Crop Yield Analysis

Ahemd Elharith Osama

2024-06-28

# Introduction

Smallholder farmers are crucial contributors to global food production, and in India often suffer most from poverty and malnutrition. These farmers face challenges such as limited access to modern agriculture, unpredictable weather, and resource constraints. To tackle this issue, Digital Green collected data via surveys, offering insights into farming practices, environmental conditions, and crop yields.

# About the Data

The data was collected through a survey conducted across multiple districts in India. It consists of a variety of factors that could potentially impact the yield of rice crops. These factors include things like the type and amount of fertilizers used, the quantity of seedlings planted, methods of preparing the land, different irrigation techniques employed, among other features. The dataset comprises more than 5000 data points, each having more than 40 features.

* ***Data source :*** [***Zindi platform***](https://zindi.africa/competitions/digital-green-crop-yield-estimate-challenge/data) ***.***
* ***Data files :*** [***Train.csv***](https://docs.google.com/spreadsheets/d/1aZwFRA2Cm2s2s7OxFwlU-D8s4vmDUO3OCjYgDq7U95A/edit?usp=sharing) ***and*** [***VariableDescription.csv***](https://docs.google.com/spreadsheets/d/1IDrSq9wI5BsXIgRMiOGFoOJNM914XB9xxOpepNQ53q8/edit?usp=sharing) ***.***
* ***powered by :*** [***Digital Green***](https://www.digitalgreen.org/) ***and*** [***Fair Forward AI***](https://www.bmz-digital.global/en/overview-of-initiatives/fair-forward/) ***.***

# The objective

The most important variable in this data is the Yield variable which is the corp yield for different farmers in India (wheat and rice crops) , we want to answer the following questions :

1. What the distribution of the yield variables ?
2. what is the crop yield for each District in India what is the largest ?
3. What is the different agriculture methods that influence the crop yield ?
4. What the best agriculture methods that implies better crop yield ?
5. What are the variables that are correlated to the yield variable ?

***NOTE :***

***We gonna use statistical inference with proper tests to draw conclusion about these questions and estimate the population parameters.***

# Workflow Description:

**We’ll go through the following steps in order to achieve the objective** :

1. Discovering the data.
2. Explanatory Data Analysis “EDA”.
3. Cleaning the data.
4. Data analysis and visualization.
5. Statistical testing
6. Insights and conclusion(Report.pdf).

# 1. Discovering the data:

*The data files are as follows :*

* *"Train.csv" : containing the data under study.*
* *"VariableDescription.csv" : containing the definitions of each variables or features in the Train.csv.*
* *"Test.csv" : containing the data without the "Yield" variable , so we'll not use this file.*

## 1.1 Importing libraries

library(psych)  
library(vcd)  
library(pander)  
library(knitr)  
library(kableExtra)  
library(ggplot2)  
library(dplyr)  
options(repr.plot.width = 12, repr.plot.height = 8)

## 1.2 Load the data

train\_df <- read.csv("Train.csv")  
variable\_defintion <- read.csv("VariableDescription.csv")

* Data shape
* Number of rows : 3870 Number of Variables : 44

## 1.3 Data types

*We Have three types of features (Numeric, Categorical and Date time) , we'll explore each of them individually for easier analysis letter.*

* ***Date time features***
* date\_time <- c("CropTillageDate" ,"RcNursEstDate" ,"Harv\_date" , "Threshing\_date" ,  
   "SeedingSowingTransplanting")  
  pander(colnames(train\_df[,date\_time]) , style = "rmarkdown")
* *CropTillageDate*, *RcNursEstDate*, *Harv\_date*, *Threshing\_date* and *SeedingSowingTransplanting*
* pander(paste("Number of date time features " , length(date\_time)))
* Number of date time features 5
* ***Numerical features***
* num\_col <- colnames(train\_df[, sapply(train\_df , is.numeric)])  
  pander(colnames(train\_df[, num\_col]))
* *CultLand*, *CropCultLand*, *CropTillageDepth*, *SeedlingsPerPit*, *TransplantingIrrigationHours*, *TransIrriCost*, *StandingWater*, *Ganaura*, *CropOrgFYM*, *NoFertilizerAppln*, *BasalDAP*, *BasalUrea*, *X1tdUrea*, *X1appDaysUrea*, *X2tdUrea*, *X2appDaysUrea*, *Harv\_hand\_rent*, *Residue\_length*, *Residue\_perc*, *Acre* and *Yield*
* pander(paste("Number of numerical features " , length(num\_col)) , style = "rmarkdown")
* Number of numerical features 21
* ***Categorical features***
* *ID*, *District*, *Block*, *LandPreparationMethod*, *CropEstMethod*, *NursDetFactor*, *TransDetFactor*, *TransplantingIrrigationSource*, *TransplantingIrrigationPowerSource*, *OrgFertilizers*, *PCropSolidOrgFertAppMethod*, *CropbasalFerts*, *MineralFertAppMethod*, *FirstTopDressFert*, *MineralFertAppMethod.1*, *Harv\_method*, *Threshing\_method* and *Stubble\_use*
* pander(paste("Number of categorical features " , length(cat\_col)))
* Number of categorical features 18

# 2. EDA :

In this section we’ll walk through some basic *frequency tables* and *summary statistics* about the data , also we’ll explore the *missing values* and the spare *categorical features*.

## 2.1 Datetime features

* First convert to Date
* library(dplyr)  
    
  train\_df <- train\_df %>%  
   mutate\_at(vars(date\_time), as.Date, format = "%Y-%m-%d")  
    
  pander(str(train\_df[,date\_time]) , style = "rmarkdown")
* ‘data.frame’: 3870 obs. of 5 variables: $ CropTillageDate : Date, format: “2022-07-20” “2022-07-18” … $ RcNursEstDate : Date, format: “2022-06-27” “2022-06-20” … $ Harv\_date : Date, format: “2022-11-16” “2022-11-25” … $ Threshing\_date : Date, format: “2022-11-16” “2022-12-24” … $ SeedingSowingTransplanting: Date, format: “2022-07-21” “2022-07-20” …
* Summary for agriculture dates
* pander(summary(train\_df[,date\_time]))

Table continues below

| CropTillageDate | RcNursEstDate | Harv\_date |
| --- | --- | --- |
| Min. :2022-05-30 | Min. :2022-06-01 | Min. :2021-12-01 |
| 1st Qu.:2022-07-02 | 1st Qu.:2022-06-20 | 1st Qu.:2022-11-02 |
| Median :2022-07-14 | Median :2022-06-27 | Median :2022-11-13 |
| Mean :2022-07-11 | Mean :2022-06-27 | Mean :2022-11-12 |
| 3rd Qu.:2022-07-22 | 3rd Qu.:2022-07-04 | 3rd Qu.:2022-11-25 |
| Max. :2022-08-27 | Max. :2022-07-31 | Max. :2023-03-18 |
| NA | NA’s :83 | NA |

| Threshing\_date | SeedingSowingTransplanting |
| --- | --- |
| Min. :2022-10-01 | Min. :2022-07-01 |
| 1st Qu.:2022-11-15 | 1st Qu.:2022-07-18 |
| Median :2022-12-13 | Median :2022-07-25 |
| Mean :2022-12-12 | Mean :2022-07-25 |
| 3rd Qu.:2023-01-02 | 3rd Qu.:2022-08-02 |
| Max. :2023-03-22 | Max. :2022-08-31 |
| NA | NA |

## 2.2 Categorical features

* Unique values in each categorical column
* pander(sapply(train\_df[,cat\_col], function(x) n\_distinct(x)) , style = "rmarkdown")

Table continues below

| ID | District | Block | LandPreparationMethod | CropEstMethod |
| --- | --- | --- | --- | --- |
| 3870 | 4 | 9 | 43 | 4 |

Table continues below

| NursDetFactor | TransDetFactor | TransplantingIrrigationSource |
| --- | --- | --- |
| 126 | 156 | 7 |

Table continues below

| TransplantingIrrigationPowerSource | OrgFertilizers |
| --- | --- |
| 4 | 32 |

Table continues below

| PCropSolidOrgFertAppMethod | CropbasalFerts | MineralFertAppMethod |
| --- | --- | --- |
| 5 | 35 | 3 |

Table continues below

| FirstTopDressFert | MineralFertAppMethod.1 | Harv\_method | Threshing\_method |
| --- | --- | --- | --- |
| 15 | 4 | 2 | 2 |

| Stubble\_use |
| --- |
| 2 |

* Missing values
* pander(cat("Number of missing values in categorical columns:",sum(is.na(train\_df[,cat\_col]))))
* Number of missing values in categorical columns: 0

## 2.3 Numerical features

* Summary of numerical features
* pander(summary(train\_df[num\_col] , na.omit = TRUE) , style = "rmarkdown")

Table continues below

| CultLand | CropCultLand | CropTillageDepth | SeedlingsPerPit |
| --- | --- | --- | --- |
| Min. : 1.00 | Min. : 1.00 | Min. :1.000 | Min. : 1.000 |
| 1st Qu.: 12.00 | 1st Qu.: 10.00 | 1st Qu.:4.000 | 1st Qu.: 2.000 |
| Median : 20.00 | Median : 20.00 | Median :4.000 | Median : 2.000 |
| Mean : 28.53 | Mean : 24.73 | Mean :4.488 | Mean : 2.707 |
| 3rd Qu.: 35.00 | 3rd Qu.: 30.00 | 3rd Qu.:5.000 | 3rd Qu.: 3.000 |
| Max. :800.00 | Max. :800.00 | Max. :8.000 | Max. :442.000 |
| NA | NA | NA | NA’s :289 |

