final\_rmd

final

2024-08-12

# Load necessary libraries  
library(readr)

## Warning: package 'readr' was built under R version 4.4.1

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.4.1

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.4.1

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(broom)

## Warning: package 'broom' was built under R version 4.4.1

# Read the CSV file  
data <- read.csv("agricultural\_yield\_train.csv")  
  
# Check for missing values using colSums  
missing\_values <- colSums(is.na(data))  
  
# Print the number of missing values for each column  
print(missing\_values)

## Soil\_Quality Seed\_Variety   
## 0 0   
## Fertilizer\_Amount\_kg\_per\_hectare Sunny\_Days   
## 0 0   
## Rainfall\_mm Irrigation\_Schedule   
## 0 0   
## Yield\_kg\_per\_hectare   
## 0

# Function to identify outliers using IQR  
find\_outliers <- function(x) {  
 Q1 <- quantile(x, 0.25)  
 Q3 <- quantile(x, 0.75)  
 IQR <- Q3 - Q1  
 lower\_bound <- Q1 - 1.5 \* IQR  
 upper\_bound <- Q3 + 1.5 \* IQR  
 outliers <- which(x < lower\_bound | x > upper\_bound)  
 return(outliers)  
}  
  
# Identify outliers for each numeric column  
numeric\_cols <- names(data)[sapply(data, is.numeric)]  
outliers\_list <- lapply(data[, numeric\_cols], find\_outliers)  
  
# Combine all outlier indices  
all\_outliers <- unique(unlist(outliers\_list))  
# Create a data frame with outlier values  
outliers\_data <- data[all\_outliers, ]  
  
# Print the data frame  
head(outliers\_data)

## Soil\_Quality Seed\_Variety Fertilizer\_Amount\_kg\_per\_hectare Sunny\_Days  
## 212 59.01316 1 65.09466 130.08629  
## 432 65.81460 1 275.03355 127.14334  
## 483 70.71398 1 201.61239 127.23752  
## 728 86.05724 1 192.22578 128.67852  
## 793 68.00166 1 193.71652 133.37125  
## 946 53.12819 1 209.89666 72.62996  
## Rainfall\_mm Irrigation\_Schedule Yield\_kg\_per\_hectare  
## 212 489.5360 6 779.9393  
## 432 613.3043 4 800.8255  
## 483 570.9249 5 757.7700  
## 728 558.2322 6 954.1423  
## 793 541.1011 6 914.2621  
## 946 442.8674 7 914.8469

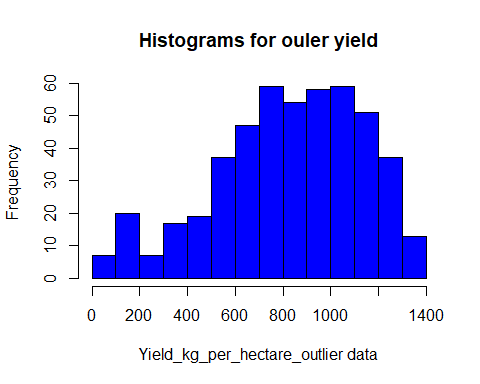
# Remove outliers from the data  
cleaned\_data <- data[-all\_outliers, ]  
  
#######################################################  
  
# Print the number of rows before and after removing outliers  
print(paste("Number of rows before removing outliers:", nrow(data)))

## [1] "Number of rows before removing outliers: 16000"

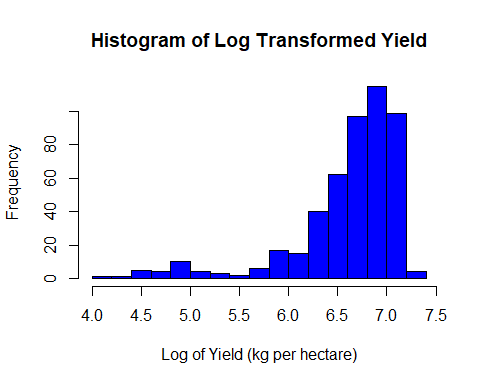
print(paste("Number of rows after removing outliers:", nrow(cleaned\_data)))

## [1] "Number of rows after removing outliers: 15515"

#consider the oulier of data frame  
hist(outliers\_data$Yield\_kg\_per\_hectare ,xlab= "Yield\_kg\_per\_hectare\_outlier data",main = "Histograms for ouler yield",breaks = 12,col = "blue")



# Logarithmic transformation  
log\_yield <- log(outliers\_data$Yield\_kg\_per\_hectare)  
hist(log\_yield, xlab = "Log of Yield (kg per hectare)", main = "Histogram of Log Transformed Yield", breaks = 12, col = "blue")



#mean of the compair yeild  
#with outlier mean of yeil  
mean(data$Yield\_kg\_per\_hectare)

## [1] 713.9997

#with out mean of yeil  
mean(cleaned\_data$Yield\_kg\_per\_hectare)

## [1] 710.5186

#ouiltr mean  
mean(outliers\_data$Yield\_kg\_per\_hectare)

## [1] 825.3593

#with oulier variyanse yeild  
var(data$Yield\_kg\_per\_hectare)

## [1] 40889.25

#with out oulier variyanse yeild  
var(cleaned\_data$Yield\_kg\_per\_hectare)

## [1] 38814.39

#oulier variyanse yeild  
var(outliers\_data$Yield\_kg\_per\_hectare)

## [1] 94665.92

######################################################  
#consider the cleaned\_data frame  
# Calculate the number of samples for the training set (75% of the data)  
train\_size <- floor(0.75 \* nrow(cleaned\_data))  
  
# Generate a vector of row indices  
indices <- 1:nrow(cleaned\_data)  
  
# Randomly sample indices for the training set  
train\_indices <- sample(indices, size = train\_size, replace = FALSE)  
  
# Create training and testing sets  
train1 <- cleaned\_data[train\_indices, ]  
test1 <- cleaned\_data[-train\_indices, ]  
  
