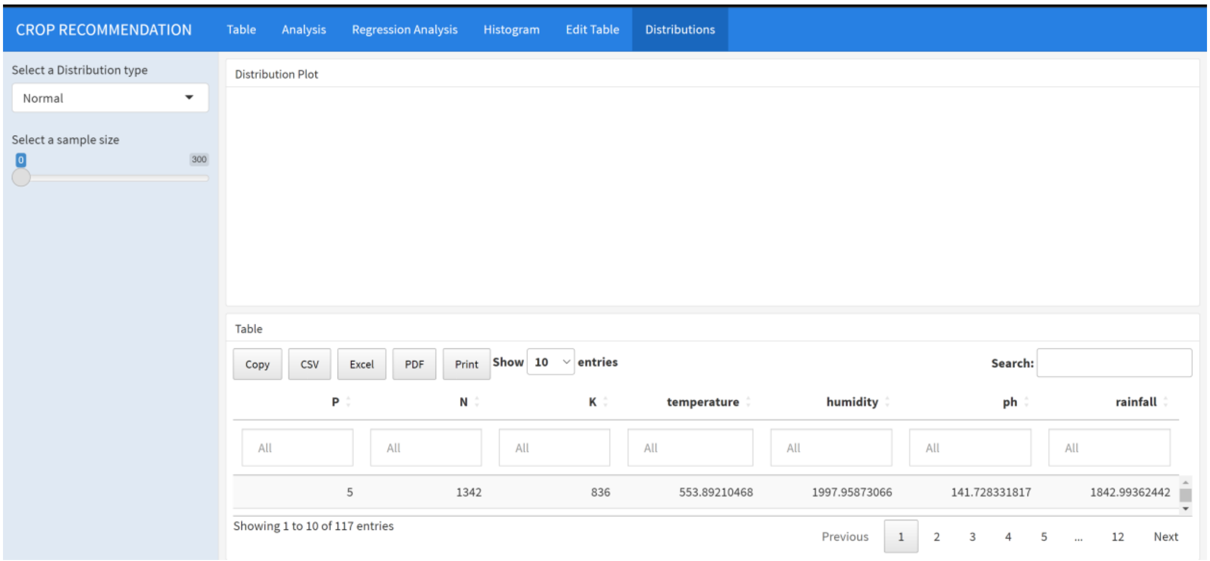
Showing 1 to 10 of 2,200 entries

*Figure 22 The editing will be saved here Dynamically*

### 7) Distributions:

~~~
# Distributions

```{r}
library(tidyverse)
library(DT)
sData <- read.csv("crop.csv")
uData <- reactive(
sData %>% group_by(!!! rlang::syms(input$GB)) %>%
summarise_if(is.numeric,sum,na.rm=T))
```

## Column {.sidebar}

```{r}
selectInput("Distributions", "Select a Distribution type", choices = c("Normal", "Uniform"))
sliderInput("sampleSize","Select a sample size", min = 0, max = 300, value= 270, step= 0)
conditionalPanel(condition = "input.Distribution == 'Normal'", textInput("Mean", "Please Select Mean", 10), textInput("sd", "Please Select Standard Deviation"))
conditionalPanel(condition = "input.Distribution == 'Uniform'", textInput("u1","Val1: ", value = 1), textInput("u2","Val2: ", value = 1))
server <- function(input, output, session){
output$myPlot <- renderPlot({
distType <- input$Distribution
size <- input$sampleSize
if(distType == "Normal"){
randomVec <- rnorm(size, mean = as.numeric(input$mean), sd=as.numeric(input$sd))
}
else{
  randomVec <- runif(size, min = 0, max = 300)
}
hist(randomVec, col="blue")})}
```

## Column
~~~

~~~
### Distribution Plot


```{r}
plotOutput("myPlot")
```
### Table


```{r}
renderDT(
 updateData(), rownames = F, extensions = 'Buttons', filter="top", editable=T,
 options = list(
   dom = 'Blfrtip',
   buttons = c('copy', 'csv', 'excel', 'pdf', 'print'),
   lengthMenu = list(c(10,50, 100, -1), c(10,50, 100, "All"))
 )
)
```
~~~



Select a Distribution type: Normal

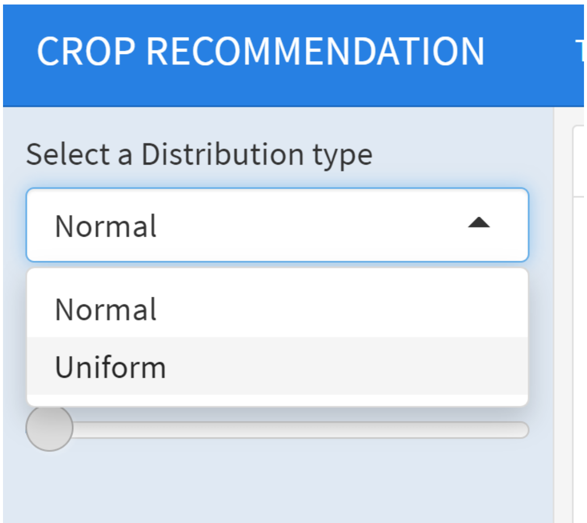
Select a sample size: 0 (0–300)

Show 10 entries

| P | N | K | temperature | humidity | ph | rainfall |
|---|---|---|---|---|---|---|
| All | All | All | All | All | All | All |
| 5 | 1342 | 836 | 553.89210468 | 1997.95873066 | 141.728331817 | 1842.99362442 |

Showing 1 to 10 of 117 entries

*Figure 23 Distribution picture*



CROP RECOMMENDATION

Select a Distribution type

Normal

*Figure 24 Applied Uniform and Normal Distribution on this data as it was continuous*