Table continues below

| TransplantingIrrigationHours | TransIrriCost | StandingWater |
| --- | --- | --- |
| Min. : 1.000 | Min. : 1.0 | Min. : 1.000 |
| 1st Qu.: 2.000 | 1st Qu.: 150.0 | 1st Qu.: 2.000 |
| Median : 4.000 | Median : 250.0 | Median : 3.000 |
| Mean : 8.018 | Mean : 379.7 | Mean : 3.248 |
| 3rd Qu.: 6.000 | 3rd Qu.: 450.0 | 3rd Qu.: 4.000 |
| Max. :2000.000 | Max. :6000.0 | Max. :15.000 |
| NA’s :193 | NA’s :882 | NA’s :238 |

Table continues below

| Ganaura | CropOrgFYM | NoFertilizerAppln | BasalDAP |
| --- | --- | --- | --- |
| Min. : 1.00 | Min. : 1.00 | Min. :1.000 | Min. : 1.00 |
| 1st Qu.: 1.00 | 1st Qu.: 1.00 | 1st Qu.:2.000 | 1st Qu.: 6.00 |
| Median : 3.00 | Median : 2.00 | Median :2.000 | Median : 10.00 |
| Mean : 29.73 | Mean : 57.45 | Mean :2.184 | Mean : 11.45 |
| 3rd Qu.: 4.00 | 3rd Qu.: 5.00 | 3rd Qu.:3.000 | 3rd Qu.: 15.00 |
| Max. :1400.00 | Max. :4000.00 | Max. :4.000 | Max. :100.00 |
| NA’s :2417 | NA’s :2674 | NA | NA’s :543 |

Table continues below

| BasalUrea | X1tdUrea | X1appDaysUrea | X2tdUrea |
| --- | --- | --- | --- |
| Min. : 1.00 | Min. : 1.00 | Min. : 1.0 | Min. : 1.000 |
| 1st Qu.: 7.00 | 1st Qu.: 6.00 | 1st Qu.: 23.0 | 1st Qu.: 4.000 |
| Median :10.00 | Median :10.00 | Median : 28.0 | Median : 6.000 |
| Mean :13.35 | Mean :11.51 | Mean : 29.2 | Mean : 7.375 |
| 3rd Qu.:16.00 | 3rd Qu.:15.00 | 3rd Qu.: 36.0 | 3rd Qu.:10.000 |
| Max. :90.00 | Max. :90.00 | Max. :332.0 | Max. :67.000 |
| NA’s :1704 | NA’s :556 | NA’s :556 | NA’s :2694 |

Table continues below

| X2appDaysUrea | Harv\_hand\_rent | Residue\_length | Residue\_perc |
| --- | --- | --- | --- |
| Min. : 1.00 | Min. : 1.0 | Min. : 2.00 | Min. :10.00 |
| 1st Qu.:58.00 | 1st Qu.: 150.0 | 1st Qu.:25.00 | 1st Qu.:10.00 |
| Median :60.00 | Median : 400.0 | Median :26.00 | Median :10.00 |
| Mean :58.77 | Mean : 536.6 | Mean :26.52 | Mean :11.77 |
| 3rd Qu.:65.00 | 3rd Qu.: 700.0 | 3rd Qu.:30.00 | 3rd Qu.:10.00 |
| Max. :97.00 | Max. :60000.0 | Max. :30.00 | Max. :40.00 |
| NA’s :2700 | NA’s :252 | NA | NA |

| Acre | Yield |
| --- | --- |
| Min. :0.04545 | Min. : 4.0 |
| 1st Qu.:0.15625 | 1st Qu.: 300.0 |
| Median :0.22727 | Median : 425.0 |
| Mean :0.29283 | Mean : 594.3 |
| 3rd Qu.:0.37037 | 3rd Qu.: 740.0 |
| Max. :2.18750 | Max. :16800.0 |
| NA | NA |

* Description with some basic statistic (mean, sd and median)
* pander(describe(train\_df[,num\_col] , trim = .15 , na.rm = TRUE)[,3:5],style = "rmarkdown")

|  | mean | sd | median |
| --- | --- | --- | --- |
| **CultLand** | 28.53 | 30.45 | 20 |
| **CropCultLand** | 24.73 | 27.99 | 20 |
| **CropTillageDepth** | 4.488 | 1.133 | 4 |
| **SeedlingsPerPit** | 2.707 | 7.624 | 2 |
| **TransplantingIrrigationHours** | 8.018 | 42.61 | 4 |
| **TransIrriCost** | 379.7 | 419.7 | 250 |
| **StandingWater** | 3.248 | 2.207 | 3 |
| **Ganaura** | 29.73 | 122.7 | 3 |
| **CropOrgFYM** | 57.45 | 328.3 | 2 |
| **NoFertilizerAppln** | 2.184 | 0.6346 | 2 |
| **BasalDAP** | 11.45 | 8.422 | 10 |
| **BasalUrea** | 13.35 | 9.702 | 10 |
| **X1tdUrea** | 11.51 | 8.716 | 10 |
| **X1appDaysUrea** | 29.2 | 12.14 | 28 |
| **X2tdUrea** | 7.375 | 5.933 | 6 |
| **X2appDaysUrea** | 58.76 | 11.36 | 60 |
| **Harv\_hand\_rent** | 536.6 | 1139 | 400 |
| **Residue\_length** | 26.52 | 3.193 | 26 |
| **Residue\_perc** | 11.77 | 7.065 | 10 |
| **Acre** | 0.2928 | 0.2069 | 0.2273 |
| **Yield** | 594.3 | 651.9 | 425 |

* Missing values
* # Filter columns with missing values in num\_col  
  columns\_with\_na <- colnames(train\_df[,num\_col])[colSums(is.na(train\_df[,num\_col])) > 0]  
  pander(sapply(train\_df[,columns\_with\_na], function(x) sum(is.na(x))) , style = "rmarkdown")

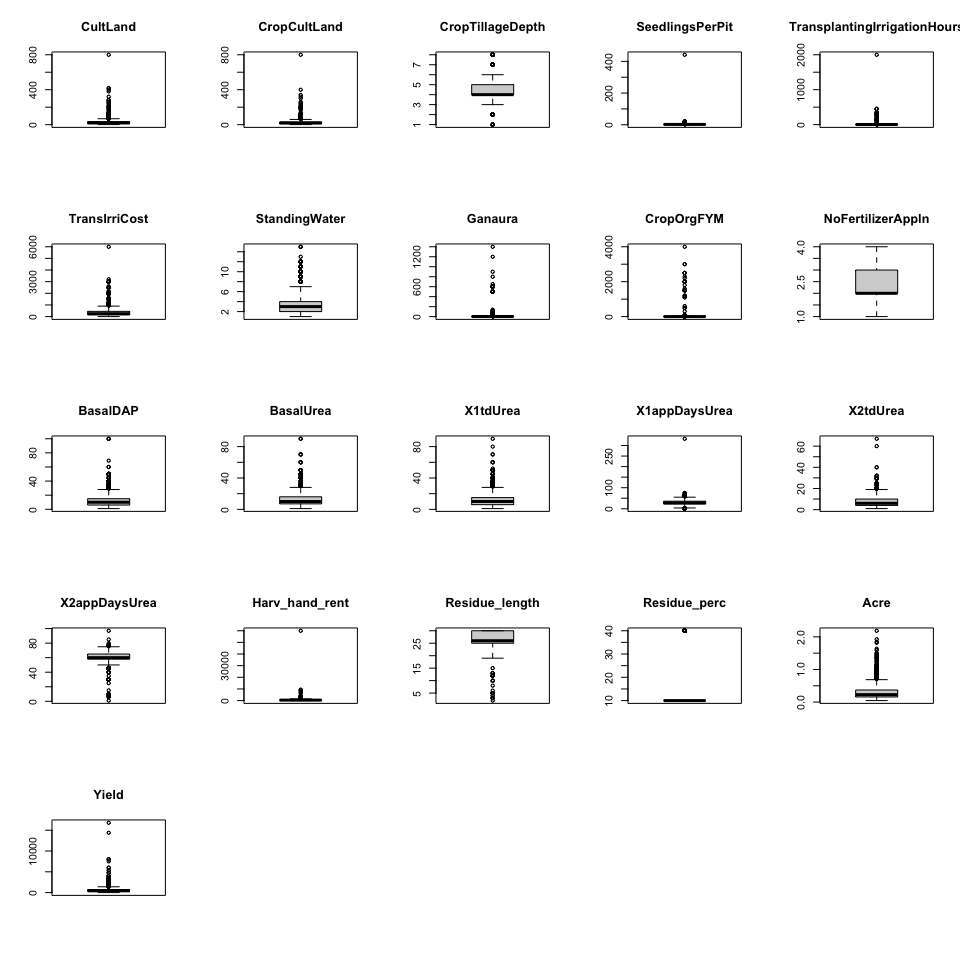
Table continues below

| SeedlingsPerPit | TransplantingIrrigationHours | TransIrriCost | StandingWater |
| --- | --- | --- | --- |
| 289 | 193 | 882 | 238 |