# Print the number of rows train and testing data set  
print(paste("Number of rows train data set:", nrow(train1)))

## [1] "Number of rows train data set: 11636"

print(paste("Number of rows teast data set:", nrow(test1)))

## [1] "Number of rows teast data set: 3879"

# Filter the train1 dataset where Seed\_Variety is 0  
train1\_filtered\_0 <- subset(train1, Seed\_Variety == 0)  
  
# Print the head of the filtered dataset to verify  
head(train1\_filtered\_0)

## Soil\_Quality Seed\_Variety Fertilizer\_Amount\_kg\_per\_hectare Sunny\_Days  
## 15887 73.77213 0 216.7756 100.36530  
## 10203 55.29475 0 205.9384 110.11012  
## 3288 74.18859 0 291.5795 77.83488  
## 10039 70.83459 0 298.1495 92.00413  
## 4265 98.47427 0 257.4484 93.08829  
## 8489 72.22469 0 122.0645 114.39440  
## Rainfall\_mm Irrigation\_Schedule Yield\_kg\_per\_hectare  
## 15887 434.1471 2 373.2112  
## 10203 432.5018 7 626.8286  
## 3288 521.5628 4 472.3144  
## 10039 560.1343 3 474.9772  
## 4265 366.0212 3 519.3002  
## 8489 351.2376 3 490.3949

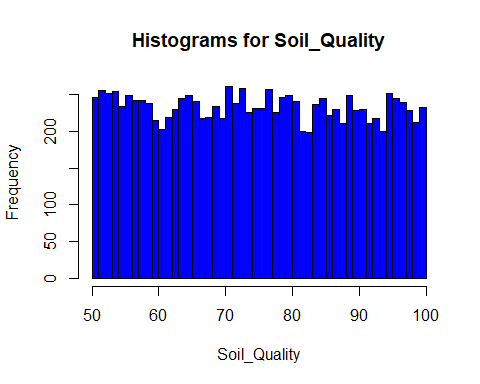
# Filter the train1 dataset where Seed\_Variety is 1  
train1\_filtered\_1 <- subset(train1, Seed\_Variety == 1)  
  
# Print the head of the filtered dataset to verify  
head(train1\_filtered\_1)

## Soil\_Quality Seed\_Variety Fertilizer\_Amount\_kg\_per\_hectare Sunny\_Days  
## 8179 73.79380 1 286.87839 78.39382  
## 2685 83.96964 1 208.89274 103.32797  
## 5640 52.26928 1 94.13869 105.75976  
## 2945 52.95116 1 264.06048 82.48267  
## 6114 51.39387 1 283.45747 113.54430  
## 10929 65.27774 1 91.05633 106.52755  
## Rainfall\_mm Irrigation\_Schedule Yield\_kg\_per\_hectare  
## 8179 510.9672 7 903.6764  
## 2685 653.4640 3 705.3779  
## 5640 746.5282 5 522.3417  
## 2945 466.4019 8 996.3509  
## 6114 453.8228 7 1090.2711  
## 10929 503.8265 7 718.7664

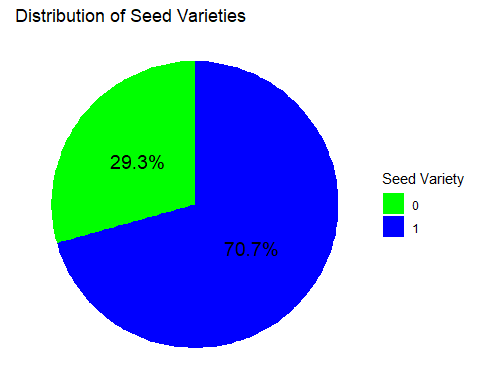
#####################################################################  
#discriptive anyis  
df <-train1 # Training data set  
# Display summary statistics  
summary(df)

## Soil\_Quality Seed\_Variety Fertilizer\_Amount\_kg\_per\_hectare  
## Min. : 50.01 Min. :0.0000 Min. : 50.05   
## 1st Qu.: 62.27 1st Qu.:0.0000 1st Qu.:112.91   
## Median : 74.64 Median :1.0000 Median :175.99   
## Mean : 74.72 Mean :0.7068 Mean :175.26   
## 3rd Qu.: 87.20 3rd Qu.:1.0000 3rd Qu.:237.46   
## Max. :100.00 Max. :1.0000 Max. :299.99   
## Sunny\_Days Rainfall\_mm Irrigation\_Schedule Yield\_kg\_per\_hectare  
## Min. : 73.03 Min. :232.8 Min. : 0.000 Min. : 157.3   
## 1st Qu.: 93.24 1st Qu.:435.2 1st Qu.: 3.000 1st Qu.: 576.7   
## Median : 99.99 Median :501.4 Median : 5.000 Median : 724.6   
## Mean : 99.91 Mean :501.3 Mean : 4.929 Mean : 709.8   
## 3rd Qu.:106.63 3rd Qu.:566.8 3rd Qu.: 6.000 3rd Qu.: 851.9   
## Max. :126.83 Max. :767.5 Max. :10.000 Max. :1277.0

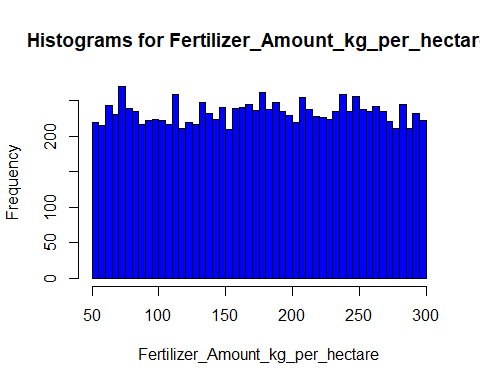
#Histograms for Soil\_Quality  
hist(df$Soil\_Quality,xlab = "Soil\_Quality",main = "Histograms for Soil\_Quality",breaks = 50,col = "blue")