Table continues below

| Ganaura | CropOrgFYM | BasalDAP | BasalUrea | X1tdUrea | X1appDaysUrea |
| --- | --- | --- | --- | --- | --- |
| 2417 | 2674 | 543 | 1704 | 556 | 556 |

| X2tdUrea | X2appDaysUrea | Harv\_hand\_rent |
| --- | --- | --- |
| 2694 | 2700 | 252 |

* pander(paste("Number of missing values in numerical features:"  
   ,sum(is.na(train\_df[,num\_col]))))
* Number of missing values in numerical features: 15698
* Outliers analysis
* In this analysis we gonna include columns with non missing values , and use the IQR to identify outliers.
* # Function to check for outliers in a column  
  has\_outliers <- function(x) {  
   Q1 <- quantile(x, 0.25, na.rm = TRUE)  
   Q3 <- quantile(x, 0.75, na.rm = TRUE)  
   IQR <- Q3 - Q1  
   lower\_bound <- Q1 - 1.5 \* IQR  
   upper\_bound <- Q3 + 1.5 \* IQR  
   if (any(x < lower\_bound | x > upper\_bound, na.rm = TRUE))  
   {  
   return(colnames(x))  
   }  
    
  }  
  columns\_with\_outliers <- names(sapply(train\_df[,num\_col], has\_outliers))  
  pander(cat("Numbers of coulumns with outliers:" , length(columns\_with\_outliers)))
* Numbers of coulumns with outliers: 21
* Boxplot to show outliers
* par(mfrow=c(5, 5))  
  for (i in columns\_with\_outliers) {  
   boxplot(train\_df[,i], main= i )  
  }
* 

# 3. Data cleaning and preprocessing:

**In this section we gonna correct some errors in the data , handling missing values and outliers for each variables type.**

## 3.1 Datetime features

we gonna drop date time features because in fact the calender of agriculture is known in these districts.

train\_df <- train\_df[, !names(train\_df) %in% date\_time]  
dim(train\_df)

## [1] 3870 39

## 3.2 Categorical features

**For these variables we’ll do the following :**

1. ***Correct some errors within the District “Jamui” an Block “Gurua” which is “Gaya” Block.***

* # Group by District and get unique Block values  
  unique\_blocks <- train\_df %>%  
   group\_by(District) %>%  
   summarise(UniqueBlocks = (unique(Block)))  
    
  # Print the resulting data frame  
  print(unique\_blocks)
* ## # A tibble: 10 × 2  
  ## # Groups: District [4]  
  ## District UniqueBlocks  
  ## <chr> <chr>   
  ## 1 Gaya Gurua   
  ## 2 Gaya Wazirganj   
  ## 3 Jamui Khaira   
  ## 4 Jamui Jamui   
  ## 5 Jamui Gurua   
  ## 6 Nalanda Noorsarai   
  ## 7 Nalanda Rajgir   
  ## 8 Vaishali Garoul   
  ## 9 Vaishali Mahua   
  ## 10 Vaishali Chehrakala
* train\_df[which(train\_df$District == 'Jamui' & train\_df$Block == 'Gurua'),  
   "District"] <- "Gaya"

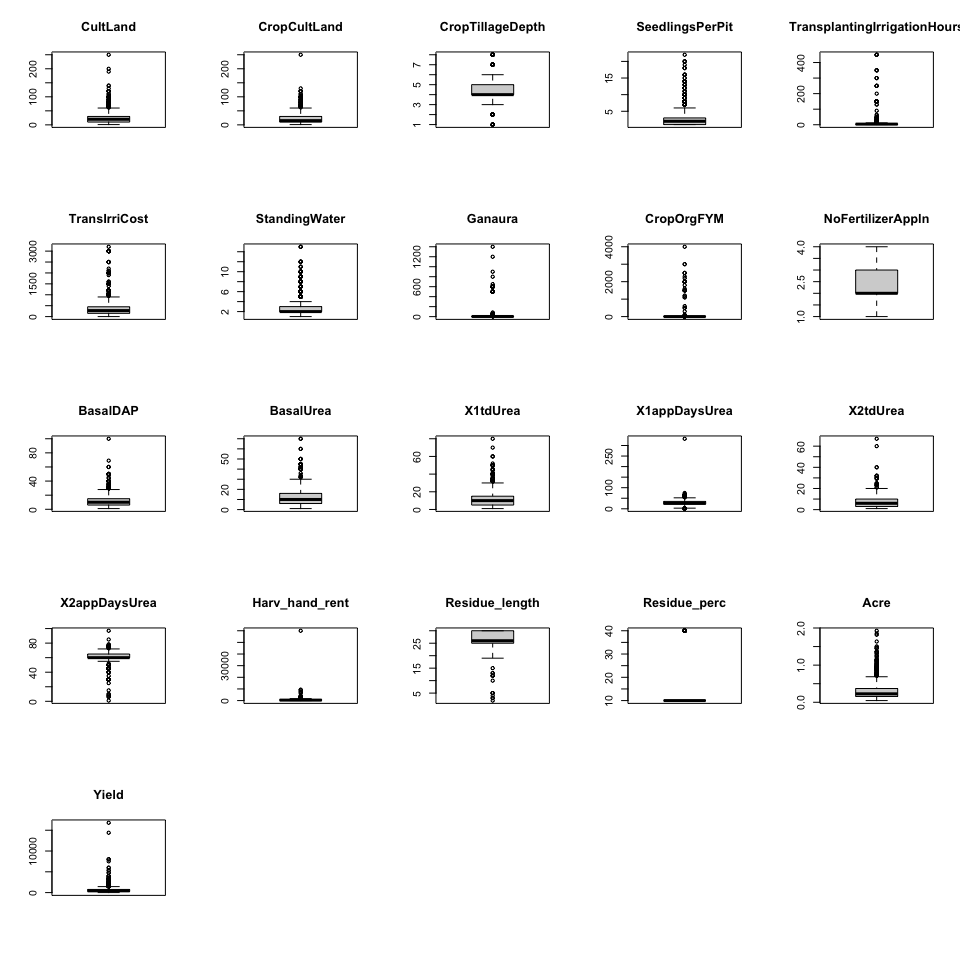
1. ***Drop columns with more than 10 unique values (this maybe open questions and cause problems in analysis and visualization letter).***

* get\_columns <- function(df, threshold = 10) {  
   col\_names <- sapply(df, function(col) length(unique(col)) > threshold)  
   names(df)[col\_names]  
  }  
  to\_drop\_cat <- get\_columns(train\_df[,cat\_col])  
    
  train\_df <- train\_df[, (!names(train\_df) %in% to\_drop\_cat) &   
   (!names(train\_df) %in% date\_time)]  
  print(dim(train\_df))
* ## [1] 3870 32
  + Update the cat\_col
* cat\_col <- cat\_col[!(cat\_col %in% to\_drop\_cat)]  
  cat\_col
* ## [1] "District" "Block"   
  ## [3] "CropEstMethod" "TransplantingIrrigationSource"   
  ## [5] "TransplantingIrrigationPowerSource" "PCropSolidOrgFertAppMethod"   
  ## [7] "MineralFertAppMethod" "MineralFertAppMethod.1"   
  ## [9] "Harv\_method" "Threshing\_method"   
  ## [11] "Stubble\_use"
* colnames(train\_df[])
* ## [1] "District" "Block"   
  ## [3] "CultLand" "CropCultLand"   
  ## [5] "CropTillageDepth" "CropEstMethod"   
  ## [7] "SeedlingsPerPit" "TransplantingIrrigationHours"   
  ## [9] "TransplantingIrrigationSource" "TransplantingIrrigationPowerSource"  
  ## [11] "TransIrriCost" "StandingWater"   
  ## [13] "Ganaura" "CropOrgFYM"   
  ## [15] "PCropSolidOrgFertAppMethod" "NoFertilizerAppln"   
  ## [17] "BasalDAP" "BasalUrea"   
  ## [19] "MineralFertAppMethod" "X1tdUrea"   
  ## [21] "X1appDaysUrea" "X2tdUrea"   
  ## [23] "X2appDaysUrea" "MineralFertAppMethod.1"   
  ## [25] "Harv\_method" "Harv\_hand\_rent"   
  ## [27] "Threshing\_method" "Residue\_length"   
  ## [29] "Residue\_perc" "Stubble\_use"   
  ## [31] "Acre" "Yield"

## 3.3 Numerical features

**For these variables we’ll do the following :**

1. ***Cutoff some extreme outliers***

* train\_df <- subset(train\_df,   
   CultLand != 800 &  
   CropCultLand != 800 &  
   SeedlingsPerPit != 442 &  
   !(TransplantingIrrigationHours == 2000.0 |  
   TransplantingIrrigationHours == 1000) &  
   TransIrriCost != 6000.0 &  
   '.1tdUrea' != 120 &  
   '.1appDaysUrea' != 332.0  
  )  
  par(mfrow=c(5, 5))  
  for (i in columns\_with\_outliers) {  
   boxplot(train\_df[,i], main= i )  
  }
* 

1. ***Drop columns with more than 40% of missing values***

* # Calculate the proportion of missing values in each column  
  missing\_percent <- colMeans(is.na(train\_df[,num\_col]))  
    