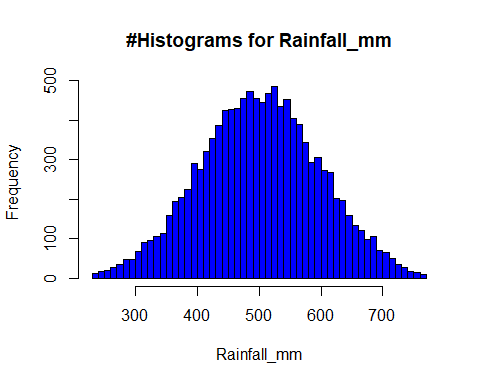
# Summarize the data to get counts and percentages for each seed variety  
seed\_variety\_counts <- df %>%  
 group\_by(Seed\_Variety) %>%  
 summarise(Count = n()) %>%  
 mutate(Percentage = Count / sum(Count) \* 100)  
  
# Plot a pie chart using ggplot  
ggplot(seed\_variety\_counts, aes(x = "", y = Count, fill = factor(Seed\_Variety))) +  
 geom\_bar(stat = "identity", width = 1) +  
 coord\_polar("y", start = 0) +  
 geom\_text(aes(label = paste0(round(Percentage, 1), "%")),   
 position = position\_stack(vjust = 0.5), size = 5) +  
 labs(title = "Distribution of Seed Varieties", fill = "Seed Variety") +  
 theme\_void() +  
 scale\_fill\_manual(values = c("green", "blue"))



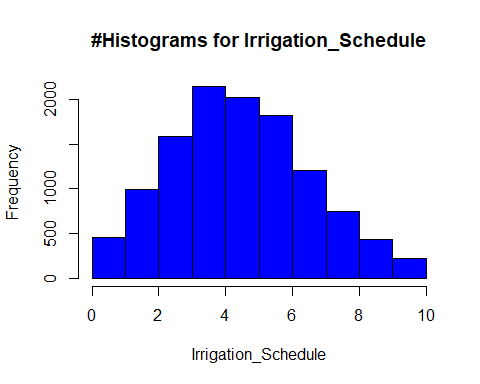
#Histograms for Fertilizer\_Amount\_kg\_per\_hectare  
  
hist(df$Fertilizer\_Amount\_kg\_per\_hectare,xlab = "Fertilizer\_Amount\_kg\_per\_hectare",main = "Histograms for Fertilizer\_Amount\_kg\_per\_hectare",breaks = 50,col = "blue")



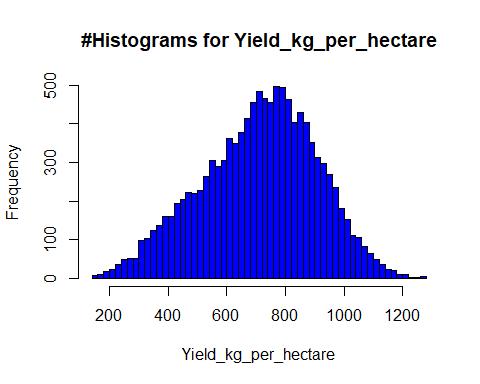
#Histograms for Rainfall\_mm  
  
hist(df$Rainfall\_mm,xlab = "Rainfall\_mm",main = "#Histograms for Rainfall\_mm",breaks = 50,col = "blue")



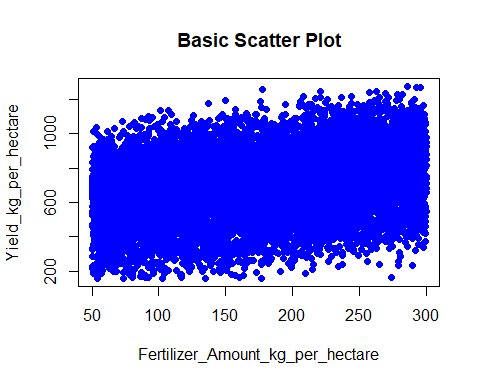
#Histograms for Irrigation\_Schedule  
  
hist(df$Irrigation\_Schedule ,xlab= "Irrigation\_Schedule",main = "#Histograms for Irrigation\_Schedule",breaks = 12,col = "blue")



#Histograms for Yield\_kg\_per\_hectare  
  
hist(df$Yield\_kg\_per\_hectare ,xlab= "Yield\_kg\_per\_hectare",main = "#Histograms for Yield\_kg\_per\_hectare",breaks = 50,col = "blue")



#consider the relasion between two variyable  
  
plot(df$Fertilizer\_Amount\_kg\_per\_hectare, df$Yield\_kg\_per\_hectare, main = "Basic Scatter Plot", xlab = "Fertilizer\_Amount\_kg\_per\_hectare", ylab = "Yield\_kg\_per\_hectare", pch = 19, col = "blue")

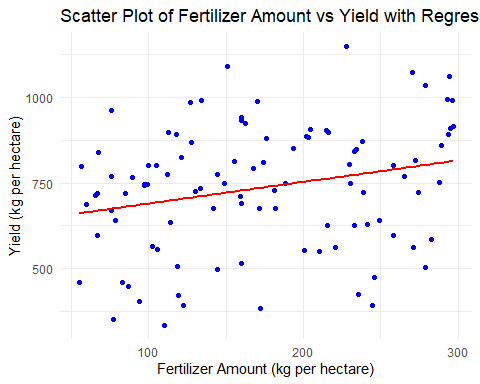


# Correlation matrix  
cor\_matrix <- cor(cleaned\_data %>% select\_if(is.numeric))  
print(cor\_matrix)

## Soil\_Quality Seed\_Variety  
## Soil\_Quality 1.000000000 -0.0031737518  
## Seed\_Variety -0.003173752 1.0000000000  
## Fertilizer\_Amount\_kg\_per\_hectare -0.004597307 -0.0114884161  
## Sunny\_Days -0.004336951 -0.0042159457  
## Rainfall\_mm 0.011597031 -0.0003057816  
## Irrigation\_Schedule 0.003363257 0.0052917529  
## Yield\_kg\_per\_hectare 0.108710202 0.6939216884  
## Fertilizer\_Amount\_kg\_per\_hectare Sunny\_Days  
## Soil\_Quality -0.004597307 -4.336951e-03  
## Seed\_Variety -0.011488416 -4.215946e-03  
## Fertilizer\_Amount\_kg\_per\_hectare 1.000000000 2.189391e-03  
## Sunny\_Days 0.002189391 1.000000e+00  
## Rainfall\_mm 0.004783484 -1.031590e-03  
## Irrigation\_Schedule 0.007026269 -3.105524e-05  
## Yield\_kg\_per\_hectare 0.289617070 9.540786e-02  
## Rainfall\_mm Irrigation\_Schedule  
## Soil\_Quality 0.0115970312 3.363257e-03  
## Seed\_Variety -0.0003057816 5.291753e-03  
## Fertilizer\_Amount\_kg\_per\_hectare 0.0047834843 7.026269e-03  
## Sunny\_Days -0.0010315900 -3.105524e-05  
## Rainfall\_mm 1.0000000000 -2.312458e-03  
## Irrigation\_Schedule -0.0023124582 1.000000e+00  
## Yield\_kg\_per\_hectare -0.2454698683 5.378587e-01  
## Yield\_kg\_per\_hectare  
## Soil\_Quality 0.10871020  
## Seed\_Variety 0.69392169  
## Fertilizer\_Amount\_kg\_per\_hectare 0.28961707  
## Sunny\_Days 0.09540786  
## Rainfall\_mm -0.24546987  
## Irrigation\_Schedule 0.53785871  
## Yield\_kg\_per\_hectare 1.00000000

#####################################################################################################################################################################################  
# Sample 100 random rows from the cleaned\_data data frame  
set.seed(123) # Set seed for reproducibility  
random\_sample <- sample\_n(cleaned\_data, 100)  
  
# Create scatter plot with a regression line for Fertilizer\_Amount\_kg\_per\_hectare vs Yield\_kg\_per\_hectare  
ggplot(random\_sample, aes(x = Fertilizer\_Amount\_kg\_per\_hectare, y = Yield\_kg\_per\_hectare)) +  
 geom\_point(color = "blue") + # Scatter plot  
 geom\_smooth(method = "lm", color = "red", se = FALSE) + # Regression line  
 labs(title = "Scatter Plot of Fertilizer Amount vs Yield with Regression Line",  
 x = "Fertilizer Amount (kg per hectare)",  
 y = "Yield (kg per hectare)") +  
 theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'



######################################################  
#discriptive analyis for comparison of the seed vari  
#train data set  
summary(train1)

## Soil\_Quality Seed\_Variety Fertilizer\_Amount\_kg\_per\_hectare  
## Min. : 50.01 Min. :0.0000 Min. : 50.05   
## 1st Qu.: 62.27 1st Qu.:0.0000 1st Qu.:112.91   
## Median : 74.64 Median :1.0000 Median :175.99   
## Mean : 74.72 Mean :0.7068 Mean :175.26   
## 3rd Qu.: 87.20 3rd Qu.:1.0000 3rd Qu.:237.46   
## Max. :100.00 Max. :1.0000 Max. :299.99   
## Sunny\_Days Rainfall\_mm Irrigation\_Schedule Yield\_kg\_per\_hectare  
## Min. : 73.03 Min. :232.8 Min. : 0.000 Min. : 157.3   
## 1st Qu.: 93.24 1st Qu.:435.2 1st Qu.: 3.000 1st Qu.: 576.7   
## Median : 99.99 Median :501.4 Median : 5.000 Median : 724.6   
## Mean : 99.91 Mean :501.3 Mean : 4.929 Mean : 709.8   
## 3rd Qu.:106.63 3rd Qu.:566.8 3rd Qu.: 6.000 3rd Qu.: 851.9   
## Max. :126.83 Max. :767.5 Max. :10.000 Max. :1277.0

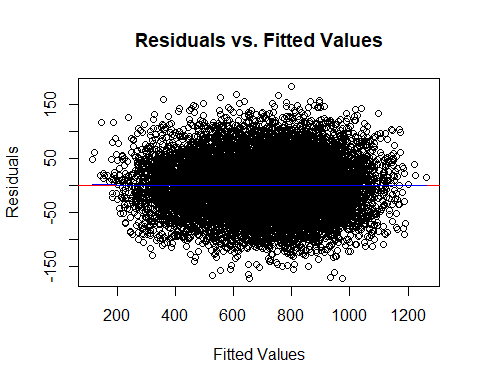
#seed varity o data set  
summary(train1\_filtered\_0)

## Soil\_Quality Seed\_Variety Fertilizer\_Amount\_kg\_per\_hectare  
## Min. : 50.01 Min. :0 Min. : 50.07   
## 1st Qu.: 62.33 1st Qu.:0 1st Qu.:114.75   
## Median : 74.60 Median :0 Median :177.66   
## Mean : 74.75 Mean :0 Mean :176.56   
## 3rd Qu.: 87.23 3rd Qu.:0 3rd Qu.:238.40   
## Max. :100.00 Max. :0 Max. :299.99   
## Sunny\_Days Rainfall\_mm Irrigation\_Schedule Yield\_kg\_per\_hectare  
## Min. : 73.12 Min. :234.4 Min. : 0.000 Min. : 157.3   
## 1st Qu.: 93.39 1st Qu.:433.2 1st Qu.: 3.000 1st Qu.: 397.6   
## Median : 99.87 Median :500.0 Median : 5.000 Median : 494.1   
## Mean : 99.93 Mean :501.0 Mean : 4.924 Mean : 498.6   
## 3rd Qu.:106.47 3rd Qu.:568.4 3rd Qu.: 6.000 3rd Qu.: 593.1   
## Max. :126.43 Max. :764.9 Max. :10.000 Max. :1057.9