  # Select columns with more than 40% missing values  
  to\_drop\_num <- names(train\_df[,num\_col])[missing\_percent > 0.4]  
  pander(cat("Column Names: \n",to\_drop\_num))
* Column Names: Ganaura CropOrgFYM BasalUrea X2tdUrea X2appDaysUrea
* # drop the columns  
  train\_df <- train\_df[, ((!names(train\_df) %in% to\_drop\_num)  
   & (!names(train\_df) %in% to\_drop\_cat)  
   & (!names(train\_df) %in% date\_time))]  
  pander(dim(train\_df))
* *2826* and *27*
* num\_col <- num\_col[(!(num\_col %in% to\_drop\_num) )]  
  num\_col
* ## [1] "CultLand" "CropCultLand"   
  ## [3] "CropTillageDepth" "SeedlingsPerPit"   
  ## [5] "TransplantingIrrigationHours" "TransIrriCost"   
  ## [7] "StandingWater" "NoFertilizerAppln"   
  ## [9] "BasalDAP" "X1tdUrea"   
  ## [11] "X1appDaysUrea" "Harv\_hand\_rent"   
  ## [13] "Residue\_length" "Residue\_perc"   
  ## [15] "Acre" "Yield"
* # number of misiing after droping  
  pander(paste("Number of missing values in numerical features:",  
   sum(is.na(train\_df[,num\_col]))))
* Number of missing values in numerical features: 1389

1. I**mpute the remaining missing values with median of columns**

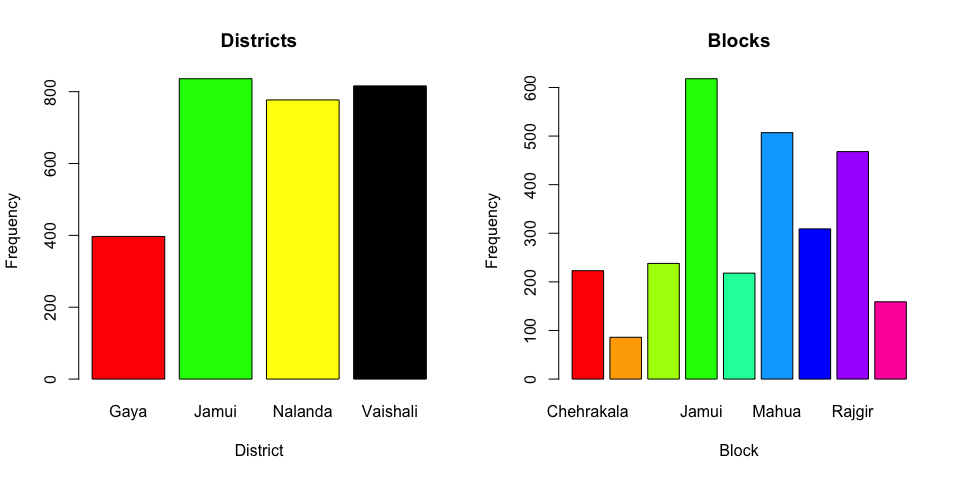
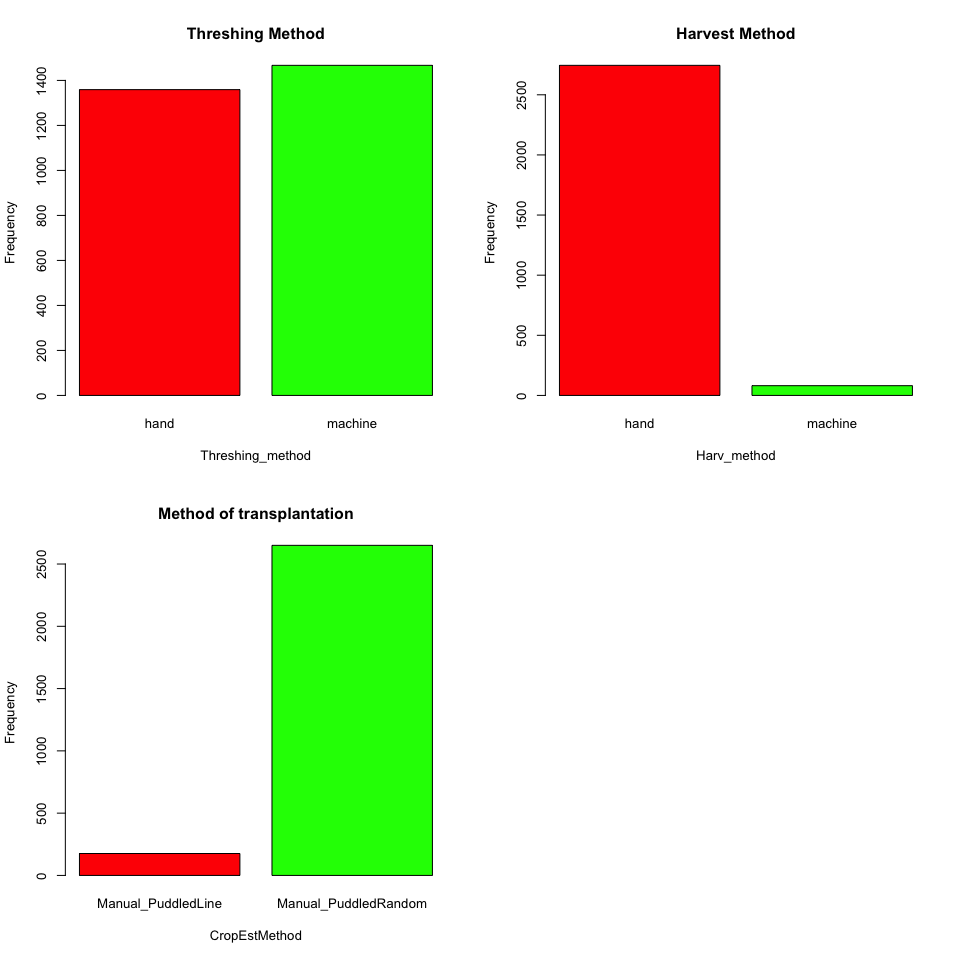
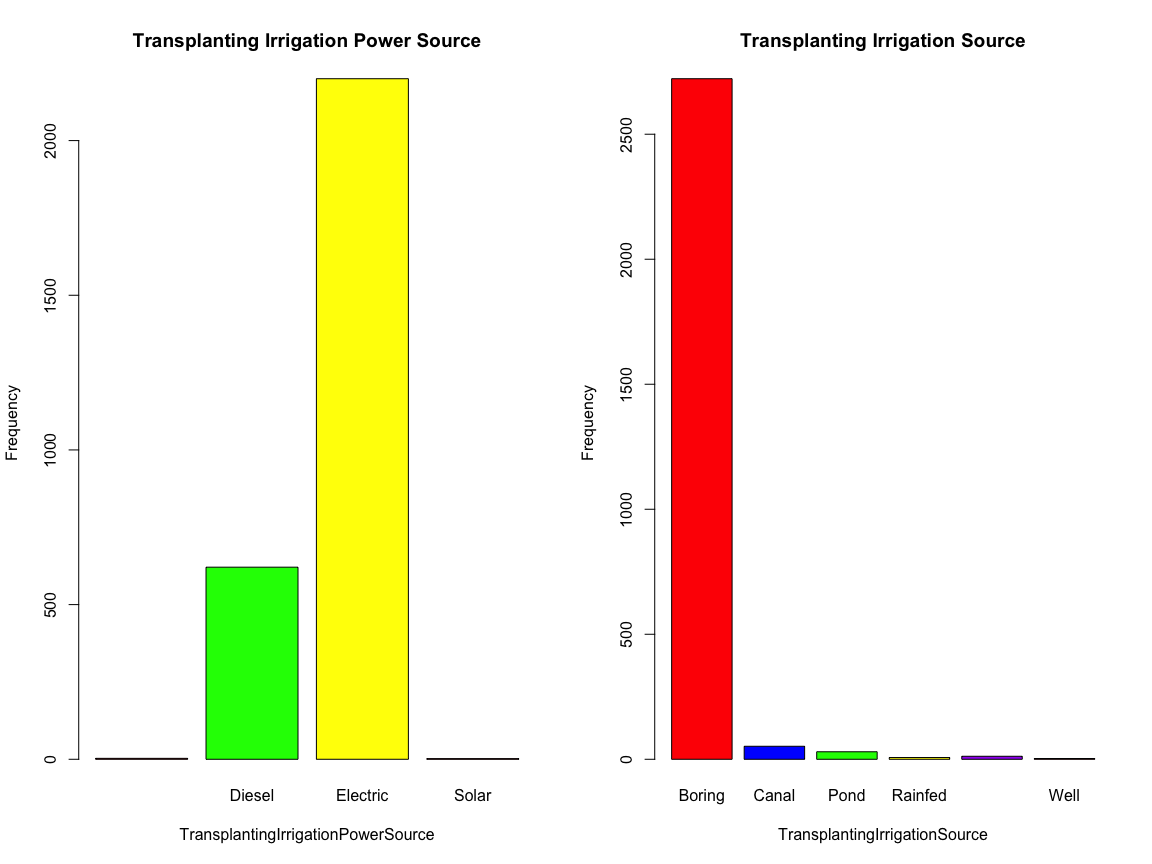
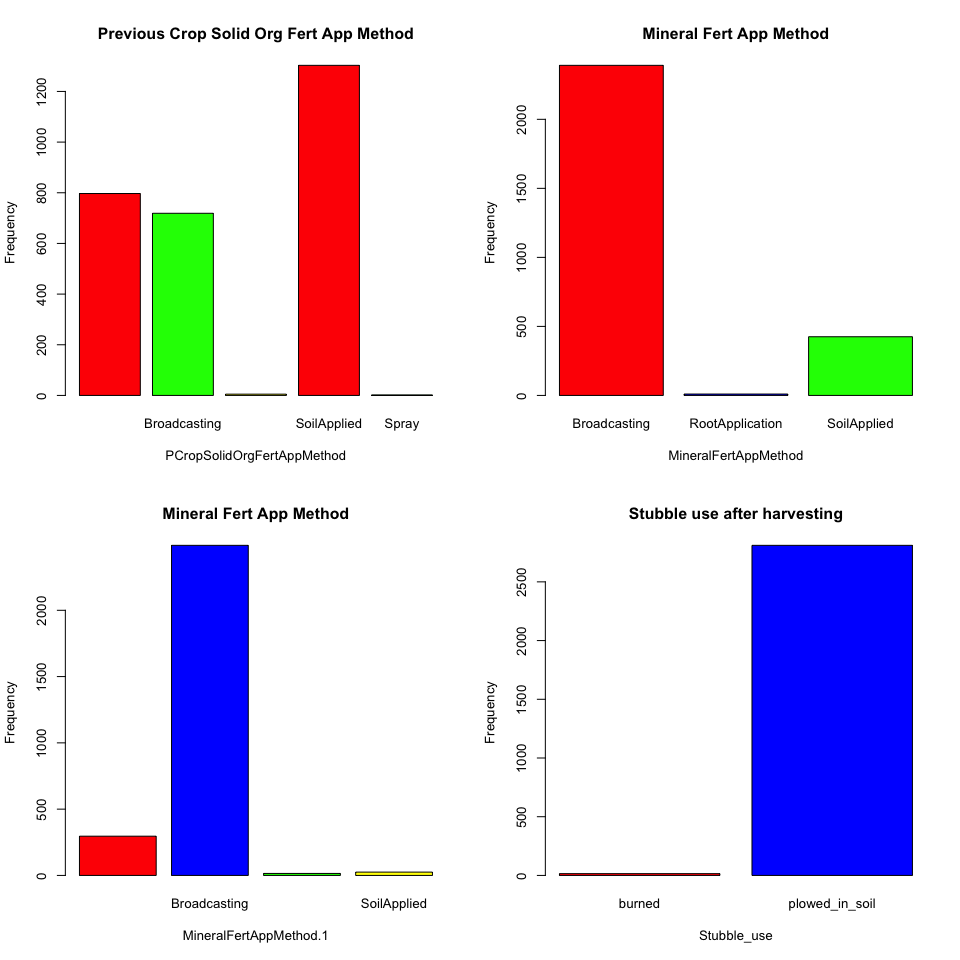
* for (i in 1:ncol(train\_df)) {  
   train\_df[, i][is.na(train\_df[, i])] <- median(train\_df[, i], na.rm = TRUE)  
  }  
    