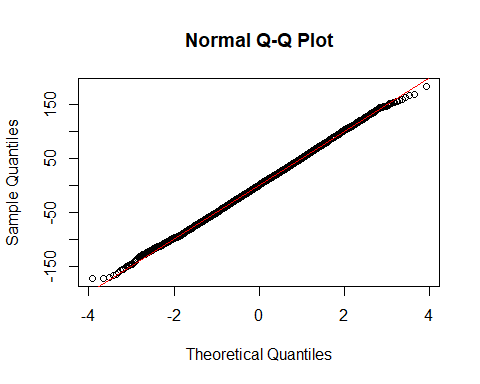
#seed varity 1 data set  
summary(train1\_filtered\_1)

## Soil\_Quality Seed\_Variety Fertilizer\_Amount\_kg\_per\_hectare  
## Min. : 50.01 Min. :1 Min. : 50.05   
## 1st Qu.: 62.26 1st Qu.:1 1st Qu.:112.36   
## Median : 74.65 Median :1 Median :175.09   
## Mean : 74.70 Mean :1 Mean :174.72   
## 3rd Qu.: 87.20 3rd Qu.:1 3rd Qu.:237.20   
## Max. :100.00 Max. :1 Max. :299.99   
## Sunny\_Days Rainfall\_mm Irrigation\_Schedule Yield\_kg\_per\_hectare  
## Min. : 73.03 Min. :232.8 Min. : 0.000 Min. : 313.4   
## 1st Qu.: 93.18 1st Qu.:436.1 1st Qu.: 3.000 1st Qu.: 698.1   
## Median :100.04 Median :501.8 Median : 5.000 Median : 794.1   
## Mean : 99.90 Mean :501.5 Mean : 4.932 Mean : 797.5   
## 3rd Qu.:106.70 3rd Qu.:566.3 3rd Qu.: 6.000 3rd Qu.: 895.2   
## Max. :126.83 Max. :767.5 Max. :10.000 Max. :1277.0

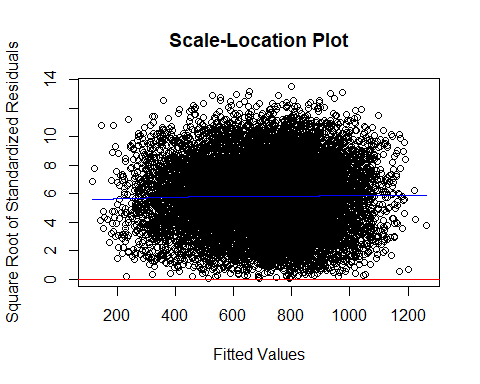
###########################################################################3  
# Fit the model  
model <- lm(Yield\_kg\_per\_hectare ~ Fertilizer\_Amount\_kg\_per\_hectare +   
 Seed\_Variety + Rainfall\_mm + Irrigation\_Schedule + Soil\_Quality +   
 Sunny\_Days, data = train1)  
#check the Diagnosing  
#Diagnosing Heteroscedasticity  
# Plot residuals vs. fitted values  
#This plot helps detect non-linearity, unequal error variances (heteroscedasticity), and outliers.  
plot(model$fitted.values, model$residuals,   
 xlab = "Fitted Values",   
 ylab = "Residuals",   
 main = "Residuals vs. Fitted Values")  
abline(h = 0, col = "red")  
  
# Add a smoothed line to identify patterns  
lines(lowess(model$fitted.values, model$residuals), col = "blue")



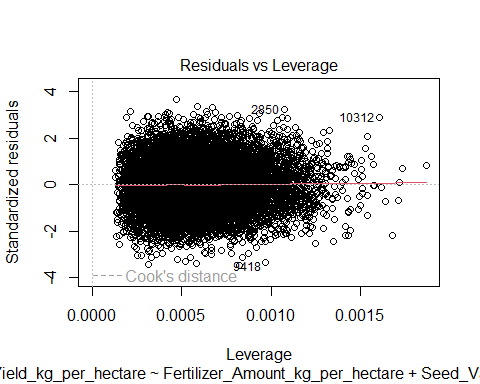
#Normal Q-Q Plot  
qqnorm(model$residuals, main = "Normal Q-Q Plot")  
qqline(model$residuals, col = "red")



#Scale-Location Plot (Spread-Location Plot)  
plot(model$fitted.values, sqrt(abs(model$residuals)),   
 xlab = "Fitted Values",   
 ylab = "Square Root of Standardized Residuals",   
 main = "Scale-Location Plot")  
abline(h = 0, col = "red")  
lines(lowess(model$fitted.values, sqrt(abs(model$residuals))), col = "blue")



#Residuals vs. Leverage Plot  
  
plot(model, which = 5)



#Variance Inflation Factor (VIF)  
#Multicollinearity  
library(car)

## Warning: package 'car' was built under R version 4.4.1

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.4.1

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

vif\_values <- vif(model)  
print(vif\_values)

## Fertilizer\_Amount\_kg\_per\_hectare Seed\_Variety   
## 1.000279 1.000149   
## Rainfall\_mm Irrigation\_Schedule   
## 1.000268 1.000292   
## Soil\_Quality Sunny\_Days   
## 1.000331 1.000231

#####################################################################  
#Model selecsion prat  
  
#with seed variety  
model1 <- lm(Yield\_kg\_per\_hectare ~ Fertilizer\_Amount\_kg\_per\_hectare + Seed\_Variety+ Rainfall\_mm + Irrigation\_Schedule + Soil\_Quality+Sunny\_Days, data = train1)  
# Print the summary of the linear model  
summary(model1)

##   
## Call:  
## lm(formula = Yield\_kg\_per\_hectare ~ Fertilizer\_Amount\_kg\_per\_hectare +   
## Seed\_Variety + Rainfall\_mm + Irrigation\_Schedule + Soil\_Quality +   
## Sunny\_Days, data = train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -172.752 -33.980 0.045 33.440 183.162   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 44.063908 6.160398 7.153 9.01e-13 \*\*\*  
## Fertilizer\_Amount\_kg\_per\_hectare 0.807624 0.006455 125.109 < 2e-16 \*\*\*  
## Seed\_Variety 300.258593 1.017991 294.952 < 2e-16 \*\*\*  
## Rainfall\_mm -0.503244 0.004809 -104.651 < 2e-16 \*\*\*  
## Irrigation\_Schedule 49.778305 0.220689 225.559 < 2e-16 \*\*\*  
## Soil\_Quality 1.553755 0.032012 48.537 < 2e-16 \*\*\*  
## Sunny\_Days 2.030183 0.048249 42.077 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 49.99 on 11629 degrees of freedom  
## Multiple R-squared: 0.9354, Adjusted R-squared: 0.9354   
## F-statistic: 2.808e+04 on 6 and 11629 DF, p-value: < 2.2e-16

#wiith out seed varity  
model2 <- lm(Yield\_kg\_per\_hectare ~ Fertilizer\_Amount\_kg\_per\_hectare +Rainfall\_mm + Irrigation\_Schedule + Soil\_Quality+Sunny\_Days, data = train1)  
# Print the summary of the linear model  
summary(model2)

##   
## Call:  
## lm(formula = Yield\_kg\_per\_hectare ~ Fertilizer\_Amount\_kg\_per\_hectare +   
## Rainfall\_mm + Irrigation\_Schedule + Soil\_Quality + Sunny\_Days,   
## data = train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -382.03 -161.03 60.72 106.79 272.21   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 260.65312 17.81174 14.63 <2e-16 \*\*\*  
## Fertilizer\_Amount\_kg\_per\_hectare 0.78540 0.01880 41.78 <2e-16 \*\*\*  
## Rainfall\_mm -0.49990 0.01400 -35.70 <2e-16 \*\*\*  
## Irrigation\_Schedule 49.90582 0.64266 77.66 <2e-16 \*\*\*  
## Soil\_Quality 1.53707 0.09322 16.49 <2e-16 \*\*\*  
## Sunny\_Days 2.01477 0.14051 14.34 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 145.6 on 11630 degrees of freedom  
## Multiple R-squared: 0.4525, Adjusted R-squared: 0.4522   
## F-statistic: 1922 on 5 and 11630 DF, p-value: < 2.2e-16

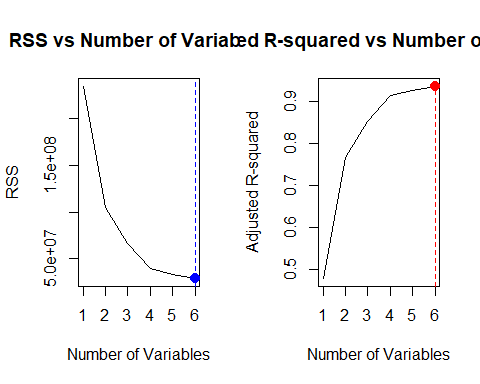
#forwer selcsion method  
library(leaps)

## Warning: package 'leaps' was built under R version 4.4.1

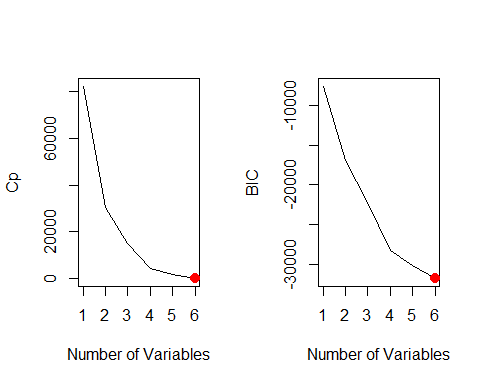
# Fit the regsubsets model with all predictors including Seed\_Variety  
Model\_forward<- regsubsets(Yield\_kg\_per\_hectare ~ Fertilizer\_Amount\_kg\_per\_hectare + Seed\_Variety+ Rainfall\_mm + Irrigation\_Schedule + Soil\_Quality+Sunny\_Days, data = train1,nvmax = 6,method="forward") # nvmax is the maximum number of predictors to include  
  
# Get the summary of the model  
model\_with\_seed\_summary <- summary(Model\_forward)  
  
# Display key information from the summary  
print(model\_with\_seed\_summary)

## Subset selection object  
## Call: regsubsets.formula(Yield\_kg\_per\_hectare ~ Fertilizer\_Amount\_kg\_per\_hectare +   
## Seed\_Variety + Rainfall\_mm + Irrigation\_Schedule + Soil\_Quality +   
## Sunny\_Days, data = train1, nvmax = 6, method = "forward")  
## 6 Variables (and intercept)  
## Forced in Forced out  
## Fertilizer\_Amount\_kg\_per\_hectare FALSE FALSE  
## Seed\_Variety FALSE FALSE  
## Rainfall\_mm FALSE FALSE  
## Irrigation\_Schedule FALSE FALSE  
## Soil\_Quality FALSE FALSE  
## Sunny\_Days FALSE FALSE  
## 1 subsets of each size up to 6  
## Selection Algorithm: forward  
## Fertilizer\_Amount\_kg\_per\_hectare Seed\_Variety Rainfall\_mm  
## 1 ( 1 ) " " "\*" " "   
## 2 ( 1 ) " " "\*" " "   
## 3 ( 1 ) "\*" "\*" " "   
## 4 ( 1 ) "\*" "\*" "\*"   
## 5 ( 1 ) "\*" "\*" "\*"   
## 6 ( 1 ) "\*" "\*" "\*"   
## Irrigation\_Schedule Soil\_Quality Sunny\_Days  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) "\*" " " " "   
## 3 ( 1 ) "\*" " " " "   
## 4 ( 1 ) "\*" " " " "   
## 5 ( 1 ) "\*" "\*" " "   
## 6 ( 1 ) "\*" "\*" "\*"

# Set up the plotting area to display two plots side by side  
par(mfrow = c(1, 2))  
  
# Plot RSS for each model  
plot(model\_with\_seed\_summary$rss,   
 xlab = "Number of Variables",   
 ylab = "RSS",   
 type = "l",   
 main = "RSS vs Number of Variables")  
  