  # number of misiing after droping  
  pander(paste("Number of missing values in numerical features:",  
   sum(is.na(train\_df[,num\_col]))))
* Number of missing values in numerical features: 0

# 4. Data analysis and visualization:

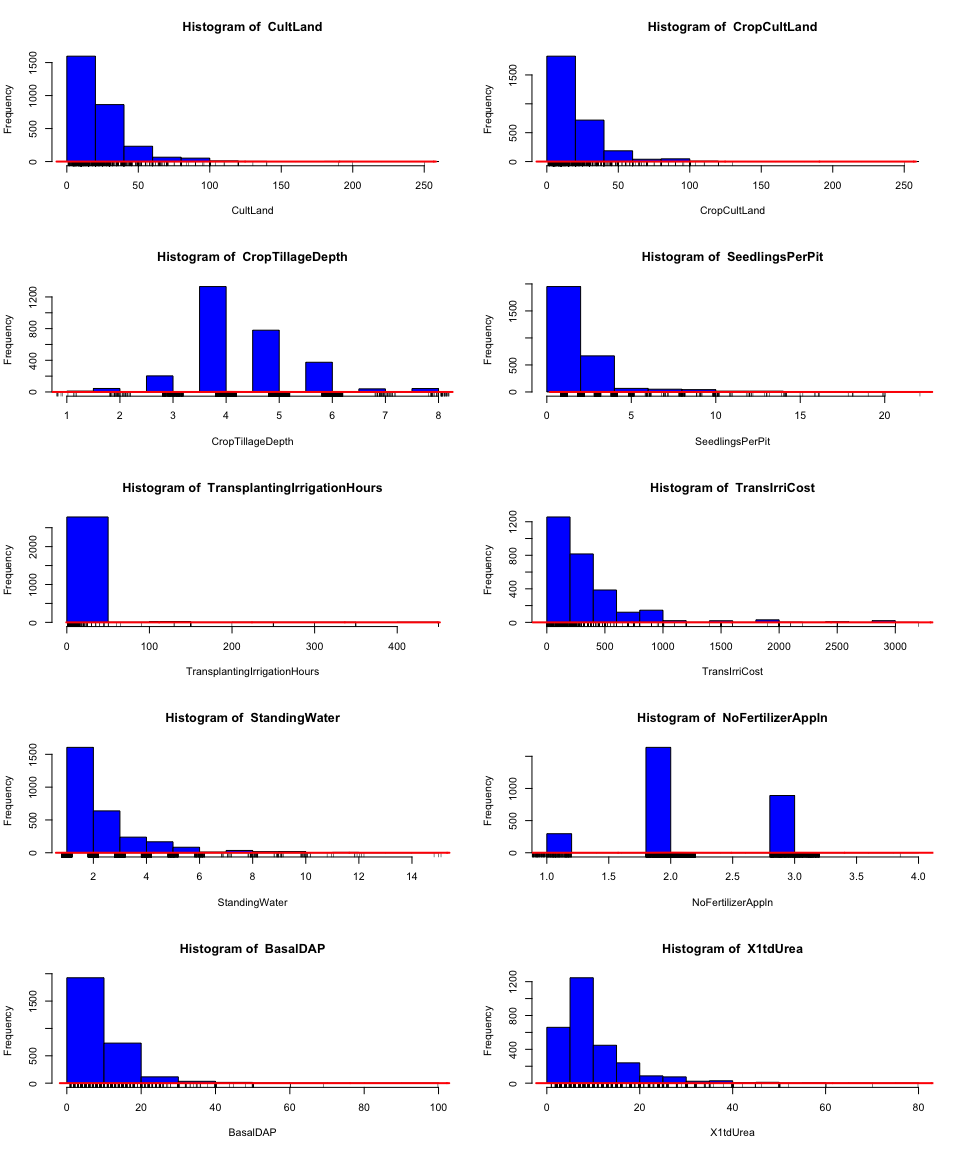
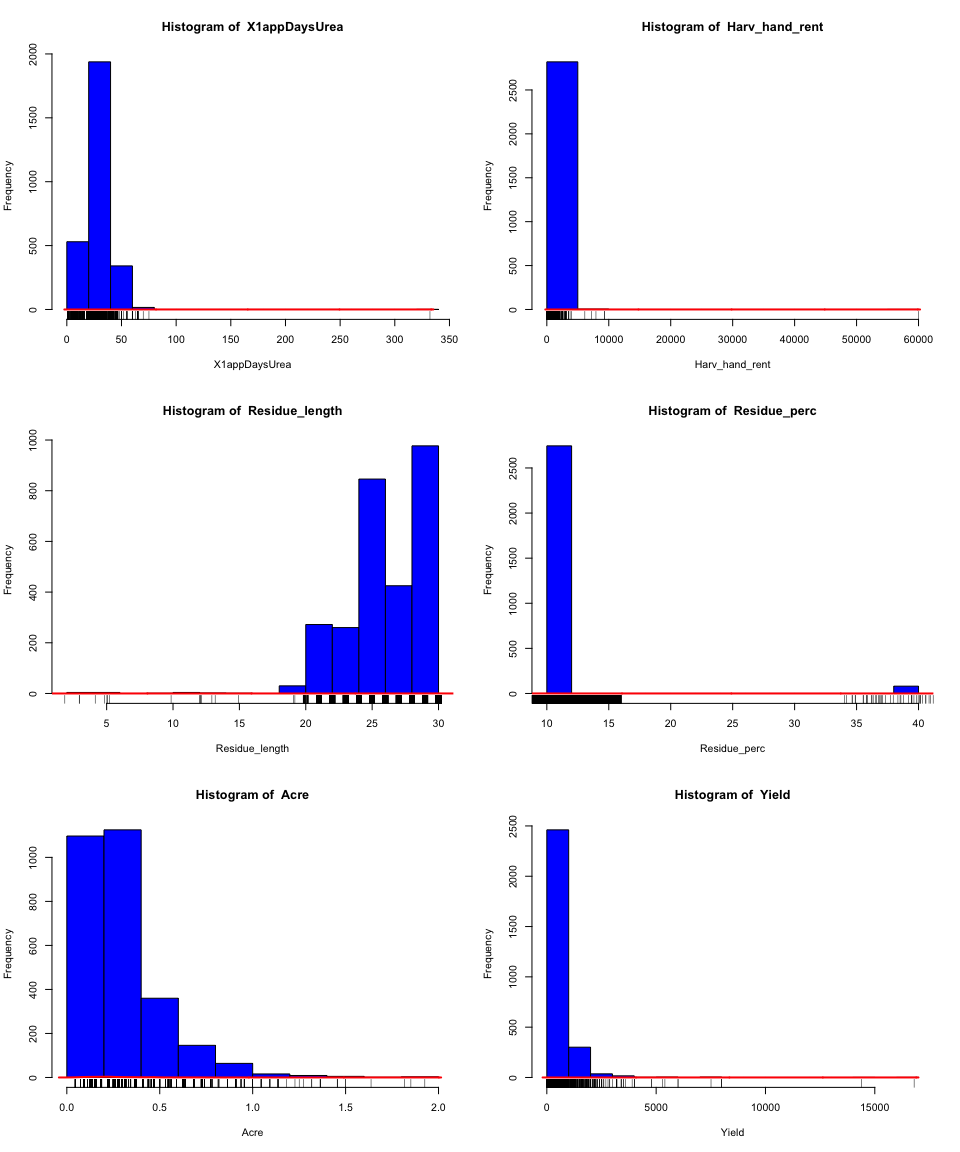
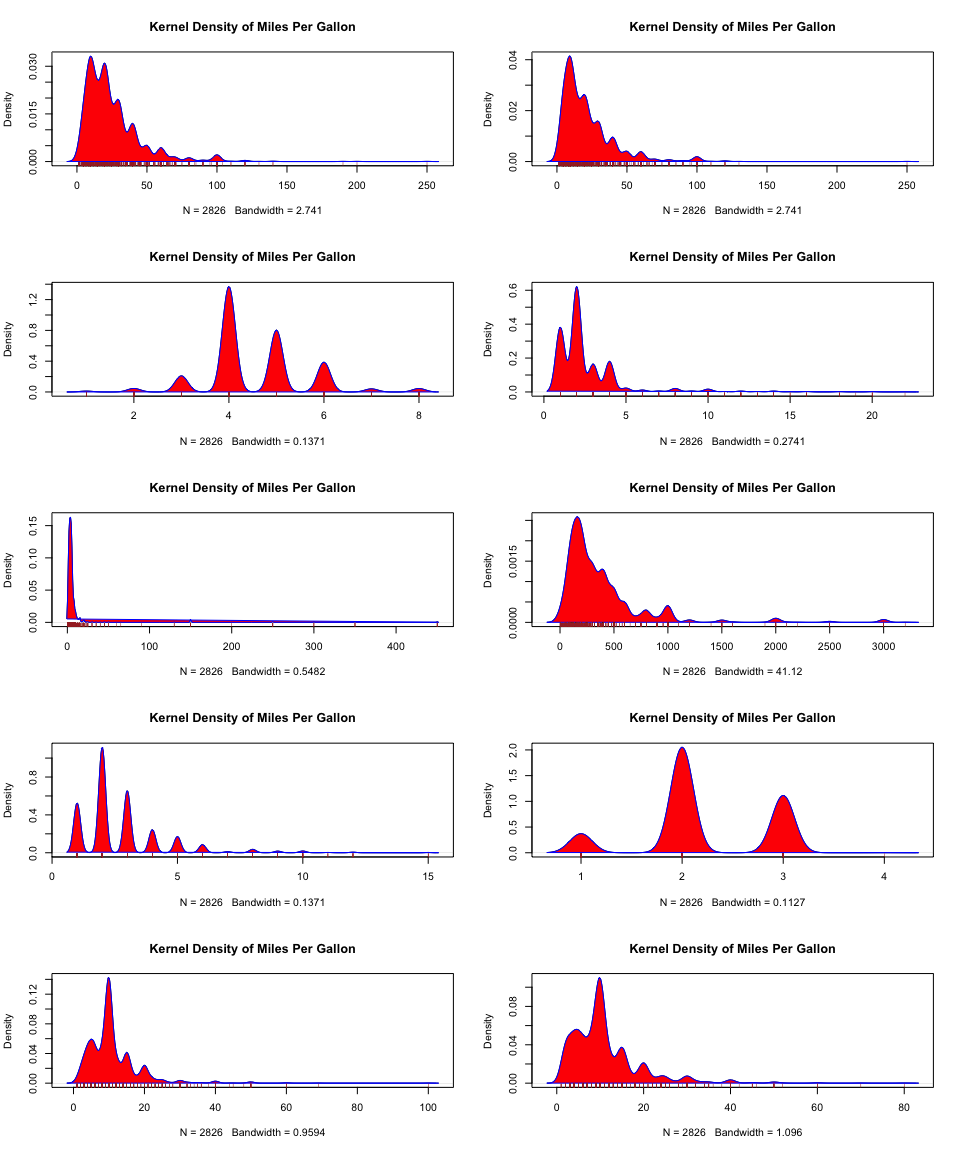
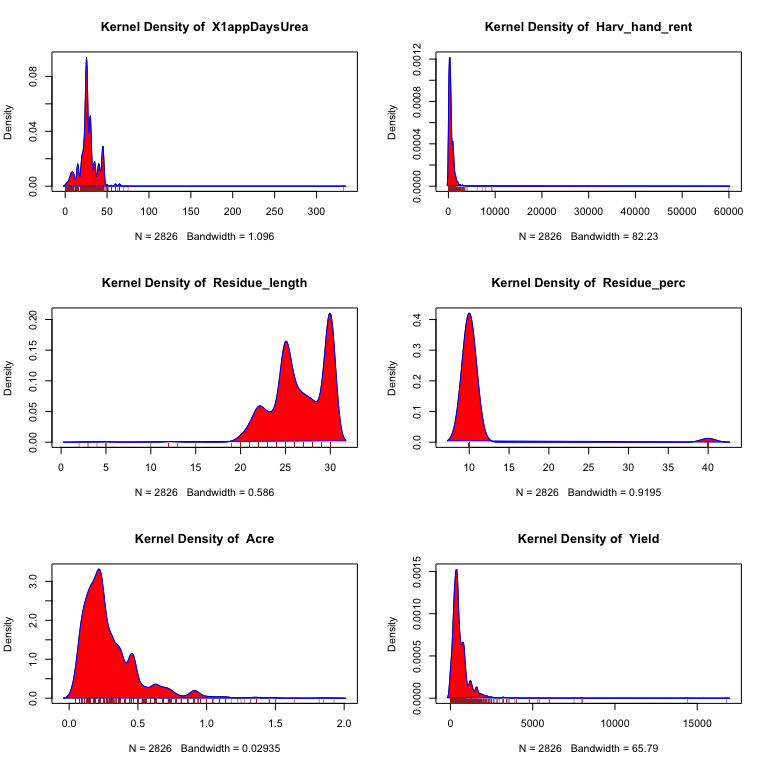
In this section we’ll go through to types of analysis ***Univariate analysis*** *and* ***Bivariate analysis****.*

## 4.1 ***Univariate analysis***

### 4.1.1 Categorical features

* District and Block
* par(mfrow=c(1,2) ) # divide graph area in 2 columns  
  barplot(table(train\_df$District),main="Districts",  
   xlab="District", ylab="Frequency" ,  
   col=c("red" , "green" , 'yellow' , "black"))  
  barplot(table(train\_df$Block),main="Blocks",  
   xlab="Block", ylab="Frequency" , col = rainbow(length(unique(train\_df$Block))))
* 
* Methods used in transplantation , harvesting and threshing
* par(mfrow=c(2, 2)) # divide graph area in 2 columns  
  barplot(table(train\_df$Threshing\_method),main="Threshing Method",  
   xlab="Threshing\_method", ylab="Frequency" , col=c("red" , "green"))  
  barplot(table(train\_df$Harv\_method),main="Harvest Method",  
   xlab="Harv\_method", ylab="Frequency" , col=c("red" , "green"))  
  barplot(table(train\_df$CropEstMethod),main="Method of transplantation",  
   xlab="CropEstMethod", ylab="Frequency", col=c("red" , "green"))
* 
* **Transplanting Irrigation Source and Transplanting Irrigation Power Source**
* par(mfrow=c(1, 2)) # divide graph area in 2 columns  
  barplot(table(train\_df$TransplantingIrrigationPowerSource),  