# Find and highlight the model with the minimum RSS  
RSS\_min <- which.min(model\_with\_seed\_summary$rss)  
points(RSS\_min, model\_with\_seed\_summary$rss[RSS\_min], col = "blue", cex = 2, pch = 20)  
abline(v = RSS\_min, col = "blue", lty = 2)  
  
# Plot Adjusted R-squared for each model  
plot(model\_with\_seed\_summary$adjr2,   
 xlab = "Number of Variables",   
 ylab = "Adjusted R-squared",   
 type = "l",   
 main = "Adjusted R-squared vs Number of Variables")  
  
# Find and highlight the model with the maximum Adjusted R-squared  
adjr2\_max <- which.max(model\_with\_seed\_summary$adjr2)  
points(adjr2\_max, model\_with\_seed\_summary$adjr2[adjr2\_max], col = "red", cex = 2, pch = 20)  
abline(v = adjr2\_max, col = "red", lty = 2)



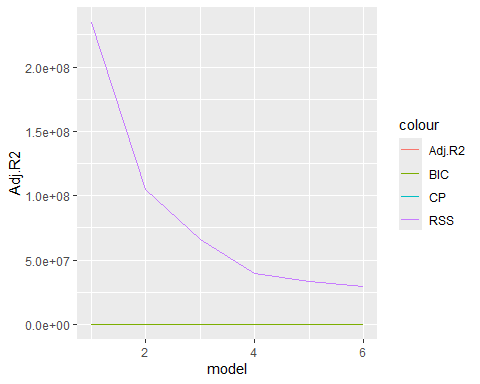
par(mfrow = c(1, 2))  
plot(model\_with\_seed\_summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")  
cp.min <- which.min(model\_with\_seed\_summary$cp)  
points(cp.min, model\_with\_seed\_summary$cp[cp.min], col = "red", cex = 2, pch = 20)  
bic.min <- which.min(model\_with\_seed\_summary$bic)  
plot(model\_with\_seed\_summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")  
points(bic.min, model\_with\_seed\_summary$bic[bic.min], col = "red", cex = 2, pch = 20)



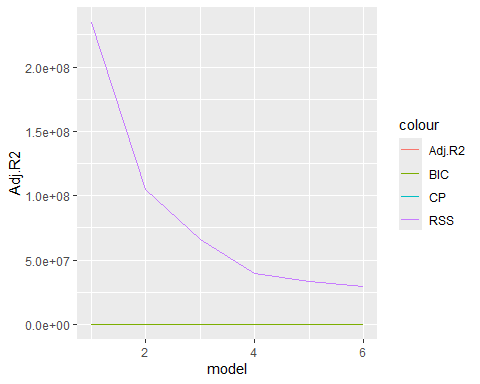
res.sum <- summary(Model\_forward)  
criterion<-data.frame(  
model=1:6,  
Adj.R2 = (res.sum$adjr2),  
CP = (res.sum$cp),  
BIC = (res.sum$bic),  
RSS=res.sum$rss  
)  
head(criterion)

## model Adj.R2 CP BIC RSS  
## 1 1 0.4783875 82318.723 -7555.337 234753559  
## 2 2 0.7667792 30373.167 -16913.251 104952816  
## 3 3 0.8528503 14871.444 -22263.678 66213847  
## 4 4 0.9127561 4083.971 -28337.945 39254317  
## 5 5 0.9255808 1775.468 -30179.645 33481091  
## 6 6 0.9354083 7.000 -31819.256 29057245

library(ggplot2)  
ggplot(criterion, aes(model)) +  
 geom\_line(aes(y = Adj.R2, colour = "Adj.R2")) +  
 geom\_line(aes(y = CP, colour = "CP"))+  
 geom\_line(aes(y = BIC, colour = "BIC"))+  
 geom\_line(aes(y = RSS, colour = "RSS"))



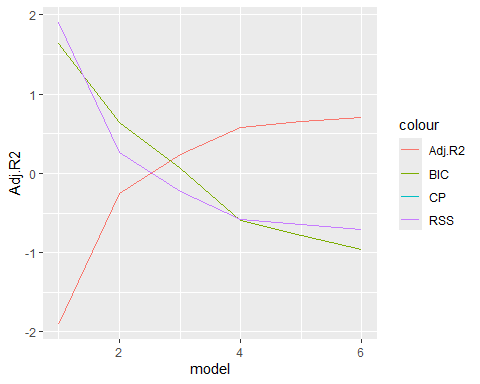
library(ggplot2)  
ggplot(criterion, aes(model)) +  
 geom\_line(aes(y = Adj.R2, colour = "Adj.R2")) +  
 geom\_line(aes(y = CP, colour = "CP"))+  
 geom\_line(aes(y = BIC, colour = "BIC"))+  
 geom\_line(aes(y = RSS, colour = "RSS"))



# standardize  
criterion\_std<-cbind(model=criterion$model, scale(criterion[,-1]))  
criterion\_std<-as.data.frame(criterion\_std)  
head(criterion\_std)

## model Adj.R2 CP BIC RSS  
## 1 1 -1.9050862 1.9051292 1.6414660 1.9050931  
## 2 2 -0.2580365 0.2579539 0.6368103 0.2580227  
## 3 3 0.2335289 -0.2336003 0.0623942 -0.2335447  
## 4 4 0.5756598 -0.5756672 -0.5897326 -0.5756401  
## 5 5 0.6489041 -0.6488691 -0.7874555 -0.6488979  
## 6 6 0.7050300 -0.7049465 -0.9634824 -0.7050330

#after stadlize  
ggplot(criterion\_std, aes(model)) +  
 geom\_line(aes(y = Adj.R2, colour = "Adj.R2")) +  
 geom\_line(aes(y = CP, colour = "CP"))+  
 geom\_line(aes(y = BIC, colour = "BIC"))+  
 geom\_line(aes(y = RSS, colour = "RSS"))