   main="Transplanting Irrigation Power Source",  
   xlab="TransplantingIrrigationPowerSource", ylab="Frequency" ,  
   col = c("red", "green", "yellow"))  
  barplot(table(train\_df$TransplantingIrrigationSource),  
   main="Transplanting Irrigation Source",  
   xlab="TransplantingIrrigationSource", ylab="Frequency" ,  
   col = c("red", "blue", "green", "yellow", "purple"))
* 
* ***Methods of fertilization***
  + *PCropSolidOrgFertAppMethod : organic fertilizer in your previous crop during land preparation we see that soil applied (تسميد التربة) is the most used, then Broadcasting (رش الأسمدة).*
  + *MineralFertAppMethod : chemical fertilizer in your current crop during land preparation.*
  + *MineralFertAppMethod.1 : chemical fertilizer in your current crop during second dose.*
  + Stubble\_use : Management practice of the stubble after harvesting.
* par(mfrow=c(2, 2)) # divide graph area in 2 columns  
  barplot(table(train\_df$PCropSolidOrgFertAppMethod),  
   main="Previous Crop Solid Org Fert App Method",  
   xlab="PCropSolidOrgFertAppMethod", ylab="Frequency" ,  
   col = c("red", "green", "yellow"))  
  barplot(table(train\_df$MineralFertAppMethod),  
   main="Mineral Fert App Method",  
   xlab="MineralFertAppMethod", ylab="Frequency" ,  
   col = c("red", "blue", "green", "yellow", "purple"))  
  barplot(table(train\_df$MineralFertAppMethod.1),  
   main="Mineral Fert App Method",  
   xlab="MineralFertAppMethod.1", ylab="Frequency" ,  
   col = c("red", "blue", "green", "yellow", "purple"))  
  barplot(table(train\_df$Stubble\_use),  
   main="Stubble use after harvesting",  
   xlab="Stubble\_use", ylab="Frequency" ,  
   col = c("red", "blue", "green", "yellow", "purple"))
* 