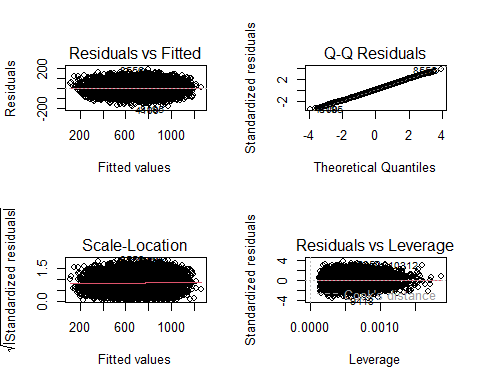
coef(Model\_forward, 6)

## (Intercept) Fertilizer\_Amount\_kg\_per\_hectare   
## 44.0639085 0.8076244   
## Seed\_Variety Rainfall\_mm   
## 300.2585927 -0.5032444   
## Irrigation\_Schedule Soil\_Quality   
## 49.7783048 1.5537554   
## Sunny\_Days   
## 2.0301833

#better model  
better\_model <- lm(Yield\_kg\_per\_hectare ~ Fertilizer\_Amount\_kg\_per\_hectare + Seed\_Variety+ Rainfall\_mm + Irrigation\_Schedule + Soil\_Quality+Sunny\_Days,data=train1 )  
summary(better\_model)

##   
## Call:  
## lm(formula = Yield\_kg\_per\_hectare ~ Fertilizer\_Amount\_kg\_per\_hectare +   
## Seed\_Variety + Rainfall\_mm + Irrigation\_Schedule + Soil\_Quality +   
## Sunny\_Days, data = train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -172.752 -33.980 0.045 33.440 183.162   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 44.063908 6.160398 7.153 9.01e-13 \*\*\*  
## Fertilizer\_Amount\_kg\_per\_hectare 0.807624 0.006455 125.109 < 2e-16 \*\*\*  
## Seed\_Variety 300.258593 1.017991 294.952 < 2e-16 \*\*\*  
## Rainfall\_mm -0.503244 0.004809 -104.651 < 2e-16 \*\*\*  
## Irrigation\_Schedule 49.778305 0.220689 225.559 < 2e-16 \*\*\*  
## Soil\_Quality 1.553755 0.032012 48.537 < 2e-16 \*\*\*  
## Sunny\_Days 2.030183 0.048249 42.077 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 49.99 on 11629 degrees of freedom  
## Multiple R-squared: 0.9354, Adjusted R-squared: 0.9354   
## F-statistic: 2.808e+04 on 6 and 11629 DF, p-value: < 2.2e-16

# Model diagnostics  
par(mfrow=c(2,2))  
plot(better\_model)



###################################################################  
  
# Make predictions on the test data  
test\_predictions <- predict(better\_model, newdata = test1)  
  
  
  
#mean yeil of the focat yeil  
mean(test\_predictions)

## [1] 711.9286

#mean of the actual yeild  
mean(test1$Yield\_kg\_per\_hectare)

## [1] 712.5346

# Calculate performance metrics  
mae <- mean(abs(test\_predictions - test1$Yield\_kg\_per\_hectare))  
rmse <- sqrt(mean((test\_predictions - test1$Yield\_kg\_per\_hectare)^2))  
r\_squared <- cor(test\_predictions, test1$Yield\_kg\_per\_hectare)^2  
  
cat("Test MAE:", mae, "\nTest RMSE:", rmse, "\nTest R-squared:", r\_squared, "\n")

## Test MAE: 39.78421   
## Test RMSE: 49.59865   
## Test R-squared: 0.9372522

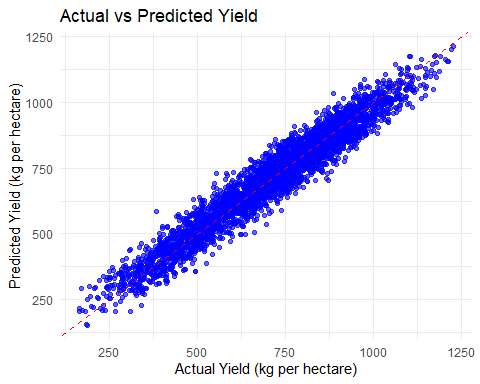
#focat\_value add teast one data set  
  
test1$focat\_yeild=test\_predictions  
head(test1)

## Soil\_Quality Seed\_Variety Fertilizer\_Amount\_kg\_per\_hectare Sunny\_Days  
## 8 69.33589 1 135.92277 119.82700  
## 16 77.14188 0 286.16030 89.06296  
## 17 71.57169 0 91.82663 101.39229  
## 19 78.76663 1 239.54935 101.53371  
## 25 83.85147 1 133.72629 97.19609  
## 26 53.63000 1 298.46915 100.00733  
## Rainfall\_mm Irrigation\_Schedule Yield\_kg\_per\_hectare focat\_yeild  
## 8 384.3504 2 750.3530 711.2333  
## 16 650.7772 5 436.2702 497.2393  
## 17 494.6965 5 372.0941 435.2135  
## 19 429.3474 3 698.1650 799.5728  
## 25 547.8296 6 877.4256 802.9113  
## 26 489.8564 9 1069.6259 1073.2219

# Plot actual vs. predicted values  
ggplot(test1, aes(x = test1$Yield\_kg\_per\_hectare, y = test1$focat\_yeild)) +  
 geom\_point(color = "blue", alpha = 0.6) +  
 geom\_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +  
 labs(title = "Actual vs Predicted Yield",  
 x = "Actual Yield (kg per hectare)",  
 y = "Predicted Yield (kg per hectare)") +  
 theme\_minimal()

## Warning: Use of `test1$Yield\_kg\_per\_hectare` is discouraged.  
## ℹ Use `Yield\_kg\_per\_hectare` instead.

## Warning: Use of `test1$focat\_yeild` is discouraged.  
## ℹ Use `focat\_yeild` instead.



#get sumriy actual vs. predicted values  
#actua  
summary(test1$Yield\_kg\_per\_hectare)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 164.2 571.9 734.2 712.5 858.9 1229.0

#predicted values  
summary(test1$focat\_yeild)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 151.8 582.5 734.3 711.9 853.3 1212.8