### 4.1.2 Numerical features

* Distributions of numerical features
* par(mfrow=c(5, 2))   
  for (i in num\_col[1:10])  
  {  
   hist(train\_df[,i],   
   main = paste("Histogram of " , i),   
   xlab = i ,  
   ylab = "Frequency" ,  
   col = c("blue"))  
   rug(jitter(train\_df[,i]))  
   lines(density(train\_df[,i]), col="red", lwd=2)  
  }
* 
* par(mfrow=c(3, 2))   
  for (i in num\_col[11:16])  
  {  
   hist(train\_df[,i],   
   main = paste("Histogram of " , i),   
   xlab = i ,  
   ylab = "Frequency" ,  
   col = c("blue"))  
   rug(jitter(train\_df[,i]))  
   lines(density(train\_df[,i]), col="red", lwd=2)  
  }
* 
* **Kernel density plots**
* ***kernel density estimation is a nonparametric method for estimating the probability density function of a random variable.***
* par(mfrow=c(5, 2))   
  for (i in num\_col[1:10]) {  
   d <- density(train\_df[,i])  
   plot(d, main="Kernel Density of Miles Per Gallon")  
   polygon(d, col="red", border="blue")  
   rug(train\_df[,i], col="brown")  
   }
* 
* par(mfrow=c(3, 2))   
  for (i in num\_col[11:16]) {  
   d <- density(train\_df[,i])  
   plot(d, main= paste("Kernel Density of " , i))  
   polygon(d, col="red", border="blue")  
   rug(train\_df[,i], col="brown")  
   }
* 

## 4.2 Bivariante analysis

### 4.2.1 Categorical features

We gonna divide the agriculture Districts to Zones acoording to this [website](https://geography4u.com/wp-content/uploads/2020/06/Agro-climatic-zones-in-Bihar.jpg) .

* Districts to Zones
* # Define a function to assign zones based on the District  
  assign\_value <- function(District) {  
   if (District %in% c('Nalanda', 'Gaya')) {  
   return('zone1')  
   } else if (District == 'Jamui') {  
   return('zone2')  
   } else if (District == 'Vaishali') {  
   return('zone3')  
   } else {  
   return(NA) # Handle other cases if needed  
   }  
  }  
    
  # Apply the function to create a new 'Zone' column  
  train\_df$Zone <- sapply(train\_df$District, assign\_value)  
    
  # show the result  
    
  pander(train\_df %>%  
   group\_by(Zone) %>%  
   summarise(Unique\_Districts = list(unique(District))))

| Zone | Unique\_Districts |
| --- | --- |
| zone1 | Nalanda, Gaya |
| zone2 | Jamui |
| zone3 | Vaishali |

* Basic statistics for Yield per Zone
* pander(train\_df %>%  
   group\_by(Zone) %>%  
   summarise(mean = mean(Yield),  
   median = median(Yield),  
   sum = sum(Yield),  
   min = min(Yield),  
   max = max(Yield)))

| Zone | mean | median | sum | min | max |
| --- | --- | --- | --- | --- | --- |
| zone1 | 645.5 | 565 | 757822 | 10 | 7510 |
| zone2 | 834.9 | 480 | 698018 | 7 | 16800 |
| zone3 | 335.6 | 240 | 273829 | 4 | 2880 |

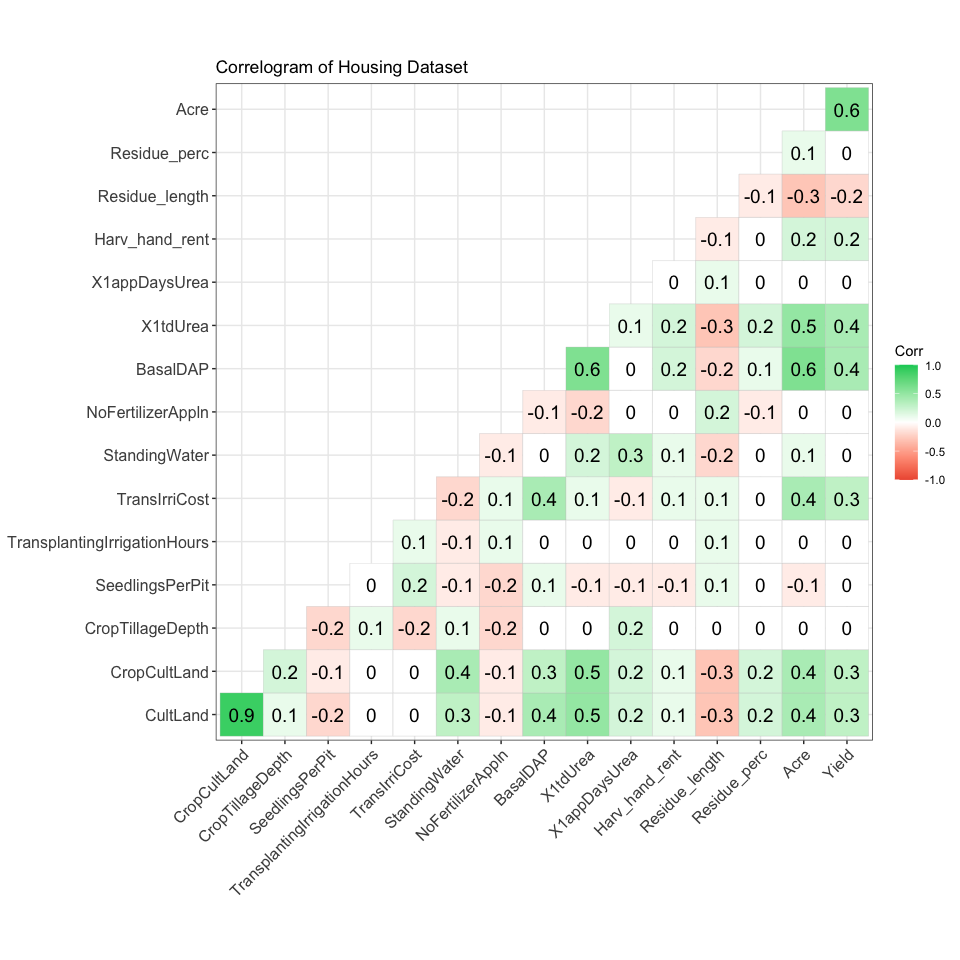
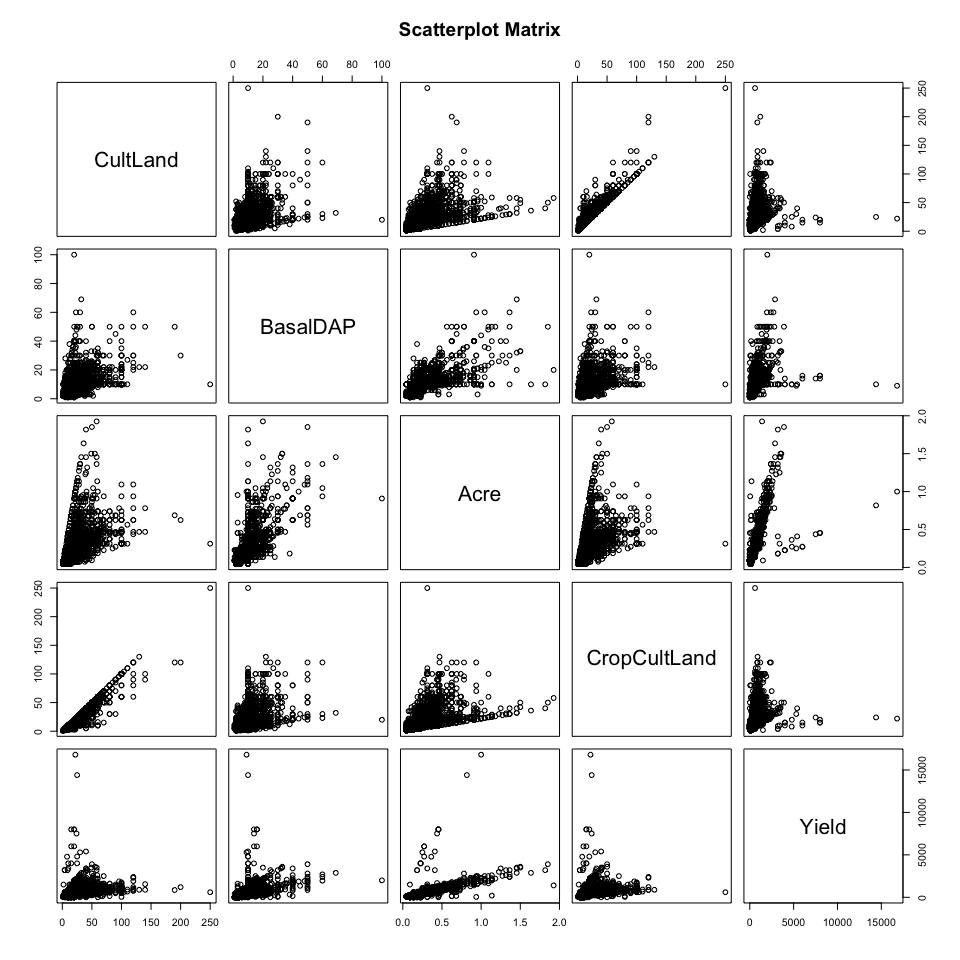
* Districts and Blocks
* pander(train\_df %>%  
   group\_by(District) %>%  
   summarise(Unique\_Blocks = list(unique(Block))))

| District | Unique\_Blocks |
| --- | --- |
| Gaya | Gurua, Wazirganj |
| Jamui | Khaira, Jamui |
| Nalanda | Noorsarai, Rajgir |
| Vaishali | Mahua, Chehrakala, Garoul |

* Yield per Acre for each Districts
* train\_df$Yield\_per\_Acre <- train\_df$Yield / train\_df$Acre   
  pander(train\_df %>%  
   group\_by(District) %>%  
   summarise(mean = mean(Yield\_per\_Acre),  
   median = median(Yield\_per\_Acre),  
   sum = sum(Yield\_per\_Acre),  
   min = min(Yield\_per\_Acre),  
   max = max(Yield\_per\_Acre)))

| District | mean | median | sum | min | max |
| --- | --- | --- | --- | --- | --- |
| Gaya | 2057 | 2160 | 816462 | 486 | 3714 |
| Jamui | 2132 | 1760 | 1782040 | 15.4 | 22000 |
| Nalanda | 2160 | 1960 | 1678244 | 32 | 21200 |
| Vaishali | 1637 | 1760 | 1335997 | 44 | 16500 |

### 4.2.2 Numerical features

* **Correlation analysis (correlation matrix)**
* library(ggcorrplot)  
  corr <- round(cor(train\_df[,num\_col]), 1)  
    
  # Plot  
  ggcorrplot(corr,  
   type = "lower",  
   lab = TRUE,   
   lab\_size = 5,   
   colors = c("tomato2", "white", "springgreen3"),  
   title="Correlogram of Housing Dataset",   
   ggtheme=theme\_bw)
* 
* **Scatterplot matrix**
* **we’ll select features more than .6 correlation coefficient**
* cor\_matrix <- cor(train\_df[,num\_col])  
  # Find pairs with correlation greater than 0.6  
  high\_cor\_pairs <- which(abs(cor\_matrix) > 0.6 & abs(cor\_matrix) < 1, arr.ind = TRUE)  
  high\_cor\_pairs <- high\_cor\_pairs[high\_cor\_pairs[, 1] < high\_cor\_pairs[, 2], ]  
  high\_cor\_features <- unique(c(rownames(cor\_matrix)[high\_cor\_pairs[, 1]], colnames(cor\_matrix)[high\_cor\_pairs[, 2]]))  
    
  # Print high correlation pairs  
  pander(high\_cor\_features)
* *CultLand*, *BasalDAP*, *Acre*, *CropCultLand* and *Yield*
* pairs(~CultLand + BasalDAP + Acre + CropCultLand + Yield, data = train\_df ,  
   main = "Scatterplot Matrix" , col = c("black"))
* 

# 5 Statistical testing

***In this section using statistical inferences and hypothesis testing we’ll try to answer the questions in The Objective section***.

**NOTE: p-value**: If the p-value is less than the chosen significance level (e.g., 0.05), reject the null hypothesis.

### **1. What the distribution of the yield variables ?**

#### **Kolmogorov-Smirnov Test**

* ***Hypotheses***
* **Null Hypothesis (H0)**: The sample comes from the Normal distribution.
* **Alternative Hypothesis (H1)**: The sample does not come from the Normal distribution.
* pander(ks\_test <- ks.test(train\_df$Yield, "pnorm", mean=mean(train\_df$Yield),  
   sd=sd(train\_df$Yield)) , style ="grid")

Asymptotic one-sample Kolmogorov-Smirnov test: train\_df$Yield

| Test statistic | P value | Alternative hypothesis |
| --- | --- | --- |
| 0.2062 | 9.166e-105 \* \* \* | two-sided |

### 2.what is the crop yield for each District in India what is the largest?

#### Kruskal-Wallis test

* ***Hypotheses***
* **Null Hypothesis (H0)**: The distributions of the groups are identical.
* **Alternative Hypothesis (H1)**: At least one of the group distributions is different.
* pander(train\_df %>%  
   group\_by(District) %>%  
   summarise(median = median(Yield)), style ="grid")

| District | median |
| --- | --- |
| Gaya | 405 |
| Jamui | 480 |
| Nalanda | 600 |
| Vaishali | 240 |

* # Perform Kruskal-Wallis test  
  pander(kruskal.test(Yield ~ District, data = train\_df) , style ="grid" )

Kruskal-Wallis rank sum test: Yield by District

| Test statistic | df | P value |
| --- | --- | --- |
| 744.1 | 3 | 5.745e-161 \* \* \* |

### 3.What is the different agriculture methods that influence the crop yield ?

#### Mann-Whitney U Test

* ***Hypotheses***
* **Null Hypothesis (H0)**: The distributions of the two groups are identical (which implies the medians are equal).
* **Alternative Hypothesis (H1)**: The distributions of the two groups are different (which implies the medians are different).
* pander(train\_df %>%  
   group\_by(Harv\_method ) %>%  
   summarise(median = median(Yield)), style ="grid")

| Harv\_method | median |
| --- | --- |
| hand | 410 |
| machine | 600 |

* pander(wilcox.test(Yield ~ Harv\_method, data = train\_df) , style ="grid" )

Wilcoxon rank sum test with continuity correction: Yield by Harv\_method

| Test statistic | P value | Alternative hypothesis |
| --- | --- | --- |
| 71588 | 4.443e-08 \* \* \* | two.sided |

* pander(train\_df %>%  
   group\_by(Threshing\_method) %>%  
   summarise(median = median(Yield)), style ="grid")

| Threshing\_method | median |
| --- | --- |
| hand | 400 |
| machine | 450 |

* pander(wilcox.test(Yield ~ Threshing\_method, data = train\_df) , style ="grid" )

Wilcoxon rank sum test with continuity correction: Yield by Threshing\_method

| Test statistic | P value | Alternative hypothesis |
| --- | --- | --- |
| 877581 | 3.688e-08 \* \* \* | two.sided |

### 4.What the best agriculture methods that implies better crop yield ?

#### Mann-Whitney U Test one side

* ***Hypotheses***
* **Null Hypothesis (H0)**: The distributions of the two groups are identical (which implies the medians are equal).
* **Alternative Hypothesis (H1)**: The values in one group are generally greater than the values in the other group.
* pander(wilcox.test(Yield ~ Harv\_method, alternative = "greater" ,data = train\_df)  
   , style ="grid")

Wilcoxon rank sum test with continuity correction: Yield by Harv\_method

| Test statistic | P value | Alternative hypothesis |
| --- | --- | --- |
| 71588 | 1 | greater |

* pander(wilcox.test(Yield ~ Threshing\_method, alternative = "greater" ,data = train\_df)  
   , style ="grid")

Wilcoxon rank sum test with continuity correction: Yield by Threshing\_method

| Test statistic | P value | Alternative hypothesis |
| --- | --- | --- |
| 877581 | 1 | greater |

### 5.What are the variables that are correlated to the yield variable ?

from section 4.2.2 we have answer the question , but we gonna use the following test for Yield and Acre variables

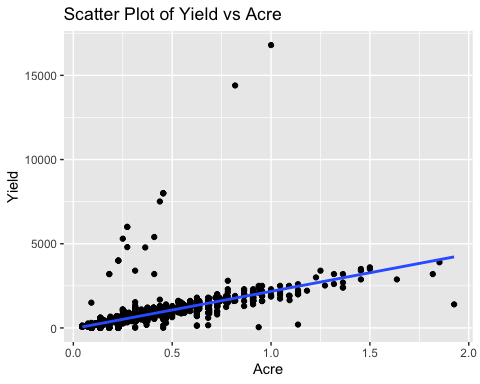
#### Spearman’s Rank Correlation

* ***Hypotheses***
* **Null Hypothesis (H0)**: There is no association between the two variables.
* **Alternative Hypothesis (H1)**: There is an association between the two variables.

pander(cor.test(train\_df$Yield, train\_df$Acre , method = "spearman"), style ="grid")

Spearman’s rank correlation rho: train\_df$Yield and train\_df$Acre

| Test statistic | P value | Alternative hypothesis | rho |
| --- | --- | --- | --- |
| 283225039 | 0 \* \* \* | two.sided | 0.9247 |

* ***Plot the result***
* ggplot(train\_df, aes(x = Acre, y = Yield)) +  
   geom\_point() +  
   geom\_smooth(method = "lm", se = FALSE) +  
   labs(title = "Scatter Plot of Yield vs Acre",  
   x = "Acre",  
   y = "Yield")
* 

# 6. Insights and conclusion

***After this long walk through these Indian districts and blocks and saw the nature of rice and wheat agriculture in this zones using this data , i see that this document is obvious from statistical over view, so I’ll provide anther Report.pdf document that contain a brief report for all what we explore and draw hypotheses about.***

### Anthor links and references:

* [github](https://github.com/ahmedalharth/Digital-Green-crop-Yield-Estimate-R-.git).
* [google drive](https://drive.google.com/drive/folders/18aWIEAIvRfXbyPN4sC2DyuodwvI0_aE4?usp=share_link).